IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 23, 2020, accepted December 6, 2020, date of publication December 15, 2020,
date of current version December 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3044954

Principal Tensor Embedding for Unsupervised Tensor Learning

CLAUDIO TURCHETTI, (Life Member, IEEE), LAURA FALASCHETTI, (Member, IEEE), AND LORENZO MANONI

Department of Information Engineering (DII), Università Politecnica delle Marche, 60131 Ancona, Italy

Corresponding author: Claudio Turchetti (c.turchetti@univpm.it)

This work was supported by the Università Politecnica delle Marche.

ABSTRACT Tensors and multiway analysis aim to explore the relationships between the variables used to represent the data and find a summarization of the data with models of reduced dimensionality. However, although in this context a great attention was devoted to this problem, dimension reduction of high-order tensors remains a challenge. The aim of this article is to provide a nonlinear dimensionality reduction approach, named principal tensor embedding (PTE), for unsupervised tensor learning, that is able to derive an explicit nonlinear model of data. As in the standard manifold learning (ML) technique, it assumes multidimensional data lie close to a low-dimensional manifold embedded in a high-dimensional space. On the basis of this assumption a local parametrization of data that accurately captures its local geometry is derived. From this mathematical framework a nonlinear stochastic model of data that depends on a reduced set of latent variables is obtained. In this way the initial problem of unsupervised learning is reduced to the regression of a nonlinear input-output function, i.e. a supervised learning problem. Extensive experiments on several tensor datasets demonstrate that the proposed ML approach gives competitive performance when compared with other techniques used for data reconstruction and classification.

INDEX TERMS Manifold learning, multiway analysis, nonlinear dimensionality reduction, tensor, tensor learning, unsupervised learning.

I. INTRODUCTION

Tensors, also referred to as multiway arrays, are high-order generalizations of vectors and matrices and have been adopted in diverse branches of data analysis, to represent a wide range of real-world data. Examples of tensor data are grayscale and color video sequences [1]–[4], gene expression [5], genome-scale signals [6], magnetic resonance imaging [7], to cite just a few.

Data modeling and classification of these data are important problems in several applications, such as human action and gesture recognition [8], tumor classifications [9], spatio-temporal analysis in climatology, geology and sociology [10], neuroimaging data analysis [11], big data representation [12], completion of big data [13], and so on. To address these problems most previous works represent a tensor by a vector in high-dimensional space and apply ordinary learning methods for vectorial data. Representative techniques in this context include feature extraction and selection [14], [15], linear discriminant analysis (LDA) [16], and support vector machine (SVM) [17]. Unfortunately this approach needs to arrange the tensor data into long vectors causing two main problems, *i)* loss of structural information of tensors, *ii)* vectors with very high dimensionality. To face these problems specific tensor learning algorithms that retain the original structure of tensor data have been recently developed [18]–[26]. However, in all these methods the problem of high dimensionality of data, also known as the *curse of dimensionality*, remains. To deal with such an issue the classical dimensionality reduction method, known as principal component analysis (PCA) [27], [28] was generalized to the second-order case (2-DPCA) [29], low-rank matrices (GLRAM) [30], and high-order cases (MPCA) [31], [32]. Likewise, the linear discriminant analysis (LDA) [33] technique was extended to a 2D case (2-DLDA) [34], [35] and multilinear discriminant analysis (MDA) [36]. Nevertheless to overcome the limitations of methods based on classical approaches, the decomposition of tensors into low-rank components, using two popular models, namely the Tucker decomposition (TD) [37] and the

The associate editor coordinating the review of this manuscript and approving it for publication was Santhosh Kumar Gopalan.

CANDECOMP/PARAFAC (CP) decomposition [38], has been one of the main concerns in tensor analysis to reduce the dimensionality of data [39], [40]. In all the above methods, data are considered as points in a multidimensional space, thus using the global structure information of the dataset alone. Instead many studies have shown that some classes of real world high-dimensional data exist in which they lie on a low-dimensional manifold (a parametrized surface), thus showing a local geometric structure.

Manifold learning (ML) is a nonlinear dimensionality reduction (NLDR) technique that, assuming the existence of an intrinsic structure, the manifold, has proven to be very effective in modeling data with reduced dimensionality [41], [42]. ML, also classified as embedding method, is based on the assumption that high-dimensional data are embedded in a nonlinear manifold of lower dimension [43]–[47], [48]–[50]. In this context several algorithms have been proposed, such as locally linear embedding (LLE) [51], local tangent space alignment (LTSA) [52], locally multidimensional scaling (LMDS) [53], and ISOMAP [54].

To deal with the high-order tensor data, some of these methods were extended by using multiway data analysis [55] and in particular higher order tensor decomposition [56], [57]. For example, Lai *et al.* [58] proposed a robust tensor learning method called sparse tensor alignment (STA) for unsupervised tensor feature extraction. Ju *et al.* [59] introduced a new tensor dimension reduction model based on the Bayesian theory. The proposed method assumes that each observation can be represented as a linear combination of some tensor bases, thus CP decomposition and variational EM algorithm are used to solve this model. He *et al.* [60] proposed tensor subspace analysis (TSA) for second-order learning. In the method suggested by Jiang *et al.* [61], given image tensor data, a k-nearest neighbour graph to encode the geometrical structure of data is constructed. Liu *et al.* [62] proposed a non-linear dimensionality reduction algorithm based on locally linear embedding called supervised locally linear embedding in tensor space (SLLE/T). SLLE/T preserves local manifold structure within each class based on locally linear embedding (LLE) and enforces separability of data points belonging to different classes. Chen *et al.* [63], assuming that data lie in a nonlinear manifold, attempted to discover the intrinsic structure of this manifold with a two-stage algorithm named tensor-based Riemannian manifold distance-approximating projection (TRIMAP). Jia and Fu [64] suggested a low-rank tensor subspace learning for RGB-D action recognition, in which the tensor samples are factorized to obtain three projection matrices by Tucker Decomposition (TD).

The central objective in ML algorithms is to determine an effective parametrization of data. This is a key issue in order to accurately capture the local geometry of the low-dimensional manifold and the following aspects are relevant to this end: *i)* nonlinearity, *ii)* explicit modeling, *iii)* intrinsic dimension (ID) estimation.

With regard to nonlinearity data generally have a nonlinear geometric structure, thus using the tangent space at each data point to locally describe its neighbour, as assumed in some of the previous techniques, is a strong limitation.

With reference to the second aspect, a main drawback of most ML methods is that no explicit mapping representing the local manifold parametrization can be obtained after the training process, as they learn high-dimensional data implicitly.

Regarding ID estimation, the intrinsic dimension (ID) may be interpreted as the minimum number of parameters required to describe the data [65], thus to derive a low-dimensional model, the dataset ID has to be discovered first.

At present, none of the methods suggested so far are able to take into account all the key requirements of nonlinearity, explicit modeling, and ID estimation for low-dimensional tensor modeling.

The aim of this article is to develop a manifold learning-based approach, named principal tensor embedding (PTE), for unsupervised tensor learning, that is able to address the first two of the aforementioned key points, while adopting the most relevant state-of-the-art methods for the ID estimation. This result represents an advancement with respect to the state-of-the-art, as the standard tensor learning techniques are not able to combine all the relevant aspects previously mentioned. The method has been derived by considering a tensor as an element of the finite-dimensional linear space of tensors, in which the inner product defines a metric for the space. Once a basis is computed using the Gram-Schmidt procedure, the coefficient vector in this basis, establishes an isomorphism between a vector space of rank-one and the space of tensors. In this way the problem of dimensionality reduction in tensor space reduces to the dimensionality reduction in vector space. To this end an effective manifold algorithm recently proposed, can be used for the parametrization of data, to accurately capture the local geometry of the low-dimensional manifold that represents the data. In such a way a nonlinear model of data with reduced dimensionality is obtained.

The model establishes an explicit one-to-one correspondence between a tensor, a point on the manifold embedded in the high dimensional space, and a vector, a point in the low-dimensional Euclidean space. Additionally a relationship for the geodesic distance of all pairs of points on the manifold, as a nonlinear function of the Euclidean distance between points in the low-dimensional space, is given.

The rest of the paper is organized as follows. Section II reviews related work on tensor learning. Section III summarizes our method highlighting the most relevant aspects. Section IV introduces some general concepts on finite dimensional linear space of tensors. Section V gives a representation of a tensor in terms of a basis derived by the tensor version of the Gram-Schmidt procedure. In Section VI a nonlinear dimensionality reduction approach, named principal tensor embedding (PTE), is developed, and the estimation of nonlinearity in such a model is treated in Section VII using a

nonparametric kernel regression (NPKR) technique. Experimental results are presented in Section VIII, in which the proposed tensor learning approach is used and compared with some other techniques for data reconstruction and classification problems.

II. RELATED WORK

Tensor learning techniques have been more widely applied to 2-order and 3-order tensors. In the following we summarize the most relevant approaches in these two main fields.

A. TENSOR LEARNING OF 2-ORDER TENSORS

Principal component analysis (PCA) is one of the most common techniques for unsupervised subspace learning, however when applied to tensor objects it requires their reshaping into vectors with high-dimensionality (vectorization). As a result this implies high processing costs in terms of computational and memory demand.

Multilinear principal component analysis (MPCA) [31] is the multilinear extension of the classical PCA that have been used both for two-order and three-order tensors. Recently a method called graph-Laplacian PCA (gLPCA) that combines a manifold learning method, i.e. the graph-Laplacian, with PCA has been proposed [66].

Tucker decomposition (TD) is a technique that reduces a given tensor to a low-rank tensor [37]. It solves an optimization problem, by minimizing the Frobenius distance between the given tensor and a tensor of lower dimensionality.

To account for the geometrical (manifold) structure of image tensor data, a technique called graph-Laplacian Tucker tensor decomposition (GLTD), that combines graph-Laplacian with a regularized version of TD, has recently been proposed in [61].

B. TENSOR LEARNING OF 3-ORDER TENSORS

Tensor neighborhood preserving embedded (TNPE) and tensor locality preserving projection (TLPP) [67] are tensor embedding techniques. They extend neighborhood preserving embedding (NPE) and locality preserving projection (LPP), which can only work with vectorized representation, to be used with more general tensor representations. More specifically, given a set of data points $\{x_i,\ i = 1,\ldots,N\}$ in higher-dimensional space, NPE and LPE seek a transformation matrix that maps each data point x_i to a corresponding lower-dimensional data point y_i. Similarly TNPE and TLPP find a set of transformation matrices for defining the embedded tensor subspaces that together give an optimal approximation to the tensor manifold, preserving some local geometric properties.

Orthogonal tensor neighborhood preserving embedded (OTNPE) [68] is a generalized tensor subspace model similar to TNPE. However, while TNPE cannot ensure the obtained transformation matrices have orthogonal column vectors, OTNPE aims to derive orthonormal basis tensor for TNPE.

Sparse tensor alignment (STA) [58] is a sparse representation incorporated into tensor alignment (TA) framework, a technique that unifies the tensor learning methods. Since a tensor $\mathcal{X}_i$ can be unfolded into a large size matrix X_i, to k nearest neighbours tensors correspond k large size matrices $X_i^{(k)}$. The alignment techniques aims to obtain the projection matrices U_k that map the unfolding tensors $X_i^{(k)}$ into a low-dimensional unfolding tensors $Y_i^{(k)}$, $U_k : X_i^{(k)} \rightarrow Y_i^{(k)}$.

Unfortunately none of the aforementioned techniques is able to satisfy the key requirements in order to accurately capture the local geometry of tensors embedded in a manifold, i.e. nonlinearity, explicit modeling and ID estimation.

III. OUR METHOD

In this article we address the problem of unsupervised learning of tensors. In this case one has a set of N observations, the data $\Omega = \{\mathcal{X}^{(1)},\ldots,\mathcal{X}^{(N)}\}$, of a random M-order tensor $\mathcal{X}$. The goal is to derive a model that depends on a reduced set of parameters, the latent variables, that is able to reconstruct the data. To be effective the dimension d of parameters in the model must be less than the dimension L of the tensor linear space.

Mathematically this problem is equivalent to determine a d-dimensional manifold $\mathcal{M}$ embedded in $\mathbb{R}^{I_1\times I_2\times\ldots\times I_M}$ ($d \ll L = \prod_{j=1}^{M} I_j$) characterized by a nonlinear map

$$\mathcal{X} = \gamma(\beta''),\ \beta'' \in U \subset \mathbb{R}^d,\ \mathcal{X} = \mathbb{R}^{I_1\times I_2\times\ldots\times I_M} \tag{1}$$

from low-dimensional space $U \subset \mathbb{R}^d$ to high-dimensional space $\mathbb{R}^{I_1\times I_2\times\ldots\times I_M}$.

The main steps of our approach are:

- Given the data set $\Omega = \{\mathcal{X}^{(1)},\mathcal{X}^{(2)},\ldots,\mathcal{X}^{(N)}\}$ the Gram-Schmidt procedure is applied to L observations $\{\hat{\mathcal{X}}^{(1)},\hat{\mathcal{X}}^{(2)},\ldots,\hat{\mathcal{X}}^{(L)}\}$ so that an orthonormal basis $\{\mathcal{U}^{(1)},\mathcal{U}^{(2)},\ldots,\mathcal{U}^{(L)}\}$ is obtained.
- The generic tensor $\mathcal{X}$ of the set Ω can be represented as the summation

$$\mathcal{X} = \sum_{i=1}^{L} \alpha_i \mathcal{U}^{(i)},\ L = \prod_{j=1}^{M} I_j \tag{2}$$

  where $\alpha = (\alpha_1,\alpha_2,\ldots,\alpha_L)^T$ is the coefficient vector. Thus to the dataset $\Omega = \{\mathcal{X}^{(1)},\mathcal{X}^{(2)},\ldots,\mathcal{X}^{(N)}\}$ corresponds a set of N vectors $\{\alpha^{(1)},\alpha^{(2)},\ldots,\alpha^{(N)}\}$ in $\mathbb{R}^L$.
- We assume these vectors are points sampled from a manifold $\mathcal{M}$ of dimension d embedded in the L-dimensional observation space, so that a local parametrization

$$\alpha = F(\theta),\ \theta \in \mathbb{R}^d,\ d \in \mathbb{R}^L, \tag{3}$$

  exists, with $d < L$, where θ is the vector of latent variables and d is the so-called intrinsic dimension.
- Since θ is hidden, i.e. not known, it will be shown that a local parametrization γ of the manifold, through a partition $\alpha = \varphi(\alpha'') = \left(\alpha'', G(\alpha'')\right)^T$ of the vector α can be derived.
  This is an explicit modeling of the manifold depending on known variables $\alpha'' \in \mathbb{R}^d$ with d = ID. The model is completely defined once α and $G(\cdot)$ have been estimated.

- By estimating the variance vector $\sigma_\alpha = (\sigma_{\alpha_1}, \ldots, \sigma_{\alpha_L})^T$ of α, their elements are put in decreasing order and the corresponding terms $\{\hat{\mathcal{U}}^{(1)}, \hat{\mathcal{U}}^{(2)}, \ldots, \hat{\mathcal{U}}^{(L)}\}$ are ordered accordingly. In this way a new representation of the tensor $\mathcal{X}$ is obtained

$$\mathcal{X} = \sum_{i=1}^{L} \beta_i \hat{\mathcal{U}}^{(i)} = \sum_{i=1}^{d} \beta_i'' \hat{\mathcal{U}}^{(i)} + \sum_{i>d} G_i(\beta'') \hat{\mathcal{U}}^{(i)} \quad (4)$$

in terms of the new vector $\beta = (\beta_1, \beta_2, \ldots, \beta_L)^T$ such that $\sigma_{\beta_1} \geq \sigma_{\beta_2} \geq \ldots \geq \sigma_{\beta_L}$. It is worth to notice that as the values σ_{β_i} are in decreasing order, then the higher the index i, the lower the importance of the corresponding component. An overview diagram that explains the main steps of the approach is reported in Fig. 1.
- On the basis of this result, a low-dimensional representation of tensor $\mathcal{X}$ can be simply obtained by truncating the summation in (4) to the first r terms

$$\mathcal{X} \cong \sum_{i=1}^{r} \beta_i \hat{\mathcal{U}}^{(i)}, \quad (5)$$

thus obtaining a truncation error

$$\mathcal{E}_r = \sum_{i>r} \sigma_{\beta_i} \quad (6)$$

that decreases as r increases. Following this property the proposed technique has been called principal tensor embedding (PTE).
- Assuming the ID has been determined with one of the methods known in literature, to estimate the function $G(\cdot)$ an effective method for nonparametric input-output nonlinear function regression in tensor space, called nonparametric regression kernel (NPKR), will be used.

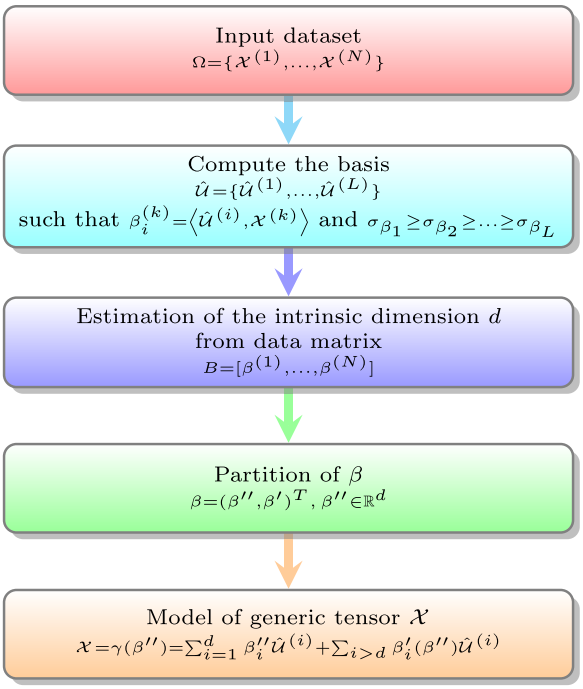


FIGURE 1. Overview diagram of the proposed method.

It is worth to notice that the proposed approach satisfy nonlinearity, explicit modeling and ID estimation, i.e. all the key aspects of ML, thus representing a real advancement to previous techniques.

IV. THE FINITE DIMENSIONAL LINEAR SPACE OF TENSORS

Let us refer to tensors, regarded as multidimensional arrays and denoted by Euler script calligraphic letters, e.g. $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$, where $\times$ represents the Cartesian product. The number of dimensions M, also known as modes, of a tensor denotes the order of a tensor. The elements of an M-order tensor $\mathcal{X}$ will be represented by

$$x_{i_1, i_2, \ldots, i_M}, \quad i_l = 1, 2, \ldots, I_l, \quad l = 1, 2, \ldots, M. \quad (7)$$

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

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \cdots \sum_{i_M=1}^{I_M} x_{i_1 i_2 \ldots i_M} y_{i_1 i_2 \ldots i_M}. \quad (8)$$

From this definition it follows that the norm of a tensor is given by

$$\|\mathcal{X}\| = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}. \quad (9)$$

The outer product of tensor $\mathcal{X} = x_{i_1\, i_2 \ldots i_M} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$ with tensor $\mathcal{Y} = y_{j_1\, j_2 \ldots j_L} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_L}$ is the $(M+L)$-order tensor $\mathcal{Z}$ defined as

$$\mathcal{Z} = \mathcal{X} \circ \mathcal{Y} \quad (10)$$

where the generic element of $\mathcal{Z}$ is given by

$$z_{i_1 i_2 \ldots i_M j_1 j_2 \ldots j_L} = x_{i_1 i_2 \ldots i_M} \cdot y_{j_1 j_2 \ldots j_L} \quad (11)$$

In particular for two vectors x and y the generic element of outer product $\mathcal{Z} = x \circ y$ is the matrix

$$z_{ij} = x_i y_j. \quad (12)$$

With reference to the canonical basis $e_1 = (1, 0, \ldots, 0), \ldots, e_{I_l} = (0, 0, \ldots, 1)$ for $\mathbb{R}^{I_l}$, an M order tensor $\mathcal{X}$ can be decomposed as

$$\mathcal{X} = \sum_{i_1=1}^{I_1} \cdots \sum_{i_M=1}^{I_M} x_{i_1 i_2 \ldots i_M}\ e_{i_1} e_{i_2} \ldots e_{i_M} \quad (13)$$

where the outer product $e_{i_1} e_{i_2} \ldots e_{i_M}$ is the M-order canonical basis tensor. As this basis is of size $\prod_{j=1}^{M} I_j$, thus the set of M-order tensors form a linear space of dimensionality $L = \prod_{j=1}^{M} I_j$. As an example for $M = 2$ and $I_1 = I_2 = M$ we have

$$e_1\, e_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$e_1\, e_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$e_1\ e_3 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \ \ldots \tag{14}$$

For easy of reference Table 1 reports some notations, that will be frequently used in the following.

TABLE 1. Basic notation.

Symbol	Description
$\mathcal{X},\ A,\ \alpha,\ \boldsymbol{\alpha}$	tensor, matrix, vector, random vector
$x_{i_1,i_2,\ldots,i_M}$	generic element of tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$
$\langle \mathcal{X}, \mathcal{Y} \rangle$	inner product of tensors $\mathcal{X}$, $\mathcal{Y}$
$\|\mathcal{X}\|$	norm of tensor $\mathcal{X}$
$\Omega = \{\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(N)}\}$	tensor dataset
$(\beta'', \beta')^T$	partition of vector β
$\{A \mid \Omega\}$	cell array, $A \in \mathbb{R}^{n \times m}$, $\Omega \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M \times N}$
$\mathcal{X} = \gamma(\beta'')$	parametrization of tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$, $\beta'' \in \mathbb{R}^d$
$k_m(x)$	kernel function

V. A BASIS FROM DATA

One of the main problems in representing elements of a linear space, is to find a basis. For vectors $v \in \mathbb{R}^{I_1}$ the problem can be solved estimating the covariance function

$$R_{vv} = E\{vv^T\} \tag{15}$$

from data. Assuming a set $\{v^{(1)}, v^{(2)}, \ldots, v^{(N)}\}$ of observations is collected, then (15) can be approximated as

$$R_{vv} \cong \frac{1}{N} VV^T \tag{16}$$

where $V = [v^{(1)}, v^{(2)}, \ldots, v^{(N)}] \in \mathbb{R}^{I_1 \times N}$ is the data matrix. Thus, once an estimation of R_{vv} is derived, the problem reduces to the decomposition of R_{vv}

$$R_{vv} = U \Sigma U^T \tag{17}$$

By noting that $vv^T = v \circ v$, (15) can be generalized to tensor space by simply substituting v with $\mathcal{X}$, thus obtaining

$$R_{\chi\chi} = E\{\mathcal{X} \circ \mathcal{X}\} \tag{18}$$

where $R_{\chi\chi}$ is a tensor of $2M$ order. Even though some techniques for the decomposition of a tensor are known in literature, the dimension of tensor $R_{\chi\chi}$ in (18) can be very large, so that these approaches cannot be used in practice.

A more effective approach for this purpose can be derived using the well known Gram-Schmidt procedure. Extracting from a given dataset $\Omega = \{\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \ldots, \mathcal{X}^{(N)}\}$, $L = \prod_{j=1}^{M} I_j$ observations $\Lambda = \{\hat{\mathcal{X}}^{(1)}, \hat{\mathcal{X}}^{(2)}, \ldots, \hat{\mathcal{X}}^{(L)}\}$ and assuming they are independent each other, the tensor version of Gram-Schmidt procedure is as follows.

$$\begin{aligned} &1. \quad \mathcal{Y}^{(1)} = \hat{\mathcal{X}}^{(1)}, \ \ \mathcal{U}^{(1)} = \frac{\mathcal{Y}^{(1)}}{\|\mathcal{Y}^{(1)}\|} \\ &2. \quad \mathcal{Y}^{(2)} = \hat{\mathcal{X}}^{(2)} - \left\langle \hat{\mathcal{X}}^{(2)}, \mathcal{U}^{(1)} \right\rangle \mathcal{U}^{(1)}, \ \ \mathcal{U}^{(2)} = \frac{\mathcal{Y}^{(2)}}{\|\mathcal{Y}^{(2)}\|} \\ &k. \quad \mathcal{Y}^{(k)} = \hat{\mathcal{X}}^{(k)} - \sum_{i=1}^{k-1} \left\langle \hat{\mathcal{X}}^{(k)}, \mathcal{U}^{(i)} \right\rangle \mathcal{U}^{(i)}, \ \ \mathcal{U}^{(k)} = \frac{\mathcal{Y}^{(k)}}{\|\mathcal{Y}^{(k)}\|}. \end{aligned} \tag{19}$$

It is straightforward to show that the tensors so obtained $\mathcal{U}^{(1)}, \mathcal{U}^{(2)}, \ldots, \mathcal{U}^{(L)}$ are orthonormal, meaning that

$$\left\langle \mathcal{U}^{(i)}, \mathcal{U}^{(j)} \right\rangle = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}. \tag{20}$$

Having derived a basis $\mathcal{U} = \{\mathcal{U}^{(1)}, \mathcal{U}^{(2)}, \ldots, \mathcal{U}^{(L)}\}$, then the generic tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$ can be represented as the summation

$$\mathcal{X} = \sum_{i=1}^{L} \alpha_i \mathcal{U}^{(i)}, \quad L = I_1 \cdot I_2 \cdot \ldots \cdot I_M \tag{21}$$

where $\alpha_i = \left\langle \mathcal{X}, \mathcal{U}^{(i)} \right\rangle$ is the *i-th* coordinate with respect to the element $\mathcal{U}^{(i)}$ of the basis. Due to the randomness of $\mathcal{X}$ thus $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_L)^T$ is a realization of a random vector $\boldsymbol{\alpha}$. Here a bold face character is used for random variables. In such a way the correspondence ξ

$$\xi : \ \alpha \in \mathbb{R}^L \Longleftrightarrow \mathcal{X} = \sum_{i=1}^{L} \boldsymbol{\alpha}_i \mathcal{U}^{(i)} \tag{22}$$

is defined, or that is the same

$$\xi : \ \alpha \in \mathbb{R}^L \Longleftrightarrow \boldsymbol{\alpha} \in \boldsymbol{H} \tag{23}$$

where $\boldsymbol{H}$ is the space of random variables. For every observation $\mathcal{X}^{(k)}$ of dataset Ω, let us determine the vector

$$\alpha^{(k)} = (\alpha_1^{(k)}, \ldots, \alpha_L^{(k)})^T, \ k = 1, \ldots, N \tag{24}$$

where

$$\alpha_i^{(k)} = \left\langle \mathcal{U}^{(i)}, \mathcal{X}^{(k)} \right\rangle, \ i = 1, \ldots, L, \ k = 1, \ldots, N \ . \tag{25}$$

Thus the following correspondence

$$A = [\alpha^{(1)}, \ldots, \alpha^{(N)}] \Longleftrightarrow \Omega = \{\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(N)}\} \tag{26}$$

holds, where $A \in \mathbb{R}^{L \times N}$ is the data matrix of the coefficient random vector $\boldsymbol{\alpha}$. To represent this correspondence in a more compact way we use the concept of *cell array* in which the elements, the *cells*, are containers that can hold arrays of different sizes. Using this concept (26) becomes

$$\{A|\Omega\} = \{\alpha^{(k)} | \ \mathcal{X}^{(k)}, \ k = 1, \ldots, N\}. \tag{27}$$

VI. LOCAL PARAMETRIZATION OF DATA

On the basis of previous results for a generic tensor $\mathcal{X}$ we can write

$$\mathcal{X} = \sum_{i=1}^{L} \alpha_i \mathcal{U}^{(i)} \tag{28}$$

which establishes a one-to-one correspondence (an isomorphism) between $\mathbb{R}^L$ and the space of tensors.

To reduce the dimensionality of the above representation, a Manifold Learning (ML) approach will be used. In this context the problem can be formalized as follows.

To the dataset $\Omega = \{\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(N)}\} \subset \mathbb{R}^{I_1 \times I_2 \times \ldots I_M}$ in tensor space corresponds a set of N data points $A = \{\alpha^{(1)}, \ldots, \alpha^{(N)}\} \subset \mathbb{R}^L$, which are assumed to be sampled from a manifold $\mathcal{M}$ of dimension d embedded in the L-dimensional observation space. We further assume that the data points do not contain either noise or outliers. This implies that a local parametrization

$$\boldsymbol{\alpha} = F(\boldsymbol{\theta}),\ \boldsymbol{\theta} \in \mathbb{R}^d,\ \boldsymbol{\alpha} \in \mathbb{R}^L \tag{29}$$

exists, with $d < L$. The dimension d represents the so-called *intrinsic dimension* (ID), which may be interpreted as the minimum number of parameters required to represent the data [65].

From differential geometry [69], [70] it can be shown that assuming the values of α lie on a manifold $\mathcal{M}$ of dimension d, then a local parametrization φ exists represented by the graph of a function $G(\cdot)$ such that

$$\boldsymbol{\alpha} = \varphi(\boldsymbol{\alpha''}) = (\boldsymbol{\alpha''}, G(\boldsymbol{\alpha''}))^T = (\boldsymbol{\alpha''}, \boldsymbol{\alpha'})^T, \quad \boldsymbol{\alpha''} \in \mathbb{R}^d,\ \boldsymbol{\alpha'} \in \mathbb{R}^m,\ m = L - d \tag{30}$$

where $(\boldsymbol{\alpha''}, \boldsymbol{\alpha'})^T$ is a partition of $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha''}, \boldsymbol{\alpha'}$ are row vectors. A proof of this result is reported in the Appendix IX for ease of reference.

Comparing (29) and (30) it clearly results $\boldsymbol{\theta} = \boldsymbol{\alpha''}$ and $F(\cdot) = \varphi(\cdot)$. In this way the data matrix A can be partitioned accordingly

$$A = \begin{bmatrix} A'' \\ A' \end{bmatrix},\ A'' \in \mathbb{R}^{d \times N},\ A' \in \mathbb{R}^{m \times N} \tag{31}$$

where A'' and A' are the data matrices of $\boldsymbol{\alpha''}$ and $\boldsymbol{\alpha'} = G(\boldsymbol{\alpha''})$, respectively.

A. PRINCIPAL TENSOR EMBEDDING

Being $\boldsymbol{\alpha}$ a random variable, the mean vector $\mu_\alpha = (\mu_{\alpha_1}, \ldots, \mu_{\alpha_L})^T$

$$\mu_\alpha \triangleq E\{\boldsymbol{\alpha}\} = \frac{1}{N}\sum_{k=1}^{N} \alpha^{(k)} \in \mathbb{R}^{L \times 1} \tag{32}$$

and the variance vector $\sigma_\alpha = (\sigma_{\alpha_1}, \ldots, \sigma_{\alpha_L})^T$

$$\begin{aligned} \sigma_\alpha &\triangleq diag\, E\{(\boldsymbol{\alpha} - \mu_\alpha)(\boldsymbol{\alpha} - \mu_\alpha)^T\} \\ &= diag\left[\frac{1}{N}\sum_{i=1}^{N}(\alpha^{(k)} - \mu_\alpha)(\alpha^{(k)} - \mu_\alpha)^T\right] \in \mathbb{R}^{L \times 1} \end{aligned} \tag{33}$$

can be computed from the set A, so that the following cell array

$$\{\sigma_\alpha | \mathcal{U}\} = \{\sigma_{\alpha_i} | \mathcal{U}^{(i)},\ i = 1, \ldots, L\} \tag{34}$$

can be defined. If the elements $(\sigma_{\alpha_1}, \ldots, \sigma_{\alpha_L})$ of σ_α are put in decreasing order and the corresponding terms $\{\mathcal{U}^{(1)}, \ldots, \mathcal{U}^{(L)}\}$ are ordered accordingly, then a new cell array is derived

$$\{\sigma_\beta | \hat{\mathcal{U}}\} \tag{35}$$

where $\boldsymbol{\beta} = (\boldsymbol{\beta_1}, \ldots, \boldsymbol{\beta_L})^T$ is a random vector such that $\sigma_{\beta_1} \geq \sigma_{\beta_2} \geq \ldots \geq \sigma_{\beta_L}$ and a bold face character is used, following the convention previously adopted for random variables. In this way for every $\mathcal{X}^{(k)}$ it results

$$\mathcal{X}^{(k)} = \sum_{i=1}^{L} \beta_i^{(k)} \hat{\mathcal{U}}^{(i)},\ k = 1, \ldots, N \tag{36}$$

where the terms

$$\beta_i^{(k)} = \left\langle \hat{\mathcal{U}}^{(i)}, \mathcal{X}^{(k)} \right\rangle, \quad i = 1, \ldots, L,\ k = 1, \ldots, N \tag{37}$$

correspond to the cell arrays

$$\{\beta^{(k)} | \hat{\mathcal{U}}\}, \quad k = 1, \ldots, N. \tag{38}$$

It is worth to notice that when β is used in normal face it represents an observation or realization of the random variable $\boldsymbol{\beta}$. Having reordered the basis elements, the correspondence (27) becomes

$$\{B | \Omega\} = \{\beta^{(k)} |\, \mathcal{X}^{(k)},\ k = 1, \ldots, N\} \tag{39}$$

where $B = \left[\beta^{(1)}, \ldots, \beta^{(N)}\right] \in \mathbb{R}^{L \times N}$ is the new data matrix of coefficients in the basis $\hat{\mathcal{U}} = \{\hat{\mathcal{U}}^{(1)}, \ldots, \hat{\mathcal{U}}^{(L)}\}$. As a main result of this reordering, to the dataset B corresponds the new random vector

$$\boldsymbol{\beta} = (\boldsymbol{\beta''}, \boldsymbol{\beta'})^T \tag{40}$$

where the components of $\boldsymbol{\beta}$ are such that $\sigma_{\beta_1} \geq \ldots \geq \sigma_{\beta_L}$. Among the possible choices the vector $\boldsymbol{\alpha}$ can be partitioned, (40) has the following useful property. By choosing a generic index $r < L$, (28) can be rewritten as

$$\mathcal{X} = \sum_{i=1}^{r} \alpha_i \mathcal{U}^{(i)} + \sum_{i>r} \alpha_i \mathcal{U}^{(i)} = \mathcal{X}_r + \mathcal{N} \tag{41}$$

where $\mathcal{X}_r = \sum_{i=1}^{r} \alpha_i \mathcal{U}^{(i)}$. Assuming $\mathcal{X}_r$ is a good approximation of $\mathcal{X}$ and $\mu_\alpha = 0$ for the sake of notation simplicity, then the truncation error in approximating $\mathcal{X}$ with the first r components is given by the norm of residual $\mathcal{N} = \mathcal{X} - \mathcal{X}_r$, i.e.,

$$\begin{aligned} \mathcal{E}_r &= E\{\langle \mathcal{N}, \mathcal{N} \rangle\} = E\{\langle \mathcal{X} - \mathcal{X}_r, \mathcal{X} - \mathcal{X}_r \rangle\} \\ &= E\{\sum_{i>r} \boldsymbol{\alpha_i^2}\} = \sum_{i>r} \sigma_{\alpha_i}. \end{aligned} \tag{42}$$

As the values σ_{β_i} are in decreasing order, thus the partition (41) ensures the truncation error (42) is minimum when $\sigma_{\alpha_i} = \sigma_{\beta_i}$.

With reference to the new partition (40), the local parametrization of corresponding data $\{\beta^{(1)}, \ldots, \beta^{(N)}\}$ can be written as

$$\boldsymbol{\beta} = \varphi(\boldsymbol{\beta''}) = (\boldsymbol{\beta''}, G(\boldsymbol{\beta''})^T = (\boldsymbol{\beta''}, \boldsymbol{\beta'})^T,\ \boldsymbol{\beta''} \in \mathbb{R}^d \tag{43}$$

and the generic observation of dataset Ω can be modeled by

$$\mathcal{X}^{(k)} = \sum_{i=1}^{d} \beta_i''^{(k)} \hat{\mathcal{U}}^{(i)} + \sum_{i>d} G_i(\beta''^{(k)}) \hat{\mathcal{U}}^{(i)}, \ k = 1, \ldots, N$$
$$\beta_i''^{(k)} = \langle \hat{\mathcal{U}}^{(i)}, \mathcal{X}^{(k)} \rangle, \ i = 1, \ldots, d, \ k = 1, \ldots, N. \quad (44)$$

The data matrix B of $\boldsymbol{\beta}$ can be partitioned accordingly

$$B = \begin{bmatrix} B'' \\ B' \end{bmatrix}, \ B'' \in \mathbb{R}^{d \times N}, \ B' \in \mathbb{R}^{m \times N} \quad (45)$$

where B'' and B' are the data matrices of $\boldsymbol{\beta''}$ and $\boldsymbol{\beta'} = G(\boldsymbol{\beta''})$ respectively. As you can see from (44) the tensor dataset Ω of dimension $L \times N$ is represented by the data matrix B'' of dimension $d \times N$, thus a reduction of complexity in representing data from $L \times N$ to $d \times N$ is obtained with the model (44). Formally this model is equivalent to the following correspondence

$$B'' = [\beta''^{(1)}, \ldots, \beta''^{(N)}] \Longleftrightarrow \Omega = \{\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(N)}\} \quad (46)$$

which can be rewritten in a more compact form as

$$\{B'' | \, \Omega\} = \{\beta''^{(k)} | \, \mathcal{X}^{(k)}, \ k = 1, \ldots, N\}. \quad (47)$$

The values $\mathcal{X}^{(k)}$ in (44) can be interpreted as observations of a random tensor $\mathcal{X}$, thus the following stochastic model

$$\mathcal{X} = \gamma(\boldsymbol{\beta''}) = \sum_{i=1}^{d} \boldsymbol{\beta_i''} \hat{\mathcal{U}}^{(i)} + \sum_{i>d} \boldsymbol{\beta_i'}(\boldsymbol{\beta''}) \hat{\mathcal{U}}^{(i)}, \ \boldsymbol{\beta'} = G(\boldsymbol{\beta''}) \quad (48)$$

that established a one-to-one correspondence between the random tensor $\mathcal{X}$ and the random vector $\boldsymbol{\beta''}$, holds. In the context of random models for tensors, $\boldsymbol{\beta''}$ represents the vector of latent variables, that is a smaller set of variables that cannot be observed directly, and $\gamma(\boldsymbol{\beta''})$ is a local parametrization of $\mathcal{X}$. Once the new basis $\hat{\mathcal{U}}$ is obtained from $\mathcal{U}$ by the reordering procedure previously described, the hyperparameter vector $\beta''^{(k)}$ for a generic observation $\mathcal{X}^{(k)}$, can be easily derived by the inner product (44).

On the basis of previous results, a low-dimensional representation of the tensor $\mathcal{X}$ can be obtained by truncating the summation in (48) to the first r terms

$$\mathcal{X} \cong \sum_{i=1}^{r} \beta_i \hat{\mathcal{U}}^{(i)} \quad , \quad (49)$$

thus giving a truncation error

$$\mathcal{E}_r \cong \sum_{i>r} \sigma_{\beta_i} \quad (50)$$

that decreased as r increases.

As the terms in (49) correspond to the most important components in the representation of the tensor $\mathcal{X}$, this approach for tensor learning can be called principal embedded tensor (PTE) technique.

B. THE METRIC OF THE MANIFOLD $\mathcal{M}$

A relationship between the metric on the manifold $\mathcal{M}$, that is the geodesic distance of all pairs of points on the manifold, and the Euclidean metric of the corresponding points in $\mathbb{R}^d$, can be derived as follows. By taking advantage of the one-to-one property of the correspondence (48), the geodesic distance between two points $\mathcal{X}^{(i)}$ and $\mathcal{X}^{(j)}$ on the manifold can be defined as

$$d(\mathcal{X}^{(i)}, \mathcal{X}^{(j)}) = \|\beta''^{(i)} - \beta''^{(j)}\| = \|\gamma^{-1}(\mathcal{X}^{(i)}) - \gamma^{-1}(\mathcal{X}^{(j)})\| \quad (51)$$

where $\beta''^{(i)}$, $\beta''^{(j)}$ are the corresponding points in low-dimensional space $\mathbb{R}^d$ and γ^{-1} is the inverse of γ. Besides a relationship for the Euclidean distance between two points $\mathcal{X}^{(i)}$ and $\mathcal{X}^{(j)}$ on the manifold is given by

$$\|\mathcal{X}^{(i)} - \mathcal{X}^{(j)}\| = \|\gamma(\beta''^{(i)}) - \gamma(\beta''^{(j)})\|. \quad (52)$$

Using the property of differentiability of local parametrization we can apply the first-order Taylor expansion at β_0'' to represent a generic point on the manifold at β''

$$\mathcal{X} = \gamma(\beta'') = \gamma(\beta_0'') + J(\gamma)(\beta'' - \beta_0'')^T + o(\|\beta'' - \beta_0''\|) \quad (53)$$

where $J(\gamma)$ is the Jacobian of γ. Choosing $\beta''^{(j)} = \beta_0''$ and $\beta''^{(i)} = \beta''$ and substituting (53) into (52) we have

$$\|\mathcal{X}^{(i)} - \mathcal{X}^{(j)}\| = \|J(\gamma)(\beta'' - \beta_0)^T + o(\|\beta'' - \beta_0''\|)\|. \quad (54)$$

This relationship clearly shows that the geodesic distance between two points $\mathcal{X}^{(i)}, \mathcal{X}^{(j)}$, defined by (51), is different from their Euclidean distance.

VII. NONPARAMETRIC KERNEL REGRESSION (NPKR)

Assuming the ID of dataset B has been determined with one of the methods known in literature [71]–[75], the unsupervised learning of the stochastic process (s.p.) $\mathcal{X}$ that generates the data Ω, reduces to the estimation of the input-output function $G(\cdot)$ in (43). In this way the initial problem of unsupervised learning reduces to a supervised learning problem as the input data B'' of $G(\cdot)$ are known. The function $G(\cdot)$ represents a mapping (in general nonlinear) from data B'' in the low-dimensional feature space to high-dimensional data B'. The estimation of this function is a regression problem that can be solved using several different approaches.

In this context an effective method for non parametric input-output nonlinear function regression in tensor space has been recently proposed [76]. The method can be summarized as follows. Given any continuous and bounded function $f(x)$ of the n-dimensional variable $x = (x^1, \ldots, x^n)$, defined in a compact subset $I \in \mathbb{R}^n$, then some sequences $k_m(x)$, named kernel functions, exists such that the convolution

$$f_m(x) = k_m * f(x) = \int_I f(t) k_m(x - t) dt \quad (55)$$

converges uniformly to $f(x)$ on I, as $m \to \infty$. Examples of these functions are:

i) the polynomial kernel defined as

$$k_m(x) = \begin{cases} \dfrac{(1-\|x\|^2)^m}{C_m}, & \|x\| \le 1 \\ 0, & \|x\| > 1 \end{cases} \tag{56}$$

where $\|x\| = (x^T x)^{1/2}$ is the norm of X and C_m is a normalized factor given by

$$C_m = \int_I (1-\|t\|^2)^m dt \tag{57}$$

ii) the Gaussian kernel defined as

$$k_m(x) = \frac{m^n}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}m^2\|x\|^2\right). \tag{58}$$

As a consequence of property (55) we have

$$f(x) \cong f_m(x), \text{ for } m \gg 1 \tag{59}$$

which can be considered as a universal approximating relationship. The main issue of (59) is that it requires calculating the integral on the right hand side of (55). To overcome this problem, suppose we want to compute the mean value of the n-dimensional function $g(t)$, $t = (t^1, \ldots, t^n)$, in the interval I

$$E(g(t)) = \int_I g(t)p(t)dt \tag{60}$$

where t is a realization of the random variable (r.v.) $\boldsymbol{t\ \mathcal{D}\ (t^1, \ldots, t^n)}$, with probability density function (pdf) $p(t)$, and $E(\cdot)$ denotes the expected value. To numerically solve the integral in (60) a Monte Carlo integration technique [77], [78] can be derived as follows. Let us select at random N points $(t^1, \ldots, t^N)$ sampled from pdf $p(t)$, then the Monte Carlo approximation of (60) is

$$E(g(t)) \cong \frac{1}{N}\sum_{i=1}^{N} g(t_i). \tag{61}$$

By applying this approach to the convolution $g_m(x) = k_m * g(x)$, where $g(x) = f(x)p(x)$, we have

$$\begin{aligned} g_m(x) &= k_m * g(x) = E(f(t)k_m(x-t)) \\ &\cong \frac{1}{N}\sum_{i=1}^{N} f(t_i)k_m(x-t_i) \end{aligned} \tag{62}$$

and in particular for $f(x) = 1(x) = \{1 | x \in I\}$

$$1_m(x) = k_m * p(x) = E(k_m(x-t)) \cong \frac{1}{N}\sum_{i=1}^{N} k_m(x-t_i). \tag{63}$$

Combining (62) and (63) we finally get

$$f(x) \cong f_m(x) = \frac{\sum_{i=1}^{N} f(t_i)k_m(x-t_i)}{\sum_{i=1}^{N} k_m(x-t_i)}. \tag{64}$$

The function approximation (64) is a non-parametric model of function $f(x)$ as it only depends on the observations $f(t_i)$ and not on parameters to be estimated. The method described above, named nonparametric kernel regression (NPKR), can be used for the approximation of the function $G(\cdot)$ in (48), thus giving the following relationship

$$G(\beta'') \cong G_m(\beta'') = \frac{\sum_{k=1}^{N} \beta'^{(k)} k_m(\beta'' - \beta''^{(k)})}{\sum_{k=1}^{N} k_m(\beta'' - \beta''^{(k)})} \tag{65}$$

where $k_m(\cdot)$ is a given kernel function, $\beta''^{(k)}$ and $\beta'^{(k)}$ are the input and output training points respectively and β'' is a testing point, chosen from data matrix $B'' = [\beta''^{(1)}, \ldots, \beta''^{(N)}]$. It is worth to notice that the previous relationship has the same form as the well-known Nadaraya-Watson nonparametric kernel estimator [79]–[81] that was proposed for the estimation of the regression function of data $(x_1, y_1), \ldots, (x_n, y_n)$ sampled from a population having a density $f(x, y)$, thus giving a link between the two theories.

VIII. EXPERIMENTS

The experiments for the validation of the proposed tensor learning approach address two different problems, namely *data reconstruction* and *classification*.

Data reconstruction aims at reconstructing the original dataset $\Omega = \{\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(N)}\}$ by the embedded vectors $B'' = \{\beta''^{(k)},\ k = 1, \ldots, N\}$ using the model given by (48) and (65).

A pseudo-code of the algorithm used for the estimation of the model from the dataset Ω is reported in Algorithm 1.

Algorithm 1 PTE

INPUT: dataset
$\Omega = \{\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(N)}\}$, $\mathcal{X}^{(i)} \in \mathbb{R}^{I_1 \times \ldots \times I_M}$, $L = \prod_{j=1}^{M} I_j$
1. Extract L observations at random
$\Lambda = \{\hat{\mathcal{X}}^{(1)}, \ldots, \hat{\mathcal{X}}^{(L)}\}$
2. Compute a basis by Gram-Schmidt procedure
$\mathcal{U} = \{\mathcal{U}^{(1)}, \ldots, \mathcal{U}^{(L)}\}$
3. Compute the data matrix $A = [\alpha^{(1)}, \ldots, \alpha^{(N)}]$ such that
$\alpha_i^{(k)} = \langle \mathcal{U}^{(i)}, \mathcal{X}^{(k)} \rangle$, $i = 1, \ldots, L,\ k = 1, \ldots, N$
4. Determine the reordered basis $\hat{\mathcal{U}} = \{\hat{\mathcal{U}}^{(1)}, \ldots, \hat{\mathcal{U}}^{(L)}\}$ such that $\sigma_{\beta_1} \ge \sigma_{\beta_2} \ge \ldots \ge \sigma_{\beta_L}$,
$\beta_i^{(k)} = \left\langle \hat{\mathcal{U}}^{(i)}, \mathcal{X}^{(k)} \right\rangle$, $i = 1, \ldots, L,\ k = 1, \ldots, N$
5. Estimate the intrinsic dimension d of data matrix $B = [\beta^{(1)}, \ldots, \beta^{(N)}]$ and extract B'' such that
$B = \begin{bmatrix} B'' \\ B' \end{bmatrix}$, $B'' \in \mathbb{R}^{d \times N}$
6. Approximate the function $\beta' = G(\beta'')$ by NPKR, with $G_m(\beta'')$ given by (65)
OUTPUT: the model of dataset Ω
$\mathcal{X} = \gamma(\beta'') = \sum_{i=1}^{d} \beta_i'' \hat{\mathcal{U}}^{(i)} + \sum_{i>d} \beta_i'(\beta'')\hat{\mathcal{U}}^{(i)}$, $\beta' = G(\beta'')$

As far as the extraction of L observations is concerned, the following considerations are useful. A complete basis can be derived provided the number N of observations is larger than the dimensionality L of tensor $\mathcal{X}$. In this case to obtain a set of non-zero orthonormal tensors the selected L data are required to be independent. In general this assumption is naturally satisfied since each element in the database is obtained

independently from each other. Nevertheless independence can be easily proven by checking that the elements obtained by the Gram-Schmidt procedure are non-zero, as they are a linear combination of dataset. In case we have less data, i.e. $N < L$, a complete basis cannot be obtained but the method for manifold nonlinear dimensionality reduction can still be applied provided the condition $d \ll N$ is satisfied. However, in this case, the main consequence of having a reduced dataset is a large error in the estimation of data ID, as it will be discussed in the experiment VIII-B for classification.

In order to visually assess the quality of Algorithm 1, a simple experiment on a synthetic dataset was preliminary performed. To this end data $\mathcal{X} \in \mathbb{R}^3$ in a low dimensional space were generated by the following parametrized function

$$\mathcal{X} = \alpha_1''\phi^{(1)} + \alpha_2''\phi^{(2)} + G(\alpha_1'', \alpha_2'')\phi^{(3)} \tag{66}$$

where

$$\alpha' = G(\alpha_1'', \alpha_2'') = (\alpha_1'')^3 - 3\alpha_1''(\alpha_2'')^2 \tag{67}$$

and the vectors

$$\begin{aligned} \phi^{(1)} &= (1 \quad 0 \quad 0)^T, \\ \phi^{(2)} &= (0 \quad 1 \quad 0)^T, \ \phi^{(3)} = (0 \quad 0 \quad 1)^T \end{aligned} \tag{68}$$

form the canonical basis $\phi = \left[\phi^{(1)}, \phi^{(2)}, \phi^{(3)}\right] \in \mathbb{R}^{3\times3}$. (66) represents a parametrized surface in $\mathbb{R}^3$ known as Monkey Saddle whose behaviour is plotted in Fig. 2. From (66) a dataset $\Omega = \{\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(N)}\}$ was achieved by N values of $\alpha'' = (\alpha_1'', \alpha_2'')$ randomly chosen in the interval $[-1.5, 1.5] \times [-1.5, 1.5]$, thus obtaining the data matrix $A = [\alpha^{(1)}, \ldots, \alpha^{(N)}] \in \mathbb{R}^{3\times N}$, where $\alpha = (\alpha'', \alpha')^T$. Using Ω as the input dataset in Algorithm 1, the model $\mathcal{X} = \gamma(\beta'')$ has been derived. Fig. 3 depicts the surface achieved with the points sampled from the model. As you can see the model is able to reconstruct the manifold embedded in the high-dimensional space of data.

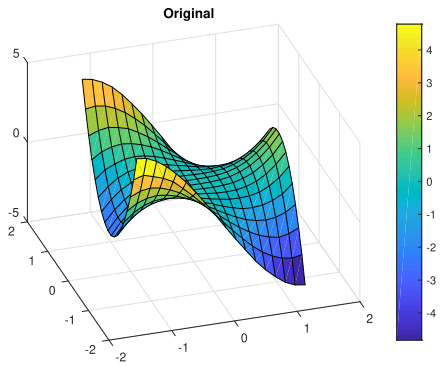


FIGURE 2. Monkey Saddle surface.

In all the experiments a Gaussian kernel was used for the regression of function $G(\cdot)$ in (48), using the NPKR method.

Classification aims at classifying the data Ω by the kNN algorithm, using the embedded vectors B'' as low-dimensional features.

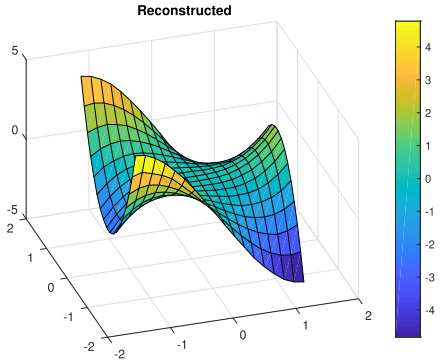


FIGURE 3. Monkey Saddle surface reconstructed.

A. DATA RECONSTRUCTION

The capability of the proposed method to model data with a reduced dimensionality, has been validated by three experiments conducted on different datasets, namely CIFAR-10, RGB-D Object Dataset, AT&T Faces Dataset.

1) EXPERIMENT ON CIFAR-10 (3D-TENSOR)

CIFAR-10 dataset [82], [83] consists of 60000, 32×32 colour images divided in 10 classes, with 6000 images per class, of which 50000 are for training and 10000 for testing.

In this experiment, we use the 50000 RGB images of the training set for image reconstruction by the model (48)-(65). For this purpose, the data has been organized in a 3D-tensor dataset $\Omega = \{\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(N)}\}$, with $N = 50000$, $\mathcal{X} \in \mathbb{R}^{32\times32\times3}$, so that the dimension of the basis is $L = 32{\cdot}32{\cdot}3 = 3072$. Once a basis $\mathcal{U}$ is derived with the Gram-Schmidt procedure, to the set Ω corresponds the data matrix $A = \left[\alpha^{(1)}, \ldots, \alpha^{(N)}\right]$ where the columns can be considered as realizations of the random vector α. For the set chosen in this experiment the estimated value of the vectors μ_α and σ_α are reported in Fig. 4.

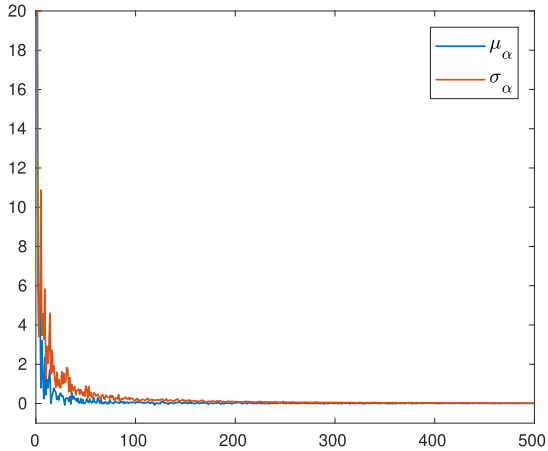


FIGURE 4. Vectors μ_α and σ_α for Experiment VIII-A1 (x-axis truncated to value 500).

To estimate the intrinsic dimension d of data matrix B we used the following relevant state-of-the-art intrin-

sic dimension (ID) estimators: Dimensionality from Angle and Norm Concentration (DANCo) and its faster variant (FastDANCo) [84], [85], Minimum Neighbor Distance - Maximum Likelihood (MiND_{ML}) and Minimum Neighbor Distance - Kullback Leibler (MiND_{KL}) [86], Maximum Likelihood Estimation (MLE) [87], Intrinsic Dimensionality Estimation of Submanifolds in $\mathcal{R}^d$ (Hein) [88]. Table 2 reports the values of intrinsic dimension as obtained with the author's Matlab implementation[1] of the above mentioned methods for ID estimation. As you can see, although the value of ID so obtained show a large spread, they are all of the same order and significantly reduce the dimensionality of tensor data, which is two-orders higher ($L = 3072$). To stress the model and prove the dimensionality reducing capability of the approach we chose the minimum value of ID.

TABLE 2. ID of the data matrix *B* achieved from CIFAR-10 dataset, as estimated with the methods DANCo, FastDANCo, MiND_{ML}, MiND_{KL}, MLE and Hein.

DANCo	FastDANCo	MiND_{ML}	MiND_{KL}	MLE	Hein
18	19	21	36	21	10

Fig. 5 shows a set of original images and the corresponding images reconstructed with the model (48)-(65) and $d = 10$. Table 3 reports the root mean squared error (RMSE) for each processed images, computed using the NPKR method for regression of function $G(\cdot)$ in (48), (here a Gaussian kernel and $m = 2$ are used in (58)), and compared with two well-known regression methods, namely Support Vector Machine (SVM) (with different kernels) and Regression Tree. Table 3 shows that the NPKR method for the regression of the function $G(\cdot)$ gives the better results, as it is able to reconstruct the data with the minimum error.

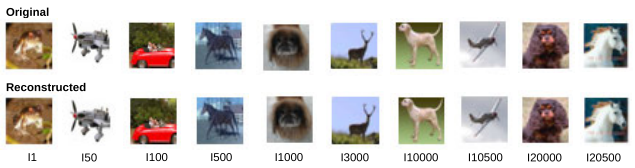


FIGURE 5. Comparison of the original images extracted from the CIFAR-10 dataset and the corresponding reconstructed images with the proposed approach.

In order to study how robust is the NPKR method with respect to hyperparameters, the sensitivity of RMSE and PSNR (peak signal-to-noise ratio) to the dimension d of β'' is reported in Fig. 6 and Fig. 7 for different values of m. The mathematical representation of the PSNR is as follows:

$$PSNR = 20\log_{10}\left(\frac{max(f)}{\sqrt{MSE}}\right) \tag{69}$$

[1]https://it.mathworks.com/matlabcentral/fileexchange/40112-intrinsic-dimensionality-estimation-techniques
http://www.stat.lsa.umich.edu/~elevina/mledim.m
https://www.ml.uni-saarland.de/code/IntDim/IntDim.htm

TABLE 3. RMSE for the reconstruction of random images extracted from CIFAR-10 dataset.

Image	Method for regression of $\beta' = G(\beta'')$				
	SVM lin	SVM poly	SVM Gau	Tree	NPKR ($m = 2$)
I1	0.1307	0.1666	0.0583	0.0632	0.0025
I50	0.1957	0.2441	0.1310	0.0918	0.0001
I100	0.2053	0.2293	0.1298	0.0947	0.0014
I500	0.1034	0.1502	0.0518	0.0732	0.0005
I1000	0.1629	0.1837	0.0809	0.0790	0.0010
I3000	0.1881	0.2323	0.1065	0.0695	0.0026
I10000	0.1458	0.1597	0.0648	0.0700	0.0006
I10500	0.0973	0.1452	0.0459	0.0663	0.0001
I20000	0.1456	0.1889	0.0760	0.0741	0.0005
I20500	0.2433	0.2798	0.1852	0.1160	0.0002

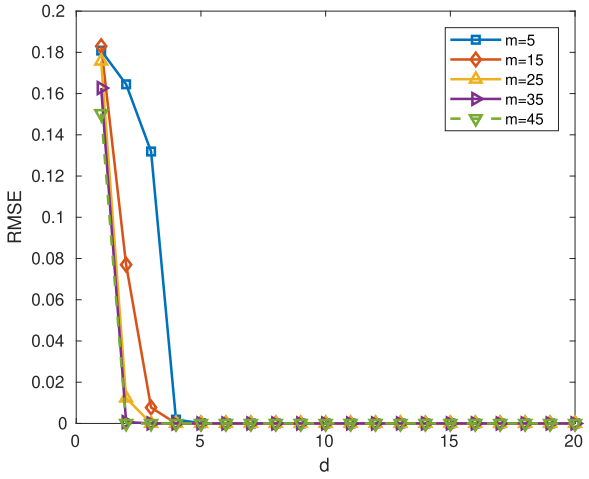


FIGURE 6. Sensitivity of RMSE to the dimension *d* of β'' for different values of *m* for Experiment VIII-A1.

where f represents the original image and MSE the mean squared error. PSNR and MSE are used to compare the squared error between the original image and the reconstructed image. There is an inverse relationship between PSNR and MSE. So a higher PSNR value indicates a higher quality of the image.

2) EXPERIMENT ON RGB-D OBJECT DATASET (4D-TENSOR)

The RGB-D Object Dataset [89] contains 300 objects divided in 51 categories. For each object the dataset provides a number of images ranging from a minimum of 506 to a maximum of 852 for a total of 207920 frames. In this experiment we used the subset *Cropped RGB and depth images with object segmentation masks* [90] that contains the cropped RGB-D frames and tightly include the object as it is spun around on a

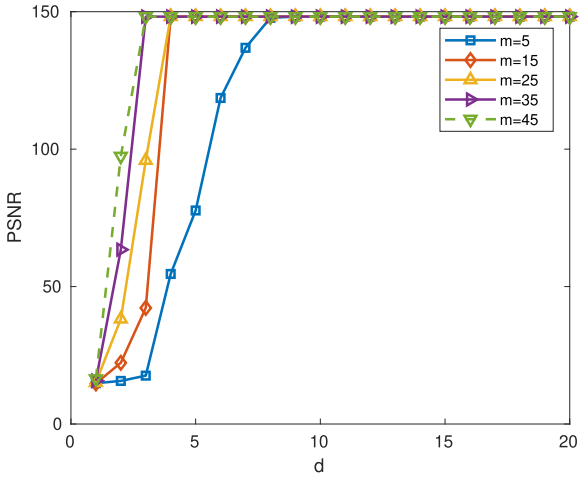


FIGURE 7. Sensitivity of PSNR to the dimension d of β'' for different values of m for Experiment VIII-A1.

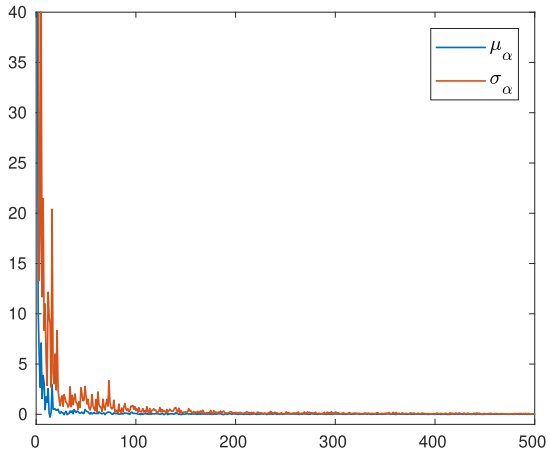


FIGURE 8. Vectors μ_α and σ_α for Experiment VIII-A2 (x-axis truncated to value 500).

turntable. We used all the 207920 images, resized into 32×32 pixel box. In particular, we divided each class in 5 frames of $32 \times 32 \times 3$ RGB images to obtain a 4D-tensor dataset $\Omega = \{\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(N)}\}$, with $N = 207920$, $\mathcal{X} \in \mathbb{R}^{32\times32\times3\times5}$, resulting in a dimension $L = 15360$ of the basis. Once a basis $\mathcal{U}$ is derived with the Gram-Schmidt procedure, to the set Ω corresponds the data matrix $A = [\alpha^{(1)}, \ldots, \alpha^{(N)}]$ where the columns can be considered as realizations of the random vector α. For the set chosen in this experiment the estimated value of the vectors μ_α and σ_α are reported in Fig. 8.

Table 4 reports the results obtained with the same ID estimators used in Experiment VIII-A1. Choosing an intrinsic dimension $d = 5$ for data matrix B'', thus a reduction of dimensionality from $L = 15360$ to $d = 5$ is obtained with the model (48). Then we applied the proposed reconstruction method to different 4D-tensors, that is 32×32 RGB videos of 5 frames each.

Figs. 9-13 show the 5 frames that composed the original videos and the corresponding reconstructed frames obtained with the proposed approach, demonstrating the validity of this approach.

TABLE 4. ID of the data matrix B achieved from the RGB-D Object Dataset, as estimated with the methods DANCo, FastDANCo, MiND_{ML}, MiND_{KL}, MLE and Hein.

DANCo	FastDANCo	MiND_{ML}	MiND_{KL}	MLE	Hein
5	5	5	9	5.85	5

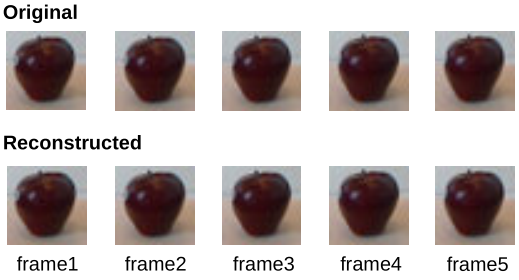


FIGURE 9. Comparison of the 5 frames (extracted from the RGB-D Object Dataset) that composed the original video V1 and the corresponding reconstructed frames obtained with the proposed approach.

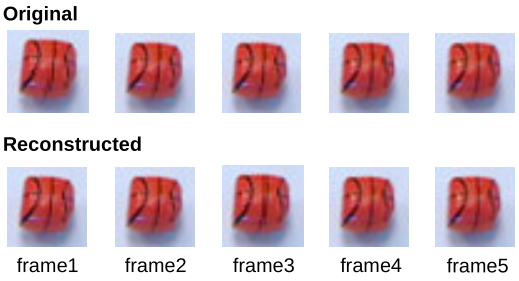


FIGURE 10. Comparison of the 5 frames (extracted from the RGB-D Object Dataset) that composed the original video V2 and the corresponding reconstructed frames obtained with the proposed approach.

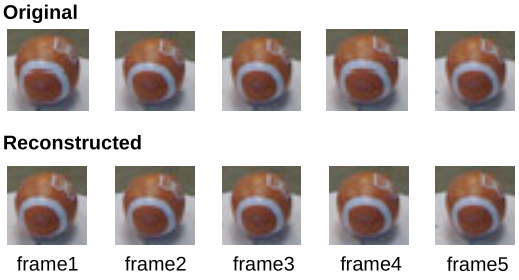


FIGURE 11. Comparison of the 5 frames (extracted from the RGB-D Object Dataset) that composed the original video V3 and the corresponding reconstructed frames obtained with the proposed approach.

Table 5 reports the RMSE for each processed 4D-tensors, obtained using NPKR method with Gaussian kernel and $m = 5$ in (58), showing that the method is able to reconstruct the data with a very low error.

In this case, as well, to assess the robustness of the PTE method with respect to hyperparameters, the sensitivity of

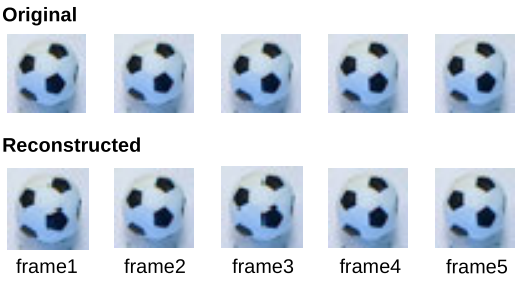


FIGURE 12. Comparison of the 5 frames (extracted from the RGB-D Object Dataset) that composed the original video V4 and the corresponding reconstructed frames obtained with the proposed approach.

FIGURE 13. Comparison of the 5 frames (extracted from the RGB-D Object Dataset) that composed the original video V5 and the corresponding reconstructed frames obtained with the proposed approach.

TABLE 5. RMSE for the reconstruction of random sequences of images extracted from RGB-D Object Dataset.

Video	PTE ($m = 5$)
V1	0.0022
V2	7.6505e-05
V3	5.2188e-04
V4	2.4069e-06
V5	5.6387e-09

RMSE and PSNR to the dimension d of β'' is reported in Fig. 14 and Fig. 15 for different values of m.

3) EXPERIMENT ON AT&T FACES DATASET (2D-TENSOR)

The AT&T Faces Dataset [91], [92] contains 10 different images for each of 40 distinct peoples. The size of each image is 92×112, with 256 gray levels per pixel.

In this experiment, we applied the proposed approach on the original and occluded images achieved from AT&T Faces Dataset, with the same procedure described in [61]. Here, 20 distinct persons are selected and each face image is resized into 56×46 format. In addition to original face data, we also test our method on the partially occluded face data. Here, 20% images were selected randomly and corrupted manually for each person class, and the size of corruption is 11×10. To apply the proposed method, the data has been organized in a 2D-tensor dataset $\Omega = \{\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(N)}\}$, with

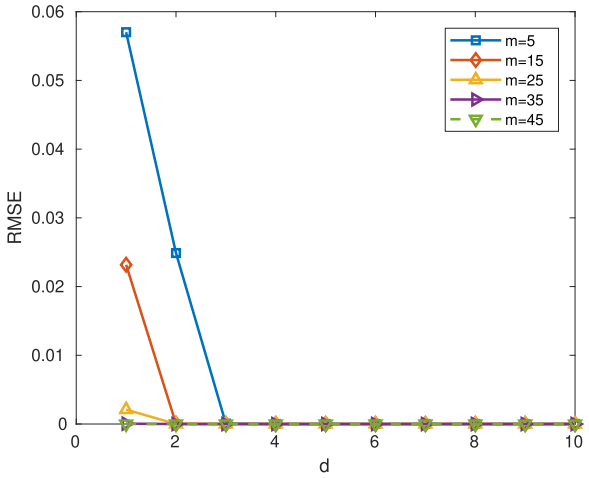


FIGURE 14. Sensitivity of RMSE to the dimension d of β'' for different values of m for Experiment VIII-A2.

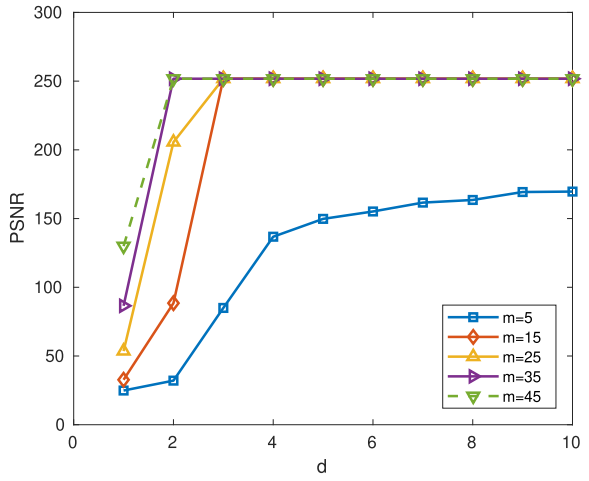


FIGURE 15. Sensitivity of PSNR to the dimension d of β'' for different values of m for Experiment VIII-A2.

$N = 200$, $\mathcal{X} \in \mathbb{R}^{56\times46}$ resulting in a dimension of the basis of $L = 2576$. Once a basis $\mathcal{U}$ is derived with the Gram-Schmidt procedure, to the set Ω corresponds the data matrix $A = [\alpha^{(1)}, \ldots, \alpha^{(N)}]$ where the columns can be considered as realizations of the random vector α. For the set chosen in this experiment the estimated value of the vectors μ_α and σ_α are reported in Fig. 16.

On the basis of the estimated ID values reported in Table 6, we select an optimal intrinsic dimension $d = 6$ for the reconstruction of this dataset.

TABLE 6. ID of the data matrix B achieved from the AT&T dataset, as estimated with the methods DANCo, FastDANCo, MiND_{ML}, MiND_{KL}, MLE and Hein.

DANCo	FastDANCo	MiND_{ML}	MiND_{KL}	MLE	Hein
6	6	5	7	6	5

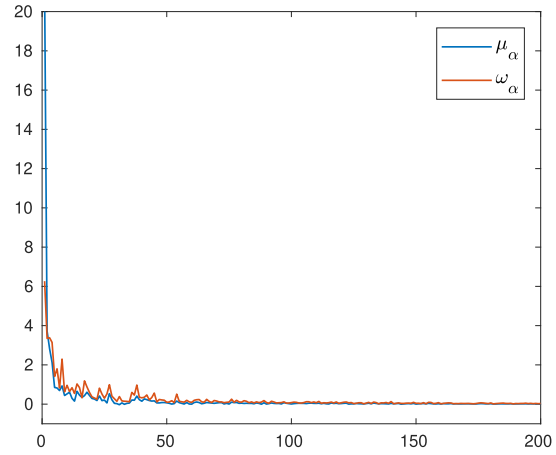


FIGURE 16. Vectors μ_α and σ_α for Experiment VIII-A3 (x-axis truncated to value 200).

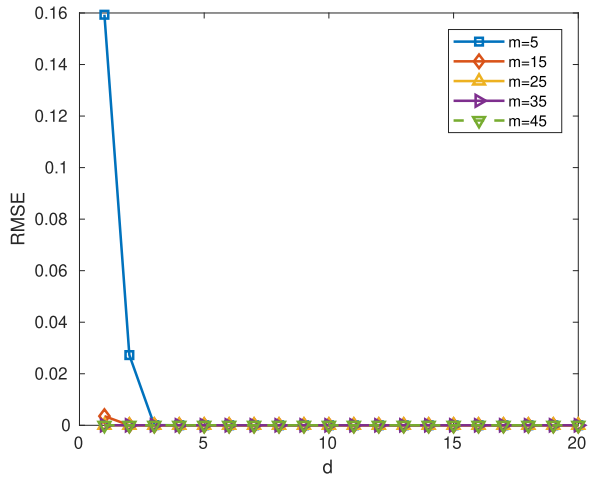


FIGURE 17. Sensitivity of RMSE to the dimension d of β'' for different values of m for Experiment VIII-A3.

Also in this case to assess the robustness of the PTE method with respect to hyperparameters, the sensitivity of RMSE and PSNR to the dimension d of β'' is reported in Fig. 17 and Fig. 18 for different values of m.

Fig. 19 and Fig. 20 show a set of original images and the corresponding reconstructed images with the model (48) - (65), demonstrating the validity of this approach. Furthermore, Fig. 21 and Fig. 22 report a set of original partially occluded images and the corresponding reconstructed images showing the good noise tolerance of the method. Table 7 reports the reconstruction residuals for different reconstruction methods. The methods used to evaluate the PTE method are: PCA-based methods (PCA [27], gLPCA [66]) and tensor decomposition methods (TD [37], GLTD [61]). In this table the average residual for reconstructed tensor $\mathrm{Res}(\mathcal{X})$ has been defined as:

$$\mathrm{Res}(\mathcal{X}) = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\|\mathcal{X}_0^{(k)} - \mathcal{X}^{(k)}\|^2}, \qquad (70)$$

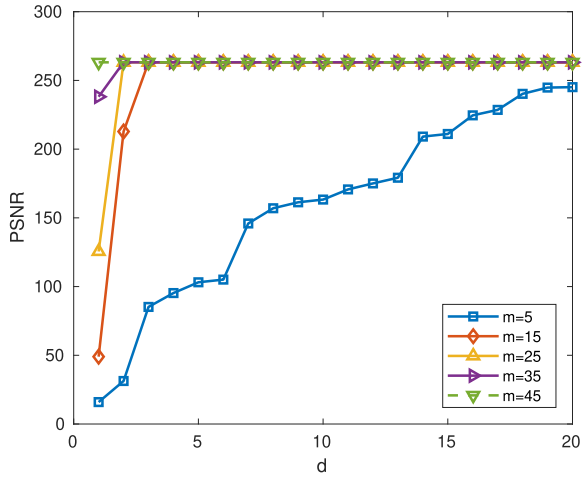


FIGURE 18. Sensitivity of PSNR to the dimension d of β'' for different values of m for Experiment VIII-A3.

FIGURE 19. Comparison of the original images extracted from the AT&T dataset and the corresponding reconstructed images with the proposed approach, on subject 8.

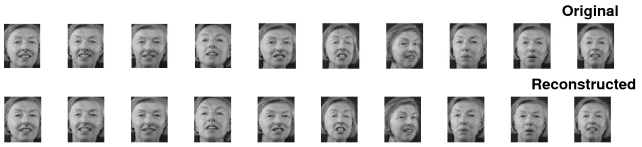


FIGURE 20. Comparison of the original images extracted from the AT&T dataset and the corresponding reconstructed images with the proposed approach, on subject 12.

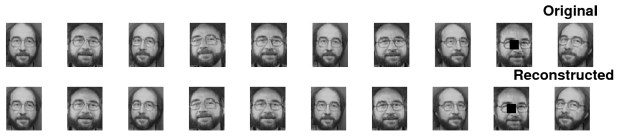


FIGURE 21. Comparison of the partially occluded original images extracted from the AT&T dataset and the corresponding reconstructed images with the proposed approach, on subject 8.

where $\mathcal{X}_0 = (\mathcal{X}_0^{(1)}, \cdots, \mathcal{X}_0^{(N)})$ represents the original N images and $\mathcal{X} = (\mathcal{X}^{(1)}, \cdots, \mathcal{X}^{(N)})$ the reconstructed images. Meanwhile, for the occluded image data, the same equation (70) defines the average noise-free residual (NF-Res), being $\mathcal{X}_0$ in this case the set of original non occluded signals.

B. CLASSIFICATION

1) EXPERIMENT ON CLASSIFICATION OF 2-ORDER TENSORS

This experiment was addressed to the classification of data collected from the following datasets.

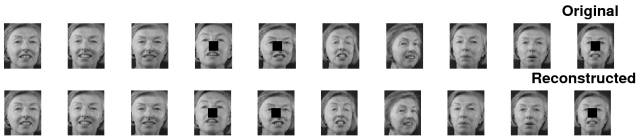


FIGURE 22. Comparison of the partially occluded original images extracted from the AT&T dataset and the corresponding reconstructed images with the proposed approach, on subject 12.

TABLE 7. Residual of different methods on both original AT&T face and AT&T noisy face datasets, respectively.

Dataset	Metric	Method				
		PCA	gLPCA	TD	GLTD	PTE
AT&T	Res	0.141	0.167	0.120	0.145	**2.000e-06**
AT&T-noisy	NF-Res	0.155	0.184	0.157	0.164	**6.000e-06**

- *AT&T Faces dataset*: as described in the experiment of Section VIII-A3.
- *MNIST dataset*: is consisted of 8-bit gray-scale images of digits from "0" to "9". There are about 6000 examples for each class [93]. Each image is centered on a 28×28 grid. In our experiments, we randomly choose 50 images for each class.
- *COIL-20 dataset*: contains 20 objects [94]. Each object has 72 images. The size of each image is 32×32 pixels, with 256 gray levels per pixel. We use the first 32 images for each object in our experiments.

It is worth to notice that for all such datasets the number of images in each class is far less the dimension of the corresponding basis. The main consequence of this mismatch is a large error in the estimation of intrinsic dimension of data, so that all the methods for ID estimation fails. However, although the intrinsic dimension of data cannot be determined with a certain degree of reliability, the model given by (48) - (65) continuous to be valid with an uncertainty on the dimension d. As a consequence, in all the experiments the dimension d was empirically determined, to obtain a good modeling of data, instead of using the method for ID estimation.

We perform semisupervised learning on different datasets, by training the classifier on the labeled data (20% of dataset) and use the rest as unlabeled data (80% of dataset). In particular 20% of data points for each class were randomly selected as labeled data, and the rest was used as unlabeled data. The classifier was trained on the labeled data and the class labels were predicted on the unlabeled data.

We performed classification using k-nearest neighbor (kNN) algorithm [28], and compared the results obtained with features extracted by our model (matrix B'') and several other tensor learning methods, including PCA-based methods (PCA [27], gLPCA [66]), tensor decomposition methods (TD [37], GLTD [61]). Although other valuable methods for classification exist, we choose to use kNN algorithm since it is common in the literature of manifold learning [60], [62], thus making comparison with other tensor learning techniques easier.

Table 8 reports the results for the classification experiment as achieved by the methods used in [61] and PTE method. Also in this case the intrinsic dimension d was empirically determined. As you can see, our method outperforms all the other methods.

TABLE 8. Comparison of classification results of 2-order tensors datasets using kNN classification algorithm (train set = 20%, test set = 80%). The best results are marked by bold font.

Dataset	Method Accuracy [%]					
	Original	PCA	gLPCA	TD	GLTD	PTE
AT&T	70.60	72.19	82.61	70.69	82.79	$\underline{d = 30}$ **87.25**
MNIST	72.86	72.06	74.47	70.38	76.53	$\underline{d = 50}$ **78.80**
COIL-20	81.67	84.05	86.27	81.38	87.35	$\underline{d = 50}$ **92.07**

2) EXPERIMENT ON CLASSIFICATION OF 3-ORDER TENSORS

In order to compare the proposed method with some other tensor-based learning methods, a last experiment was performed on the following datasets.

- *Weizmann Action Database*: an high-order dataset commonly used for human action recognition. The database includes 90 videos coming from 10 categories of actions - a) bending (bend), b) jacking (jack), c) jumping (jump), d) jumping in places (pjump), e) running (run), f) galloping sideways (side), g) skipping (skip), h) walking (walk), i) single-hand waving (wave1), g) both-hands waving (wave2) - which were performed by nine subjects [95], [96]. A tensor samples of size $32 \times 24 \times 10$, represented in a spatio-temporal domain, is formed by 10 successive frames of each action, each of which was normalized to the size 32×24 pixels.
- *Cambridge Hand Gesture Database*: consists of 900 image sequences of nine gesture classes, which are defined by three primitive hand shapes and three primitive motions [97], [98]. Each class contains 100 image sequences (5 different illuminations $\times$ 10 arbitrary motions $\times$ 2 subjects). The procedure to format data is the same as in the Weizmann action database.

In these experiments, we randomly selected six action tensors of each category for training and the remaining tensors were used for testing. The experiments were independently performed 10 times using the kNN algorithm with Euclidean distance for classification, following the procedure in [58].

TABLE 9. Comparison of classification results of 3-order tensors datasets using kNN classification algorithm. The best results are marked by bold font.

Dataset	Method Accuracy [%]						
	Original	MPCA	TLPP	TNPE	OTNPE	STA	PTE
Weizmann Action Database	70.03	70.14	76.33	75.62	77.56	79.33	$d=50$ **82.72**
Cambridge Hand Gesture Database	75.11	79.61	80.18	81.07	76.24	82.74	$d=30$ **83.33**

The methods used to evaluate the proposed PTE are: the baseline method (nearest classifier on the original data), multilinear PCA (MPCA) [31], tensor locality preserving projection (TLPP) and tensor neighborhood preserving embedded (TNPE) [67], orthogonal tensor neighborhood preserving embedded (OTNPE) [68], sparse tensor alignment (STA) [58].

Table 9 reports the results for the classification experiment as achieved by the aforementioned methods and PTE method. Also in this case the intrinsic dimension d was empirically determined. As you can see the proposed PTE outperforms the other methods in terms of recognition rate.

IX. CONCLUSION

In this article a nonlinear, explicit model of tensor data that depends on a reduced set of latent variables is derived. The main steps required for the estimation of the model from data are:

i) compute a basis by a Gram-Schmidt procedure;
ii) reorder the basis in such a way the variances of coefficients are in decreasing order;
iii) estimate the intrinsic dimension d of data;
iv) define a data parametrization of dimension d;
v) approximate the nonlinearity in the parametrization by a regression model.

The capability of the proposed approach for data reconstruction has been validated by performing several experiments on datasets with tensors of different orders. In these experiments several methods for regression, i.e. SVM with different kernels and NPKR method, have been adopted. In all cases the proposed tensor learning approach gives good performance for data reconstruction, nevertheless the PTE method is able to reconstruct data with minimum error. Additionally the proposed tensor learning approach has proven to be effective for classification problem, using data of reduced dimensionality. To show this ability, classification on several different datasets has been performed. The comparison of the results obtained with feature extracted by the proposed approach and state-of-the-art tensor learning methods (PCA, gLPCA, TD, GLTD, MPCA, TLPP, TNPE, OTNPE, STA), shows the effectiveness of the suggested model.

APPENDIX
LOCAL PARAMETRIZATION OF DATA EMBEDDED IN A MANIFOLD

Assume all the values of α embedded in the manifold $\mathcal{M}$ are described by the following parametrized surface in $\mathbb{R}^L$

$$\boldsymbol{\alpha} = F(\boldsymbol{\theta}), \quad \boldsymbol{\theta} \in U \subset \mathbb{R}^d, \, \boldsymbol{\alpha} \in \mathbb{R}^L, \, d < L. \tag{71}$$

This means that a bijective and differentiable function $f(\boldsymbol{x})$ defined on a subset $V = U \times \mathbb{R}^m \subset \mathbb{R}^L$, given by

$$f(\boldsymbol{x}) = f(\boldsymbol{\theta}, \boldsymbol{t}) = F(\boldsymbol{\theta}) + \begin{pmatrix} 0 \\ \boldsymbol{t} \end{pmatrix}, \, f : U \times \mathbb{R}^m \to \mathbb{R}^L \tag{72}$$

exists, where $\boldsymbol{x} = (\boldsymbol{\theta}, \boldsymbol{t})^T \in V, \quad \boldsymbol{t} \in \mathbb{R}^m$, $m = L - d$. This ensures that a one-to-one correspondence is established between a point $\boldsymbol{x} = (\boldsymbol{\theta}, 0)^T$ in V and a point $\boldsymbol{\alpha}$ in the manifold $\mathcal{M}$. As a consequence f is invertible and unique. Rearranging (72) according to the dimension d of $\boldsymbol{\theta}$ we have

$$f(\boldsymbol{x}) = \begin{pmatrix} F''(\boldsymbol{\theta}) \\ F'(\boldsymbol{\theta}) + \boldsymbol{t} \end{pmatrix}, \; F'' \in \mathbb{R}^d \tag{73}$$

and from differentiability of f one gets

$$J(f) = (J_{\boldsymbol{\theta}}(f), J_{\boldsymbol{t}}(f)) = \begin{pmatrix} J(F'') & 0 \\ J(F') & I_{mm} \end{pmatrix} \tag{74}$$

where I_{mm} is an $(m \times m)$ diagonal identity matrix and $J(f)$ is the Jacobian of f. In order that $f(\boldsymbol{x})$ to be invertible, as it is a bijective mapping, the condition $\det J(f) \neq 0$ on Jacobian must be satisfied. As a consequence from (74) we have $\det J(F'') \neq 0$, meaning that the function $F''(\boldsymbol{\theta}) = \boldsymbol{\alpha''}$ is invertible. Thus the inverse F''^{-1} of F'' exists such that

$$\boldsymbol{\theta} = F''^{-1}(\boldsymbol{\alpha''}). \tag{75}$$

Combining (71) and (75) it results

$$\boldsymbol{\alpha} = \begin{pmatrix} \boldsymbol{\alpha''} \\ F'\left(F''^{-1}(\boldsymbol{\alpha''})\right) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\alpha''} \\ G(\boldsymbol{\alpha''}) \end{pmatrix} \tag{76}$$

where $G(\cdot) = F'\left(F''^{-1}(\cdot)\right)$, and this proves (30).

REFERENCES

[1] J. Zhang, C. Xu, P. Jing, C. Zhang, and Y. Su, "A tensor-driven temporal correlation model for video sequence classification," *IEEE Signal Process. Lett.*, vol. 23, no. 9, pp. 1246–1249, Sep. 2016.
[2] X. Gao, X. Li, J. Feng, and D. Tao, "Shot-based video retrieval with optical flow tensor and HMMs," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 140–147, Jan. 2009.
[3] D. Xu, S. Yan, L. Zhang, S. Lin, H.-J. Zhang, and T. S. Huang, "Reconstruction and recognition of tensor-based objects with concurrent subspaces analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 1, pp. 36–47, Jan. 2008.
[4] K. Li, J. Yang, and J. Jiang, "Nonrigid structure from motion via sparse representation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1401–1413, Aug. 2015.
[5] Y. Li and A. Ngom, "Non-negative matrix and tensor factorization based classification of clinical microarray gene expression data," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2010, pp. 438–443.

[6] O. Alter and G. H. Golub, "Reconstructing the pathways of a cellular system from genome-scale signals by using matrix and tensor computations," *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 49, pp. 17559–17564, 2005.

[7] A. Lay-Ekuakille, P. Vergallo, D. Stefano, A. Massaro, A. Trabacca, M. Cacciola, D. Labate, F. C. Morabito, and R. Morello, "Diffusion tensor imaging measurements for neuro-detection," in *Proc. IEEE Int. Symp. Med. Meas. Appl.*, May 2012, pp. 1–4.

[8] T.-K. Kim, S.-F. Wong, and R. Cipolla, "Tensor canonical correlation analysis for action classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[9] P. Sankaranarayanan, T. E. Schomay, K. A. Aiello, and O. Alter, "Tensor GSVD of patient- and platform-matched tumor and normal DNA copy-number profiles uncovers chromosome arm-wide patterns of tumor-exclusive platform-consistent alterations encoding for cell transformation and predicting ovarian cancer survival," *PLoS ONE*, vol. 10, no. 4, pp. 1–21, Apr. 2015.

[10] M. T. Bahadori, Q. R. Yu, and Y. Liu, "Fast multivariate spatio-temporal analysis via low rank tensor learning," in *Proc. Adv. Neural Inf. Process. Syst. 27*. Red Hook, NY, USA: Curran Associates, 2014, pp. 3491–3499.

[11] H. Zhou, L. Li, and H. Zhu, "Tensor regression with applications in neuroimaging data analysis," *J. Amer. Stat. Assoc.*, vol. 108, no. 502, pp. 540–552, Jun. 2013.

[12] D. Kaur, G. S. Aujla, N. Kumar, A. Y. Zomaya, C. Perera, and R. Ranjan, "Tensor-based big data management scheme for dimensionality reduction problem in smart grid systems: SDN perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 10, pp. 1985–1998, Oct. 2018.

[13] Q. Song, H. Ge, J. Caverlee, and X. Hu, "Tensor completion algorithms in big data analytics," *ACM Trans. Knowl. Discovery from Data*, vol. 13, no. 1, pp. 1–48, Jan. 2019.

[14] Y. Han, Y. Yang, F. Wu, and R. Hong, "Compact and discriminative descriptor inference using multi-cues," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5114–5126, Dec. 2015.

[15] Y. Han, Y. Yang, Y. Yan, Z. Ma, N. Sebe, and X. Zhou, "Semisupervised feature selection via spline regression for video semantic recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 252–264, Feb. 2015.

[16] Y. Yan, E. Ricci, R. Subramanian, G. Liu, and N. Sebe, "Multitask linear discriminant analysis for view invariant action recognition," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5599–5611, Dec. 2014.

[17] Y. Zhang, J. Ren, and J. Jiang, "Combining MLC and SVM classifiers for learning based decision making: Analysis and evaluations," *Comput. Intell. Neurosci.*, vol. 2015, p. 44, Jan. 2015.

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

[19] J. Zhang and J. Jiang, "Decomposition-based tensor learning regression for improved classification of multimedia," *J. Vis. Commun. Image Represent.*, vol. 41, pp. 260–271, Nov. 2016.

[20] W. K. Wong, Z. Lai, Y. Xu, J. Wen, and C. P. Ho, "Joint tensor feature analysis for visual object recognition," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2425–2436, Nov. 2015.

[21] S. Van Eyndhoven, M. Bousse, B. Hunyadi, L. De Lathauwer, and S. Van Huffel, "Single-channel EEG classification by multi-channel tensor subspace learning and regression," in *Proc. IEEE 28th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Sep. 2018, pp. 1–6.

[22] Q. Zheng, Y. Wang, and P. A. Heng, "Multitask feature learning meets robust tensor decomposition for EEG classification," *IEEE Trans. Cybern.*, early access, Oct. 31, 2019, doi: 10.1109/TCYB.2019.2946914.

[23] Z. Chen, K. Batselier, J. A. K. Suykens, and N. Wong, "Parallelized tensor train learning of polynomial classifiers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4621–4632, Oct. 2018.

[24] Y. Liu, F. Shang, W. Fan, J. Cheng, and H. Cheng, "Generalized higher order orthogonal iteration for tensor learning and decomposition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2551–2563, Dec. 2016.

[25] K. Wimalawarne, R. Tomioka, and M. Sugiyama, "Theoretical and experimental analyses of tensor-based regression and classification," *Neural Comput.*, vol. 28, no. 4, pp. 686–715, Apr. 2016.

[26] D. Tao, X. Li, X. Wu, W. Hu, and S. J. Maybank, "Supervised tensor learning," *Knowl. Inf. Syst.*, vol. 13, no. 1, pp. 1–42, 2007.

[27] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Cham, Switzerland: Springer, 2002.

[28] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2012.

[29] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, Jan. 2004.

[30] J. Ye, "Generalized low rank approximations of matrices," *Mach. Learn.*, vol. 61, nos. 1–3, pp. 167–191, Nov. 2005.

[31] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 18–39, Jan. 2008.

[32] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear principal component analysis for unsupervised multilinear subspace learning," *IEEE Trans. Neural Netw.*, vol. 20, no. 11, pp. 1820–1836, Nov. 2009.

[33] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *J. Amer. Stat. Assoc.*, vol. 97, no. 458, pp. 611–631, Jun. 2002.

[34] J. Yang, D. Zhang, X. Yong, and J.-Y. Yang, "Two-dimensional discriminant transform for face recognition," *Pattern Recognit.*, vol. 38, no. 7, pp. 1125–1129, Jul. 2005.

[35] M. Li and B. Yuan, "2D-LDA: A statistical linear discriminant analysis for image matrix," *Pattern Recognit. Lett.*, vol. 26, no. 5, pp. 527–532, Apr. 2005.

[36] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 212–220, Jan. 2007.

[37] T. G. Kolda, "Multilinear operators for higher-order decompositions," Sandia Nat. Laboratories, Albuquerque, NM, USA, Tech. Rep. SAND2006-2081, 2006.

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

[39] S. Rabanser, O. Shchur, and S. Günnemann, "Introduction to tensor decompositions and their applications in machine learning," *CoRR*, vol. abs/1711.10781, pp. 1–13, Nov. 2017.

[40] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, Jul. 2017.

[41] J. Zhang, H. Huang, and J. Wang, "Manifold learning for visualizing and analyzing high-dimensional data," *IEEE Intell. Syst.*, vol. 25, no. 4, pp. 54–61, Jul. 2010.

[42] P. Zhang, H. He, and L. Gao, "A nonlinear and explicit framework of supervised manifold-feature extraction for hyperspectral image classification," *Neurocomputing*, vol. 337, pp. 315–324, Apr. 2019.

[43] A. Jamshidi, M. Kirby, and D. Broomhead, "Geometric manifold learning," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 69–76, Mar. 2011.

[44] H. S. Seung, "COGNITION: The manifold ways of perception," *Science*, vol. 290, no. 5500, pp. 2268–2269, Dec. 2000.

[45] Z. Zhang, J. Wang, and H. Zha, "Adaptive manifold learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 253–265, Feb. 2012.

[46] H. Qiao, P. Zhang, D. Wang, and B. Zhang, "An explicit nonlinear mapping for manifold learning," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 51–63, Feb. 2013.

[47] R. Talmon, S. Mallat, H. Zaveri, and R. R. Coifman, "Manifold learning for latent variable inference in dynamical systems," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 3843–3856, Aug. 2015.

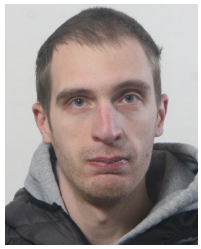
[48] Y. Bai, Z. Sun, B. Zeng, J. Long, L. Li, J. V. de Oliveira, and C. Li, "A comparison of dimension reduction techniques for support vector machine modeling of multi-parameter manufacturing quality prediction," *J. Intell. Manuf.*, vol. 30, no. 5, pp. 2245–2256, Jun. 2019.

[49] K. Gajamannage, R. Paffenroth, and E. M. Bollt, "A nonlinear dimensionality reduction framework using smooth geodesics," *Pattern Recognit.*, vol. 87, pp. 226–236, Mar. 2019.

[50] C. Turchetti and L. Falaschetti, "A manifold learning approach to dimensionality reduction for modeling data," *Inf. Sci.*, vol. 491, pp. 16–29, Jul. 2019.
[51] S. T. Roweis, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
[52] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimension reduction via local tangent space alignment," *SIAM J. Sci. Comput.*, vol. 26, no. 1, pp. 313–338, 2002.
[53] L. Yang, "Alignment of overlapping locally scaled patches for multidimensional scaling and dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 438–450, Mar. 2008.
[54] J. B. Tenenbaum, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.
[55] E. Acar and B. Yener, "Unsupervised multiway data analysis: A literature survey," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 1, pp. 6–20, Jan. 2009.
[56] T. G. Kolda, "Orthogonal tensor decompositions," *SIAM J. Matrix Anal. Appl.*, vol. 23, no. 1, pp. 243–255, Jan. 2001.
[57] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.
[58] Z. Lai, W. K. Wong, Y. Xu, C. Zhao, and M. Sun, "Sparse alignment for robust tensor learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1779–1792, Oct. 2014.
[59] F. Ju, Y. Sun, J. Gao, Y. Hu, and B. Yin, "Vectorial dimension reduction for tensors based on Bayesian inference," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4579–4592, Oct. 2018.
[60] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in *Proc. Adv. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, 2006, pp. 499–506.
[61] B. Jiang, C. Ding, J. Tang, and B. Luo, "Image representation and learning with graph-Laplacian tucker tensor decomposition," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1417–1426, Apr. 2019.
[62] C. Liu, J. Zhou, K. He, Y. Zhu, D. Wang, and J. Xia, "Supervised locally linear embedding in tensor space," in *Proc. 3rd Int. Symp. Intell. Inf. Technol. Appl.*, vol. 3, 2009, pp. 31–34.
[63] C. Chen, J. Zhang, and R. Fleischer, "Distance approximating dimension reduction of Riemannian manifolds," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 40, no. 1, pp. 208–217, Feb. 2010.
[64] C. Jia and Y. Fu, "Low-rank tensor subspace learning for RGB-D action recognition," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4641–4652, Oct. 2016.
[65] K. Fukunaga and D. R. Olsen, "An algorithm for finding intrinsic dimensionality of data," *IEEE Trans. Comput.*, vols. C–20, no. 2, pp. 176–183, Feb. 1971.
[66] B. Jiang, C. Ding, B. Luo, and J. Tang, "Graph-laplacian PCA: Closed-form solution and robustness," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3492–3498.
[67] G. Dai and D.-Y. Yeung, "Tensor embedding methods," in *Proc. AAAI*, vol. 6, 2006, pp. 330–335.
[68] S. Liu and Q. Ruan, "Orthogonal tensor neighborhood preserving embedding for facial expression recognition," *Pattern Recognit.*, vol. 44, no. 7, pp. 1497–1513, Jul. 2011.
[69] A. S. Mishchenko and A. T. Fomenko, *A Course of Differential Geometry and Topology*. Moscow, Russia: Mir Publishers, 1988.
[70] W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*, vol. 120. New York, NY, USA: Academic, 1986.
[71] F. Camastra, "Data dimensionality estimation methods: A survey," *Pattern Recognit.*, vol. 36, no. 12, pp. 2945–2954, Dec. 2003.
[72] G. V. Trunk, "Stastical estimation of the intrinsic dimensionality of a noisy signal collection," *IEEE Trans. Comput.*, vol. C-25, no. 2, pp. 165–171, Feb. 1976.
[73] P. J. Verveer and R. P. W. Duin, "An evaluation of intrinsic dimensionality estimators," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 1, pp. 81–86, Jan. 1995.
[74] C. Kuan Chen and H. C. Andrews, "Nonlinear intrinsic dimensionality computations," *IEEE Trans. Comput.*, vol. C-23, no. 2, pp. 178–184, Feb. 1974.
[75] B. Kégl, "Intrinsic dimension estimation using packing numbers," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*. Cambridge, MA, USA: MIT Press, 2002, pp. 697–704.
[76] C. Turchetti and L. Falaschetti, "A GPU parallel algorithm for non parametric tensor learning," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, Dec. 2018, pp. 286–290.
[77] R. E. Caflisch, "Monte Carlo and quasi-Monte Carlo methods," *Acta Numerica*, vol. 7, p. 1–49, Jan. 1998.
[78] J. Dick, F. Y. Kuo, and I. H. Sloan, "High-dimensional integration: The quasi-Monte Carlo way," *Acta Numerica*, vol. 22, pp. 133–288, May 2013.
[79] É. A. Nadaraya, "On non-parametric estimates of density functions and regression curves," *Theory Probab. Appl.*, vol. 10, no. 1, pp. 186–190, Jan. 1965.
[80] J. Fan, "Design-adaptive nonparametric regression," *J. Amer. Stat. Assoc.*, vol. 87, no. 420, pp. 998–1004, Dec. 1992.
[81] J. Fan and Q. Yao, *Nonlinear Time Series: Nonparametric and Parametric Methods*. New York, NY, USA: Springer, 2003.
[82] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. TR-2009, 2009.
[83] *The CIFAR-10 Dataset*. Accessed: Dec. 12, 2020. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html
[84] C. Ceruti, S. Bassis, A. Rozza, G. Lombardi, E. Casiraghi, and P. Campadelli, "DANCo: An intrinsic dimensionality estimator exploiting angle and norm concentration," *Pattern Recognit.*, vol. 47, no. 8, pp. 2569–2581, Aug. 2014.
[85] C. Ceruti, S. Bassis, A. Rozza, G. Lombardi, E. Casiraghi, and P. Campadelli, "DANCo: Dimensionality from angle and norm concentration," *CoRR*, vol. abs/1206.3881, pp. 1–16, Jun. 2012.
[86] G. Lombardi, A. Rozza, C. Ceruti, E. Casiraghi, and P. Campadelli, "Minimum neighbor distance estimators of intrinsic dimension," in *Proc. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2011, pp. 374–389.
[87] E. Levina and P. J. Bickel, "Maximum likelihood estimation of intrinsic dimension," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*. Cambridge, MA, USA: MIT Press, 2005, pp. 777–784.
[88] M. Hein and J.-Y. Audibert, "Intrinsic dimensionality estimation of submanifolds in RD," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, 2005, pp. 289–296.
[89] *The RGB-D Object Dataset*. Accessed: Dec. 12, 2020. [Online]. Available: http://rgbd-dataset.cs.washington.edu/dataset/
[90] *The RGB-D Object Dataset-Cropped RGB-D*. Accessed: Dec. 12, 2020. [Online]. Available: http://rgbd-dataset.cs.washington.edu/dataset/rgbd-dataset/
[91] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Dec. 1994, pp. 138–142.
[92] *AT&T—The Database of Faces*. Accessed: Dec. 12, 2020. [Online]. Available: http://cam-orl.co.uk/facedatabase.html
[93] Y. LeCun, C. Cortes, and C. J. Burges. *The MNIST Database of Handwritten Digits*. Accessed: Dec. 12, 2020. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[94] S. A. Nene, S. K. Nayar, and H. Murase. *Columbia University Image Library (COIL-20)*. Accessed: Dec. 12, 2020. [Online]. Available: http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php
[95] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.
[96] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. *Actions as Space-Time Shapes*. Accessed: Dec. 12, 2020. [Online]. Available: http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html
[97] T.-K. Kim, S.-F. Wong, and R. Cipolla, "Tensor canonical correlation analysis for action classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[98] T. Kim. *Cambridge Hand Gesture Data Set*. Accessed: Dec. 12, 2020. [Online]. Available: https://labicvl.github.io/ges_db.htm

CLAUDIO TURCHETTI (Life Member, IEEE) received the Laurea degree in electronics engineering from the University of Ancona, Ancona, Italy, in 1979. He joined Università Politecnica delle Marche, Ancona, in 1980, where he was the Head of the Department of Electronics, Artificial Intelligence and Telecommunications for five years and is currently a Full Professor of micro-nanoelectronics and design of embedded systems. His current research interests include statistical device modeling, RF integrated circuits, device modeling at nanoscale, computational intelligence, signal processing, pattern recognition, system identification, machine learning, and neural networks. He has published more than 160 journal and conference papers, and two books. The most relevant papers were published in IEEE JOURNAL OF SOLID-STATE CIRCUITS, IEEE TRANSACTIONS ON ELECTRON DEVICES, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, and *Information Sciences*. He has held a variety of positions as the Project Leader in several applied research programs developed in cooperation with small, large, and multinational companies in the field of microelectronics. He has served as a Program Committee Member for several conferences and as a Reviewer for several scientific journals. He is a member of the Computational Intelligence and Signal processing Society. He has been an Expert Consultant of the Ministero dell'Università e Ricerca.

LAURA FALASCHETTI (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electronics engineering from Università Politecnica delle Marche, Ancona, Italy, in 2008, 2012, and 2016, respectively. She collaborated as a Research Fellow with the Department of Information Engineering (DII), Università Politecnica delle Marche, from 2012 to 2013. She is currently a Postdoctoral Research Fellow with DII and a Contract Professor for the course electronic systems, at electronic and biomedical engineering with Università Politecnica delle Marche. Her current research interests include embedded systems, machine learning, neural networks, manifold learning, pattern recognition, signal processing, image processing, speech and speaker recognition, and speech synthesis.



LORENZO MANONI received the B.Sc. and M.Sc. degrees in electronics engineering from Università Politecnica delle Marche, Ancona, Italy, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the Department of Information Engineering (DII). His current research interests include signal processing, embedded systems, machine learning, algorithms analysis and design, and bio-signal analysis.