Double Shrinking Sparse Dimension Reduction

Tianyi Zhou, and Dacheng Tao, *Senior Member, IEEE*

***Abstract*—Learning tasks such as classification and clustering usually perform better and cost less (time and space) on compressed representations rather than on the original data. Previous works mainly compress data via dimension reduction. In this paper, we propose "double shrinking" to compress image data on both dimensionality and cardinality via building either sparse low dimensional representations or a sparse projection matrix for dimension reduction. We formulate double shrinking model (DSM) as an ℓ_1 regularized variance maximization with constraint $\|x\|_2 = 1$, and develop double shrinking algorithm (DSA) to optimize DSM. DSA is a path-following algorithm that can build the whole solution path of locally optimal solutions of different sparse levels. Each solution on the path is the "warm start" for searching the next sparser one. In each iterate of DSA, the direction, the step size and the Lagrangian multiplier are deduced from the Karush-Kuhn-Tucker (KKT) conditions. The magnitudes of trivial variables are shrunk and the importances of critical variables are simultaneously augmented along the selected direction with the determined step length. Double shrinking can be applied to manifold learning and feature selection for better interpretation of features, and can be combined with classification and clustering to boost their performance. The experimental results suggest that double shrinking produces efficient and effective data compression.**

***Index Terms*—Sparse learning, compressed sensing, image compression, dimension reduction, ℓ_1 regularization, manifold embedding, path-following.**

I. Introduction

SPARSITY has been widely exploited to compress information, obtain efficient coding of massive data and select important features. In signal processing, compressed sensing [1][2][3][4][5] has proved that a sparse signal can be exactly recovered from a small number of its random projections. In statistics, *lasso* [6] and other ℓ_1 regularized regression models [7][8][4] are proposed to select important variables for building a parsimonious prediction model of a response. The success of sparsity and the ℓ_1 regularization for various applications is supported by various facts. For example, images often have sparse representations [9] after specific transformations such as cosine transform (DCT) and multi-scale geometric analysis (MGA) [10]; and many types of data, such as face images and non-coding RNAs, are usually obtained in the form of redundant features yet insufficient samples.

T. Zhou and D. Tao are with the Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia.

Email: tianyi.zhou@student.uts.edu.au, dacheng.tao@uts.edu.au

Office number: +61 2 9514 1829, Fax number: +61 2 9514 4517

Dimension reduction has been broadly applied to machine learning for compressing image data and preserving important information in the low-dimensional subspace. For example, principal components analysis (PCA) [11][12] maximizes the mutual information between the original high-dimensional Gaussian distributed samples and the projected low-dimensional samples. Fisher's linear discriminant analysis (FLDA) [13] retains the discriminative information by maximizing the between-class scatter and minimizing the within-class scatter. Manifold learning algorithms [14][15] preserve the important geometric structure of images. In order to reduce the computational complexity and improve the performance, learning tasks such as classification [16] and clustering are usually conducted on the dimensionality reduced subspace instead of the original high dimensional space. Dimensionality reduction using feature selection has been formally shown to be important and effective in high dimensional classification in [17], who characterized the impact of dimensionality on classification and proposed the FAIR method for high-dimensional classification.

Compressed sensing compresses images by exploiting their sparsity. Dimension reduction compresses images by preserving the important informations. In this paper, we seamlessly integrate them together in a sparse learning framework "double shrinking" that compresses image data on both dimensionality and cardinality. To retain preferred information underlain in the high dimensional and dense features, it either directly compresses the data to low dimensional and sparse representations or finds a low dimensional and sparse projection matrix to obtain low dimensional approximations. Our experimental results suggest that each case has its own advantages on different applications. The sparse representation reduces the space cost and offers explicit interpretations to new coordinates. Moreover, sparse representations of samples in the same class or cluster tend to share the same support set, and thus the subsequent classification and clustering can be improved. The sparse projection matrix for linear dimension reduction saves the time cost of projection and provides explicit interpretations to selected features. Our experimental results suggest that double shrinking can produce competitive performance in many learning tasks comparing with dimension reduction algorithms that we are aware of.

A. Double shrinking model

In this paper, we formulate double shrinking model (DSM) by introducing the ℓ_1 regularization to the conventional dimension reduction problem. Thus DSM can be

written as

$$\min_x \; x^T P x + \mu \|x\|_1 \quad s.t.\; x^T x = 1, \tag{1}$$

where $x^T P x$ refers to the conventional manifold embedding. This P is borrowed from [18] to retain important information for different applications and includes popular dimension reduction algorithms as special cases, such as locally linear embedding (LLE) [14], ISOMAP [15], Hessian eigenmaps [19], Laplacian eigenmaps [20], and their respective linear approximations [21]. For example, if P is the normalized graph Laplacian $L = I - D^{-1/2} W D^{-1/2}$ [22], wherein W is the adjacency matrix and D is the degree matrix, the solution x is the low dimensional and sparse representation. If P is $X^T L X$, wherein X is the data matrix and L is the normalized graph Laplacian, the solution x is the sparse projection matrix and the corresponding low dimensional representation is Xx. The sparsity of the representation or the projection matrix is due to the ℓ_1 regularization in (1). Sparse eigenvalue maximization problem can be equivalently transformed to the sparse eigenvalue minimization problem (1) by changing the object from maximizing $x^T P x - \mu \|x\|_1$ to minimizing $x^T(-P)x + \mu \|x\|_1$.

DSM provides the first explanation of "double shrinking", i.e., compressing data or finding a projection matrix by simultaneously shrinking dimensionality and cardinality. Most optimization problems solved by existing sparse PCA methods are relaxations of DSM or have similar forms with DSM. However, DSM is characteristic in its equality constraint $x^T x = 1$.

B. Previous works

Compared with existing works, the main challenge for optimizing DSM comes from the simultaneous appearance of the ℓ_1 regularization and the equality constraint $x^T x = 1$. Although either of them has been independently tackled in special optimization algorithms, such as *lasso* [6] and PCA [11], the direct optimization without relaxations for a problem simultaneously containing both is rarely found.

Since the ℓ_1 norm is not differentiable, most of the frequently used optimization methods are not applicable to the ℓ_1 regularized problems. In compressed sensing and statistics, various algorithms have been developed to address the ℓ_1 regularized least square regression or the ℓ_1 norm minimization with a measurement constraint. Popular algorithms can be classified into the following four groups.

1) Greedy algorithms: Orthogonal matching pursuit (OMP) [23] and compressive sampling matching pursuit (CoSaMP) [24] sequentially select important variables by using the greedy search. The sparse solution for the compressed sensing problem is obtained by optimizing the selected variables.
2) Convex optimization based algorithms: Basis pursuit [25] doubles variables in the ℓ_1 norm minimization and then the ℓ_1 norm is replaced by the sum of all the variables. Thus the objective becomes differentiable and the problem can be solved by linear programming. NESTA [26] adopts Nesterov's method [27] to minimize the smoothed approximation of the ℓ_1 norm and converges at rate $\mathcal{O}\left(1/k^2\right)$. An interior point algorithm based on the preconditioned conjugate gradient method is applied in [28] for solving the large scale compressed sensing problem. Coordinate gradient descent [29] and gradient projection [30] also have been introduced to the compressed sensing problem.
3) Iterative thresholding algorithms, e.g., message passing [31], iterative splitting and thresholding (IST) [32] and iterated hard shrinking [33], conduct soft or hard thresholding on the solution at each iteration round and finally obtain the sparse solution;
4) Fixed point method based algorithms: Bregman iterative algorithm [34], fixed point continuation [35] and iteratively re-weighted least squares (IRLS) [36] derive a fixed-point equation from the optimality condition of the compressed sensing problem. They can yield accurate sparse solution within a small number of iteration rounds.

However, pure greedy algorithms hardly ensure the optimality of DSM. The additional equality constraint $x^T x = 1$ in DSM makes the existing convex relaxation methods of the ℓ_1 regularized optimization invalid. Furthermore, the iterative thresholding algorithms and the fixed point method based algorithms have to change the ℓ_2 norm of the solution in each iteration round and thus violates the equality constraint $x^T x = 1$ in DSM. In sum, most of the optimization methods for optimizing the ℓ_1 regularized problem cannot be directly applied to DSM.

The problems solved by existing sparse PCA algorithms are similar to DSM. Most sparse PCA methods find sparse principal components (PCs) with large explained variance by solving two kinds of optimization: 1) sparsity constrained/regularized regression-type problem with an inequality constraint $x^T x \leq 1$ or normalization of the obtained x, e.g., Sparse PCA (SPCA) [37], sPCA-rSVD [38]; 2) sparsity constrained/regularized SDP that maximizes the explained variance of sparse PC with an inequality constraint $x^T x \leq 1$, e.g., DSPCA [39], Path SPCA [40] and SPC in PMD [41]. Greedy method is also applied to sparse PCA in [42]. However, they cannot directly solve DSM.

In summary, it is essential to develop an effective and efficient algorithm to directly optimize DSM.

C. Main contribution

The main challenge for solving DSM comes from the ℓ_1 regularization and the equality constraint $x^T x = 1$. In this paper, we propose DSA that builds a solution path for DSM from the dense solution to sparse ones. Each solution on the path is local optimal with respect to its associated regularization parameter. DSA starts from a point on the path and two initial sets of critical variables and trivial ones, respectively. In each iteration round, it proceeds on a direction along which the importances of the critical variables

are augmented and the magnitudes of the trivial variables are shrunk until one of the following three events happens: 1) the magnitude of a trivial variable is shrunk to zero; 2) a critical variable is transferred to the trivial variable set once its magnitude is shrunk to zero; and 3) a trivial variable is transferred to the critical variable set once its importance reaches the minimum importance of the critical ones. The first two events provide the second explanation of the name "double shrinking". The direction, step size and Lagrangian multiplier in each iteration round are determined by the Karush-Kuhn-Tucker (KKT) conditions. Such continuation technique utilizes the current solution as the "warm start" of the sparser one in the next iteration round and thus accelerates the optimization. The time complexity of each iteration round is less than $\mathcal{O}(s_A^3 + s_B^2)$, wherein s_A is the number of the critical variables and s_B is the number of the trivial ones. DSA has only one free parameter.

Double shrinking can be applied to manifold learning and feature selection [43][44], and can be combined with classification [45] and clustering algorithms to boost performance. It also provides an effective scheme for other ℓ_1 regularized optimization with equality constraints. We apply double shrinking to different machine learning tasks on image datasets of face recognition, hand-written character classification, object categorization, UCI and gene expression, and obtain promising performance. Some critical properties of double shrinking, i.e., variance-cardinality trade-off and speed, are evaluated and compared with existing sparse PCA methods and sparse coding methods, on both real datasets and artificial ones. The experimental results suggest that double shrinking provides an efficient and effective method for data compression.

The rest of the paper is organized as follows. Section 2 defines concepts used in DSM and DSA. Section 3 presents DSA and related proofs. Section 4 shows the experimental results of double shrinking for classification, clustering and feature selection. Section 5 concludes the paper.

II. Definitions

In this paper, lower-case letter denotes a vector or a constant and capital letter denotes a matrix or a set. Let x_i be the i^{th} entry of a vector x and X_{ij} be the entry that lies in the i^{th} row and the j^{th} column of the matrix X. Given an index set S for a vector x, we define x_S that satisfies $(x_S)_i = x_{S_i}$. Given a row index set A and a column index set B for a matrix X, we define X_{AB} that satisfies $(X_{AB})_{ij} = X_{A_iB_j}$. We use superscripts $\cdot^k$ and $\cdot^*$ to signify a variable in the k^{th} iteration round and the final solution of DSA, respectively.

DSM (1) is an ℓ_1 regularized eigenvalue maximization with an equality constraint $x^Tx = 1$, and thus it has a locally optimal solution that satisfies two KKT conditions. In this section, we first show the KKT conditions of DSM by defining the subgradient of the ℓ_1 norm. Based on the KKT conditions, we define the importance and magnitude of a variable in x. Since a sparse solution x is composed of zero and nonzero variables, we then correspondingly define critical and trivial variables that will be sequentially determined and updated through DSA.

A. Karush-Kuhn-Tucker conditions

We start from the KKT conditions [46] of DSM defined in (1). The lagrangian L associated with (1) is

$$L(x, \eta) = x^T P x + \eta \left(x^T x - 1\right) + \mu \|x\|_1. \tag{2}$$

Thus the KKT conditions of (1) are

$$\begin{cases} (P + \eta I)\, x = -\frac{\mu}{2} \partial \|x\|_1, \\ x^T x = 1, \end{cases} \tag{3}$$

where $\partial \|x\|_1$ signifies the subgradient of $\|x\|_1$ and has the form

$$\partial \|x_i\|_1 = \begin{cases} \operatorname{sign}(x_i), & x_i \neq 0; \\ \delta \in [-1, 1], & x_i = 0. \end{cases} \tag{4}$$

According to the definition of subgradient, the ℓ_1 norm is not differentiable at 0, and thus δ in (4) could be any real number between -1 and 1. If there exist a Lagrangian multiplier η and an x that satisfy (3), x is at least a local optimum of DSM (1). However, it is difficult to obtain the solution x and the corresponding η from (3), because $\partial \|x_i\|_1$ is unknown. The proposed DSA can sequentially determine zero variables in x and update the Lagrangian multiplier η in its path-following scheme.

B. Definitions

DSA finds sparse local solutions of DSM via dynamically selecting and updating critical and trivial variables in x. The final critical and trivial variables determine the nonzero and zero variables in the solution x^*, respectively. We define the importance and magnitude of a variable in x. They and KKT conditions together decide the updating rules for critical and trivial variables in DSA.

Definition 1: **(Importance)** The importance of a variable x_i is defined as the absolute value of the partial derivative of $x^T P x + \eta \left(x^T x - 1\right)$ w.r.t. x_i. Thus the importance vector c of x is

$$c = \left| (P + \eta I)\, x \right|. \tag{5}$$

According to the first equation in the KKT conditions (3), on a locally optimal solution, c can also be represented as

$$c = \left| \frac{\mu}{2} \partial \|x\|_1 \right|. \tag{6}$$

The Lagrangian of DSM (2) can be decomposed as the sum of a loss function $x^T P x + \eta \left(x^T x - 1\right)$ and its ℓ_1 regularization $\mu \|x\|_1$. Hence the importance of a variable x_i measures the contribution of the variable x_i to the reduction of the loss function. The gradient g of the loss function, is also used in the subsequent derivations,

$$g = (P + \eta I)\, x = -\frac{\mu}{2} \partial \|x\|_1, \tag{7}$$

$$c = \operatorname{sign}(g) \cdot g, g = \operatorname{sign}(g) \cdot c. \tag{8}$$

Definition 2: **(Magnitude)** The magnitude of a variable x_i is its absolute value. Thus the magnitude vector m of x is

$$m = |x| . \tag{9}$$

We use A and $B = A^C$ (the complement of A) to denote the index sets of critical and trivial variables in x, respectively.

Definition 3: **(Critical variable)** A critical variable is a variable with larger importance than trivial ones and nonzero magnitude in the current iteration round:

$$A = \{i : c_i \geq c_{j:j\in B}, m_i \neq 0\} . \tag{10}$$

Set A is dynamically updated throughout DSA. The final A^* in DSA is the nonzero set in the final solution x^*.

In DSA, A is initialized at the beginning of the algorithm. The importances of critical variables c_A are augmented and their corresponding magnitudes m_A are kept nonzero throughout DSA. A Critical variable will be transferred to B once its magnitude shrinks to zero.

Definition 4: **(Trivial variable)** A trivial variable is a variable with smaller importance than critical ones in the current iteration round:

$$B = \{i : c_i \leq c_{j:j\in A}\} . \tag{11}$$

Set B is dynamically updated throughout DSA. The final B^* in DSA is the zero set in the final solution x^*.

In DSA, B is initialized at the beginning of the algorithm. The magnitudes of trivial variables m_B are shrunk throughout DSA until reaching zeros. A trivial variable will be transferred to A once its importance reaches the minimum importance in c_A.

III. Double Shrinking Algorithm

In this section, we develop DSA to optimize DSM. DSA is a path-following algorithm that finds the locally optimal solutions to a sequence of DSM (1) with the tradeoff parameter μ increasing from small to large. This μ controls the sparsity of the learned model. Each point on the solution path is a "warm start" for searching the subsequent sparser solution.

DSA starts from the initial solution x, the critical variable set A and the trivial variable set B. In each iteration round of DSA, the KKT conditions of the current and subsequent locally optimal solutions determine the directions of x_A and x_B. Afterward, the corresponding step size a is determined by three events. Occurrence of either event will violate the definition of A or B. Thus the optimization should be paused immediately and then both A and B will be modified accordingly. Finally, the locally optimal solution x, the ℓ_1 regularization weight μ and the Lagrangian multiplier η are updated. DSA stops when x reaches preferred sparsity.

A. Initialization

The solution x, the critical variable set A and the trivial variable set B are initialized at the beginning of DSA. The initial x must satisfy the KKT conditions (3) with a certain weight μ. The initial sets A^0 and B^0 are preferred to be close to the final nonzero and zero variable sets respectively, and thus many iteration rounds can be saved.

In this paper, we select the initial solution x^0 as the dense solution of (1) by setting $\mu = 0$, i.e., the eigenvector of P associated with the smallest eigenvalue. Let the target sparse solution x^* has $s \leq p$ nonzero variables. The initial critical variable set A^0 is set as the variables with the first s largest importances in c^0, and $B^0 = (A^0)^C$. There will be no difference to set A^0 as the variables with the first s largest magnitudes in m^0 and $B^0 = (A^0)^C$. That is because

$$c^0 = \left|Px^0\right|, m^0 = \left|x^0\right|, Px^0 = \lambda x^0, \tag{12}$$

where λ is the corresponding eigenvalue. Thus, we have

$$c^0 = \lambda m^0. \tag{13}$$

Thus the order of variable importances in c^0 is the same as the order of variable magnitudes in m^0. According to (10) and (3), the initial Lagrangian multiplier $\eta^0 = -\lambda$ and the initial tradeoff $\mu^0 = 0$.

B. Direction

In the k^{th} iteration round, DSA starts from the current solution x^k, proceeds along a direction ∇x with a particular step size a, and fetches a sparser locally optimal solution x^{k+1} satisfying the KKT conditions of DSM (3) with $\mu = \mu^{k+1}$. In this subsection, we apply the KKT conditions and the definitions of critical/trivial variables to obtain a state transformation equation that reveals how importances and magnitudes of variables in x change between x^k and x^{k+1}. Thus the equation leads to the computation of the direction.

According to (3), x^k satisfies the KKT conditions

$$\mathfrak{R}^k : \begin{cases} \left(P + \eta^k I\right) x^k = -\frac{\mu^k}{2} \partial \|x^k\|_1 \\ {x^k}^T x^k = 1. \end{cases} \tag{14}$$

Consider the subgradient of ℓ_1 norm given in (4), $\partial\|x^k\|_1$ will keep the same until arbitrary variables in x^k change their signs. However, the updating rule in DSA find the next solution on the path once a variable becoming zero and thus avoid the change of solution signs before the update of variable set. In particular, since events 1) and 2) in the step size computation (presented in Section 3.3) will pause the current iteration round once the magnitude of a variable is shrunk to zero, the signs of positive/negative variables in x^k will not be inverted in x^{k+1} (note positive/negative variables are permitted to turn to zeros). According to (4), we have

$$\partial\|x^k\|_1 = \partial\|x^{k+1}\|_1. \tag{15}$$

However, it is worthy noting that after the update of variable sets, the zero variables are allowed to alter to positive/negative values in the next iterate, hence the sign vectors of the solutions on the path can be changed in DSA. In summary, the inverting of variable sign are not direct in DSA and needs a pause on a intermediate state, i.e., zero. And a sparse solution is obtained when reaching such intermediate state. Since x^k and x^{k+1} are sequel sparse

solutions on their associated intermediate states, we can derive (15). Therefore,

$$\mathfrak{R}^{k+1} - \mathfrak{R}^k : \begin{cases} \left(P + \eta^k I\right) \Delta x \doteq -\Delta\eta x^k - \frac{\Delta\mu}{2}\partial\|x^k\|_1, \\ {x^k}^T \Delta x = 0, \end{cases} \tag{16}$$

where $\Delta x = x^{k+1} - x^k$, $\Delta\eta = \eta^{k+1} - \eta^k$ and $\Delta\mu = \mu^{k+1} - \mu^k$.

We ignore two small quantities of the second order $\Delta\eta\Delta x$ and $\Delta x^T \Delta x$ in calculating $\mathfrak{R}^{k+1} - \mathfrak{R}^k$. The first equation of (16) can be decomposed by A and B into

$$\begin{aligned} &\begin{pmatrix} Q^k_{AA} & Q^k_{AB} \\ Q^k_{BA} & Q^k_{BB} \end{pmatrix} \begin{pmatrix} \Delta x_A \\ \Delta x_B \end{pmatrix} \\ =& -\begin{pmatrix} \Delta\mu\partial\|x^k_A\|_1/2 \\ \Delta\mu\partial\|x^k_B\|_1/2 \end{pmatrix} - \Delta\eta \begin{pmatrix} x^k_A \\ x^k_B \end{pmatrix} \\ =& -\begin{pmatrix} \Delta g_A \\ \Delta g_B \end{pmatrix} - \Delta\eta \begin{pmatrix} x^k_A \\ x^k_B \end{pmatrix}, \end{aligned} \tag{17}$$

where $Q^k = P + \eta^k I$ and $\Delta g = g^{k+1} - g^k$. The last equivalence is due to (7). We compute the direction ∇x by solving Δx from (17).

The following theorem determines Δx_B and Δg_A appeared in (17).

Theorem 1: If x^k and x^{k+1} are two consecutive solutions in DSA that satisfy the KKT conditions of DSM (3) with $\mu = \mu^k$ and $\mu = \mu^{k+1}$, respectively, the corresponding Δg_A and Δx_B are

$$\Delta g_A = \Delta\mu/2 \cdot \operatorname{sign}\left(g^k_A\right) = \Delta c_A \cdot \operatorname{sign}\left(g^k_A\right), \tag{18}$$

$$\Delta x_B = -t \cdot \operatorname{sign}\left(x^k_B\right) = \Delta m_B \cdot \operatorname{sign}\left(x^k_B\right), \tag{19}$$

where $\Delta c = c^{k+1} - c^k$, $\Delta m = m^{k+1} - m^k$, and t is a constant.

Proof: When the event 2) in the step size criteria (see Section 3.3) happens, i.e., the magnitude of a critical variable is shrunk to zero, DSA will stop at the current iteration round and this variable with zero magnitude will be transited from A to B. This ensures that all the critical variables are nonzero in DSA. According to definitions of g in (7) and the subgradient of $\|x\|_1$ in (4), we have

$$g^k_A = \mu^k\partial\|x^k_A\|_1/2 = \mu^k/2 \cdot \operatorname{sign}\left(x^k_A\right), \tag{20}$$

$$\Delta g_A = \Delta\mu\partial\|x^k_A\|_1/2 = \Delta\mu/2 \cdot \operatorname{sign}\left(x^k_A\right). \tag{21}$$

The ℓ_1 regularization weight μ is increasing throughout DSA to pursue solutions from dense to sparse, so $\Delta\mu > 0$. By combining (8), (20) and (21), we obtain

$$\Delta c_A = \left|g^k_A + \Delta g_A\right| - \left|g^k_A\right| = \Delta\mu/2. \tag{22}$$

Thus the importances of critical variables in A are equally augmented in DSA. This completes the proof of (18).

In DSA, the ℓ_1 norm $\|x\|_1$ is decreased along the gradient decent direction of the trivial variables in B. According to (4), the negative partial derivative of $\|x\|_1$ w.r.t. a nonzero variable x_i is $-\operatorname{sign}(x_i)$. Thus Δx_B in DSA is

$$\Delta x_B = -t \cdot \operatorname{sign}\left(x^k_B\right). \tag{23}$$

Due to the definition of magnitude in (9), we have

$$\Delta m_B = -t. \tag{24}$$

This completes the proof of (19). ■

Theorem 1 indicates that in each iteration round, the importances of critical variables in A are equally augmented, while magnitudes of trivial variables in B are equally shrunk. This is consistent with the definition of critical and trivial variables in Section 2.2.

In order to simplify the future derivation, we use a, a_1 and a_2 to represent μ, η and t, and thus we have:

$$\begin{cases} a = \Delta\mu/2, \\ a_1 a = t, \\ a_2 a = \Delta\eta. \end{cases} \tag{25}$$

By submitting (18), (19) and (25) into (17), we can obtain the state transformation equation

$$\begin{aligned} &\begin{pmatrix} Q^k_{AA} & Q^k_{AB} \\ Q^k_{BA} & Q^k_{BB} \end{pmatrix} \begin{pmatrix} \Delta x_A \\ -a_1 a \cdot \operatorname{sign}\left(x^k_B\right) \end{pmatrix} \\ =& -\begin{pmatrix} a \cdot \operatorname{sign}\left(x^k_A\right) \\ \Delta g_B \end{pmatrix} - a_2 a \begin{pmatrix} x^k_A \\ x^k_B \end{pmatrix}. \end{aligned} \tag{26}$$

Both Δx_A and Δg_B in (17) are determined by solving the state transformation equation (26).

We decompose the state transformation equation (26) into the following two equations:

$$Q^k_{AA}\Delta x_A - a_1 a Q^k_{AB}\operatorname{sign}\left(x^k_B\right) = -a\operatorname{sign}\left(x^k_A\right) - a_2 a x^k_A, \tag{27}$$

$$Q^k_{BA}\Delta x_A - a_1 a Q^k_{BB}\operatorname{sign}\left(x^k_B\right) = -\Delta g_B - a_2 a x^k_B. \tag{28}$$

Both Δx_A and Δg_B can then be obtained by solving the above two equations.

We summarize the obtained Δx_A, Δg_A, Δx_B and Δg_B:

$$\Delta x_A = -a\left(Q^k_{AA}\right)^{-1}\left(\operatorname{sign}\left(x^k_A\right) + a_2 x^k_A - a_1 Q^k_{AB}\operatorname{sign}\left(x^k_B\right)\right), \tag{29}$$

$$\Delta g_A = a \cdot \operatorname{sign}\left(x^k_A\right), \tag{30}$$

$$\Delta x_B = -a \cdot a_1 \operatorname{sign}\left(x^k_B\right), \tag{31}$$

$$\Delta g_B = -a\left(Q^k_{BA}\Delta x_A + a_2 x^k_B - a_1 Q^k_{BB}\operatorname{sign}\left(x^k_B\right)\right). \tag{32}$$

The above results reveal the changes of variable importances and magnitudes between two consecutively obtained solutions in DSA. However, there still exist three unknown constants a, a_1 and a_2 in (29)-(32).

This a is a multiplier shared by Δx_A, Δg_A, Δx_B and Δg_B, and thus we define a as the step size of DSA and the corresponding direction vectors are

$$\Delta x_A = a\nabla x_A, \Delta g_A = a\nabla g_A \tag{33}$$

$$\Delta x_B = a\nabla x_B, \Delta g_B = a\nabla g_B, \tag{34}$$

where the direction vectors ∇x_A, ∇g_A, ∇x_B and ∇g_B are

$$\nabla x_A = -\left(Q^k_{AA}\right)^{-1}\left(\operatorname{sign}\left(x^k_A\right) + a_2 x^k_A - a_1 Q^k_{AB}\operatorname{sign}\left(x^k_B\right)\right), \tag{35}$$

$$\nabla g_A = \operatorname{sign}\left(x^k_A\right), \tag{36}$$

$$\nabla x_B = -a_1 \operatorname{sign}\left(x^k_B\right), \tag{37}$$

$$\nabla g_B = -\left(Q^k_{BA}\nabla x_A + a_2 x^k_B - a_1 Q^k_{BB}\operatorname{sign}\left(x^k_B\right)\right). \tag{38}$$

The step size a is determined by the step size criteria in Section 3.3.

In order to calculate the directions in (35)-(38), it is necessary to further determine a_1 and a_2. This a_1 is the only predefined algorithm parameter. We derive a_2 to ensure the second equation in KKT conditions (3), i.e., the equality constraint $x^T x = 1$. Since the initial x^0 satisfies $\|x^0\|_2 = 1$, a_2 can be derived from the second equation ${x^k}^T \Delta x = 0$ in (16):

$$\begin{aligned} {x^k}^T \Delta x &= {x_A^k}^T \Delta x_A - a_1 a {x_B^k}^T \operatorname{sign}\left(x_B^k\right) \\ &= {x_A^k}^T \Delta x_A - a_1 a \left\|x_B^k\right\|_1 = 0. \end{aligned} \tag{39}$$

By substituting (29) into (39), we have

$$\begin{aligned} {x_A^k}^T \left(Q_{AA}^k\right)^{-1} &\left(\operatorname{sign}\left(x_A^k\right) + a_2 x_A^k - a_1 Q_{AB}^k \operatorname{sign}\left(x_B^k\right)\right) \\ &+ a_1 \left\|x_B^k\right\|_1 = 0. \end{aligned} \tag{40}$$

Thus a_2 is obtained by solving (40):

$$\begin{aligned} a_2 = &\frac{a_1 \left(\left\|x_B^k\right\|_1 + {x_A^k}^T \left(Q_{AA}^k\right)^{-1} Q_{AB}^k \operatorname{sign}\left(x_B^k\right)\right)}{{x_A^k}^T \left(Q_{AA}^k\right)^{-1} x_A^k} - \\ &\frac{{x_A^k}^T \left(Q_{AA}^k\right)^{-1} \operatorname{sign}\left(x_A^k\right)}{{x_A^k}^T \left(Q_{AA}^k\right)^{-1} x_A^k}. \end{aligned} \tag{41}$$

Since a_2 defined in (41) is a necessary and sufficient condition of (39), the second equation $x^T x = 1$ in KKT conditions (3) and the second equation ${x^k}^T \Delta x = 0$ in (14) are both satisfied throughout DSA.

The direction ∇x is then obtained by substituting a_2 (41) into (35) and (37).

C. Step size and update of A, B

One appealing property of DSA is that the step size a of each iteration round can be adaptively determined by the following step size criteria without any expensive computations such as the line search method. The variable sets A and B are automatically updated in the computation of a.

According to the derivation of direction in Section 3.2 and the definition of critical/trivial variables in Section 2.2, in each iteration round, DSA proceeds along the directions (35)-(38) until one of the following two criteria are violated,

- the sign of each variable in x does not change;
- the importance of each trivial variable in B is less than the importance of any critical variable in A.

The violations of the above two criteria result in the following three events. Thus the iteration round will pause immediately when one of the following three events happens.

1) The magnitude of a trivial variable is shrunk to zero, i.e.,

$$x_i^{k+1} = x_i^k + a \nabla x_i = 0, i \in B. \tag{42}$$

Thus the minimum step size that makes event 1) happen is

$$a(1) = \min\nolimits_{i \in B}^{+} -\frac{x_i^k}{\nabla x_i}, \tag{43}$$

where $a(1)$ is positive, and ∇x_i is defined in (37).

2) The magnitude of a critical variable is shrunk to zero, i.e.,

$$x_i^{k+1} = x_i^k + a \nabla x_i = 0, i \in A. \tag{44}$$

Thus the minimum step size that makes event 2) happen is

$$a(2) = \min\nolimits_{i \in A}^{+} -\frac{x_i^k}{\nabla x_i}, \tag{45}$$

where $a(2)$ is positive, and ∇x_i is defined in (35). Since the magnitude of each critical variable in A is nonzero throughout DSA, the critical variable x_i will be transferred to the trivial variable set B once the event 2) happens and it is directly shrunk to zero.

3) The importance of a nonzero trivial variable reaches the minimum importance of the critical ones, i.e.,

$$c_j^{k+1} = \min_{i \in A} c_i^{k+1} = \min_{i \in A} c_i^k + a, j \in B. \tag{46}$$

Since the trivial variable x_j^{k+1} is nonzero, according to the definitions of the gradient g in (7) and the subgradient of $\|x\|_1$ in (4), we have

$$g_j^{k+1} = \operatorname{sign}\left(x_j^{k+1}\right) \cdot c_j^{k+1}. \tag{47}$$

By substituting (46) into (47), since $g_j^{k+1} = g_j^k + a \nabla g_j$, we have

$$\operatorname{sign}\left(x_j^{k+1}\right) \cdot \left(\min_{i \in A} c_i^k + a\right) = g_j^k + a \nabla g_j. \tag{48}$$

Thus the minimum step size that makes event 3) happen is

$$a(3) = \min\nolimits_{j \in B}^{+} \frac{\operatorname{sign}\left(x_j^{k+1}\right) \cdot \min_{i \in A} c_i^k - g_j^k}{\nabla g_j - \operatorname{sign}\left(x_j^{k+1}\right)}, \tag{49}$$

where $a(3)$ is positive, and ∇g_j is defined in (38). Since the importance of each trivial variable in B is less than the minimum importance of the critical variables in A, the trivial variable x_j will be transferred to the critical variable set A once the event 3) happens.

Therefore the step size a is determined by

$$a = \min\left\{a(1), a(2), a(3)\right\}. \tag{50}$$

D. Update of x, μ and η

Given the directions ∇x_A, ∇x_B and the step size a, the solution x^{k+1} is

$$\begin{cases} x_A^{k+1} = x_A^k + a \nabla x_A \\ x_B^{k+1} = x_B^k + a \nabla x_B, \end{cases} \tag{51}$$

where the step size a is obtained from (43)(45)(49)(50), directions ∇x_A and ∇x_B are calculated according to (35) and (37), respectively.

At the end of each iteration round, η is updated for the computation of $Q = (P + \eta I)$ of the next iteration round. According to the KKT conditions (3), we have

$$(P + \eta I)\, x = -\frac{\mu}{2} \partial \|x\|_1. \tag{52}$$

By multiplying x^T to both sides of (52) simultaneously, we have

$$x^T (P + \eta I) x = x^T P x + \eta = -\frac{\mu}{2}\|x\|_1. \quad (53)$$

Thus η is updated according to

$$\eta^{k+1} = -{x^{k+1}}^T P x^{k+1} - \frac{\mu^{k+1}}{2} \left\|x^{k+1}\right\|_1, \quad (54)$$

where $\mu^{k+1}/2$ is updated according to (25):

$$\frac{\mu^{k+1}}{2} = \frac{\mu^k}{2} + \frac{\Delta\mu}{2} = \frac{\mu^k}{2} + a. \quad (55)$$

E. Algorithm

DSA builds a solution path for DSM from the dense solution to sparse ones. The stopping criterion of DSA is defined as:

$$\text{cardinality}(x) \leq s, \quad (56)$$

where s is the target cardinality of the final sparse solution x^*.

Algorithm 1 Double shrinking Algorithm (DSA)

Input: P, s, a_1, x^0, A^0, B^0.
Output: The solution x^*.
Initialize: $x := x^0$, $A := A^0$, $B := B^0$, $\eta := -\lambda$, $\mu := 0$, $k := 1$.
repeat
  Step 1: Calculate the direction ∇x via calculating ∇x_B, ∇g_A, a_2, ∇x_A and ∇g_B by (37), (36), (41), (35) and (38), respectively.
  Step 2: Calculate the step size a by (43), (45), (49) and (50). Update A and B when event 2) or 3) happens.
  Step 3: Update $x^{k+1} = x^k + a \cdot \nabla x$, η^{k+1} by (50) and $Q^{k+1} = \left(P + \eta^{k+1} I\right)$.
  Step 4: $k := k + 1$.
until $\text{cardinality}(x^k) \leq s$.
return $x^* = x^k$.

Algorithm 1 shows how to obtain a sparse solution x of DSM by DSA. However, in practice, a sparse projection matrix or a sparse representation of more than one dimensionality is preferred. A sparse matrix $X = [X_1; X_2; \cdots; X_d]$ for DSM can be obtained by utilizing a sequence of DSAs. Before the $(i+1)^{th}$ DSA, we update P as the residual $P := P - (X_i^T P X_i) \cdot X_i X_i^T$. The initial solution x^0 for the $(i+1)^{th}$ DSA algorithm can be computed as the first eigenvector of the updated P. Algorithm 2 shows how to use DSA to obtain d sparse vectors. The update of P in sparse PCA has been broadly studied in previous literatures [47] as "deflation methods". We adopts the most commonly used deflation method in Algorithm 2. Please refer to [47] for the detailed introduction of other deflation methods.

Algorithm 2 DSA for d sparse vectors

Input: P, s, a_1.
Output: The sparse matrix X.
for $i = 1$ to d **do**
  Step 1: $x := x^0$, the first eigenvector of P.
  Step 2: Conduct DSA, and obtain the solution x^*.
  Step 3: Update $X = [X; x^*]$.
  Step 4: Update $P := P - ({x^*}^T P x^*) \cdot x^* {x^*}^T$.
end for
return X.

F. Analyses and Proofs

We theoretically analyze some crucial properties of DSA. The corresponding proofs suggest that DSA converges to at least a local minimum of DSM (3) with preferred sparsity in a small number of iteration rounds[1]. The time complexity of each iteration round in DSA is not more than $\mathcal{O}({s_A}^3 + {s_B}^2)$ and is possible to be further reduced. We also analyze the influence of the unique parameter a_1 to the convergence of DSA.

DSA is derived from the KKT conditions of DSM, and thus the solution is at least a local optimum of the optimization (1). Theorem 2 proves that DSA converges to an approximate local optimum.

Theorem 2: (**Convergence**) In DSA, the solution of its k^{th} iteration round x^k satisfies the KKT conditions (3) of the DSM (1) with an ℓ_1 penalty weight μ^k.

Proof: The solution of the first iteration round is the dense solution x^0, which is an eigenvalue of P, and thus we have

$$\begin{cases} P x^0 = \lambda x^0, \\ {x^0}^T x^0 = 1. \end{cases} \quad (57)$$

Equation (58) can be written as

$$\begin{cases} (P - \lambda) x^0 = -\frac{0}{2} \partial \|x\|_1, \\ {x^0}^T x^0 = 1. \end{cases} \quad (58)$$

Refer to (3), the initial x^0 satisfies the KKT conditions of DSM with an ℓ_1 penalty weight $\mu^k = 0$. Without loss of generality, the solution of the first iteration round can be set as any x that satisfies the KKT conditions of DSM (1) with an arbitrary penalty weight μ^k.

Assume the solution of the k^{th} iteration round x^k satisfies the KKT conditions of DSM (1) with an ℓ_1 regularization weight μ^k, i.e.,

$$\begin{cases} \left(P + \eta^k I\right) x^k = -\frac{\mu^k}{2} \partial \|x^k\|_1, \\ {x^k}^T x^k = 1, \end{cases} \quad (59)$$

We initially consider the first equation in the KKT conditions (3). The state transform equation (26) is satisfied when Δx_A and Δg_B are obtained according to (29) and (32), respectively. Equation (26) derives (17). Since (17) is an equal decomposition of (16), (16) is automatically

[1]The number of iteration rounds is the number of occurrences of the three events in Section 3.3. The first two events happens $p - s$ times, and the occurrences of event 3) are much less than that of the first two events.

satisfied. Since the step size a ensures that the sign vector of x keeps unchanged inside each iteration round (may change between two consecutive iteration rounds), we have $\partial\|x^k\|_1 = \partial\|x^{k+1}\|_1$. Thus the first equation in (16) can be rewritten as

$$\begin{aligned}
&\left(P+\eta^k I\right)\left(x^{k+1}-x^k\right)\\
=&-\Delta\eta x^k-\frac{\mu^{k+1}}{2}\partial\|x^{k+1}\|_1+\frac{\mu^k}{2}\partial\|x^k\|_1\\
\doteq&-\Delta\eta\left(x^k+\Delta x\right)-\frac{\mu^{k+1}}{2}\partial\|x^{k+1}\|_1+\frac{\mu^k}{2}\partial\|x^k\|_1\\
=&-\Delta\eta x^{k+1}-\frac{\mu^{k+1}}{2}\partial\|x^{k+1}\|_1+\frac{\mu^k}{2}\partial\|x^k\|_1. \qquad (60)
\end{aligned}$$

By combining the first equation of (59) and (60), we arrive at

$$\left(P+\eta^{k+1}I\right)x^{k+1}=-\frac{\mu^{k+1}}{2}\partial\|x^{k+1}\|_1. \qquad (61)$$

Thus the first equation in the KKT conditions (3) is satisfied.

We then consider the second equation in the KKT conditions (3). By substituting a_2 defined in (41) into Δx_A in (29), we have

$$x^{k^T}\Delta x=x_A^{k^T}\Delta x_A+x_B^{k^T}\Delta x_B=0. \qquad (62)$$

By combining the first equation of (59) and (62), we arrive at

$$x^{k+1^T}x^{k+1}=1. \qquad (63)$$

Thus (61) and (63) compose the KKT conditions of DSM (1) with an ℓ_1 regularization weight μ^{k+1}, and they are satisfied when $x=x^{k+1}$. According to the above analyses, we conclude that the solution of the k^{th} iteration x^k approximately satisfies the KKT conditions (3) of DSM with an ℓ_1 penalty weight μ^k in DSA. This completes the proof. ■

In DSA, the only predefined algorithm parameter is a_1. Inappropriate selection of a_1 increases the number of iteration rounds for the rectification of the critical and trivial variables. However, different choice of a_1 will not influence the convergence to local optimums because Theorem 2 is independent of a_1.

The time complexity of each iteration round in DSA is determined by the cardinality of the critical variable set A in this iteration round. Theorem 3 shows the time complexity of DSA.

Theorem 3: (**Complexity**) The time complexity of each iteration round of DSA is not more than $\mathcal{O}\left(s_A^3+s_B^2\right)$, wherein s_A and s_B are the cardinalities of A and B respectively in this iteration round.

Proof: In each iteration round of DSA, the directions ∇x_B, ∇g_A, a_2, ∇x_A and ∇g_B are calculated according to (37), (36), (41), (35) and (38), respectively. The step size a is calculated by using (43), (45), (49) and (50). Both x^{k+1} and η^{k+1} are updated by using (51) and (54), respectively. The main computational costs of these operations are the matrix inverse calculation $\left(Q_{AA}^k\right)^{-1}$ in (35) with complexity $\mathcal{O}\left(s_A^3\right)$ and the matrix multiplication $Q_{BB}^k\mathrm{sign}\left(x_B^k\right)$ in (38) with complexity $\mathcal{O}\left(s_B^2\right)$. Thus, the time complexity of each iteration round is $\mathcal{O}\left(s_A^3+s_B^2\right)$, wherein s_A and s_B are the cardinalities of A and B respectively in this iteration round. This completes the proof. ■

It is possible to further reduce the time complexity of each iteration round by accelerating the matrix inverse computation. In particular, $\left(Q_{AA}^{k+1}\right)^{-1}$ can be updated from $\left(Q_{AA}^k\right)^{-1}$ approximately. If $\Delta\eta$ is small compared with Q_{AA}^k and I, according to [48], we have

$$\begin{aligned}
\left(Q_{AA}^{k+1}\right)^{-1}&=\left(Q_{AA}^k+\Delta\eta I\right)^{-1}\\
&\cong\left(Q_{AA}^k\right)^{-1}-\Delta\eta\left(Q_{AA}^k\right)^{-1}\left(Q_{AA}^k\right)^{-1}. \qquad (64)
\end{aligned}$$

If A is not updated at the end of the k^{th} iteration round, then $\left(Q_{AA}^{k+1}\right)^{-1}$ can be updated from $\left(Q_{AA}^k\right)^{-1}$ according to (64). If one variable x_i is removed from A at the end of the k^{th} iteration, assume the updated A is A^*, it is still possible to update $\left(Q_{A^*A^*}^k\right)^{-1}$ from $\left(Q_{AA}^k\right)^{-1}$ by using the block matrix inverse [49]. The above analyses show preliminary results on the acceleration.

IV. Experiments

In this section, we show double shrinking can benefit several machine learning tasks including classification, nonlinear manifold learning, clustering and feature selection. Experiments are conducted on different kinds of datasets, such as face datasets [50][51][52][53][15], COIL-20 object dataset [54], UCI machine learning repository [55], Pitprops data [56] and gene expression data [57][58]. Double shrinking is applied to classification and feature selection experiments for producing sparse projection matrix, while it is employed to derive sparse low dimensional representations in nonlinear manifold learning and clustering experiments. We show that the obtained sparse solutions perform competitively compared against the corresponding dense solutions. We also compare double shrinking against existing sparse PCA solvers [37][42][40][38][41] on Pitprops data, gene expression data and artificial data. The experimental results demonstrate that double shrinking is able to produce better solution with less time cost. We implement all the algorithms in MatLab and run all the experiments on a 3.0GHz Intel Xeon processor with 32GB main memory under Windows XP. In DSA, the only free parameter $a_1=0.4$ is fixed in all experiments.

A. Image classification

Double shrinking can obtain proper representations of data samples and thus benefit the subsequent classification. We evaluate double shrinking by testing the classification performance of the sparse projection matrix obtained by DSA on 4 human face datasets, including FERET [50], UMIST [52], YALE [51], and ORL [53].

The FERET dataset consists of $13,539$ face images from $1,565$ individuals. The images vary in size, gender, pose, illumination, facial expression and age. In our experiment, we randomly select 100 individuals, each of which has 7 images, 5 of which are randomly chosen for training

and the rest for test. There are 565 face images from 20 individuals in the UMIST dataset. The samples change in race, gender, pose and appearance. We randomly choose 7 images of each individual for training and the rest for test. The YALE dataset contains 165 face images of 15 individuals. Lighting conditions, gender, facial expressions and configurations are different among these images. The ORL dataset includes 400 face images from 10 individuals. The images were taken at different times, varying the lighting, facial expressions and facial details.

Three linear dimension reduction methods, i.e., principal component analysis (PCA) [11], linear discriminant analysis (LDA) [13], neighborhood preserving embedding (NPE) [21] and their double shrinking versions, i.e., DS-PCA, DS-LDA, are compared on both datasets. For DS-PCA, P is the negative covariance matrix $-X^T X$. For DS-LDA, P is $-[S_W^\dagger S_B + (S_W^\dagger S_B)^T]/2$, wherein S_W is within-class scatter matrix, S_B is between-class scatter matrix, and $\cdot^\dagger$ denotes the Moore-Penrose matrix inverse.

In each experiment, we initially obtain the dense projection matrix from a linear dimension reduction method and a corresponding 66% sparse projection matrix (2/3 entries are zeros) from its double shrinking version. Then the test data is projected onto these two low dimensional subspaces defined by the two projection matrices, respectively. Finally, the nearest neighbor classifier is used for classification.

Fig. 1-Fig. 4 show the classification performance of these 3 linear dimension reduction methods and their corresponding double shrinking versions on the 4 datasets, respectively. All the figures show the recognition rates of double shrinking versions are comparable to the corresponding linear dimension reduction methods on each dataset. This observation indicates that a very sparse projection matrix obtained by using DSA includes sufficient discriminative information with much less storage requirements ($1/3$ of the dense one's in the experiments).

For human face classification experiments, In each plot below the performance evaluation curves, we show the first 10 projection vectors of a linear dimension reduction method and the first 10 projection vectors of its double shrinking version. Comparing against the dense ones, although the sparse projection vectors are blank at most areas, the important biological features, e.g., eyebrows, eyes, nose, mouth, mustache and profile, are usually selected for subsequent classification. Since these selected features have clear physical meanings, the sparse projection vectors can provide an explicit interpretation to the new coordinate. Therefore, by building a sparse projection matrix, double shrinking provides a more efficient strategy to compress the useful information for subsequent classification and a more explicit interpretation to the obtained subspace than the conventional linear dimension reduction algorithms.

B. Nonlinear manifold learning on images

Nonlinear manifold learning remains the manifold structure of the high dimensional data in its low dimensional representations. In order to explore the advantages of double shrinking for nonlinear manifold learning, we design two experiments based on the double shrinking versions of ISOMAP [15] and LLE [14]. In each experiment, the matrix P is calculated according to [18]. Then DSA is applied to the given data for producing a sparse low dimensional representation.

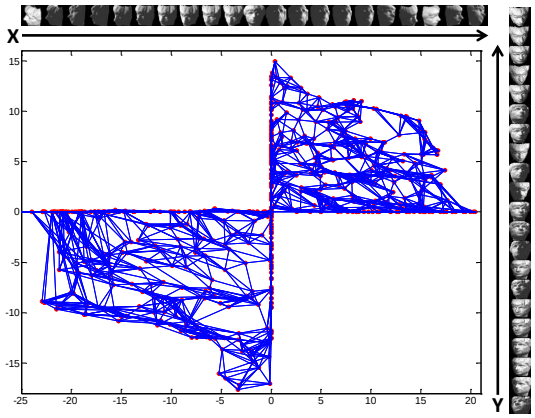

Fig. 5. (3D face) Two-dimensional embedding (with neighborhood graph of the original data) of 698 64×64 face images via double shrinking-ISOMAP. The images were sampled from a face rendered with different poses. Illumination differences were artificially eliminated. 50% of the face images have sparse representations in the two-dimensional subspace and thus are projected on the two coordinate axes X and Y. We sample 21 images from each axis and show them on the top and right of this figure, respectively.

We run the double shrinking version of ISOMAP (DS-ISOMAP) on the face dataset used in [15]. ISOMAP preserves global geodesic distances of all sample pairs. The face dataset is composed of 698 images of size 64×64 with different poses and light directions. In Fig 5, the two dimensional embedding of the face data obtained via DS-ISOMAP is presented with the neighborhood graph of the original data. The short distance between connected samples indicates that the neighborhood structure of the original data is well preserved in the two dimensional embedding. Because of the sparsity obtained by using double shrinking, 50% of the samples are projected onto the two axes in their low dimensional representations. The sample images on the two axes exhibited in Fig 5 imply that DS-ISOMAP entirely recovers the intrinsic geometric structure of the face data. For instance, the samples on X axis encode the poses from top-left to bottom-right with smoothing changing illumination. Thus the global geometric structure of the original data is well preserved. Note that the original ISOMAP results assign left-right pose change to the X axis, which is an obvious difference produced by dense representation. Therefore, double shrinking can find the intrinsic dimension of the data with favor of sparsity and inherit advantages from the used nonlinear manifold learning method.

We run the double shrinking version of LLE (DS-LLE) on a subset of COIL-20 [54], including two objects duck and cat. Therefore, the intrinsic dimension of the subset is two and we can embed the samples in a two dimensional

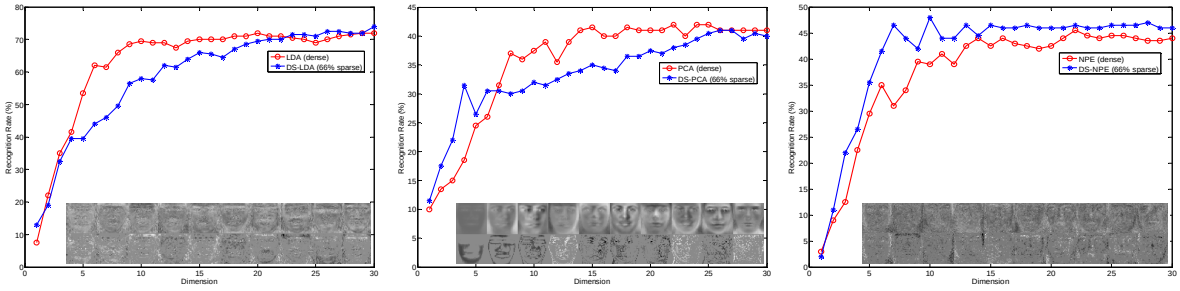

Fig. 1. (FERET) Recognition rate vs. Subspace dimensions curves of LDA, PCA, NPE and their double shrinking versions on FERET face dataset. The first 10 dense eigenfaces obtained via eigenvalue decomposition (the top row) and the corresponding 10 66% sparse eigenfaces obtained via double shrinking (the bottom row) are shown on the bottom of each plot.

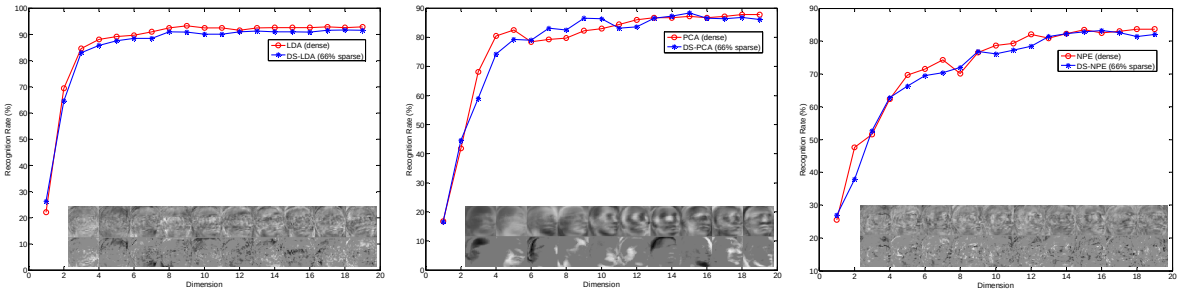

Fig. 2. (UMIST) Recognition rate vs. Subspace dimensions curves of LDA, PCA, NPE and their double shrinking versions on UMIST face dataset. The first 10 dense eigenfaces obtained via eigenvalue decomposition (the top row) and the corresponding 10 66% sparse eigenfaces obtained via double shrinking (the bottom row) are shown on the bottom of each plot.

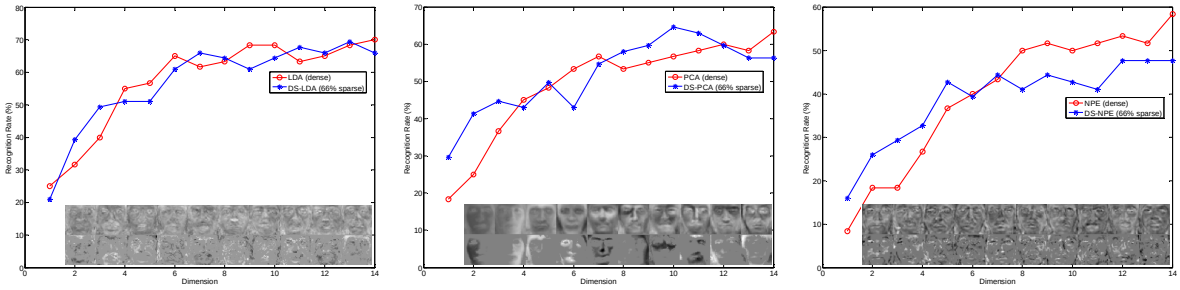

Fig. 3. (YALE) Recognition rate vs. Subspace dimensions curves of LDA, PCA, NPE and their double shrinking versions on YALE face dataset. The first 10 dense eigenfaces obtained via eigenvalue decomposition (the top row) and the corresponding 10 66% sparse eigenfaces obtained via double shrinking (the bottom row) are shown on the bottom of each plot.

subspace for visualization. LLE [14] preserves the local geometry by retaining the neighbor reconstruction coefficients. The images of each object are taken 5 degrees apart as the object is rotated on a turning table and each object has 72 images with size 32×32.

Fig. 6 presents the two dimensional DS-LLE embedding by using the neighborhood graph of the original data. Because of the sparsity obtained by double shrinking, most samples (90%) are projected onto the two axes. The sample images on the two axes are shown at the bottom of Fig. 6. They imply that most duck images are distributed along the X axis, while most cat images are distributed along the Y axis. This observation indicates that double shrinking is able to enhance the separability of low dimensional representations. Moreover, images on each axis preserve local similarity and smoothness over neighbor samples. This property is inherited from LLE. Therefore, compared against nonlinear manifold learning method, double shrinking is able to provide sematic and more compact representations.

C. Image clustering

We apply DSA to PCA and obtain sparse low dimensional representations of given data for k-means [59] based clustering. The matrix P is $-XX^T$. The clustering results of PCA and DS-PCA are compared with each other in subspaces of different dimensions by using three different evaluation metrics, including sum of squares, accuracy and the normalized mutual information. Sum-of-squares adds the square deviation of each sample from its cluster center. Smaller sum-of-squares implies better clustering perfor-

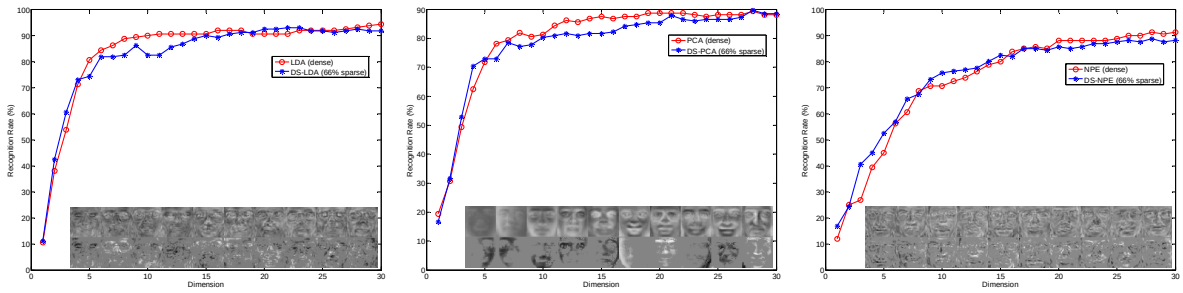

Fig. 4. (ORL) Recognition rate vs. Subspace dimensions curves of LDA, PCA, NPE and their double shrinking versions on ORL face dataset. The first 10 dense eigenfaces obtained via eigenvalue decomposition (the top row) and the corresponding 10 66% sparse eigenfaces obtained via double shrinking (the bottom row) are shown on the bottom of each plot.

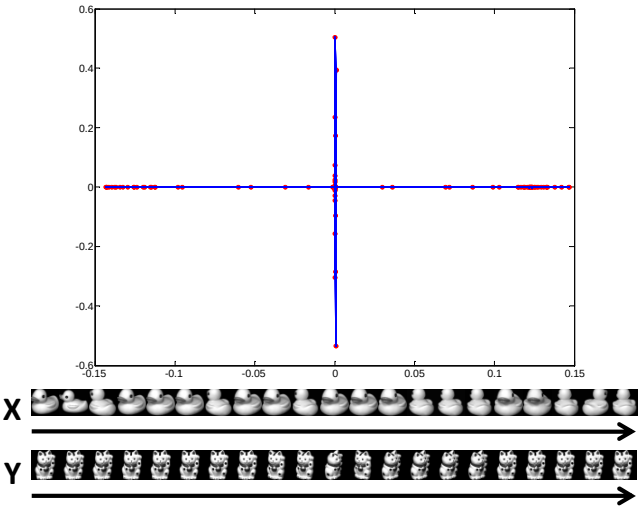

Fig. 6. (COIL-20) Two-dimensional embedding (with neighborhood graph of the original data) of 144 32×32 images of two objects (a toy cat and a toy duck) via double shrinking-LLE. The images were sampled from a toy cat and a toy duck rendered with different poses. 90% of the images have sparse representations in the two-dimensional subspace and thus are projected on the two coordinate axes X and Y. We sample 21 images from each axis and show them on the top and right of this figure, respectively.

mance. Accuracy and normalized mutual information are defined in [60] and [61], respectively. Higher accuracy and larger normalized mutual information correspond to better clustering result. We test the clustering performance of double shrinking on Semeion handwritten digit dataset from the UCI machine learning repository [55]. Semeion dataset includes 1593 256-dimensional samples in 10 classes. The sparsity (the proposition of zero entries) in DSA is set as 60%.

Fig. 7 shows the clustering results of the three evaluation metrics obtained by using PCA and DS-PCA. The sparse low dimensional representations obtained by DS-PCA outperform the dense representations obtained by PCA in most situations. This observation indicates that doubles shrinking can effectively compress the original data and simultaneously preserve the important information for subsequent clustering. This advantage should be attributed to the sparsity of the low dimensional representations, which enhance the separability of the data.

D. Comparison to sparse PCA methods on feature selection

We test the performance of DSA for solving sparse PCA and selecting critical features from a collection of high dimensional data. In this experiment, the matrix P is $-X^T X$. We compare the explained variance of DSA and popular existing sparse PCA algorithms, i.e., Sparse PCA [37], Greedy SPCA [42], Path SPCA [40], sPCA-rSVD [38] and SPC [41] on different cardinalities. We do not include DSPCA [39] in our experiments, because it relaxes the sparse PCA problem to an SDP problem and thus has an expensive computational complexity of $\mathcal{O}(n^3)$ per iteration round. We choose to use Path SPCA [40], because it is a faster alternative of DSPCA.

We use the explained variance as the evaluation criterion of the obtained sparse loading vectors. When only one sparse loading vector v is considered, the variance explained by the corresponding component Xv is

$$Var(v) = v^T X^T X v. \tag{65}$$

When the obtained sparse loading vectors are more than one, for example, V including k sparse loading vectors, the corresponding components are possibly to be correlated. Thus, summing up the variance explained individually by each of the components overestimates the variance explained by all the components. In this case, we use the QR decomposition of the first k sparse PCs $XV = QR$, and define the variance explained by the k corresponding components XV as

$$Var(V) = \sum_{i=1}^{k} R_{i,i}^2. \tag{66}$$

The proportion of the explained variance is defined as $Var(V)/Var(V^{'})$, wherein $V^{'}$ is the first k loading vectors obtained by the classical PCA.

We compare DSA with other sparse PCA algorithms on two gene expression data sets, i.e., Colon cancer [57] and Lymphoma [58]. On both datasets, we consider the 500 genes with largest variance. For each sparse PCA method except SPC, the solution path of the first sparse

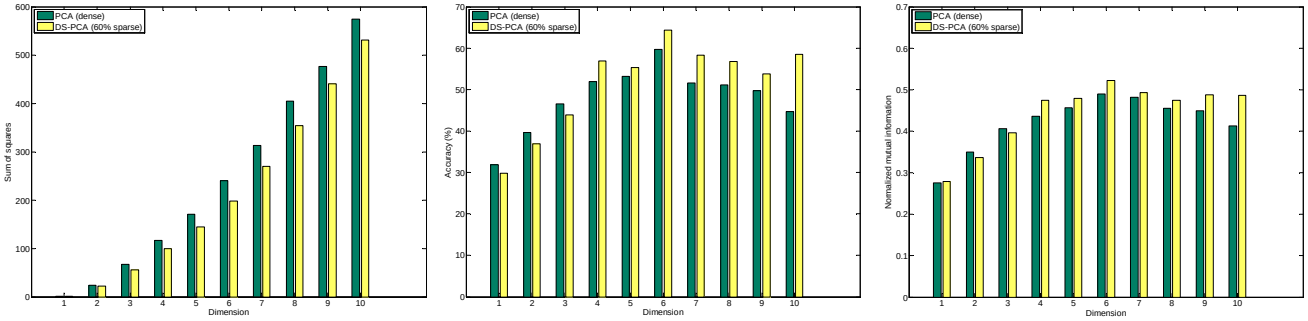

Fig. 7. (Semeion) Sum of squares vs. Subspace dimensions (left), Accuracy vs. Subspace dimensions (middle), Normalized mutual information vs. Subspace dimensions (right) of clustering results on low dimensional representations of Semeion handwritten digit data via PCA and double shrinking-PCA. There are 60% samples owing zero representations on each coordinate obtained via double shrinking.

loading vector (including 500 solutions) is computed. For SPC, because it adjusts the cardinality of the solution by tuning the parameter in the ℓ_1 constraint, the whole solution path is difficult to obtain. Thus we compute 10 sparse solutions with different cardinalities by adjusting the parameter c_2 of constraint $\|v\|_1 \leq c_2$ within the range $[1, \sqrt{500}]$. We show their variance vs. cardinality trade-off curves in Fig 8, together with the corresponding computation time. Note that SPC computes 10 sparse solutions in the shown computation time, while each of the other method computes 500 solutions to build the whole solution path. The curves demonstrate that comparing with other sparse PCA algorithms, DSA can obtain sparse solution with comparable variance in much less CPU seconds on different cardinalities.

We compare the scalability of DSA with other sparse PCA solvers on two artificial datasets, including a 100×100 Gaussian random matrix and a 500×500 Gaussian random matrix. The variance-cardinality trade-off curves of different methods and the corresponding time costs are shown in Fig 9. We also compute 10 solutions in each of SPC experiments and show the total time cost for obtaining the 10 solutions. The results imply that Greedy SPCA, Path SPCA and DSA have comparable performance on the explained variance of the obtained sparse loading vectors, while the explained variances of the sparse loading vector obtained by SPCA and SPC are smaller. Double shrinking has the lowest time cost among all the algorithms. In addition, the CPU seconds of DSA increase slowly with the increasing data size compared with the other solvers. This observation is consistent with the time complexity analysis of DSA. The appealing scalability of DSA suggests its priority in solving large scale problems.

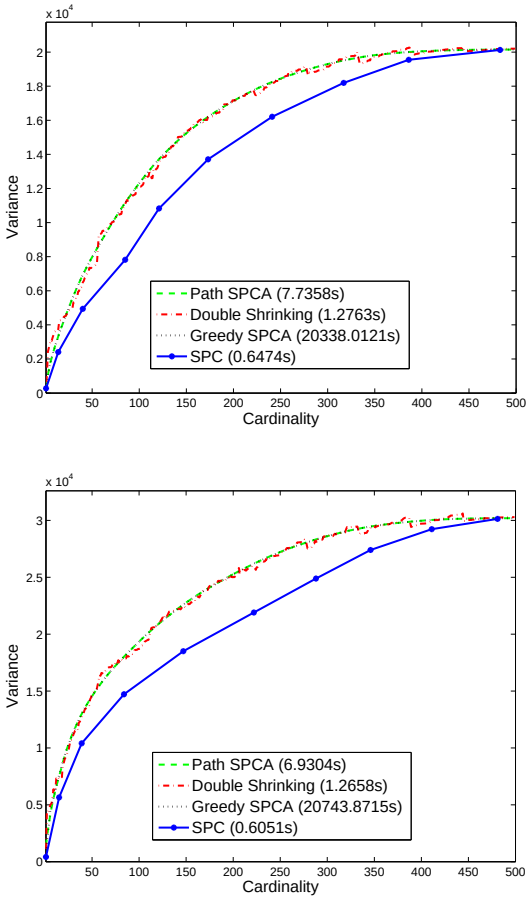

Fig. 8. Trade-off curves between explained variance and cardinality for the first sparse principal component of colon cancer data (top) and lymphoma data (bottom). Different Sparse PCA methods (Greedy search, Path SPCA, SPC, Double shrinking) are compared with each other. SPC computes 10 sparse solutions of different cardinalities, while the other methods computes 500 solutions to build their solution paths. Their corresponding time costs are listed on the bottom of each plot.

V. Conclusion

This paper proposed double shrinking for sparse dimension reduction [62] of image data. Different from existing dimension reduction methods, double shrinking compresses data by simultaneously shrinking both dimensionality and cardinality. It improves the low dimensional representation by exploiting the promising properties of sparsity, which has been successfully applied to problems in signal processing and statistics. Double shrinking model (DSM) is an ℓ_1 penalized eigenvalue maximization/minimization with an unitary constraint. It consists of manifold embedding and the ℓ_1 norm penalty to shrink the data dimensionality and cardinality, respectively.

We then developed double shrinking algorithm (DSA) to optimize DSM. DSA is a path-following algorithm that

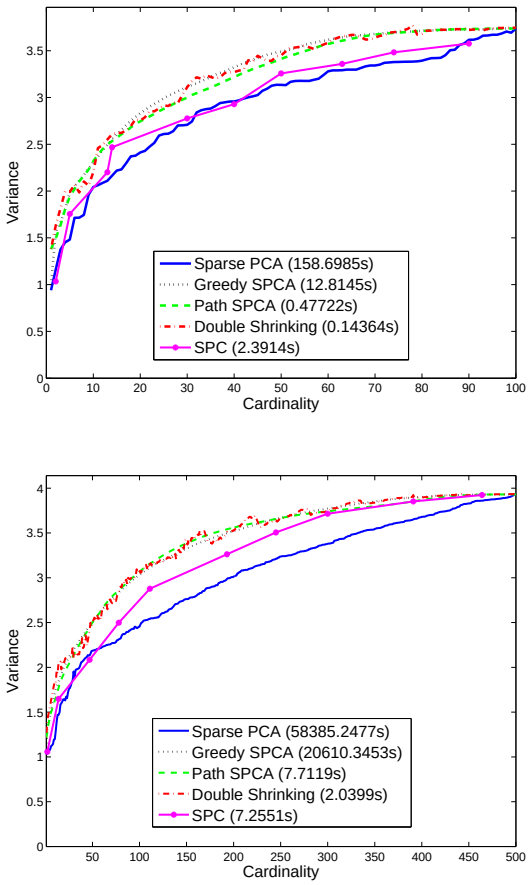

Fig. 9. Trade-off curves between explained variance and cardinality for the first sparse principal component of a 100×100 gaussian random matrix (top) and a 500×500 gaussian random matrix (bottom), each entry of the matrix is sampled from an independent standard gaussian distribution. Different Sparse PCA methods (Sparse PCA, Greedy search, Path SPCA, SPC, Double shrinking) are compared with each other. SPC computes 10 sparse solutions of different cardinalities, while the other methods computes 100 (left) or 500 (right) solutions to build their solution paths. Their corresponding time costs are listed on the bottom of each plot.

can build the whole solution path of DSM efficiently. We analyzed the essential properties of DSA. Each solution on the solution path is proved to be at least a local optimum on the corresponding sparse level. The time complexity of each iteration round is about $\mathcal{O}({s_A}^3 + {s_B}^2)$, wherein s_A and s_B are the number of the critical variables and trivial ones, respectively. The step size of each iteration round has a closed form, and thus its computation is efficient. DSA has only one free parameter a_1 that can be conveniently determined, so it can be applied in practice conveniently. Compared against the corresponding dimension reduction method, double shrinking has promising priorities in providing explicit interpretation to selected features, decreasing the computational costs, improving the data representation for subsequent classification and clustering.

ACKNOWLEDGMENT

The authors would like to thank the handling Associate Editor and the five anonymous reviewers for their constructive comments on this paper. This work was fully supported by Australian Research Council Discovery Project with number DP-120103730.

REFERENCES

[1] E. J. Candès, J. K. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[2] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] L. Sun, J. Liu, J. Chen, and J. Ye, "Efficient recovery of jointly sparse vectors," in *Advances in Neural Information Processing Systems 23*, 2009.

[4] J. Lv and Y. Fan, "A unified approach to model selection and sparse recovery using regularized least squares," *The Annals of Statistics*, vol. 37, pp. 3498–3528, 2009.

[5] T. Zhou and D. Tao, "1-bit hamming compressed sensing," in *ISIT '12: IEEE International Symposium on Information Theory*, 2012.

[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society (Series B)*, vol. 58, pp. 267–288, 1996.

[7] J. Liu, J. Chen, and J. Ye, "Large-scale sparse logistic regression," in *KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 547–556.

[8] J. Liu, S. Ji, and J. Ye, *SLEP: Sparse Learning with Efficient Projections*, Arizona State University, 2009. [Online]. Available: http://www.public.asu.edu/ jye02/Software/SLEP

[9] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008.

[10] E. Arias-Castro, D. L. Donoho, and X. Huo, "Near-optimal detection of geometric objects by fast multiscale methods," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2402–2425, 2005.

[11] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Phychology*, vol. 24, pp. 417–441, 1936.

[12] T. Zhou and D. Tao, "Bilateral random projections," in *ISIT '12: IEEE International Symposium on Information Theory*, 2012.

[13] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[14] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[15] J. B. Tenenbaum, V. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[16] T. Zhou, D. Tao, and X. Wu, "Nesvm: A fast gradient method for support vector machines," in *ICDM '10: Proceedings of the 2010 IEEE International Conference on Data Mining*, 2010, pp. 679–688.

[17] J. Fan and Y. Fan, "High dimensional classification using features annealed independence rules," *The Annals of Statistics*, vol. 36, pp. 2605–2637, 2008.

[18] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet, "Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering," in *Advances in Neural Information Processing Systems 16 (NIPS '03)*, vol. 16, 2004, pp. 177–184.

[19] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *PNAS*, vol. 100, no. 10, pp. 5591–5596, 2003.

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

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

[22] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems 14 (NIPS '02)*, 2002.

[23] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, pp. 4655–4666, 2007.

[24] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, pp. 301–321, 2008.

[25] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1999.
[26] J. Bobin, S. Becker, and E. Candès, "Nesta: A fast and accurate first-order method for sparse recovery," *technical report*, 2009.
[27] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 103, no. 1, pp. 127–152, 2005.
[28] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l1-regularized least squares," *IEEE Journal of In Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, 2007.
[29] P. Tseng and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization," *Mathematical Programming*, vol. 117, no. 1-2, pp. 387–423, 2009.
[30] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.
[31] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, 2009.
[32] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.
[33] K. Bredies and D. A. Lorenz, "Iterated hard shrinkage for minimization problems with sparsity constraints," *SIAM Journal on Scientific Computing*, vol. 30, no. 2, pp. 657–683, 2008.
[34] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for l1-minimization with applications to compressed sensing," *SIAM Journal on Imaging Sciences*, vol. 1, no. 1, pp. 143–168, 2008.
[35] E. T. Hale, W. Yin, and Y. Zhang, "Fixed-point continuation for ℓ_1-minimization: Methodology and convergence," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1107–1130, 2008.
[36] B. Rao, K. Engan, S. Cotter, J. Palmer, and K. Kreutz-Delgado, "Subset selection in noise based on diversity measure minimization," *IEEE Transactions on Signal Processing*, vol. 51, pp. 760–770, 2003.
[37] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 262–286, 2006.
[38] H. Shen and J. Z. Huang, "Sparse principal component analysis via regularized low rank matrix approximation," *Journal of Multivariate Analysis*, vol. 99, no. 6, pp. 1015–1034, 2008.
[39] A. d'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet, "A direct formulation for sparse pca using semidefinite programming," *SIAM Review*, vol. 49, no. 3, pp. 434–448, 2007.
[40] A. d'Aspremont, F. Bach, and L. E. Ghaoui, "Optimal solutions for sparse principal component analysis," *Journal of Machine Learning Research*, vol. 9, pp. 1269–1294, 2008.
[41] D. M. Witten, R. Tibshirani, and T. Hastie, "A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis," *Biostatistics*, vol. 10, no. 3, pp. 515–534, 2009.
[42] B. Moghaddam, Y. Weiss, and S. Avidan, "Spectral bounds for sparse pca: Exact and greedy algorithms," in *Advances in Neural Information Processing Systems 20*, 2006, pp. 915–922.
[43] J. Lv and J. S. Liu, "Model selection principles in misspecified models," *Manuscript*, 2010.
[44] J. Fan and J. Lv, "A selective overview of variable selection in high dimensional feature space (editor's invited paper)," *invited review article*, vol. 20, pp. 101–148, 2010.
[45] T. Zhou, D. Tao, and X. Wu, "Compressed labeling on distilled labelsets for multi-label learning," *Machine Learning (Springer)*, 2012.
[46] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.
[47] L. Mackey, "Deflation methods for sparse pca," in *Advances in Neural Information Processing Systems 22 (NIPS '08)*, 2008.
[48] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," oct 2008, version 20081110. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?3274
[49] G. H. Golub and C. F. Van Loan, *Matrix computations (3rd ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.
[50] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
[51] P. N. Belhumeur, J. a. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 711–720, 1997.
[52] D. B. Graham and N. M. Allinson, "Characterizing virtual eigensignatures for general purpose face recognition," *Face Recognition: From Theory to Applications, NATO ASI series F, Computer and System Science*, vol. 163, pp. 446–456, 1936.
[53] F. S. Samaria, A. Harter, and O. A. Site, "Parameterisation of a stochastic model for human face identification," *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, 1994.
[54] S. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-20)," Tech. Rep., 1996.
[55] A. Asuncion and D. Newman, "UCI machine learning repository," *Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html*, 2007.
[56] N. Trendafilov, I. T. Jolliffe, and M. Uddin, "A modified principal component technique based on the lasso," *Journal of Computational and Graphical Statistics*, vol. 12, pp. 531–C547, 2003.
[57] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Cell Biology*, vol. 96, pp. 6745–6750, 1999.
[58] A. Alizadeh, M. Eisen, R. Davis, C. Ma, I. Lossos, and A. Rosenwald, "Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, pp. 503–511, 2000.
[59] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, 2nd ed., ser. Springer Series in Statistics. Springer, 2009.
[60] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *SIGIR '03: ACM SIGIR conference on Research and development in informaion retrieval*, 2003, pp. 267–273.
[61] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1624–1637, 2005.
[62] T. Zhou, D. Tao, and X. Wu, "Manifold elastic net: a unified framework for sparse dimension reduction," *Data Mining and Knowledge Discovery (Springer)*, vol. 22, no. 3, pp. 340–371, 2011.

Tianyi Zhou Tianyi Zhou is pursuing the Ph.D degree of Computer Science at University of Technology, Sydney, Australia. He received the BEng degree in automation in 2008 from Beijing Institute of Technology, China. His main research interest is in statistics, machine learning, data mining, signal processing and information theory. He has authored 14 scientific articles at top venues including IEEE T-IP, Machine Learning (Springer), DMKD (Springer), ICML, AISTATS, ISIT, ICDM, and NIPS workshops.

Tao Dacheng Tao (M'07-SM'12) is Professor of Computer Science with the Centre for Quantum Computation and Information Systems and the Faculty of Engineering and Information Technology in the University of Technology, Sydney. He mainly applies statistics and mathematics for data analysis problems in data mining, computer vision, machine learning, multimedia, and video surveillance. He has authored and co-authored more than 100 scientific articles at top venues including IEEE T-PAMI, T-IP, T-NNLS, AISTATS, ICDM, CVPR, and ECCV, with the best theory/algorithm paper runner up award in IEEE ICDM07.