Image Recognition via Two-dimensional Random Projection and Nearest Constrained Subspace

Liang Liao[*], Yanning Zhang[†], Stephen John Maybank[‡], Zhoufeng Liu[§]

Abstract

We consider the problem of image recognition via two-dimensional random projection and nearest constrained subspace. First, image features are extracted by a two-dimensional random projection. The two-dimensional random projection for feature extraction is an extension of the 1D compressive sampling technique to 2D and is computationally more efficient than its 1D counterpart and 2D reconstruction is guaranteed. Second, we design a new classifier called NCSC (Nearest Constrained Subspace Classifier) and apply it to image recognition with the 2D features. The proposed classifier is a generalized version of NN (Nearest Neighbor) and NFL (Nearest Feature Line), and it has a close relationship to NS (Nearest Subspace). For large datasets, a fast NCSC, called NCSC-II, is proposed. Experiments on several publicly available image sets show that when well-tuned, NCSC/NCSC-II outperforms its rivals including NN, NFL, NS and the orthonormal ℓ_2-norm classifier. NCSC/NCSC-II with the 2D random features also shows good classification performance in noisy environment.

Keywords: Supervised image classification, Two-dimensional random projection, Compressive sampling, ℓ_1-norm minimization, ℓ_0-norm sparse representation, Constrained subspace, Affine hull

[*] Liang Liao is with the Shaanxi Provincial Key Laboratory of Speech and Image Information Processing (SAIIP), School of Computer Science, Northwestern Polytechnic University, Xi'an, Shaanxi, 710129, P. R. China. He is also with the School of Electronics and Information, Zhongyuan University of Technology, Zhengzhou, Henan, 450007, P. R. China. Email: liaoliangis@126.com.

[†] Yanning Zhang (corresponding author) is with the Shaanxi Provincial Key Laboratory of Speech and Image Information Processing (SAIIP), School of Computer Science, Northwestern Polytechnic University, Xi'an, Shaanxi, 710129, P. R. China. Email: ynzhang@nwpu.edu.cn

[‡] Stephen John Maybank is with the Department of Computer Science and Information Systems, Birkbeck College, University of London, Bloomsbury, London, WC1E 7HX, UK. Email: sjmaybank@dcs.bbk.ac.uk

[§] Zhoufeng Liu is with the School of Electronics and Information, Zhongyuan University of Technology, Zhengzhou, Henan, 450007, P. R. China. Email: lzhoufeng@hotmail.com

1. Introduction

For most practical pattern recognition scenarios, feature extraction and classification methods are equally important. Feature extraction should retain most if not all of the useful information in the data while keeping the dimension of the features as low as possible. A careful choice of features is required to achieve low complexity in the classifier and a high accuracy in classification.

1.1 Compressive Sampling

Recent developments of compressive sampling (CS) theory give us clues for new methods of feature extraction. Namely, if the sparsity of the data is appropriately harnessed, then the data can be highly compressed by an underdetermined random projection (defined by a full rank random matrix whose row number is less than its column number), to achieve a sampling rate even lower than the classical Nyquist rate without any information loss. The original data can be exactly recovered from the highly compressed measurements by the ℓ_1-norm minimization techniques [1–9].

More specifically, let $\mathbf{x} \in \mathbb{R}^D$ be a κ-sparse ($\kappa < D$) vector, i.e., $\mathbf{x}$ has at most κ nonzero entries, and let $\mathbf{\Phi} \in \mathbb{R}^{d\times D}$ ($d < D$) be a matrix, whose entries are Gaussian distributed (or more generally, Restricted Isometry Property compatible). Then $\mathbf{x}$ can be compressed as follows.

$$\widehat{\mathbf{x}} = \mathbf{\Phi}\mathbf{x} \tag{1}$$

where $\widehat{\mathbf{x}} \in \mathbb{R}^d$ is the vector of CS measurements.

Given $\widehat{\mathbf{x}}$ and $\mathbf{\Phi}$, there are an infinite number of vectors $\mathbf{x}$ that satisfy Equation (1). However, it has been proved that if $d \geqslant O(\kappa \log(\frac{D}{\kappa}))$, then with overwhelming high probability

$$p \geqslant 1 - \exp O(-d) \tag{2}$$

$\mathbf{x}$ can be exactly recovered from $\widehat{\mathbf{x}}$ by minimizing the ℓ_0-norm of $\mathbf{x}$ as follows [1,2].

$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^D} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \widehat{\mathbf{x}} = \mathbf{\Phi}\mathbf{x} \tag{3}$$

where $\mathbf{x}^*$ is the recovered version of $\mathbf{x}$.

Because the optimization problem of Equation (3) is NP-hard, the recovery of $\mathbf{x}$ is equivalently reformulated as the ℓ_1-norm minimization problem as follows.

$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^D} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \widehat{\mathbf{x}} = \mathbf{\Phi}\mathbf{x} \tag{4}$$

This problem can be solved by algorithms such as Basis Pursuit [10] or Orthogonal Matching Pursuit [11].

Since the data dimension can be efficiently reduced without significant information loss, the above mentioned projection technique serves as a tool for feature extraction. Mathematical analyses show that compressive recognition, detection and other processings in compression domain $\mathbb{R}^d$ are feasible [12–20].

Note that the above mentioned projection technique is applied to vectors. For image data which are naturally represented by matrices, the 1D representation discards structural information about the image.

Due to this concern, different 2D (matrix) representations are exploited for feature extraction. For example, the 2D representation methods include 2DPCA [21] and its variants [22–24], 2DLDA [25], bilinear subspace learning [26,27], tensor analysis [28–31], and the recent common interest in 2D random projection [32–34].

Among the 2D representations, either supervised or unsupervised, 2DPCA, 2DLDA and bilinear subspace learning, etc., are obtained by deterministic two-dimensional projection. Another category of 2D representation include those obtained by random linear projection [32–34]. Both categories are actually the order-two tensor analyses, which exploit the correlations among image pixels with different dimensions and in this sense is believe to lead to good classification performance for different applications such as image recognition and human gait recognition [28–31].

1.2 NN, NFL and NS

Besides the feature extraction, classifier design is equally important. Classical but still popular subspace-based classifiers include NN (Nearest Neighbor), NFL (Nearest Feature Line, proposed by Stan Z. Li et al [35]) and NS (Nearest Subspace).

NN, NFL and NS share some common traits and can be summarized in a generalized way — given a query sample $\mathbf{y}$ and n training samples belonging to K classes, NN, NFL and NS all use on the same strategy to determine the class of $\mathbf{y}$ as follows.

$$\begin{cases} r_i(\mathbf{y}) = \min_{\mathbf{x}\in\mathbb{M}_i} \|\mathbf{y}-\mathbf{x}\|_2, \quad \forall i = 1, \cdots, K \\ \text{class}(\mathbf{y}) = \text{argmin}_{i\in\{1,\cdots,K\}}\, r_i(\mathbf{y}) \end{cases} \tag{5}$$

where $r_i(\mathbf{y})$ is the distance of $\mathbf{y}$ to class i and $\mathbb{M}_i$ is a classifier-specific dataset defined by training set i.

Denoting the i-th training set by $\mathbb{X}_i = \left\{\mathbf{x}_i^{(1)}, \cdots, \mathbf{x}_i^{(n_i)}\right\}$, in NN, $\mathbb{M}_i$ is the i-th training set itself, namely

$$\mathbb{M}_i = \mathbb{X}_i \tag{6}$$

In NFL, $\mathbb{M}_i$ is a set of feature lines defined by $\mathbb{X}_i$, namely,

$$\mathbb{M}_i = \left\{\alpha\mathbf{x}^{(a)} + (1-\alpha)\mathbf{x}^{(b)} |\ \alpha \in \mathbb{R}, \ \ \mathbf{x}^{(a)}, \mathbf{x}^{(b)} \in \mathbb{X}_i\right\} \tag{7}$$

In NS, $\mathbb{M}_i$ is the linear subspace spanned by $\mathbf{x}_i^{(1)}, \cdots, \mathbf{x}_i^{(n_i)}$. Denote the matrix whos columns are the training samples of the i-th class by

$$\mathbf{A}_i = \left[\mathbf{x}_i^{(1)}, \cdots, \mathbf{x}_i^{(n_i)}\right] \tag{8}$$

then, in NS, $\mathbb{M}_i$ can be written as follows.

$$\mathbb{M}_i = \{\mathbf{A}_i\boldsymbol{\alpha} |\ \boldsymbol{\alpha} \in \mathbb{R}^{n_i}\} \tag{9}$$

It follows from Equations (6)–(9) that in all cases we have $\mathbb{X}_i \subseteq \mathbb{M}_i$. For notation convenience, we respectively denote the training superset $\mathbb{M}_i$ of NN, NFL and NS as $\mathbb{M}_i^{\text{NN}}$, $\mathbb{M}_i^{\text{NFL}}$ and $\mathbb{M}_i^{\text{NS}}$. It is not difficult to see that $\mathbb{M}_i^{\text{NN}} \subset \mathbb{M}_i^{\text{NFL}} \subset \mathbb{M}_i^{\text{NS}}$. Since $\mathbb{M}_i^{\text{NS}}$ is a linear subspace and $\mathbb{M}_i^{\text{NN}}$ and $\mathbb{M}_i^{\text{NFL}}$ are just the appropriate subsets of it, we call $\mathbb{M}_i^{\text{NN}}$ and $\mathbb{M}_i^{\text{NFL}}$ *the constrained subspaces* for the i-th class.

1.3 NM and its Relationship to NN, NFL, NS

In NM (Nearest Manifold), it is assumed that the data of a class lie on or near to a manifold, and that the dimension of the manifold is much less than the dimension of the feature space.

NM uses the same strategy of Equation (5) to classify $\mathbf{y}$ with

$$\mathbb{M}_i = \mathcal{M}_i, \quad i = 1, \cdots, K. \tag{10}$$

where $\mathcal{M}_i$ is the data manifold of the i-th class.

If suitable manifolds for all $i = 1, \cdots, K$ can be found, then NM has a high classification accuracy.

Note that $\mathbb{M}_i$ in Equations (6), (7) and (9) can be viewed as different approximations to $\mathcal{M}_i$ for the i-th class. From this perspective, we contend that NN, NFL and NS are all approximations to NM and propose later a novel classifier, called NCSC (Nearest Constrained Subspace Classifier), and show by experiments that NCSC is a better approximation to NM than NN, NFL and NS.

1.4 Contributions of This Study

Based on our previous work [34], we discuss the technique of 2DCS (two-dimensional compressive sampling), which is inspired by 1DCS (traditional compressive sampling) and 2DPCA [21]. The 2D (matrix based) approach is computationally less complex than the 1D (vector based) approach to image data. The reconstruction of the original data is still guaranteed with a high probability. In this sense, 2DCS is more efficient than 1DCS for feature extraction. Our experiments show that when 2DCS features are exploited by some state-of-the-art classifiers, the performance of image recognition is improved.

This interest is somehow shared almost at the same time by A. Eftekhari et al [32] and L. Leng et al [33]. Although addressing the same problem, the focuses of A. Eftekhari, L. Leng et al. [32,33] and ours are different. Besides the theoretical analysis of 2D random projection and the assumption that 2D signal is sparse, A. Eftekhari et al. reported a reconstruction algorithm of 2D sparse signal based on smoothed ℓ_0-norm minimization. A reconstruction method of natural images (not explicitly sparse) and the problem of designing a cutting-edge classifier exploiting the 2D random projection features were not addressed in [32]. On the other hand, in [33], 2D random projection and its variations combined with PCA, LDA etc, were studied and compared with other feature extractors but without mention of the problems of 2D reconstruction and classifier design.

In our work, we propose a two-steps (including row processing and column processing) 2DCS reconstruction scheme for natural images via TV minimization. We also propose a classifier called NCSC (Nearest Constrained Subspace Classifier) and its fast version called NCSC-II, in which the subspace associated with the target class is constrainedly spanned by training samples. The constrained subspace is a union of a series of affine hulls.

We prove that NCSC is a generalized version of NN (Nearest Neighbor), NFL (Nearest Feature Line) and has a close relationship with NS (Nearest Subspace). Employing the intrinsic dimension as a freedom degree parameter, the constrained subspace, rather than the unconstrained one, is believed to be a more accurate approximation to the data manifold. The intrinsic dimension of the constrained subspace in NCSC is defined by a ℓ_0-norm sparse representation, and NCSC itself is in fact an approximation to the conceptual NM (Nearest Manifold) classifier, which is believed to be the optimal classifier using the nearest distance as the proximity measurement.

2. 2DCS: Two-Dimensional Compressive Sampling

In 1DCS, images are first recast as vectors and then projected to a lower dimensional space, namely image $\boldsymbol{x} \in \mathbb{R}^{M\times N}$ is represented by vector $\boldsymbol{x}_{1\mathrm{D}} \in \mathbb{R}^{MN}$.

Alternatively, we propose that the matrix $\boldsymbol{x} \in \mathbb{R}^{M\times N}$ can be projected by a column-wise approach using a matrix $\boldsymbol{\Phi}_1 \in \mathbb{R}^{m\times M}$ $(m < M)$ as follows [34].

$$\boldsymbol{z} = \boldsymbol{\Phi}_1 \boldsymbol{x} \tag{11}$$

After Equation (11), the row number of $\boldsymbol{z}$ is reduced to m. In the context of 2DCS, we call Equation (11) the step of “row compression”.

Similarly, the right multiplication of $\boldsymbol{z}$ by $\boldsymbol{\Phi}_2 \in \mathbb{R}^{N\times n}$ $(n < N)$ leads to “column compression”, yielding a matrix $\boldsymbol{y} \in \mathbb{R}^{m\times n}$ as follows.

$$\boldsymbol{y} = \boldsymbol{\Phi}_1 \boldsymbol{x} \boldsymbol{\Phi}_2. \tag{12}$$

Due to its similarity to 1DCS, we call Equation (12) 2DCS (two-dimensional compressive sampling). As a kind of stepwise implementation of 1DCS, 2DCS reduces feature extraction to a series of subtasks. Thus, the computational complexity of 2DCS is significantly less than that of 1DCS, which is superlinear function of the input scale.

If the sparsity of $\boldsymbol{x}$ is appropriately harnessed, the reconstruction of $\boldsymbol{x}$ from $\boldsymbol{y}$ is guaranteed.

The 2DCS reconstruction requires two steps of reconstructions, i.e., the column reconstruction and row reconstruction as follows.

(S1) Column reconstruction

$$\begin{aligned} \boldsymbol{z}^*_{\mathrm{row},i} &= \mathrm{argmin}_{\boldsymbol{z}_{\mathrm{row},i} \in \mathbb{R}^{1\times N}} \|\Psi(\boldsymbol{z}_{\mathrm{row},i})\|_1 \\ \text{subject to} \quad & \boldsymbol{y}_i = \boldsymbol{z}_{\mathrm{row},i} \boldsymbol{\Phi}_2 \quad \forall i = 1, 2, \cdots, m. \end{aligned} \tag{13}$$

where $\boldsymbol{z}^*_{\mathrm{row},i}$ is the i-th recovered row of $\boldsymbol{z} \doteq \boldsymbol{\Phi}_1 \boldsymbol{x}$, $\boldsymbol{y}_i$ is the i-th row of $\boldsymbol{y}$ and $\Psi(\cdot)$ is a sparsifying transformation, which transforms a target vector or matrix (not explicitly sparse) to sparse one. For image data, Ψ could be TV (Total Variation) transform. If the target vector $\boldsymbol{x}$ is already sparse itself, then Ψ is the identity transformation.

After the above step, $\boldsymbol{z} \doteq \mathbf{\Phi}_1\boldsymbol{x}$ is recovered.

(S2) Row reconstruction

$$\begin{aligned} \boldsymbol{x}_j^* &= \operatorname{argmin}_{\boldsymbol{x}_j \in \mathbb{R}^M} \|\Psi(\boldsymbol{x}_j)\|_1 \\ \text{subject to} \quad \boldsymbol{z}_{\text{col},j} &= \mathbf{\Phi}_1\boldsymbol{x}_j \quad \forall j = 1, 2, \cdots, N. \end{aligned} \tag{14}$$

where $\boldsymbol{x}_j{}^*$ is the recovered j-th column of $\boldsymbol{x}$ and $\boldsymbol{z}_{\text{col},j}$ is the j-th column of $\boldsymbol{z}$.

After the two steps, $\boldsymbol{x}$ is recovered.

To be more specific, given (column) vector $\mathbf{u}$, which is not explicitly sparse (e.g., $\mathbf{u}$ is a vector from image data), and its measurements $\mathbf{b} = \mathbf{Du}$, the reconstruction of $\mathbf{u}$ via $\mathbf{b}$, $\mathbf{D}$ can be implemented via TV (Total Variation) minimization [36]. TV minimization is defined as follows.

$$\mathbf{u}^* = \underset{\mathbf{u}}{\operatorname{argmin}} \sum\nolimits_i \|\Delta_i(\mathbf{u})\|_1 \quad \text{subject to} \quad \mathbf{Du} = \mathbf{b} \tag{15}$$

where $\Delta_i(\mathbf{u})$ is the discrete gradient vector of $\mathbf{u}$ at position i.

Hereinafter, given projection matrix $\mathbf{D}$ and vector $\mathbf{b}$, we denote the solution of Equation (15) by $\text{TV}(\mathbf{D}, \mathbf{b})$. Thus, we summarize our algorithm of 2DCS image reconstruction via TV minimization as Algorithm 1.

Algorithm 1 2DCS Image Reconstruction via TV minimization

```
Input: Projection matrices $\mathbf{\Phi}_1 \in \mathbb{R}^{m\times M}$ , $\mathbf{\Phi}_2 \in \mathbb{R}^{N\times n}$ and $\boldsymbol{y} \in \mathbb{R}^{m\times n}$.
Output: Reconstructed $\boldsymbol{x} \in \mathbb{R}^{M\times N}$.
$\mathbf{Y} \leftarrow \boldsymbol{y}^T$, $\mathbf{D} \leftarrow \mathbf{\Phi}_2^T$;
for $i \leftarrow 1$ to $m$ do                      ▷ Column Reconstruction
    $\mathbf{b} \leftarrow \mathbf{Y}(i)$;          ▷ $\mathbf{Y}(i)$ is the $i$-th column of matrix $\mathbf{Y}$
    $\mathbf{U}(i) \leftarrow \text{TV}(\mathbf{D}, \mathbf{b})$;   ▷ $\mathbf{U}(i)$ is the $i$-th column of matrix $\mathbf{U}$
end for
$\boldsymbol{z} \leftarrow \mathbf{U}^T$;            ▷ Column Reconstruction Completed
$\mathbf{D} \leftarrow \mathbf{\Phi}_1$;
for $i \leftarrow 1$ to $N$ do                      ▷ Row Reconstruction
    $\mathbf{b} \leftarrow \boldsymbol{z}(i)$;      ▷ $\boldsymbol{z}(i)$ is the $i$-th column of matrix $\boldsymbol{z}$
    $\boldsymbol{x}(i) \leftarrow \text{TV}(\mathbf{D}, \mathbf{b})$;   ▷ $\boldsymbol{x}(i)$ is the $i$-th column of matrix $\boldsymbol{x}$
end for
return $\boldsymbol{x}$;                             ▷ Row Reconstruction Completed
```

3. NCSC: Nearest Constrained Subspace Classifier

In this section, we extend NN, NFL and NS to a unified classifier called NCSC (Nearest Constrained Subspace Classifier), in which, the employed *constrained subspaces* with the tuned intrinsic dimension parameter are better approximations to the data manifolds than those of NN, NFL and NS.

3.1 Manifold Perspective and Manifold Approximation

From the geometric point of view, the vectors representing the natural images of the same class generally reside on (or near to) a low dimensional geometric structure known as manifold, embedded in the high dimensional feature space [37–39]. If the data manifolds for all the classes can be learned, then it would be possible to design more effective classifiers. The concept of manifold has long been a powerful analytical tool for understanding image classes, for example images of human face or handwritten digits [40–42].

In the last decade, some well-known manifold learning algorithms have emerged, such as ISOMAP [37], LLE (Local Linear Embedding) [38], Laplacian Eigenmap [39], Hessian Eigenmaps (HLLE) [43], Maximum Variance Unfolding (MVU) [44] and Local Tangent Space Alignment (LTSA) [45]. However they are not designed to solve the problem of classifying new images. Although there are some works which attempt to deal with this problem [46–48], the algorithms are all unsupervised and designed for a single manifold, not for multiple manifolds. These algorithms are unsuitable for supervised multi-class classification, in which each class is modeled by a manifold.

As discussed in Section 1.3, NM (Nearest Manifold), with the nearest distance criterion, is believed to be optimal in terms of classification accuracy. But due to the unavailability of NM, we argue that some approximation strategies should be exploited. Since the training data are the points on manifolds, if there are enough well-distributed training data, then the manifold can be accurately approximated.

From this viewpoint, we argue, to achieve an accurate manifold approximation, it is necessary to make the intrinsic dimension of $\mathbb{M}_i$ equal to the intrinsic dimension of manifold $\mathcal{M}_i$ ($\forall i = 1, \cdots, K$, given K classes). Otherwise the accuracy of the approximation to the manifold can not be guaranteed. We call this criterion the dimension equality.

3.2 Nearest Constrained Subspace Classifier

We call the subspace generated with the dimension equality constraint the constrained subspace and the corresponding classifier with the nearest distance criterion the Nearest Constrained Subspace Classifier (NCSC). Here, we discuss the concepts of constrained subspace and NCSC in detail as follows.

Denoting the i-th training set by matrix $\mathbf{A}_i = \left[\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \cdots, \mathbf{x}_i^{(n_i)}\right]$, the points in the constrained subspace are given by $\mathbf{A}_i\boldsymbol{\alpha}$ (where $\boldsymbol{\alpha} \in \mathbb{R}^{n_i}$) with some constraints imposed on $\boldsymbol{\alpha}$.

The first constraint is

$$\mathbf{1}^T\boldsymbol{\alpha} = 1 \tag{16}$$

This constraint, preventing the vector rotation and rescaling effect, means each training data point $\mathbf{x}_i^{(1)}, \cdots, \mathbf{x}_i^{(n_i)}$ serves as a descriptor of $\mathbf{A}_i\boldsymbol{\alpha}$.

We also contend that $\boldsymbol{\alpha}$ should be sparse in order to make the intrinsic dimension of the constrained subspace equal to that of the manifold. We use the ℓ_0-norm of $\boldsymbol{\alpha}$ to model this sparsity, namely, $\|\boldsymbol{\alpha}\|_0 \leqslant \kappa$, where κ is a sparsity parameter and defines the intrinsic dimension (freedom degree) of the constrained subspace.

Based on the above two constraints on $\boldsymbol{\alpha}$, in NCSC, $r(\mathbf{y})$ is written as follows.

$$r_i(\mathbf{y}) = \min_{\boldsymbol{\alpha}} \|\mathbf{y} - \mathbf{A}_i\boldsymbol{\alpha}\|_2 \quad \text{subject to} \quad \mathbf{1}^T\boldsymbol{\alpha} = 1 \quad \text{and} \quad \|\boldsymbol{\alpha}\|_0 \leqslant \kappa \leqslant n_i \tag{17}$$

The constraint $\|\boldsymbol{\alpha}\|_0 \leqslant \kappa$ ensures that at most κ columns in $\mathbf{A}_i$ at the same time contribute to point $\mathbf{A}_i\boldsymbol{\alpha}$. Since $\boldsymbol{\alpha}$ has n_i entries, there are $\binom{n_i}{\kappa}$ κ-combinations of the training vectors of class i. Given the training set of class i denoted by $\left\{\mathbf{x}_i^{(1)}, \cdots, \mathbf{x}_i^{(n_i)}\right\}$, for the j-th κ-combination, we define the base matrix as follows.

$$\mathbf{W}_{i,j} = \left[\mathbf{w}_{i,j}^{(1)}, \cdots, \mathbf{w}_{i,j}^{(\kappa)}\right] \tag{18}$$

where $\left\{\mathbf{w}_{i,j}^{(1)}, \cdots, \mathbf{w}_{i,j}^{(\kappa)}\right\}$ is the j-th κ-combination of the training vectors of class i.

Then, the subproblem of Equation (17) for $\mathbf{W}_{i,j}$ can be written as follows.

$$r_i^{(j)}(\mathbf{y}) = \min_{\boldsymbol{\beta}\in\mathbb{R}^\kappa} \|\mathbf{y} - \mathbf{W}_{i,j}\boldsymbol{\beta}\|_2 \quad \text{subject to} \quad \mathbf{1}^T\boldsymbol{\beta} = 1 \tag{19}$$

There are very mature algorithms for solving Equation (19). Interested readers are referred to [49–51] for more details.

After calculating Equation (19) for all $j = 1, \cdots, \binom{n_i}{\kappa}$, $r_i(\mathbf{y})$ is defined by

$$r_i(\mathbf{y}) = \min_j r_i^{(j)}(\mathbf{y}) \tag{20}$$

Based on the above discussion, we summarize NCSC as Algorithm 2.

Algorithm 2 NCSC: Nearest Constrained Subspace Classifier

```
Input: A query sample $\mathbf{y}$, training vectors partitioned to $K$ classes and parameter $\kappa$.
Output: Class ID of $\mathbf{y}$.
for $i \leftarrow 1$ to $K$ do
    for $i \leftarrow 1$ to $\binom{n_i}{\kappa}$ do
        Obtain $\mathbf{W}_{i,j}$ as in Equation (18);
        Calculate $r_i^{(j)}(\mathbf{y})$ as in Equation (19);
    end for
end for
return class($\mathbf{y}$) $\leftarrow$ argmin$_i$ $r_i(\mathbf{y})$;
```

Note the ℓ_0-norm sparse representation employed by NCSC is closely associated with the intrinsic dimension of manifold and we discuss this issue in detail in Section 3.4.

3.3 Union of Affine Hulls

In NCSC, each constrained subspace is a union of affine hulls. Here we give the explanation of this claim.

Given K classes, there are K constrained subspaces. The i-th constrained subspace $\mathbb{M}_i^{\mathrm{NCSC}}$ is written as follows.

$$\mathbb{M}_i^{\mathrm{NCSC}} = \left\{\mathbf{A}_i\boldsymbol{\alpha} \middle|\ \boldsymbol{\alpha} \in \mathbb{R}^{n_i}, \|\boldsymbol{\alpha}\|_0 \leqslant \kappa \leqslant n_i \text{ and } \mathbf{1}^T\boldsymbol{\alpha} = 1\right\} \tag{21}$$

Since the solution of Equation (17) can be divided into the $\binom{n_i}{\kappa}$ solutions of Equation (19), it is not difficult to see that $\mathbb{M}_i^{\mathrm{NCSC}}$ can be rewritten as follows.

$$\begin{cases} \mathbb{M}_i^{\mathrm{NCSC}} = \bigcup_{j=1}^{\binom{n_i}{\kappa}} \mathcal{H}_{i,j} \\ \mathcal{H}_{i,j} = \left\{\mathbf{W}_{i,j}\boldsymbol{\beta} \middle|\ \boldsymbol{\beta} \in \mathbb{R}^{\kappa} \text{ and } \mathbf{1}^T\boldsymbol{\beta} = 1\right\} \end{cases} \tag{22}$$

where $\mathcal{H}_{i,j}$ is the affine hull of the column vectors in $\mathbf{W}_{i,j}$.

Note that affine hull (also known as affine subspace) is also referred to as "linear manifold". This means that $\mathbb{M}_i^{\mathrm{NCSC}}$ can be viewed as an approximation to $\mathcal{M}_i$ by a series of linear manifolds.

It is also worth noticing that when $\kappa = 1$, $\mathbb{M}_i^{\mathrm{NCSC}}$ becomes $\mathbb{M}_i^{\mathrm{NN}}$, whose intrinsic dimension is 0. When $\kappa = 2$, $\mathbb{M}_i^{\mathrm{NCSC}}$ becomes $\mathbb{M}_i^{\mathrm{NFL}}$, whose intrinsic dimension is 1. Thus, NN and NFL are just two low-dimensional special cases of NCSC.

In NN (i.e., NCSC with $\kappa = 1$), the constrained subspace for a specific class (e.g., class i) is a vector set, i.e., the training set of that class. In NFL (i.e., NCSC with $\kappa = 2$), for a specific class, the constrained subspace is a set of feature lines, where each feature line is interpolated and extrapolated from a pair of training samples. When $\kappa = 3$, the constrained subspace for a specific class is a set of feature planes spanned by any triplet of the training samples. From this sense, NCSC with $\kappa = 3$ can be called NFP (Nearest Feature Plane).

Figure 1 gives the constrained subspace interpretations of NN, NFL and NFP. Without losing generality and for demonstration convenience, in Figure 1, we set $n_i = 3$. The intrinsic dimensions of the constrained subspaces in Figure 1(a)–Figure 1(c) are respectively $0, 1$ and 2.

3.4 Intrinsic Dimension

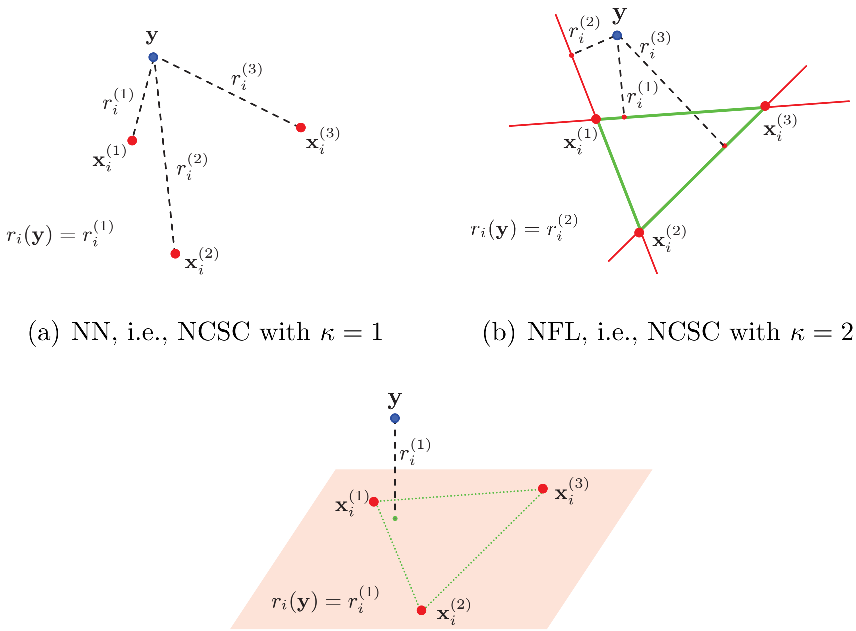
Based on the above discussions, it is easy to find that if the training samples $\mathbf{x}_i^{(1)}, \cdots, \mathbf{x}_i^{(n_i)}$ of class i are linearly independent, the intrinsic dimension of $\mathbb{M}_i^{\mathrm{NCSC}}$ is given by

$$\mathrm{Dim}(\mathbb{M}_i^{\mathrm{NCSC}}) = \kappa - 1 \geqslant 0, \quad \forall i = 1, \cdots, K \tag{23}$$

The linear independence of $\mathbf{x}_i^{(1)}, \cdots, \mathbf{x}_i^{(n_i)}$ is satisfied in many pattern recognition problems if the feature dimension D is large enough. Hereinafter, unless otherwise stated, we assume that the training samples of each class are linear independent.

As mentioned in Section 3.1, in order to make $\mathbb{M}_i^{\mathrm{NCSC}}$ a more accurate approximation to $\mathcal{M}_i$, at least their intrinsic dimensions should be equal, namely,

$$\mathrm{Dim}(\mathbb{M}_i^{\mathrm{NCSC}}) = \mathrm{Dim}(\mathcal{M}_i) \tag{24}$$



(a) NN, i.e., NCSC with $\kappa = 1$

(b) NFL, i.e., NCSC with $\kappa = 2$

(c) NCSC with $\kappa = 3$

Figure 1: Constrained subspaces respectively with $\kappa = 1, 2, 3$ for class i where $n_i = 3$ for demonstration convenience. The blue point stands for a query sample, the red points are the training samples of class i and $r_i(\mathbf{y})$ is the distance from the query point to that class. (a) The NN (Nearest Neighbor) case is intrinsically zero-dimensional and equivalent to the NCSC case where no data interpolation and extrapolation is employed. (b) The NFL (Nearest Feature Line) case is actually the one-dimensional NCSC case with $\kappa = 2$. The green line segments are from the data interpolations of the training samples. The red segments are from the data extrapolations. (c) The NCSC case with $\kappa = 3$ is intrinsically two-dimensional. The interpolated points are confined inside the green borders of the feature point triplets.

In this study, we focus on the scenario in which the intrinsic dimensions of all classes are identical, i.e., $\mathrm{Dim}(\mathcal{M}_1) = \mathrm{Dim}(\mathcal{M}_2) = \cdots = \mathrm{Dim}(\mathcal{M}_K)$ for all K classes. We call the corresponding dataset homogeneous and denote the identical intrinsic dimension as D_m.

Thus, the optimal parameter κ in Algorithm 2 is given as follows.

$$\kappa = D_\mathrm{m} + 1 \leqslant n_i \tag{25}$$

where D_m can be estimated in advance from the dataset by other algorithms [52–57].

Based on Equation (25), we argue that for effectiveness of NCSC or even other classifiers, the number of training samples of a class should not be less than the intrinsic manifold dimension D_m. More concretely, the condition

$$n_i > \mathrm{Dim}(\mathcal{M}_i) \quad \forall i = 1, \cdots, K \tag{26}$$

should be satisfied for effective classifications. Otherwise, the training samples are not sufficient to capture the critical manifold properties and the classification accuracy are not guaranteed.

On the other hand, in NS, all subspaces are unconstrained. The intrinsic dimension of $\mathbb{M}_i^{\text{NS}}$ is given by

$$\text{Dim}(\mathbb{M}_i^{\text{NS}}) = n_i \tag{27}$$

For convenience, in this study we assume $n_1 = n_2 = \cdots = n_K = \mathcal{N}$ and denote the intrinsic dimension of subspaces in NS by D_{s}. Then, we have $D_{\text{s}} = \mathcal{N}$, which is the highest and closely related with the highest dimensional extreme of NCSC in which the constrained subspaces are $(\mathcal{N} - 1)$ dimensional. We argue that NS is generally not optimal in terms of classification accuracy because the intrinsic subspace dimension is unnecessarily high, especially when training set is large.

3.5 Computational Complexity of NCSC and Fast NCSC

As mentioned in Section 3.2 and Section 3.3, given n_i training samples of the i-th class, the solution of Equation (17) is divided into $\binom{n_i}{\kappa}$ solutions of Equation (19). When n_i is large and $\kappa \simeq \frac{n_i}{2}$, $\binom{n_i}{\kappa}$ can be huge. This makes Algorithm 2 computationally intractable.

To reduce the computational complexity, one strategy is to replace $\binom{n_i}{\kappa}$ with a smaller number. We observe that from the viewpoint of local manifold approximation, many of the κ-combinations of training samples are not necessary. Thus, some of the combinations can be removed. To do this, we assume that a target sample $\mathbf{x}_j$ and its nearest $(\kappa - 1)$ neighbors in the same class together define a local linear manifold. This neighborhood assumption is also employed in [52, 55] to estimate the intrinsic dimension of a dataset.

Via this assumption, we define the base matrix, whose columns are $\mathbf{x}_j$ and its $(\kappa - 1)$ nearest neighbors in the same class, as follows.

$$\mathbf{A}(\mathbf{x}_j) = [\mathbf{v}_1, \cdots, \mathbf{v}_\kappa] \tag{28}$$

where $\mathbf{x}_j \in \{\mathbf{v}_1, \cdots, \mathbf{v}_\kappa\}$ and $\{\mathbf{v}_1, \cdots, \mathbf{v}_\kappa\} \setminus \{\mathbf{x}_j\}$ is the neighborhood set, containing the $(\kappa - 1)$ nearest neighbors of $\mathbf{x}_j$ in the same class.

Based on Equation (28), we summarize fast NCSC, called NCSC-II, as Algorithm 3.

Algorithm 3 NCSC-II: fast NCSC via neighborhood representation

```
Input: A query sample $\mathbf{y}$, training vectors $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ partitioned to $K$ classes and parameter $\kappa$.
Output: Class ID of $\mathbf{y}$.
for $i \leftarrow 1$ to $n$ do
    Obtain $\mathbf{A}(\mathbf{x}_j)$ as in Equation (28);
    $r(\mathbf{y}, \mathbf{x}_j) \leftarrow \min_{\boldsymbol{\beta} \in \mathbb{R}^\kappa} \|\mathbf{y} - \mathbf{A}(\mathbf{x}_j)\boldsymbol{\beta}\|_2$ subject to $\mathbf{1}^T\boldsymbol{\beta} = 1$;
end for
return class($\mathbf{y}$) $\leftarrow$ class($\mathbf{x}_m$);
```

By the neighborhood representation, the computational complexity of Algorithm 3 is reduced to $O(n)$. But the properties/definitions (23), (24), (25) and (26) of NCSC still holds in NCSC-II.[1]

As mentioned before — given parameter κ and training samples $\mathbf{x}_i^{(1)}, \cdots, \mathbf{x}_i^{(n_i)}$ of class i, $\mathbb{M}_i^{\text{NCSC}}$ is formulated as the union of $\binom{n_i}{\kappa}$ affine hulls. In NCSC-II, most of the affine hulls are removed and the remaining ones define the constrained subspace $\mathbb{M}_i^{\text{NCSC-II}}$ as follows.

$$\begin{cases} \mathbb{M}_i^{\text{NCSC-II}} = \bigcup_{j=1}^{n_i} \mathcal{H}_{i,j} \\ \mathcal{H}_{i,j} = \left\{ \mathbf{A}(\mathbf{x}_i^{(j)})\boldsymbol{\beta} | \; \boldsymbol{\beta} \in \mathbb{R}^\kappa \text{ and } \mathbf{1}^T\boldsymbol{\beta} = 1 \right\} \end{cases} \tag{29}$$

Since most of the affine hulls in $\mathbb{M}_i^{\text{NCSC}}$ are removed to obtain $\mathbb{M}_i^{\text{NCSC-II}}$, $\mathbb{M}_i^{\text{NCSC-II}}$ is a sparse representation of $\mathbb{M}_i^{\text{NCSC}}$. Thus, NCSC-II is a sparse version of NCSC.

Moreover, note that in Algorithm 2 and Algorithm 3, there is only one parameter κ. This ensures that all constrained subspaces in NCSC/NCSC-II have the same intrinsic dimension. We call Algorithm 2/Algorithm 3 homogeneous NCSC/NCSC-II. It is possible to extend homogeneous NCSC/NCSC-II to inhomogeneous NCSC/NCSC-II by adopting multiple parameters for all classes or even by varying κ for different data samples. But in this work, we focus on homogeneous NCSC/NCSC-II and leave inhomogeneous NCSC/NCSC-II to a future investigation.

For homogeneous NCSC and NCSC-II, we contend that their classification accuracy is a function of the intrinsic dimension D_c of constrained subspaces where D_c is defined as follows.

$$D_\text{c} = \kappa - 1 \tag{30}$$

If denoting the classification accuracy function by $f(D_\text{c})$, then we have a empirical scheme for estimating D_m of a labeled data set as follows.

$$D_\text{m} = \underset{D_\text{c}}{\text{argmax}}\, f(D_\text{c}) \tag{31}$$

Equation (31) gives rise to two observations. First, given a labeled data set, D_m can be estimated by NCSC/NSCS-II. The second is that when D_m is learned, we have a tuned NCSC/NCSC-II, which yields a high classification accuracy.

4. Intrinsic Dimension Estimator

Although intrinsic dimension can be estimated by NCSC/NCSC-II as in Equation (31) on a training set, there are more sophisticated estimators available. These estimators can be broadly divided into two categories: eigen projection

[1] More specifically, in Equations (23) and (24), if replacing $\mathbb{M}_i^{\text{NCSC}}$ with $\mathbb{M}_i^{\text{NCSC-II}}$, the corresponding properties and definitions still hold.

methods [58, 59] and geometric methods [52–57]. Eigen projection methods estimate intrinsic dimension from the eigen decomposition of the covariance matrix of the give data. Their estimates are given as the number of eigenvalues not less than a predefined threshold. Geometric methods, including Corr.Dim (Correlation Dimension) [53, 54], MLE (Maximum Likelihood Estimate) [52] and their variations [55–57], exploit the intrinsic geometry of the dataset and are more sophisticated than their eigen projection counterparts [52].

4.1 Correlation Dimension Estimator

Given a data set $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, the correlation integral function of the Corr.Dim estimator is defined as the following.

$$C(r) = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} H(r - \|\mathbf{x}_i - \mathbf{x}_j\|_2) \tag{32}$$

where $H(\cdot)$ is a unit step function satisfying if $x \geqslant 0$, then $H(x) = 1$, otherwise, $H(x) = 0$.

The intrinsic dimension estimate by Corr.Dim is given by plotting $\ln C(r)$ against $\ln r$ and calculating the slope $\frac{\partial \ln C(r)}{\partial \ln r}$ of its linear part [53, 54]. If there is no prominent linear part in the curve of $\ln C(r)$ as a function of $\ln r$, we use the following estimate $\hat{D}_{\mathrm{m}}$.

$$\begin{cases} \hat{D}_{\mathrm{m}} = \dfrac{\ln C(r_2) - \ln C(r_1)}{\ln r_2 - \ln r_1} \\ r_1 = \min_{i,j \in \{1,\cdots,n\}, i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\|_2 \\ r_2 = \max_{i,j \in \{1,\cdots,n\}, i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\|_2 \end{cases} \tag{33}$$

Estimate $\hat{D}_{\mathrm{m}}$ is actually an average of the slopes of the curve of $\ln C(r)$ against $\ln C(r)$ at different locations.

4.2 Maximum Likelihood Estimator

Another estimator, MLE, estimates the intrinsic dimension under the assumption that the closest k neighbors to a given point $\mathbf{x}_i \in \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ lie on the same manifold (where k is a fixed number and $k > 2$). Estimate $\hat{D}_{\mathrm{m}}$ is given as follows [52].

$$\begin{cases} \hat{D}_{\mathrm{m}} = \dfrac{1}{(k_2 - k_1 + 1)n} \displaystyle\sum_{i=1}^{n} \sum_{k=k_1}^{k_2} \hat{C}_k(\mathbf{x}_i) \\ \hat{C}_k(\mathbf{x}_i) = \left[\dfrac{1}{k-2} \displaystyle\sum_{j=1}^{k-1} \ln \dfrac{T_k(\mathbf{x}_i)}{T_j(\mathbf{x}_i)} \right]^{-1} \end{cases} \tag{34}$$

where $T_k(\mathbf{x}_i)$ is the distance from $\mathbf{x}_i$ to its k-th nearest neighbor. $\hat{C}_k(\mathbf{x}_i)$ is the local dimension estimate at $\mathbf{x}_i$ using parameter k. $\hat{D}_{\mathrm{m}}$ is the average of $\hat{C}_k(\mathbf{x}_i)$ for $k \in \{k_1, \cdots, k_2\}$ (where $k_1 \leqslant k_2$) on samples $\mathbf{x}_1, \cdots, \mathbf{x}_n$.

There are many other estimators, but a comprehensive comparison of their performance is still an open problem. We focus on the Corr.Dim estimator and the MLE estimator and make a brief comparison of their effects on NCSC/NCSC-II.

5. Experimental Verifications and Discussions

In this section, we present our experimental results for a range of classifiers and features, on several publicly available datasets.

5.1 Experiments on NCSC/NCSC-II

First, we evaluate whether NCSC with $D_{\rm c} = D_{\rm m}$ outperforms its rivals including NN, NFL and NS. In order to evaluate NCSC/NCSC-II without biases due to complexities in the features, we keep the features simple. More specifically, in the following experiments on NCSC/NCSC-II, we subtract the means from each vectorized image (by concatenating the columns of the target image) and normalize the image vectors to have a unit ℓ_2-norm. Unless otherwise stated, the zero-mean vectors with a unit ℓ_2-norm are taken as the image features for the following classification experiments.

5.1.1 Evaluation of NCSC/NCSC-II for Face Recognition

First, we evaluate NCSC/NCSC-II on the PICS/PES dataset [60] and the ORL dataset [61] for face recognition. The PICS/PES dataset is relatively small and contains 84 cropped facial images from 12 subjects (7 images/subject × 12 subjects). The image size is 241 × 181 pixels. Figure 2 shows the image samples of two subjects from the PICS/PES dataset.

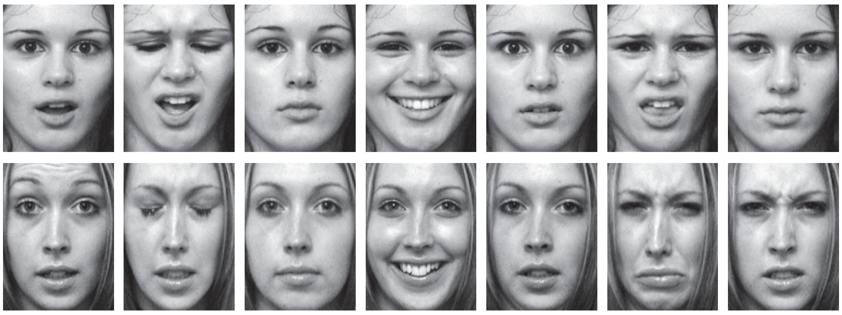


Figure 2: Image examples of two subjects in the PICS/PES dataset.

The ORL dataset contains 400 facial images from 40 subjects (10 images/subject × 40 subjects). The image size is 112 × 92 pixels. Figure 3 shows the image examples of one subject from the ORL data set.

In order to have a significant number of queries to obtain the classification accuracy, the experiment contains multiple classification rounds. In each round, n_i images are randomly chosen from each subject as the training samples, the

Figure 3: Image examples of a subject in the ORL dataset.

remaining images are chosen as query samples. In order to obtain sufficient information about the data manifold, we keep the training set as large as possible. More specifically, $n_i = 6$ for the PICS/PES dataset, namely, the training set contains 72 images (6 images/subject × 12 subjects). The remaining 12 images (1 image/subject × 12 subjects) are query images. For the ORL dataset, $n_i = 9$, namely, the training set contains 360 random images (9 images/subject × 40 subjects), the remaining 40 images (1 image/subject × 40 subjects) are query images. After a classification round, we randomly select the training samples and query samples again for another round. After enough rounds, the classification accuracy $f(D_\mathrm{c})$ is given as follows.

$$f(D_\mathrm{c}) = \frac{w}{W} \tag{35}$$

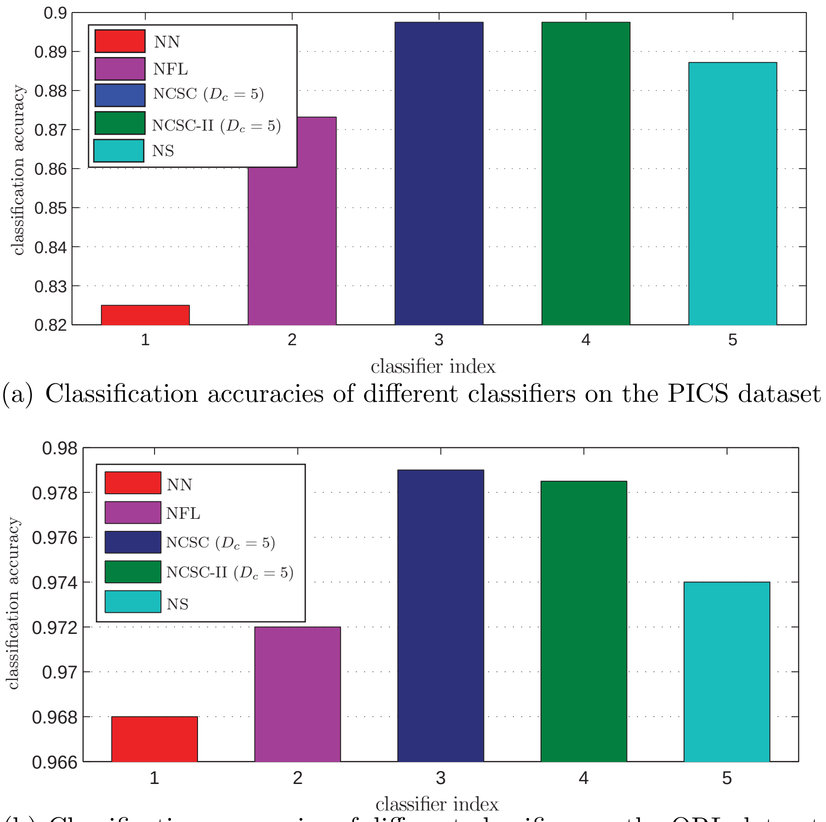
where w is the number of the correctly classification query samples and W is the total number of query samples in all classification rounds.

For a fair comparison and to avoid unnecessary classification accuracy perturbation, the random selections of training set and query set are preserved and repeated for different classifiers. For the PICS data set, $W = 6000$ (i.e., 12 samples/round × 500 rounds) and for the ORL data set, $W = 8000$ (i.e., 40 samples/round × 200 rounds).

Figure 4 gives the classification accuracies of NN, NFL, NCSC, NCSC-II and NS on the PICS and ORL data set. It is evident that NCSC and NCSC-II with $D_\mathrm{c} = 5$ estimated by MLE (Corr.Dim gives the same estimate of D_c) outperform NN, NFL and NS in terms of classification accuracy. The classification accuracies of NCSC and NCSC-II are comparably the highest.

5.1.2 Evaluation of NCSC-II for Digit Recognition

In the following experiment, we compare the classification accuracies of different classifiers on the MNIST dataset [62] with a large training set for digit recognition. The MNIST dataset of handwritten digits contains 60000 training images and 10000 test images from 10 classes. Each of the images has been size-normalized and centered. The image size is 28 × 28 pixels. Figure 5 shows some samples of the MNIST dataset.



(a) Classification accuracies of different classifiers on the PICS dataset

(b) Classification accuracies of different classifiers on the ORL dataset

Figure 4: Classification accuracies for different classifiers on the PICS and ORL dataset. Classifier with index from 1 to 5 is respectively NN, NFL, NCSC ($D_\mathrm{c} = 5$), NCSC-II ($D_\mathrm{c} = 5$) and NS.

The MNIST dataset has a large number of training samples. Due to the concern of computational complexity, we use NCSC-II rather than NCSC to classify the query samples. We use the first 10% samples of the MNIST dataset for classifier evaluations. More specifically, the training set contains the first 6000 images (600 images/class × 10 classes) and the query set contains the first 1000 images (100 images/class × 10 classes).

A variety of classifiers are evaluated for classifying not only the above mentioned 1000 query images, but also their corrupted versions under different noise levels $\rho = 0.1$, 0.2 and 0.3.

The corrupted pixels are randomly and uniformly chosen in a target query image. The number of corrupted pixels is the round integer of $28 \times 28 \times \rho$. The corruption intensities are uniformly distributed in $\{i_{\min}, \cdots, i_{\max}\}$, where $i_{\min}$ and $i_{\max}$ respectively denote the minimum and maximum intensity of the uncorrupted image.

Figure 6 gives some image samples and their corrupted versions.

Table 1 gives the classification accuracies of a variety of classifiers under noisy environment. Since NFL is NCSC with $\kappa = 2$, the combination number of NFL is $\binom{n_i}{\kappa} = \binom{600}{2} = 179700$. This number is too large for classifying query samples in

Figure 5: Examples of the MNIST dataset.

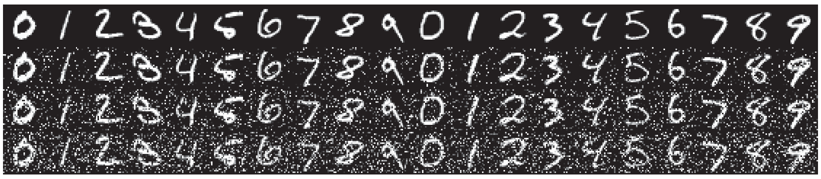


Figure 6: Some query samples and their corrupted versions. First row: uncorrupted samples. Second row: corrupted samples with noise level $\rho = 0.1$. Third row: corrupted samples with noise level $\rho = 0.2$. Bottom row: corrupted samples with noise level $\rho = 0.3$.

an acceptable time. Thus, alternatively, we evaluate NCSC-II with $\kappa = 2$, which is actually the fast version of NFL and yields the third highest classification accuracy (0.934) in Table 1.

Table 1: Comparison of classification accuracy of several subspace-based classifiers.

Noise Level		0%	10%	20%	30%
NN		0.923	0.922	0.920	0.919
NFL #		—			
NCSC-II	$D_c = 1$ †	0.934	0.932	0.927	0.920
	$D_c = 4$ (Corr.Dim)	0.949	0.946	0.945	0.922
	$D_c = 7$ (MLE) *	0.955 ‡	0.948	0.947	0.923
NS		0.547	0.125	0.123	0.108

# Unable to obtain experimental results in an acceptable time.
† Corresponding to the fast NFL classifier, i.e., NCSC-II with $\kappa = 2$.
* Corresponding to the optimal classification performance, using D_c estimated by MLE.
‡ The highest classification accuracy in this experiment.

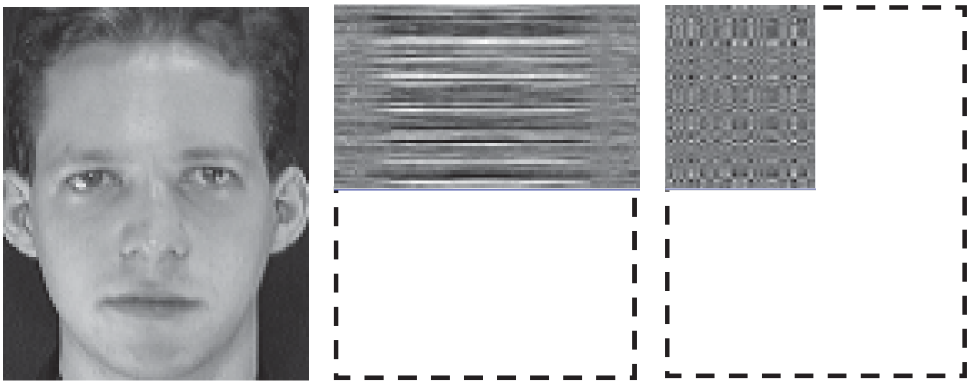
The intrinsic dimension estimates on the MNIST dataset by Corr.Dim and MLE are respectively 4 (by Corr.Dim) and 7 (by MLE). It is shown in Table 1 that with $D_c = 4$ (i.e., $\kappa = 5$) and 7 (i.e., $\kappa = 8$), NCSC-II yields better classification accuracies than the other algorithms. The highest classification accuracy is obtained by NCSC-II with $D_c = 7$. The classification accuracy of NCSC-II (or NCSC) depends on the accuracy of the intrinsic dimension estimate. Intrinsic dimension estimates via Equations (33) and (34) are actually estimate averages. Thus, we argue that the classification accuracy of NCSC-II with $D_{\mathrm{c}} = 7$ is not necessarily the upper-bound accuracy.

Also worth noticing is that with the large training training set (600 images/class × 10 classes), NS yields the worst results with its classification accuracies not larger than 0.547. We argue that it is primarily due to the unnecessary high intrinsic dimensions of the spanned subspaces employed by NS. The subspaces have nontrivial intersections which lead to poor classification accuracies.

5.1.3 Experiments on 2DCS Features

In this section, we evaluate the 2DCS features for image classification.

Figure 7 gives an example of 2DCS row and column compression. The row and column numbers have both been halved. The compression ratio $r = 25\%$ is defined as the ratio of the compressed size to the original size (i.e., $r = \frac{mn}{MN}$).



(a) Original image (b) Row compression (c) 2DCS result

Figure 7: An example of 2DCS compression. (a) Original image, image size = $M \times N = 112 \times 92$ pixels (b) Result of row compression with $r = 50\%$ (c) Final 2DCS result with $r = 25\%$

Table 2 gives the average run time of 1DCS and 2DCS projections of an ORL image (112 × 92 pixels). Each time is an average of the times from 1000 independent experiments.

The run time of 2DCS is much less than that of 1DCS. Our experiment also shows that when $r > 50\%$, 2DCS can be computed, but 1DCS needs a very large projection matrix, which leads MATLAB throw an "out of memory" exception.[2]

Figure 8 shows an original image and a variety of its reconstructions for different compression ratios r. Figure 8(a) is the original image and Figure 8(b)–Figure 8(f) are the reconstructions respectively with r equal to 10%, 30%, 50%, 70%, 90%. It is evident that a larger yields a higher reconstruction quality.

[2] The environment for this experiment is MATLAB R2009a on a SAMSUNG x86 notebook PC.

Table 2: Time Comparison of 1DCS and 2DCS

Method	Times (ms)					
	$r = 0.3\%$	$r = 0.5\%$	$r = 1.0\%$	$r = 1.5\%$	$r = 2.5\%$	$r = 10\%$
1DCS	16.41	30.51	63.69	117.97	226.49	837.74
2DCS	0.28	0.29	0.32	0.45	0.58	1.47

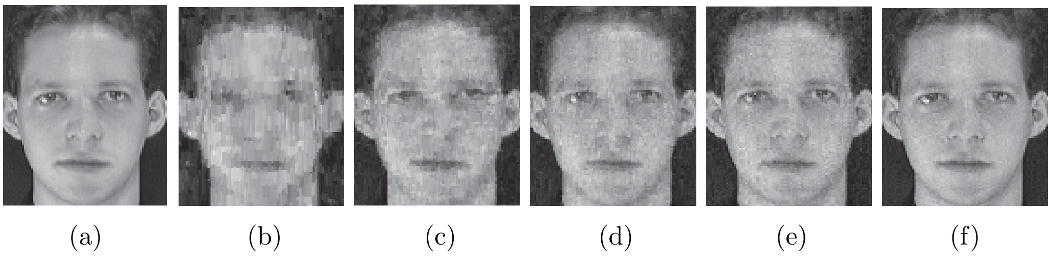


Figure 8: 2DCS reconstructions with different r. The size of the 2DCS features is $m \times n$. (a) Original image with image size $M \times N = 112 \times 92$. (b) Reconstruction with $r = 10\%$, $m \times n = 35 \times 29$. (c) Reconstruction with $r = 30\%$, $m \times n = 61 \times 50$. (d) Reconstruction with $r = 50\%$, $m \times n = 79 \times 65$. (e) Reconstruction with $r = 70\%$, $m \times n = 94 \times 77$. (f) Reconstruction with $r = 90\%$, $m \times n = 104 \times 87$.

5.2 Comparison with the Orthonormal ℓ_2-norm Method

Besides the above mentioned local least-squares approach employed by NS, we are also particularly interested in a recently reported global least-squares classifier called the orthonormal ℓ_2-norm method [63], whose formulation, similar to that of NS, is written as follows.

$$\begin{cases} \boldsymbol{\alpha}^* = \operatorname{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^n} \|\mathbf{y} - \mathbf{A}\boldsymbol{\alpha}\|_2 \\ \text{class}(\mathbf{y}) = \operatorname{argmin}_{i\in\{1,\cdots,K\}} \|\mathbf{y} - \mathbf{A}_i\delta_i(\boldsymbol{\alpha}^*)\|_2 \end{cases} \tag{36}$$

where $\sum_{i=1}^{K} n_i = n$, $\mathbf{A} = [\mathbf{A}_1, \cdots, \mathbf{A}_K]$ and $\delta_i : \mathbb{R}^n \to \mathbb{R}^{n_i}$. $\delta_i(\boldsymbol{\alpha}^*)$ is a new coefficient vector whose entries are associated with the n_i training vectors of the i-th class (i.e., the columns of $\mathbf{A}_i$).

It was reported that in some circumstances the orthonormal ℓ_2-norm classifier even outperforms a state-of-the-art classifier known as standard SRC , which is based on the ℓ_1-norm minimization criterion [19, 63].

The superiority of the orthonormal ℓ_2-norm classifier over the standard SRC is primarily due to that the latter is essentially designed for an underdetermined linear system rather than an overdetermined one. Thus, for the small training set scenario, the feature dimension employed by the standard SRC had to be substantially reduced to make its linear model underdetermined. The substantially reduced feature dimension inevitably causes classification accuracy lose. But this dilemma can be gracefully alleviated by the extended SRC, which gives an impressive classification accuracy (especially on corrupted images) even in the small training set scenario, but is on the expense of a substantially increased computational complexity [19]. But a comprehensive comparison of ℓ_2-norm based classifiers and ℓ_1-norm based ones is beyond the focus of this paper. We leave it as a possible future work.

In this section, we focus our interest on NCSC-II and the orthonormal ℓ_2-norm (hereinafter referred to as Global-L2 for short) classifier and evaluate them on the ORL dataset.

The evaluation is conducted in 20 classification rounds on corrupted images (query images). In each round, we randomly choose 40 query images (1 image/subject × 40 subjects) and 360 training images (9 images/subject × 40 subjects). Thus, 800 query images (40 images/rounds × 20 rounds) are classified.

The corruption of query images is set as follows. The corrupted pixels are uniformly distributed in a given image. For a noise level β, the number of corrupted pixels is a rounded integer of $M \times N \times \beta$. For the ORL dataset, $M = 112$ and $N = 92$. The intensities of corrupted pixels are uniformly distributed in $\{0, \cdots, 255\}$.

The classification accuracies of NCSC-II (with $\kappa = 6$) and Global-L2 respectively using the 1DCS features and 2DCS features are given in Figure 9.

Figure 9(a), Figure 9(c) and Figure 9(e) (i.e. the first column) give the classification accuracies of using the 1DCS features. Figure 9(b), Figure 9(d) and Figure 9(f) (i.e. the second column) give the classification accuracies of using the 2DCS features.

In Figure 9(a) and Figure 9(b), the 1DCS dimension and the 2DCS dimension are both equal to 644 ($r = \frac{644}{112\times 92} = 6.25\%$). In Figure 9(c) and Figure 9(d), the employed 1DCS and 2DCS dimensions are 2576 ($r = \frac{2576}{112\times 92} = 25\%$) . In Figure 9(e) and Figure 9(f), the employed dimensions are 10304 ($r = \frac{10304}{112\times 92} = 100\%$). [3]

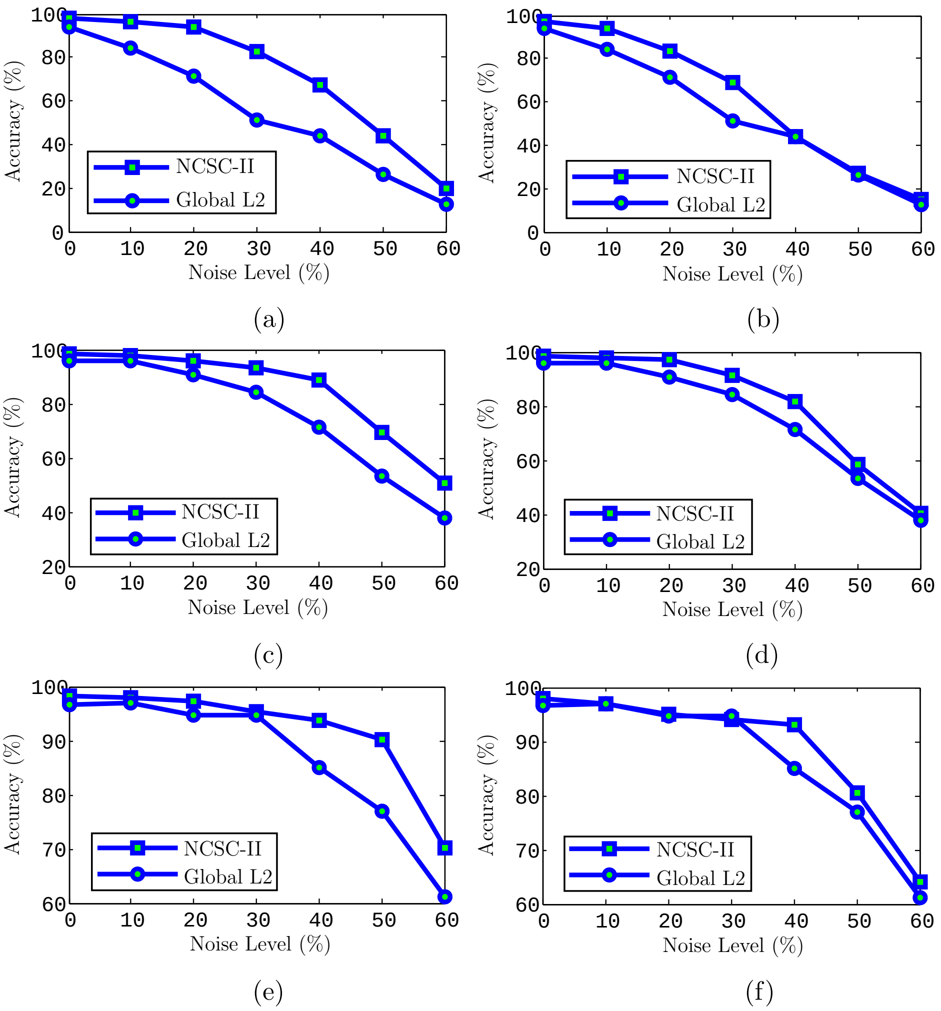
It shows that, on the ORL dataset and using the same features, the tuned NCSC-II generally outperforms Global-L2 in terms of classification accuracy, and when the feature dimensions of the 1DCS features and 2DCS features are equal, the classification accuracies of NCSC-II and Global-L2 are comparable.

6. Conclusions and Future Work

In this paper, a two-dimensional random projection technique for image feature extraction, called 2DCS (two dimensional compressive sampling), is proposed. The proposed 2DCS is more efficient than 1DCS. The proposed 2DCS is a two-stage implementation of 1DCS by manipulating image rows and columns separately. The 2DCS reconstruction via TV (Total Variation) minimization is also demonstrated.

For image recognition, the 2DCS features are used with our proposed subspace-based classifier called NCSC (Nearest Constrained Subspace Classifier). The NCSC employs the techniques of constrained least-squares and ℓ_0-norm sparse representation. The NCSC classifier includes as its low dimensional special cases

[3] Due to the complexity complexity, 10304-dimensional 1DCS features can not be directly obtained without trouble. Thus, we use a piecewise strategy to generate several segments of 1DCS data and then piece them together to get the 10304 dimensions.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 9: Classification accuracy comparison of NCSC-II and Global-L2 on the ORL dataset. First column: results by using the 1DCS features; Second column: results by using the 2DCS features; (a)-(b) Results respectively using 644-dimensional 1DCS and 2DCS features; (c)-(d) Results respectively using 2576-dimensional 1DCS and 2DCS features; (e)-(f) Results respectively using 10304-dimensional 1DCS and 2DCS features.

the classical classifiers NN (Near Neighbor) and NFL (Near Feature Line) and is also closely related to the NS (Nearest Subspace) classifier.

In order to reduce the computational complexity of NCSC, we further propose a fast version of NCSC, called NCSC-II. Under the assumption that the nearest neighbors of a target data point can capture the local intrinsic dimension, NCSC-II employs a κ-neighbors representation to formulate the local linear manifold. Using the κ-neighbors representation, the computational complexity is significantly reduced.

In NCSC/NCSC-II, we contend that with a well-tuned intrinsic subspace dimension, equal to the intrinsic dimension of the data manifold, NCSC/NCSC-II outperforms a variety of algorithms including NN, NFL and NS. Our experiments also suggest that, using the same random features (1DCS or 2DCS), NCSC outperforms the recently-reported orthonormal ℓ_2-norm method (another least-squares-based classifier), which was reported that in some situations outperforms the state-of-the-art SRC method [63].

Since the constrained subspaces employed by NCSC/NCSC-II have the same intrinsic dimension, we call them homogeneous NCSC/NCSC-II. It is possible to extend homogeneous NCSC/NCSC-II to inhomogeneous NCSC/NCSC-II by adopting multiple parameters for all classes or even varying intrinsic dimension parameter(s) for different data samples. We leave the investigation of inhomogeneous NCSC/NCSC-II as well as the possibility of simultaneous image alignment/transformation and sparse representation [64, 65] to our future research.

Another future research is to compare the SRC method, which is a global ℓ_1-norm based method, and our proposed NCSC-II, which is a local ℓ_2-norm based method, for their effectiveness and efficiency, and probably exploit their strengths for designing a new classifier.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Nos. 61152004 and 61379113), China Postdoctoral Science Foundation (No. 20110490173), Key Science and Technology Research Project of Henan Province (No. 12A510028), Foundational Frontier Technology Research Project of Henan Province (No. 132300410163) and High-end Foreign Experts Recruitment Program (No. GDW20134100119)

References

[1] E. Candès and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Process*, vol. 25, no. 2, pp. 21–30, Mar 2008.

[2] E. Candès, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Comm. on Pure and Applied Math*, vol. 59, no. 8, pp. 1207–1223, Aug 2006.

[3] ——, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb 2006.

[4] E. Candès and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?” *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[5] ——, “Decoding by linear programming,” *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec 2005.

[6] Y. Tsaig and D. L. Donoho, “Compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, 2006.

[7] R. Baraniuk, “Compressive sensing,” *IEEE Signal Processing Mag*, pp. 118–120, 2007.

[8] E. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, p. 969, 2007.

[9] D. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," in *Proceedings of the National Academy of Sciences*, vol. 100, no. 5. National Acad Sciences, 2003, pp. 2197–2202.

[10] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[11] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[12] R. Baraniuk and M. Wakin, "Random projections of smooth manifolds," *Foundations of Computational Mathematics*, vol. 9, no. 1, pp. 51–77, 2009.

[13] R. Calderbank, S. Jafarpour, and R. Schapire, "Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain," *Preprint*, 2009.

[14] M. Davenport, P. Boufounos, M. Wakin, and R. Baraniuk, "Signal processing with compressive measurements," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 445–460, 2010.

[15] M. Davenport, M. Duarte, M. Wakin, J. Laska, D. Takhar, K. Kelly, and R. Baraniuk, "The smashed filter for compressive classification and target recognition," in *Proc. SPIE Symposium*, 2007, p. 6498.

[16] D. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution," *Comm. on Pure and Applied Math*, vol. 59, no. 6, pp. 797–829, Mar 2006.

[17] M. Duarte, M. Davenport, M. Wakin, J. Laska, D. Takhar, K. Kelly, and R. Baraniuk, "Multiscale random projections for compressive classification," in *2007 IEEE International Conference on Image Processing*, vol. 6. IEEE, 2007, pp. 161–164.

[18] A. Majumdar and R. K. Ward, "Robust classifiers for data reduced via random projections," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 5, pp. 1359–1371, 2010.

[19] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 31, no. 2, pp. 210–227, Feb 2009.

[20] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Feature selection in face recognition: A sparse representation perspective," Electrical Engineering and Computer Sciences, University of California at Berkeley, Tech. Rep. UCB/EECS-2007-99, Aug 2007.

[21] 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.

[22] Y. Pang, D. Tao, Y. Yuan, and X. Li, "Binary two-dimensional PCA," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 4, pp. 1176–1180, 2008.

[23] X. Li, Y. Pang, and Y. Yuan, "L1-norm-based 2DPCA," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 4, pp. 1170–1175, 2010.

[24] D. Q. Zhang and Z. H. Zhou, "$(2D)^2$PCA: Two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, no. 1–3, pp. 224–231, Dec 2005.

[25] J. Li, R. Janardan, and Q. Li, "Two-dimensional linear discriminant analysis," *Advances in Neural Information Processing Systems*, vol. 17, pp. 1569–1576, 2004.

[26] D. Xu and S. Yan, "Semi-supervised bilinear subspace learning," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1671–1676, 2009.

[27] D. Xu, S. Yan, S. Lin, T. S. Huang, and S.-F. Chang, "Enhancing bilinear subspace learning by element rearrangement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1913–1920, 2009.

[28] D. Tao, X. Li, X. Wu, and S. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 29, no. 10, pp. 1700–1715, 2007.

[29] D. Xu, S. Lin, S. Yan, and X. Tang, "Rank-one projections with adaptive margins for face recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 5, pp. 1226–1236, 2007.

[30] Y. Fu and T. Huang, "Image classification using correlation tensor analysis," *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 226–234, 2008.

[31] X. Li, S. Lin, S. Yan, and D. Xu, "Discriminant locally linear embedding with high-order tensor data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 2, pp. 342–352, 2008.

[32] A. Eftekhari, M. Babaie-Zadeh, and H. Abrishami Moghaddam, "Two-dimensional random projection," *Signal Processing*, vol. 91, no. 7, pp. 1589–1603, 2011.

[33] L. Leng, J. Zhang, G. Chen, M. K. Khan, and K. Alghathbar, "Two-directional two-dimensional random projection and its variations for face and palmprint recognition," in *Computational Science and Its Applications-ICCSA 2011*, 2011, pp. 458–470.

[34] L. Liao, Y. Zhang, and C. Zhang, "2DCS: Two dimensional random underdetermined projection for image representation and classification," in *2011 International Conference on Multimedia Technology (ICMT)*, Hangzhou, China, July 2011, pp. 1–5.

[35] S. Z. Li and J. Lu, "Face recognition using the nearest feature line method," *IEEE Transactions on Neural Networks*, vol. 10, pp. 439–443, 1999.

[36] C. Li, W. Yin, and Y. Zhang. TVAL3: TV minimization by Augmented Lagrangian and ALternating direction ALgorithms. [Online]. Available: http://www.caam.rice.edu/ optimization/L1/TVAL3/

[37] J. Tenenbaum, V. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, p. 2319, 2000.

[38] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, p. 2323, 2000.

[39] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," *Advances in Neural Information Processing Systems*, vol. 14, pp. 585–591, 2001.

[40] G. Hinton, P. Dayan, and M. Revow, "Modeling the manifolds of images of handwritten digits," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 65–74, 1997.

[41] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[42] D. Broomhead and M. Kirby, "The Whitney reduction network: A method for computing autoassociative graphs," *Neural Computation*, vol. 13, no. 11, pp. 2595–2616, 2001.

[43] D. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," in *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 10. National Acad Sciences, 2003, p. 5591.

[44] K. Weinberger and L. Saul, "Unsupervised learning of image manifolds by semidefinite programming," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2004, p. 988.

[45] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *Journal of Shanghai University (English Edition)*, vol. 8, no. 4, pp. 406–424, 2004.

[46] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet, "Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering," *Advances in Neural Information Processing Systems*, vol. 16, pp. 177–184, 2004.

[47] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face recognition using laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, 2005.

[48] X. He, D. Cai, S. Yan, and H. Zhang, "Neighborhood preserving embedding," in *Proc. of the Tenth IEEE International Conference on Computer Vision, 2005.*, vol. 2. IEEE, 2005, pp. 1208–1213.

[49] Mathworks, "Solve constrained linear least-squares problems," http://www.mathworks.cn/help/toolbox/optim/ug/lsqlin.html.

[50] T. Coleman and Y. Li, "A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables," *SIAM Journal on Optimization*, vol. 6, no. 4, pp. 1040–1058, 1996.

[51] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization.* London, UK: Academic Press, 1981.

[52] 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: MIT Press, 2005, pp. 777–784.

[53] P. Grassberger and I. Procaccia, "Measuring the strangeness of strange attractors," *Physica D: Nonlinear Phenomena*, vol. 9, no. 1, pp. 189–208, 1983.

[54] 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.

[55] K. Carter, R. Raich, and A. Hero, "On local intrinsic dimension estimation and its applications," *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 650–663, 2010.

[56] B. Kégl, "Intrinsic dimension estimation using packing numbers," in *Advances in neural information processing systems.* 681–688, 2002, vol. 15.

[57] A. massoud Farahmand, C. Szepesvári, and J. Audibert, "Manifold-adaptive dimension estimation," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 265–272.

[58] K. Fukunaga and D. Olsen, "An algorithm for finding intrinsic dimensionality of data," *IEEE Transactions on Computers*, vol. 100, no. 2, pp. 176–183, 1971.

[59] J. Bruske and G. Sommer, "Intrinsic dimensionality estimation with optimally topology preserving maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 572–575, 1998.

[60] Psychological image collection at stirling (PICS). [Online]. Available: http://pics.psych.stir.ac.uk/2D_face_sets.htm

[61] AT&T. (2002) The database of faces. [Online]. Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[62] Y. LeCun and C. Cortes. The MNIST database of handwritten digits. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[63] 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.

[64] 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.

[65] J. Huang, X. Huang, and D. Metaxas, "Simultaneous image transformation and sparse representation recovery," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008*, 2008, pp. 1–8.