Supervised Classification via Constrained Subspace and Tensor Sparse Representation

Liang Liao[1,2], Stephen John Maybank[3], Yanning Zhang[2], Xin Liu[4]
[1]School of Electronics and Information, Zhongyuan University of Technology, Zhengzhou, P. R. China
[2]School of Computer Science, Northwestern Polytechnical University, Xi'an, P. R. China
[3]Birkbeck College, University of London, London, UK
[4]Information Center of Yellow River Conservancy Commission, Zhengzhou, P. R. China
liaoliangis@126.com (L. Liao), sjmaybank@dcs.bbk.ac.uk (S. J. Maybank),
ynzhang@nwpu.edu.cn (Y. Zhang), liuxinis@126.com (X. Liu),

***Abstract*—SRC, a supervised classifier via sparse representation, has rapidly gained popularity in recent years and can be adapted to a wide range of applications based on the sparse solution of a linear system. First, we offer an intuitive geometric model called constrained subspace to explain the mechanism of SRC. The constrained subspace model connects the dots of NN, NFL, NS, NM. Then, inspired from the constrained subspace model, we extend SRC to its tensor-based variant, which takes as input samples of high-order tensors which are elements of an algebraic ring. A tensor sparse representation is used for query tensors. We verify in our experiments on several publicly available databases that the tensor-based SRC called tSRC outperforms traditional SRC in classification accuracy. Although demonstrated for image recognition, tSRC is easily adapted to other applications involving underdetermined linear systems.**

I. Introduction

The concept of a data manifold plays an important role in a wide range of problems. Take image representation for example. An image of size $m \times n$ pixels with 256 gray-scales for each pixel has 256^{mn} pixel configurations, but only a few correspond to particular classes of objects. Due to this redundancy in the raw representation of images, it is often assumed that images belong to a manifold with a low intrinsic dimension (id). The intrinsic dimension is a measure of the number of degrees of freedom in the data [11], [14], [4]. The intrinsic dimension is the minimum number of parameters needed to describe the data structure such that the fundamental properties of the data are preserved. Briefly speaking, when data are represented by vectors in a high dimensional feature space, the manifold associated with the data is a nonlinear geometric structure which describes the distribution of the observed data and serves as a universal data set for the class in question [12].

A classifier called SRC (Sparse Representation Classifier) represents each query sample by a linear sum of a few highly associated training samples. In this way, SRC exploits the redundancy of the data and enjoys a high classification accuracy [19], [17]. Although first proposed for image classification, SRC can be easily adapted to a wide range of applications in which a sparse solution of an underdetermined linear model is established [18], [5].

Later papers reinterpreted or challenged the mechanism of SRC [3], [13], [21], [20], [15]. For example, Zhang et al contend that collaboration offered by training samples of different classes contributes to the success of SRC. Shi et al propose a class-collaborative classifier called orthonormal ℓ_2-norm algorithm to challenge the mechanism of SRC. Yang et al propose an affine sparse representation to improve SRC. Although these works have advanced our knowledge of SRC, a simple yet effective intuitive geometric model to interpret the mechanism of SRC is still unavailable.

We propose a geometric model called constrained subspace, which not only connects the dots of NN (Nearest Neighbor), NFL (Nearest Feature Line), NS (Nearest Subspace) and NM (Nearest Manifold) but also interprets SRC from the perspective of approximation by linear manifolds. Inspired by the recently reported technique called t-product [7], [1], [6], we extend SRC to its tensor-based variant while still retaining our constrained subspace model. We demonstrate in experiments that tSRC outperforms SRC and its non-tensor variant (Yang's method) in classification accuracy.

The reminder of this paper is organized as follows. In Section II, the constrained subspace is introduced. Our tensor model built on a high-order algebraic ring and the tensor-based classifier tSRC are discussed in Section III. In Section IV, a sparse tensor solution of an underlying linear system is obtained. The choice of features is discussed for tSRC. Section V presents experimental results obtained by applying the new classifier to publicly available databases. Section VI concludes this paper.

II. Constrained subspace

A. Notations and indexing

Before any detailed discussion, we first clarify some notations. (i) Vectors unless otherwise stated are column vectors. Some MATLAB notations are adopted as follows. (ii-A) Comma notation: $[\mathbf{A}_1, \mathbf{A}_2]$ is a new matrix concatenating vectors/matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ "from left to right" column by column. (ii-B) Semicolon notation: $[\mathbf{A}_1; \mathbf{A}_2]$ is a new matrix concatenating $\mathbf{A}_1$ and $\mathbf{A}_2$ "from top to bottom" row by

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

row. Similarly, we can concatenate a finite number of vectors/matrices together in one way or another as $[\mathbf{A}_1, \mathbf{A}_2, \cdots]$ or $[\mathbf{A}_1; \mathbf{A}_2; \cdots]$, as long as the sizes of the given entry vectors/matrices are compatible with each other. (iii) The values of the indices for tensors, including vectors and matrices, begin at 0. For example, $\mathbf{A}(0,0)$ rather than $\mathbf{A}(1,1)$ denotes the first scalar entry of matrix/tensor $\mathbf{A}$. (iv) Colon notation: $\mathbf{A}(i,:)$ denotes the i-th row of matrix/tensor $\mathbf{A}$. $\mathbf{A}(:,j)$ denotes the j-th column of matrix/tensor $\mathbf{A}$. Note that $\mathbf{A}(0,:)$ rather than $\mathbf{A}(1,:)$ is the top row and $\mathbf{A}(:,0)$ rather than $\mathbf{A}(:,1)$ is the leftmost column.

B. Model

For the supervised classification problem, we first propose a geometric model called constrained subspace — given K classes and N_i training vectors of class i denoted by $\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)}, \ldots, \boldsymbol{x}_i^{(N_i)}$, the constrained subspace $\mathbb{M}_i$ for class i is defined as a union of a set of of affine hulls as follows.

$$\mathbb{M}_i = \left\{ \mathbf{A}_i \boldsymbol{\alpha}_i \,\middle|\, \begin{array}{c} \boldsymbol{\alpha}_i \in \mathbb{R}^{N_i} \\ \mathbf{1}^T \boldsymbol{\alpha}_i = 1 \text{ and } \|\boldsymbol{\alpha}\|_0 \leqslant \kappa \leqslant N_i \end{array} \right\} \quad (1)$$

where $\mathbf{A}_i \doteq \left[\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)}, \ldots, \boldsymbol{x}_i^{(N_i)}\right]$, $\|\boldsymbol{\alpha}\|_0$ is the ℓ_0-norm of $\boldsymbol{\alpha}$, which is defined as the number the non-zero entries in $\boldsymbol{\alpha}$, $\mathbf{1}$ denotes a vector with all its entries equal to 1 and the intrinsic dimension parameter κ is a positive integer.

Based on equation (1), we define a generalized classifier called NCSC (Nearest Constrained Subspace Classifier), which includes classifiers NN and NFL as low-dimensional special cases — Given a query vector $\boldsymbol{y}$ belonging to one of the above mentioned K classes, NCSC classifies $\boldsymbol{y}$ as follows.

$$\text{class}(\boldsymbol{y}) = \text{argmin}_i \min_{\boldsymbol{x} \in \mathbb{M}_i} \|\boldsymbol{y} - \boldsymbol{x}\|_2 \quad (2)$$

C. Connecting the dots

Note that when $\kappa = 1$, $\mathbb{M}_i$, as in equation (1), becomes a set of feature points and NCSC, as in equation (2), becomes NN. When $\kappa = 2$, $\mathbb{M}_i$ becomes a set of feature lines and NCSC becomes NFL [10]. Moreover, if the training samples are linearly independent, the intrinsic dimension of $\mathbb{M}_i$ is given by

$$\texttt{id}(\mathbb{M}_i) = \kappa - 1. \quad (3)$$

In other words, NN and NFL are just low-dimensional special cases of NCSC respectively with $\texttt{id} = 0$ and $\texttt{id} = 1$. On the other hand, if constraints $\mathbf{1}^T \boldsymbol{\alpha}_i = 1$ and $\|\boldsymbol{\alpha}\|_0 \leqslant \kappa \leqslant N_i$ are removed from equation (1), NCSC becomes NS and $\mathbb{M}_i$ is relaxed to a linear (unconstrained) subspace $\mathbb{S}$ with $\texttt{id}(\mathbb{S}) = N_i$. Since condition $\mathbb{M}_i \subset \mathbb{S}$ is always satisfied for all $\kappa = 1, 2, \cdots, N_i$, we call our model the constrained subspace model.

NCSC is also an approximation to the well-known classifier NM. Simply speaking, if the manifold assumption is true, manifold $\mathcal{M}_i$ serves as the universal sample set of class i. Namely, given query sample $\boldsymbol{y}$ belonging to one of the above mentioned K classes, NM employs a similar approach to equation (2), classifying $\boldsymbol{y}$ as follows.

$$\text{class}(\boldsymbol{y}) = \text{argmin}_i \min_{\boldsymbol{x} \in \mathcal{M}_i} \|\boldsymbol{y} - \boldsymbol{x}\|_2 \quad (4)$$

By virtue of $\mathcal{M}_1, \cdots, \mathcal{M}_K$ serving as universal sets of samples, NM has a high classification accuracy. In order to improve NCSC's accuracy, we contend that $\mathbb{M}_i$ should be an accurate approximation to a manifold $\mathcal{M}_i$ for all $i = 1, 2, \cdots, K$.

Since the affine hull is also referred to as linear manifold, the essence of equation (1) is to use a union of a series of affine hulls with $\texttt{id} = \kappa - 1$ to approximate the corresponding manifold. Since $\mathbb{M}_i|_{\kappa=1} \subset \mathbb{M}_i|_{\kappa=2} \subset \cdots \subset \mathbb{M}_i|_{\kappa=N_i}$, we contend that in order to make $\mathbb{M}_i$ a more accurate approximation to $\mathcal{M}_i$, one criterion is

$$\texttt{id}(\mathbb{M}_i) \approxeq \texttt{id}(\mathcal{M}_i). \quad (5)$$

Equation (5) leads to the problem of estimating $\texttt{id}(\mathcal{M}_i)$ from a finite number of observed samples. Although there are some interesting estimators available [4], [2], [9], their estimates vary with the choice of estimator and estimation parameters. A comprehensive evaluation of the estimators for a range of parameter values is still an open problem. The sparse representation employed by SRC offers an approach without the explicit estimation of intrinsic dimension.

SRC is class-collaborative while NCSC discussed in Section II-B is class-wise in the sense that, in SRC, training samples from different classes are allowed to represent a query sample but in NCSC, only training samples from a same class are used to represent a query sample.

It is easy to update NCSC from class-wise to class-collaborative. To obtain a class-collaborative version of NCSC, construct a constrained subspace $\mathbb{M}$ to approximate to $\mathcal{M} \doteq \bigcup_i \mathcal{M}_i$ under the criterion $\texttt{id}(\mathbb{M}) \approxeq \texttt{id}(\mathcal{M})$. Similar to equation (1), $\mathbb{M}$ is given by

$$\mathbb{M} = \left\{ \mathbf{A}\boldsymbol{\alpha} \,\middle|\, \begin{array}{c} \boldsymbol{\alpha} \in \mathbb{R}^N \\ \mathbf{1}^T \boldsymbol{\alpha} = 1 \text{ and } \|\boldsymbol{\alpha}\|_0 \leqslant \kappa \leqslant N \end{array} \right\} \quad (6)$$

where $\mathbf{A} \doteq [\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_K]$, $\boldsymbol{\alpha} \doteq [\boldsymbol{\alpha}_1; \boldsymbol{\alpha}_2; \cdots; \boldsymbol{\alpha}_K]$ and $N \doteq \sum_i N_i$.

Given $\boldsymbol{y}$, we seek a solution $\boldsymbol{\alpha}^*$ for $\boldsymbol{y} = \mathbf{A}\boldsymbol{\alpha}$ subject to $\mathbf{1}^T \boldsymbol{\alpha} = 1$ and $\|\boldsymbol{\alpha}\|_0 \leqslant \kappa$ where κ is a small positive integer satisfying $\kappa \approxeq 1 + \texttt{id}(\mathcal{M})$. Once $\boldsymbol{\alpha}^*$ is obtained, the class-collaborative NCSC classifies $\boldsymbol{y}$ as follows.

$$\text{class}(\boldsymbol{y}) = \text{argmin}_{i \in \{1,2,\cdots,K\}} \left\|\boldsymbol{y} - \mathbf{A}\lambda_i(\boldsymbol{\alpha}^*)\right\|_2 \quad (7)$$

where $\lambda_i : \mathbb{R}^N \to \mathbb{R}^N$ is the characteristic function. $\lambda_i(\boldsymbol{\alpha}^*)$ returns a sparse vector of length N whose nonzero entries are the entries of $\boldsymbol{\alpha}^*$ associated with the training samples of class i.

D. Yang's method

To avoid an explicit estimation of κ for NCSC, one can seek $\boldsymbol{\alpha}^*$ with the minimal ℓ_0-norm with the expectation $\|\boldsymbol{\alpha}^*\|_0 \leqslant \kappa$.

Namely,

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \|\boldsymbol{\alpha}\|_0 \text{ subject to } \boldsymbol{y} = \mathbf{A}\boldsymbol{\alpha} \text{ and } \mathbf{1}^T\boldsymbol{\alpha} = 1 \quad (8)$$

Furthermore, one can solve the following ℓ_1-norm minimization problem rather than the NP-hard problem in equation (8).

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \|\boldsymbol{\alpha}\|_1 \text{ subject to } \boldsymbol{y} = \mathbf{A}\boldsymbol{\alpha} \text{ and } \mathbf{1}^T\boldsymbol{\alpha} = 1 \quad (9)$$

Equation (9) is Yang's method [20], which imposes the constraint $\mathbf{1}^T\boldsymbol{\alpha} = 1$ on SRC, in order to improve its classification performance. However Yang did not provide a concise geometric model to explain the essence of the constraint $\mathbf{1}^T\boldsymbol{\alpha} = 1$. From the perspective of manifold approximation, our constrained subspace model supports Yang's method over SRC if the manifold assumption is true.

III. Tensors

A. Definitions

In this section, we propose a tensor sparse representation for tensor data such as images (second-order tensors), which naturally come in the form of arrays rather than vectors. It is easy to extend the tensor sparse representation to high-order tensors, such as videos (third-order tensors). For presentation conciseness and without loss of generality we focus the discussion on second-order tensors such as images. First, we give some definitions as follows.

Definition 1. Tensor addition. Given tensors $\boldsymbol{A}$ and $\boldsymbol{B}$ of size $m \times n$, the sum $\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{B}$ is a new tensor of same size, satisfying $\boldsymbol{C}(i,j) = \boldsymbol{A}(i,j) + \boldsymbol{B}(i,j)$ for all $i = 0, 1, \cdots, m-1$ and $i = 0, 1, \cdots, n-1$.

Definition 2. Tensor multiplication. Given tensors $\boldsymbol{A}$ and $\boldsymbol{B}$ of size $m \times n$, the product $\boldsymbol{D} = \boldsymbol{A} \circ \boldsymbol{B}$ is a new tensor of same size defined as the result of 2D circular convolution between $\boldsymbol{A}$ and $\boldsymbol{B}$, satisfying $\boldsymbol{D}(i,j) = \sum_{k_1=0}^{m-1} \sum_{k_2=0}^{n-1} \boldsymbol{A}(k_1,k_2)\boldsymbol{B}((i-k_1) \bmod m, (j-k_2) \bmod n)$ for all $i = 0, 1, \cdots, m-1$ and $j = 0, 1, \cdots, n-1$.

Given tensors $\boldsymbol{A}$ and $\boldsymbol{B}$ of the same size, the product $\boldsymbol{C}$ can be computed efficiently by performing the 2D Fast Fourier Transform (2DFFT) and its inverse 2DFFT transform, because the following theorem holds.

Theorem 3. Fourier Transform. *Given tensors $\boldsymbol{A}$, $\boldsymbol{B}$ and the product $\boldsymbol{D} = \boldsymbol{A} \circ \boldsymbol{B}$, $\hat{\boldsymbol{D}}(i,j) = \hat{\boldsymbol{A}}(i,j)\hat{\boldsymbol{B}}(i,j)$, $\forall$ i and j, where $\hat{\boldsymbol{D}}$, $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$ are respectively the 2D Fast Fourier Transforms of $\boldsymbol{D}$, $\boldsymbol{A}$ and $\boldsymbol{B}$.*

By Theorem 3, we have $\boldsymbol{A} \circ \boldsymbol{B} \equiv \boldsymbol{B} \circ \boldsymbol{A}$.

Definition 4. Zero tensor. The zero tensor $\boldsymbol{Z}$ is defined by $\boldsymbol{Z}(i,j) = 0$ for all i and j.

Definition 5. Identity tensor. The identity tensor $\boldsymbol{E}$ is defined by $\boldsymbol{E}(0,0) = 1$ and $\boldsymbol{E}(i,j) = 0$ for $(i,j) \neq (0,0)$.

It is not difficult to prove that the tensors of size $m \times n$ defined above form an algebraic ring $\boldsymbol{\mathcal{R}}$. Furthermore, we also define a tensor vector/matrix as follows.

Definition 6. Tensor vector/matrix. Tensor vector/matrix is defined as vector/matrix whose atomic entries are tensors over $\boldsymbol{\mathcal{R}}$ rather than scalars over $\mathbb{R}$. Namely, a tensor vector is a list of tensors with the same size and a tensor matrix is a two-dimensional array of tensors with the same size. Operations between entries of tensor vector/matrix comply with the operations defined over $\boldsymbol{\mathcal{R}}$. Other manipulations of tensor vectors/matrices are analogous to these defined for vectors/matrices over $\mathbb{R}$.

For example, given a row tensor vector $\boldsymbol{A} \doteq [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N] \in \boldsymbol{\mathcal{R}}^{1\times N}$ and a column tensor $\boldsymbol{\beta} = [\boldsymbol{\beta}_1; \boldsymbol{\beta}_2; \ldots; \boldsymbol{\beta}_N] \in \boldsymbol{\mathcal{R}}^N$, where $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N$, $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_N$ are tensors of the same size, the multiplication $\boldsymbol{Y} = \boldsymbol{A} \circ \boldsymbol{\beta} \in \boldsymbol{\mathcal{R}}$ is defined as follows.

B. Connection to t-product.

The recently emerged t-product [7], [1], [6], [22], which is based on 1D circular convolution, can be described by our tensor model. Let's take the t-product defined in [22] for example — Given tensors $\boldsymbol{A}$ and $\boldsymbol{B}$ of size $m \times n$, if we leave $\boldsymbol{A}$ alone and constrain $\boldsymbol{B}$ satisfying $\boldsymbol{B}(i,j) \equiv 0$ if $i \neq 0$, then $\boldsymbol{C} = \boldsymbol{A} \circ \boldsymbol{B}$, a tensor of size $m \times n$, is equivalent to t-product $\boldsymbol{C}^{(t)} = \boldsymbol{A}^{(t)} * \boldsymbol{B}^{(t)}$ of size $m \times 1 \times n$, where $\boldsymbol{A}^{(t)}$ is a t-product tensor (we call tensors defined by t-product t-product tensors) of size $m \times 1 \times n$ satisfying $\boldsymbol{A}^{(t)}(i,0,j) \equiv \boldsymbol{A}(i,j)$ and $\boldsymbol{B}^{(t)}$ is a t-product tensor of size $1 \times 1 \times n$ satisfying $\boldsymbol{B}^{(t)}(0,0,j) \equiv \boldsymbol{B}(0,j)$ for all $i = 0, 1, \cdots, m$ and $j = 0, 1, \cdots, n$. $\boldsymbol{C}$ is equivalent to $\boldsymbol{C}^{(t)}$ in the sense that $\boldsymbol{C}(i,j) \equiv \boldsymbol{C}^{(t)}(i,0,j)$ for all i and j.

Since our approach defined in Section III-A manipulates tensors from multiple directions, (i.e., row and column directions for second-order tensors) while t-product manipulates tensor only from one direction, our tensor model is more generalized.

C. Tensor sparse representation and tSRC

Given training tensors $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_N$ belonging to K classes, a query tensor $\boldsymbol{Y}$ can be represented by a sparse linear sum of the training tensors as follows.

$$\boldsymbol{Y} = \sum_{k=1}^{N} \boldsymbol{\beta}_k \circ \boldsymbol{X}_k \text{ subject to } \sum_{k=1}^{N} \boldsymbol{\beta}_k = \boldsymbol{E} \quad (10)$$

On defining the column tensor vector $\boldsymbol{\beta} \doteq [\boldsymbol{\beta}_1; \boldsymbol{\beta}_2; \cdots; \boldsymbol{\beta}_N]$ and the row tensor vector $\boldsymbol{A} \doteq [\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_N]$, equation (10) can be rewritten in its tensor vector/matrix form. A sparse solution $\boldsymbol{\beta}^*$ is given by the following ℓ_1-norm minimization.

$$\begin{gathered} \boldsymbol{\beta}^* = \operatorname{argmin}_{\boldsymbol{\beta}} \left\|\boldsymbol{\beta}\right\|_1 \\ \text{subject to } \boldsymbol{Y} = \boldsymbol{A} \circ \boldsymbol{\beta} \text{ and } \textstyle\sum_{k=1}^{N} \boldsymbol{\beta}_k = \boldsymbol{E} \end{gathered} \quad (11)$$

where $\left\|\boldsymbol{\beta}\right\|_1 \doteq \sum_{k=1}^{N} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left|\boldsymbol{\beta}_k(i,j)\right|$.

We define the ℓ_2-norm of a second-order tensor $\boldsymbol{X}$ by $\|\boldsymbol{X}\|_2 \doteq \sqrt{\sum_i \sum_j |\boldsymbol{X}(i,j)|^2}$. Then, we propose a tensor based classifier called tSRC in Algorithm 1.

Algorithm 1 tSRC — tensor SRC on algebraic ring $\mathcal{R}$

Input: Query tensor $\boldsymbol{Y}$ and training tensors $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_N$ of K classes.

Output: Class label of $\boldsymbol{Y}$.

1: Calculate $\boldsymbol{\beta}^* \doteq [\boldsymbol{\beta}_1^*; \cdots; \boldsymbol{\beta}_N^*]$ as in equation (11);
2: $r_i \leftarrow \|\boldsymbol{Y} - \boldsymbol{A} \circ \delta_i(\boldsymbol{\beta}^*)\|_2, \quad \forall i = 1, 2, \cdots, K$;
3: **return** $\text{class}(\boldsymbol{Y}) \leftarrow \text{argmin}_{i \in \{1,2,\cdots,K\}} r_i$;

In Algorithm 1, $\delta_i : \boldsymbol{\mathcal{R}}^N \rightarrow \boldsymbol{\mathcal{R}}^N$ is the characteristic function. $\delta_i(\boldsymbol{\beta}^*)$ returns a sparse tensor vector of length N, whose nonzero entries are the coefficient tensors associated with the training tensors of class i. On denoting the index set of the training tensors of class i by $\mathcal{S}_i$ and defining $\boldsymbol{\delta}_N^j \in \boldsymbol{\mathcal{R}}^N$ as the sparse column tensor vector with entry j equal to $\boldsymbol{E}$ and all other entries equal to $\boldsymbol{Z}$, $\delta_i(\boldsymbol{\beta})$ is defined as follows.

$$\delta_i(\boldsymbol{\beta}) = \sum_{j \in \mathcal{S}_i} \text{diag}\left(\boldsymbol{\delta}_N^j \circ \boldsymbol{\beta}^T\right) \tag{12}$$

where $\boldsymbol{\beta}^T \doteq [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_N] \in \boldsymbol{\mathcal{R}}^{1 \times N}$.

IV. ℓ_1-NORM MINIMIZER AND FEATURE EXTRACTION

A. Minimizer

We give a solver for equation (10), which recasts the tensor-based optimization problem as a traditional ℓ_1-norm optimization problem as follows.

First, we give the following notations — given a tensor $\boldsymbol{X}$ of size $m \times n$ and integers k_m, k_n, $\boldsymbol{X}^{(k_m,k_n)}$ denotes a tensor of the same size, satisfying $\boldsymbol{X}^{(k_m,k_n)}(i,j) = \boldsymbol{X}((i - k_m) \bmod m, (j - k_n) \bmod n)$, for all $i = 0, 1, \cdots, m-1$ and $j = 0, 1, \cdots, n-1$. We denote the vector versions of $\boldsymbol{X}$ and $\boldsymbol{X}^{(k_m,k_n)}$ respectively by $\boldsymbol{x}$ and $\boldsymbol{x}^{(k_m,k_n)}$.

Then, equation (11) can be transformed into the form of equation (9) — given training tensors $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_N$ of size $m \times n$, the columns of $\mathbf{A}$ in equation (9) are given by

$$\mathbf{A}(:, (k-1)mn + k_m n + k_n) = \boldsymbol{x}_k^{(k_m,k_n)} \tag{13}$$

where $\boldsymbol{x}_k^{(k_m,k_n)}$ is the vector version of $\boldsymbol{X}_k^{(k_m,k_n)}$ for all $k = 1, 2, \cdots, N$, $k_m = 0, 1, \ldots, m-1$ and $k_n = 0, 1, \ldots, n-1$.

Furthermore, setting $\boldsymbol{y}$ as the vector version of $\boldsymbol{Y}$ and with the original constraint $\mathbf{1}^T\boldsymbol{\alpha} = 1$ replaced by

$$\begin{gathered}\sum_{k=1}^{N} \boldsymbol{\alpha}((k-1)mn + k_m n + k_n) = \boldsymbol{E}(k_m, k_n) \\ \forall k_m = 0, 1, \cdots, m-1 \text{ and} \\ \forall k_n = 0, 1, \cdots, n-1,\end{gathered} \tag{14}$$

$\boldsymbol{\alpha}^*$ obtained as in equation (9) is equivalent to $\boldsymbol{\beta}^*$ obtained as in equation (11).

Since $\mathbf{A}(:, (k-1)mn + k_m n + k_n)$ corresponds to $\boldsymbol{\beta}_k(k_m, k_n)$, the tensor vector $\boldsymbol{\beta}^* \doteq [\boldsymbol{\beta}_1^*; \boldsymbol{\beta}_2^*; \cdots; \boldsymbol{\beta}_N^*]$ can be easily restored by

$$\boldsymbol{\beta}_k^*(k_m, k_n) = \boldsymbol{\alpha}^*((k-1)mn + k_m n + k_n). \tag{15}$$

Equations (13) and (15) indicate that tSRC is an extension of SRC (more accurately Yang's method) which enlarges the original training volume by extending a single sample $\boldsymbol{X}_k$ to multiple samples $\boldsymbol{X}_k^{(k_m,k_n)}$ for all $k_m = 0, 1, \cdots, m-1$ and $k_n = 0, 1, \cdots, n-1$.

B. Feature extraction

There exist a wide range of popular and effective feature extractors, such as PCA (Principle Component Analysis), which extract features as vectors rather than tensors. To work with these feature vector extractors, one can extract a collection of feature vectors from a set $\left\{\boldsymbol{X}^{(k_m,k_n)} \middle| \forall k_m, k_n\right\}$ rather than just one feature vector from one sample $\boldsymbol{X}$.

With the minimizer discussed in Section IV-A, which flattens tensors to vectors, tSRC can conveniently handle feature vectors. More specifically, tSRC uses the following equation to work with feature vectors.

$$\mathbf{A}(:, (k-1)mn + k_m n + k_n) = \hat{\boldsymbol{x}}_k^{(k_m,k_n)} \tag{16}$$

where $\hat{\boldsymbol{x}}_k^{(k_m,k_n)}$ denotes the feature vector extracted from $\boldsymbol{X}_k^{(k_m,k_n)}$.

C. Computational complexity

Due to the tensor formalization, the proposed tSRC is computationally more complex than SRC and its other. When tSRC is flatten to its non-tensorial version for solving equation (11), the number of the columns of $\mathbf{A}$, denoted by N_{tSRC}, is $mn \times N$, which is a significant increase, compared with the original training number N. Nevertheless, if the number of the prospect non-zero entries of each tensor coefficient $\boldsymbol{\beta}_i$ for all i is confined to be small, the computational complexity of tSRC is significantly reduced by discarding the columns of $\mathbf{A}$ associated with the zero entries of each $\boldsymbol{\beta}_i$ and only solving for the prospect non-zero entries of each $\boldsymbol{\beta}_i$. Some fast ℓ_1-norm minimizers are specially designed to take advantage of the properties of ℓ-norm minimization and have a much lower computational complexity. For example, the computational complexity of Homotopy method is bounded by $O(\kappa d^2 + \kappa dN)$, where d is the feature dimension and N the number of training samples, if it correctly recovers a κ-sparse vector in κ steps. Using the fast ℓ_1-norm minimizers and constraining the which tensor entries are potentially non-zero, one can implement tSRC at the cost of a scalable computational complexity.

V. EXPERIMENTS

A. Settings

Although tSRC and SRC-like classifiers can be used for purposes including but not limited to supervised classification, in this section, we evaluate tSRC for supervised image classification on several publicly available databases.

In order to get statistically stable classification accuracies by classifying a sufficient number of query samples for each training set, our experiments contain multiple rounds of classification. In order to avoid unnecessary perturbations of classification accuracy, in each round, query samples and training samples are randomly chosen and then kept unchanged in the evaluation of the relevant classifiers. After all

rounds, the classification accuracy of each classifier is given by $accuracy = w/W$ where w is the total number of correctly classified query samples and W is the total number of query samples from all rounds.

B. Evaluations on the raw MNIST data

In this section, we apply the relevant classifiers to the raw MNIST database [8]. The MNIST database of handwritten digits contains a large number of image samples of size 28×28 pixels belonging to 10 classes. There are $60,000$ training images (roughly $6,000$ images per class) and $10,000$ query images (roughly $1,000$ images per class) in the MNIST database. Figure 1 shows some examples of the MNIST database.

Fig. 1. Some image samples of the MNIST database.

Due to its large number of samples, small image size and unvarying illumination condition, it is ideal to evaluate the manifold assumption, constrained subspace model and the relevant classifiers.

To use SRC and Yang's method, which handle vectors rather than tensors, on the raw images of the MNIST database, images have been flattened without feature extraction to 784-dimensional vectors. Although tSRC can directly handle images as second-order tensors, we use the vector-flattening ℓ_1-norm minimizer discussed in Section IV-A to solve equation (9). To construct the underdetermined linear system in equation (9), the column number of $\mathbf{A}$ should be larger than 784. We observe the classification performance when the number of training samples is changed. Thus, in each classification round, we set the number of training samples of class i respectively by $N_i = 100$, 150, 200, 250, 300 and 350. The experiment contains 10 classification rounds. In each round, $1,000$ query images (100 images/class $\times 10$ classes) are randomly chosen from the MNIST test set and N training samples (N samples = N_i samples/class $\times$ 10 classes) are randomly chosen from the MNIST training set.

To evaluate the efficacy of the t-product [7], [1], [6], our tSRC is developed as two variants, tSRC-1D and tSRC-2D. Classifier tSRC-1D is tSRC via tensor sparse representation offered by the t-product, but with the constraint that the only non-zero scalar entries of tensor $\boldsymbol{\beta}_k$ are in $\boldsymbol{\beta}_k(0,:)$ for all $k = 1, 2, \cdots, N$. More specifically, in the t-product model, tensor $\boldsymbol{X}_k$ of size $m \times n$ is recast to t-product tensor $\boldsymbol{X}_k^{(t)}$ of size $m \times 1 \times n$, tensor $\boldsymbol{\beta}_k$ is recast to t-product tensor $\boldsymbol{\beta}_k^{(t)}$ of size $1 \times 1 \times n$, satisfying, $\forall\ i = 0, 1, \cdots, m-1$ and $j = 0, 1, \cdots, n-1$

$$\begin{cases} \boldsymbol{X}_k^{(t)}(i,0,j) = \boldsymbol{X}_k(i,j) \\ \boldsymbol{\beta}_k^{(t)}(0,0,j) \ = \boldsymbol{\beta}_k(0,j) \end{cases}. \tag{17}$$

Thus, $\boldsymbol{C} \doteq \boldsymbol{X}_k \circ \boldsymbol{\beta}_k$ is equivalent to $\boldsymbol{C}^{(t)} \doteq \boldsymbol{X}_k^{(t)} * \boldsymbol{\beta}_k^{(t)}$ in the sense that $\boldsymbol{C}(i,j) \equiv \boldsymbol{C}^{(t)}(i,0,j)$.

In tSRC-2D, the constraint imposed on the nonzero entries in $\boldsymbol{\beta}_k$ found in tSRC-1D is removed. In comparison with SRC and Yang's method, the computational complexity of solving the linear system in tSRC is dramatically increased primarily because tensors are employed.

In order to obtain results of tSRC in a reasonable time, the computational complexity of tSRC is reduced. We constrain tSRC-2D by setting $\boldsymbol{\beta}_k(i,j)$ equal to zero outside the set of values defined by $i \in \{(m-1), 0, 1\}$ and $j \in \{(n-1), 0, 1\}$. Similarly, we constrain tSRC-1D by setting $\boldsymbol{\beta}_k(i,j)$ equal to zero outside the set of values defined by $i = 0$ and $j \in \{(n-1), 0, 1\}$. These constraints on tSRC-1D and tSRC-2D are also kept unchanged in the experiments on other databases.

This constraint imposed on tSRC exploits the fact that in a small neighborhood of an image, the grayscales of the pixels are highly correlated.

Figure 2 shows the classification accuracy curves of the relevant classifiers SRC, Yang's, tSRC-1D and tSRC-2D on the raw MNIST data.

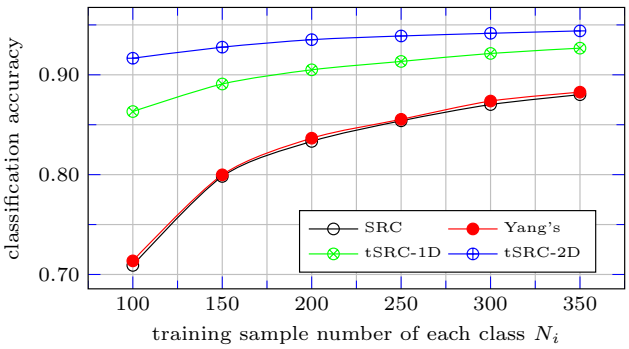

Fig. 2. Accuracy curves with an increasing number of training samples obtained by SRC, Yang's method, tSRC-1D and tSRC-2D on the raw MNIST data.

Several observations are made on Figure 2.

(i) As N_i increases, the classification accuracy increases. Since the constrained subspace is a linear approximation to manifold of low intrinsic dimension, with N_i increasing, we contend that the approximation to the underlying manifold becomes more accurate and leads to a higher classification accuracy.

(ii) The second observation is that the classification accuracy of Yang's method is higher than that of SRC, albeit marginally. This versifies the report by Yang [20] and supports the manifold assumption and our constrained subspace model.

(iii) The classification accuracy of tSRC is higher than those of SRC and Yang's method.

(iv) The accuracy of tSRC-2D is higher than that of tSRC-1D. We contend that this is because tSRC-2D exploits more image structure information both in the row direction and the column direction while tSRC-1D only exploits structure information in the column direction.

C. Evaluations with feature extraction

Since tSRC (tSRC-1D and tSRC-1D) can be transformed to its flattened version involving vectors, we are particularly interested in classification accuracies obtained on feature vectors. Given the effectiveness and popularity of PCA in data analysis, we evaluate SRC, Yang's method, tSRC with PCA features. For tSRC, given tensor $\boldsymbol{X}_k^{(k_m,k_n)}$ and after vectorizing it, we extract its PCA feature vector $\hat{\boldsymbol{x}}_k^{(k_m,k_n)}$ as in equation (16).

1) MNIST data: Figure 3 gives the classification accuracy curves of SRC, Yang's method and tSRC by repeating the experiment in Section V-B but with the PCA feature extractor. The feature dimension is $d = 100$.

Compared to the classification results shown in Figure 2, the accuracies obtained with PCA are significantly increased. The observations drawn from Figure 2 also apply to Figure 3.

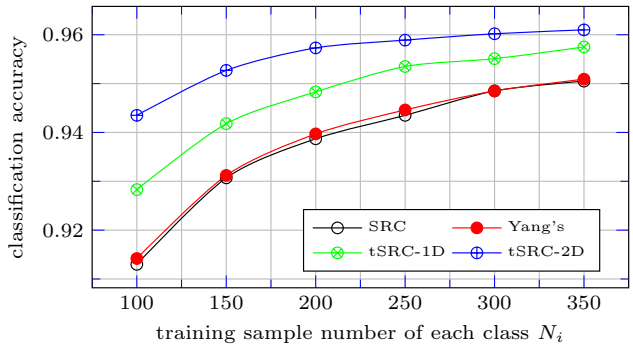

Fig. 3. Accuracy curves with an increasing number of training samples obtained by SRC, Yang's method, tSRC-1D and tSRC-2D on the PCA features of the MNIST data.

2) ORL data: We are also interested in classification accuracies on image datasets with a larger image size but with a smaller number of images. The ORL database is one such database. The ORL database contains 400 facial images (10 images/class $\times$ 40 classes) of size 112 $\times$ 92 pixels, taken at different times with variations of facial expressions, details and poses etc.

Figure 4 shows the images from a sample subject (class) of the ORL data.

Fig. 4. Images of a sample subject (class) of the ORL data.

Figure 5 gives the accuracy curves of SRC, Yang's method and tSRC on this database.

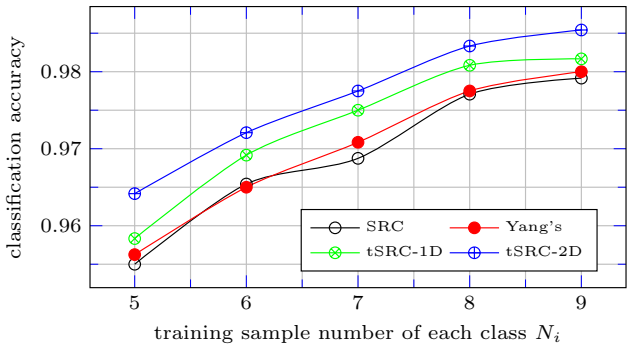

Fig. 5. Accuracy curves with an increasing number of training samples obtained by SRC, Yang's method, tSRC-1D and tSRC-2D on features extracted by PCA on the ORL data.

In our experiments, the number of training samples for each class is respectively taken as $N_i = 5, 6, 7, 8, 9$. For each value of N_i, the accuracy of each classifier is obtained from multiple classification rounds. In each round, N_i samples/class $\times$ 40 classes are randomly chosen as the training samples. The remaining samples are taken as the query samples. Thus, in each classification round, the number of query samples is $(10 - N_i)$ samples/class $\times$ 40 classes. Both training samples and query samples are extracted by PCA to yield 100-dimensional feature vectors.

In order to have the same number of query samples in multiple rounds for each value of N_i, corresponding to $N_i = 5, 6, 7, 8, 9$, the number of classification rounds is respectively set as $12, 15, 20, 30, 60$. In other words, each accuracy point in Figure 5 is the result of classifying 2400 random query samples.

The observations on the curves in Figure 4 are the same as the observations drawn in Section V-B on the raw MNIST data .

3) Extended Yale B data: We also evaluate the classifiers on the Extended Yale B database [16]. The Extended Yale B database contains $2,414$ cropped frontal face images of size 192×168 pixels from 38 subjects (classes), roughly 64 image per class. The images from this database are taken under varying illumination.

Figure 6 shows some samples of the Extended Yale B database. It is clear that the average grayscales of the samples even in the same class are perceivably different.

Fig. 6. Image samples of the Extended Yale B database under varying illumination condition.

Since the effect of the illumination can be modeled by

a scale factor, it is believed that varying illumination tends to reduce the accuracy of the approximation to the data by the underlying constrained subspace. This might challenge the effectiveness of the affine linear representation offered by the constrained subspace model.

To reduce the influence of varying illumination, we first enhance the grayscale contrast of images by means of grayscale histogram equalization. Then, we extract the PCA features of the enhanced images with feature dimension $d = 100$. There are 10 classification rounds for each value of N_i, . In each round, 912 images (912 images $=$ 24 images/class $\times$ 38 classes) are randomly chosen as query samples. N training samples (where N samples $= N_i$ samples/classes $\times$ 38 classes) are randomly chosen from the rest of the samples respectively with $N_i = 16$, 20, 24 and 28.

Figure 7 shows the accuracy curves, with an increasing number of training samples, for SRC, Yang's method, tSRC-1D and tSRC-1D. Each point on the curves is the accuracy of classifying $9,120$ random query images ($9,120$ images $= 912$ images/round $\times$ 10 rounds).

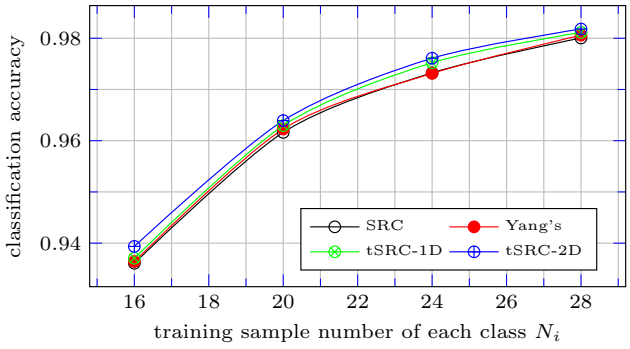

Fig. 7. Accuracy curves with an increasing number of training samples obtained by SRC, Yang's method, tSRC-1D and tSRC-2D on the PCA features of the Extended Yale B data.

The observations found in Figure 7 are consistent with the ones found in Figures 2, 3 and 5 in the sense that the accuracy of tSRC (tSRC-1D and tSRC-2D) is higher than those of SRC and Yang's method.

However, compared to the results shown in Figures 2, 3, and 5, the accuracy differences between SRC, Yang's and tSRC are less predominant although image enhancement and PCA feature extraction have been used.

Although our present work only focuses on evaluations and comparisons of the performances of relevant classifiers, we argue that, to enlarge the classification difference of tSRC and its non-tensor counterparts, besides more sophisticated image enhancement and feature extraction techniques, a reduced region of the entries constrained to zero in $\boldsymbol{\beta}_k$ for all k (discussed in Section V-B) might be helpful, but at the cost of a significant increase of computational complexity. Our present experiments clearly show the differences in the classification accuracies of SRC, Yang's method and tSRC.

4) Evaluations with increasing feature dimension: In the experiments in this section, the training samples are fixed but feature dimension is changed. The experiment is on the PCA feature vectors extracted from the ORL database. The feature dimension is respectively set by $d = 20$, 60, 100 and 140. In each classification round, 200 training samples (5 samples/class $\times$ 40 classes) are randomly chosen. The remaining 200 samples (also 5 samples/class $\times$ 40 classes) are taken as the query samples. There are 10 classification rounds. In other words, each point in Figure 8 is the accuracy of classifying $2,000$ samples (200 samples/round $\times$ 10 rounds).

Figure 8 gives the accuracy curves over different values of feature dimension. The curves shown in Figure 8 verify the accuracy of tSRC over its non-tensor counterparts.

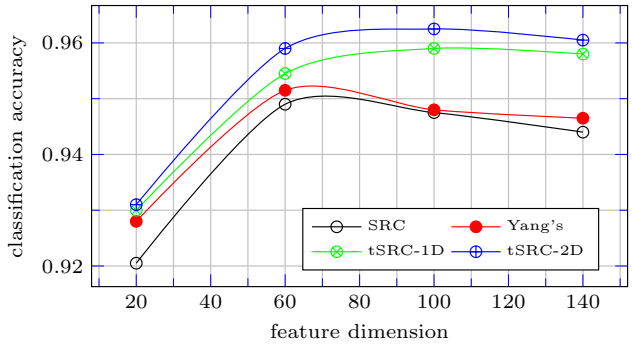

Fig. 8. Accuracy curves with an increasing feature dimension obtained by SRC, Yang's method, tSRC-1D and tSRC-2D on the PCA features of the ORL data.

Another interesting observation found from Figure 8 is that the increase in feature dimension does not always lead to an increase in the classification accuracies of SRC, Yang's method and tSRC. When feature dimension d is very small (for example, in Figure 8, when feature dimension $d < 60$), an increase of d does increase classification accuracy since more useful information in the data is exploited. When d is relatively high (for example when $d > 100$ in Figure 8), the features contain over-detailed information from the data which can cause the classification accuracy to decline. Similar results are obtained in classifying samples from other databases. The results are not reported here in order to reduce the length of the paper. However, we would like to add another interpretation to this observation from the perspective of constrained subspace — Each classifier in our experiments requires the training samples to be over-complete in order to construct an underdetermined linear system $\boldsymbol{y} = \mathbf{A}\boldsymbol{\alpha}$, which has a low-rank matrix $\mathbf{A}$. The over-completeness of training samples is highly associated with feature dimension d. When the number of training samples is fixed, an increase of d of a query sample usually make it less likely that the given training samples will be complete for the query sample. When d increases, in order to find a constrained space $\mathbb{M}$ to include the query sample, the intrinsic dimension $\mathtt{id}(\mathbb{M})$ must increase. In other words, the solution $\boldsymbol{\alpha}^*$ of equation (11) or $\boldsymbol{\beta}^*$ of equation (8) becomes less sparse. We call this phenomenon the "inflation of the constrained subspace". The

inflated constrained subspace with $\mathtt{id}(\mathbb{M})$ much larger than that of the manifold is a less accurate linear approximation to the underlying manifold and therefore causes a reduction in the accuracy of the classification.

VI. Conclusions

A constrained subspace model is proposed for a tensor-based classifier called tSRC. In this model, the constrained subspace is defined as a union of a series of affine hulls. Each affine hull is spanned by training samples via sparse representation, and serves as a local linear approximation of the corresponding manifold.

This model establishes a generalized framework for some classical classifiers including NM, NN, NFL and NS. In this model, NN, NFL and NS are all approximations to NM, which in principle enjoys a high classification accuracy. The constrained subspaces of NN and NFL have respectively intrinsic dimensions 0 and 1.

To make the constrained subspace a more accurate approximation to the corresponding manifold, we contend that the intrinsic dimension of the constrained subspace should be equal to that of manifold. Based on this assumption, we contend that the sparsity parameter κ of the sparse representation, employed with the training samples to span the constrained subspace, should be carefully tuned. Thus, the searching of the nearest constrained subspace point to a query data point is formulated as a constrained least-squares problem.

To circumvent the explicit intrinsic dimension estimation and exploit the collaboration offered by different classes, the constrained subspace model helps transform the least-squares problem to a spare representation problem via ℓ_1-norm optimization.

Thus, the proposed constrained subspace model not only connects the dots of NN, NFL, NS, SRC and NM but also offers an intuitive explanation to the mechanism of SRC and Yang's method which is a variant of SRC.

Then, we replace the vector representation of data by a high-order tensor representation. The multiplication between two tensors is defined via high-order circular convolution.

Then we propose a novel classifier called tSRC. The proposed classifier is a tensor variant of SRC subject to the conditions inspired from the constrained subspace model.

Our experiments on several publicly available databases verify our claims on the constrained subspace model and tSRC. The proposed tSRC is scalable in its computational complexity control. At the cost of a manageable complexity increase, tSRC is more accurate than competing classifiers.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. U1404607 and 61379113), the High-end Foreign Experts Recruitment Program (No. GDW20134100119), and the open foundation program of the Shaanxi Provincial Key Laboratory of Speech and Image Information Processing (No. SJ2013001).

References

[1] K. Braman, "Third-order tensors as linear operators on a space of matrices," *Linear Algebra and its Applications*, vol. 433, no. 7, pp. 1241–1253, 2010.

[2] F. Camastra and A. Vinciarelli, "Estimating the intrinsic dimension of data with a fractal-based method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 10, pp. 1404–1407, 2002.

[3] M. Elad, M. A. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," in *Proceedings of the IEEE*, vol. 98. IEEE, 2010, pp. 972–982.

[4] M. Fan, H. Qiao, and B. Zhang, "Intrinsic dimension estimation of manifolds by incising balls," *Pattern Recognition*, vol. 42, no. 5, pp. 780–787, 2009.

[5] X. Huang, D. P. Dione, C. B. Compas, X. Papademetris, B. A. Lin, A. Bregasi, A. J. Sinusas, L. H. Staib, and J. S. Duncan, "Contour tracking in echocardiographic sequences via sparse representation and dictionary learning," *Medical image analysis*, vol. 18, no. 2, pp. 253–271, 2014.

[6] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 148–172, 2013.

[7] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and its Applications*, vol. 435, no. 3, pp. 641–658, 2011.

[8] Y. LeCun and C. Cortes, "The MNIST database of handwritten digits," http://yann.lecun.com/exdb/mnist/.

[9] E. Levina and P. J. Bickel, "Maximum likelihood estimation of intrinsic dimension," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and l. Bottou, Eds., Cambridge, MA, December 2005, pp. 777–784.

[10] S. Z. Li and J. Lu, "Face recognition using the nearest feature line method," *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp. 439–443, 1999.

[11] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*. Springer, 2005, ch. Introduction, pp. 141–142.

[12] L. Liao, Y. Zhang, S. J. Maybank, and Z. Liu, "Intrinsic dimension estimation via nearest constrained subspace classifier," *Pattern Recognition*, vol. 47, no. 3, p. 1485, 1485–1493 2014.

[13] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[14] G. Peyré, "Manifold models for signals and images," *Computer Vision and Image Understanding*, vol. 113, no. 2, pp. 249–260, 2009.

[15] Q. Shi, A. Eriksson, A. van den Hengel, and C. Shen, "Is face recognition really a compressive sensing problem?" in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011, pp. 553–560.

[16] UCSD, "Extended Yale face database B," http://vision.ucsd.edu/extyaleb/CroppedYaleBZip/CroppedYale.zip.

[17] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma, "Toward a practical face recognition system: Robust alignment and illumination by sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 372–386, 2012.

[18] L. Wang, F. Shi, Y. Gao, G. Li, J. H. Gilmore, W. Lin, and D. Shen, "Integration of sparse multi-modality representation and anatomical constraint for isointense infant brain mr image segmentation," *NeuroImage*, vol. 89, pp. 152–164, 2014.

[19] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[20] J. Yang, L. Zhang, Y. Xu, and J.-y. Yang, "Beyond sparsity: The role of ℓ_1-optimizer in pattern classification," *Pattern Recognition*, vol. 45, no. 3, pp. 1104–1118, 2012.

[21] D. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *2011 IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 471–478.

[22] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-svd," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3842–3849.