Multi-Subspace Representation and Discovery

Dijun Luo, Feiping Nie, Chris Ding, and Heng Huang

Department of Computer Science and Engineering,
University of Texas, Arlington, Texas, USA
{dijun.luo,feipingnie}@gmail.com, {chqding,heng}@uta.edu

Abstract. This paper presents the multi-subspace discovery problem and provides a theoretical solution which is guaranteed to recover the number of subspaces, the dimensions of each subspace, and the members of data points of each subspace simultaneously. We further propose a data representation model to handle noisy real world data. We develop a novel optimization approach to learn the presented model which is guaranteed to converge to global optimizers. As applications of our models, we first apply our solutions as preprocessing in a series of machine learning problems, including clustering, classification, and semi-supervised learning. We found that our method automatically obtains robust data presentation which preserves the affine subspace structures of high dimensional data and generate more accurate results in the learning tasks. We also establish a robust standalone classifier which directly utilizes our sparse and low rank representation model. Experimental results indicate our methods improve the quality of data by preprocessing and the standalone classifier outperforms some state-of-the-art learning approaches.

1 Introduction

The linear sparse representation approaches recently attract attentions from the researchers in statistics and machine learning. By providing robustness, simpleness, and sound theoretical foundations, sparse representation models have been widely considered in various applications [1,2,3,4].

In most previous models, we impose on the data an assumption that the data points can be linearly represented by other data points in the same class or data points nearby. This assumption will further lead to another assumption that subspace of each class has to include the original point. Our major argument in this paper is that this assumption is too loose in real world applications. For this reason, we further impose the affine properties of the subspaces and present a challenging affine subspace discovery problem. To be more specific, given a set of data points, which lie on multiple unknown spaces, we want to recover the membership of data points to subspaces, *i.e.* which data point belongs to which subspace. The major challenge here is that not only the subspaces and membership are unknown, but also the number of subspaces and the dimensions of the subspaces are unknown.

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part II, LNAI 6912, pp. 405–420, 2011.

In this paper we will (1) present a sparse representation learning model to obtain the solutions automatically, which is theoretically guaranteed to recover all the unknown information listed above, (2) extended our model to handle noisy data and apply the sparse representation as a preprocessing in various machine learning tasks, such as unsupervised learning, classification and semi-supervised learning, and (3) develop a standalone classifier directly based on the sparse representation model. To handle the noisy data with robust performance, we introduce a mixed-norm optimization problem which involves trace, ℓ_2/ℓ_1, and ℓ_1 norms. We further develop an efficient algorithm to optimize the induced problem which is guaranteed to converge to a global optimizer.

Our model explicitly imposes both sparse and low rank requirements on the data presentation. We apply our model as preprocessing in various machine learning applications. The extensive and sound empirical results suggest that one might benefit from taking sparsity and low rank into consideration simultaneously.

2 Problem Description and Our Solution

Consider K groups data points $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K]$ and assume that there are $n_1, n_2, \cdots, n_K$ data points in each group, respectively ($\sum_{k=1}^{K} n_k = n$). We assume that for each group, the data points belong to independent affine subspaces. And the dimensions of the affine subspaces are $d_1, d_2, \cdots, d_K$. To be more specific, for each affine subspace $\mathbf{X}_k$, there exist $d_k + 1$ bases $\mathbf{U}^k = [\mathbf{u}_1^k, \mathbf{u}_2^k, \cdots, \mathbf{u}_{d_k}^k, \mathbf{u}_{d_k+1}^k]$ and for each data point $\mathbf{x} \in \mathbf{X}_k$, there exists $\boldsymbol{\beta}$ such that $\mathbf{x} = \mathbf{U}^k \boldsymbol{\beta}^k$ and that $\boldsymbol{\beta}^T \mathbf{1} = 1$. In this paper, by the dimension of the affine subspace, we mean the characteristic dimension, *i.e.* from the manifold point of view. Even though there are $d_k + 1$ bases in $\mathbf{U}^k$, we still consider that $\mathbf{U}^k$ defines a d_k-dimensional affine subspace.

2.1 Multi-Subspace Discovery Problem

The problem of **Multi-Subspace Discovery** is given $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K]$ to recover (1) the number of affine space K, (2) the dimension of each subspace d_k, and (3) the membership of the data points to the affine subspaces. The challenge in this problem is that the only known information is the input $\mathbf{X}$, where the data points are typically disordered, and all other information is unknown.

Will illustrate the Multi-Subspace Discovery problem in Figure 1. In this paper, we first derive a solution of this problem and provide several theoretical analysis of our solution on non-noisy data, then extend our model to handle noisy real-world case by adding ℓ_2/ℓ_1 norms which are convex but non-smooth regularizations. We develop an efficient algorithm to solve the problem.

2.2 A Constructive Solution

We cast the multi-subspace discovery problem into a trace norm optimization, in which the optimizer directly gives the number of affine subspace and the membership of the clustering. The results are theoretically guaranteed.

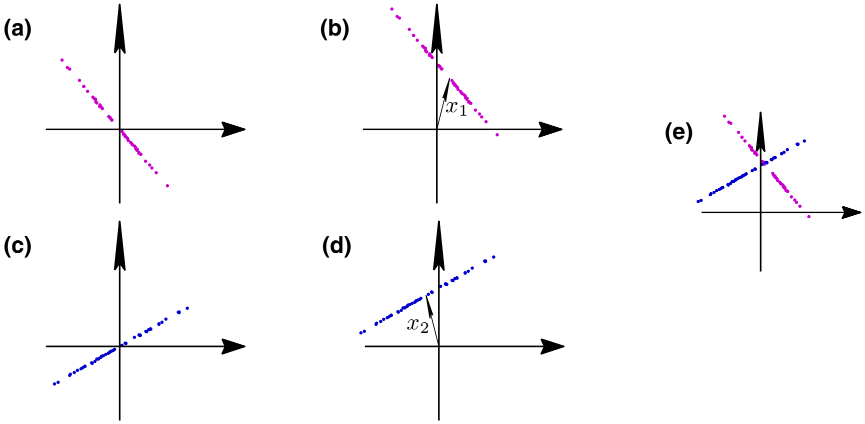


Fig. 1. A demonstration of the Multi-Subspace Discovery problem. **(a)** and **(c)**: Two groups of data points lying on two 1-dimension subspaces. **(b)**: All data points shifted by x_1 from **(a)**. **(d)**: All data points shifted by x_2 from **(c)**. **(e)**: A mixture of data points from **(b)** and **(d)**. The affine subspace clustering problem is to recover the number of subspaces (2 in this case), the membership of the data points to the subspaces (indicated by the color of the data points in **(e)**, the dimensions of the subspaces (1 for both of the subspace in this cases).

Representation of One Subspace

In order to introduce our solution in a more interpretable way, we first solve a simple problem in which there is only one affine subspace. Let $\mathbf{X}_1 = (\mathbf{x}_1, \cdots, \mathbf{x}_{n_1})$ be in a d_1-dimensional affine subspace spanned by the basis $\mathbf{U}_1$, $d_1 + 1 < n_1$, *i.e.* for each data points $\mathbf{x}_i$, there exists $\boldsymbol{\alpha}_i$,

$$\mathbf{x}_i = \mathbf{U}_1 \boldsymbol{\alpha}_i,\ \boldsymbol{\alpha}_i \in \mathbb{R}^{d_1+1},\ \boldsymbol{\alpha}_i^T \mathbf{1} = 1,\ 1 \le i \le n_1 \tag{1}$$

or more compactly, $\mathbf{X}_1 = \mathbf{U}_1\mathbf{A},\ \mathbf{A}^T\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is a column vector with all elements one in proper size and $\mathbf{A} = (\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_{n_1})$. We define

$$\tilde{\mathbf{X}}_1 = \begin{pmatrix} \mathbf{U}_1\mathbf{A} \\ \mathbf{1}^T \end{pmatrix} \tag{2}$$

Then we have,

Lemma 1. *If* $\mathbf{X}_1$ *satisfies Eq. (1) and let*

$$\mathbf{Z}_1 = \tilde{\mathbf{X}}_1^+ \tilde{\mathbf{X}}_1 \tag{3}$$

where $\tilde{\mathbf{X}}_1$ *is defined in Eq. (2) and* $\tilde{\mathbf{X}}_1^+$ *is the* Moore-Penrose *pseudo inverse of* $\tilde{\mathbf{X}}_1$*, then*

$$\mathbf{X}_1 = \mathbf{X}_1\mathbf{Z}_1,\ \mathbf{1}^T\mathbf{Z}_1 = \mathbf{1}^T, \tag{4}$$

and rank$(\mathbf{Z}_1) = d_1 + 1$.

Proof. By making use of the property of *Moore-Penrose* pseudo inverse, we immediately have

$$\tilde{\mathbf{X}}_1 = \tilde{\mathbf{X}}_1 \tilde{\mathbf{X}}_1^{+} \tilde{\mathbf{X}}_1,$$

Thus,

$$\begin{pmatrix} \mathbf{U}_1\mathbf{A} \\ \mathbf{1}^T \end{pmatrix} = \begin{pmatrix} \mathbf{U}_1\mathbf{A} \\ \mathbf{1}^T \end{pmatrix} \mathbf{Z},$$

which is equivalent to two equations of

$$\mathbf{X}_1 = \mathbf{X}_1 \mathbf{Z}_1,$$

$$\mathbf{1}^T \mathbf{Z}_1 = \mathbf{1}^T.$$

It is obvious that $\text{rank}(\mathbf{Z}_1) = \text{rank}(\tilde{\mathbf{X}}_1)$. On the other hand, by the definition of $\mathbf{A}$ in Eq. (2), we have $\mathbf{1}^T\mathbf{A} = \mathbf{1}^T$, thus

$$\tilde{\mathbf{X}}_1 = \begin{pmatrix} \mathbf{U}_1\mathbf{A} \\ \mathbf{1}^T \end{pmatrix} = \begin{pmatrix} \mathbf{U}_1\mathbf{A} \\ \mathbf{1}^T\mathbf{A} \end{pmatrix} = \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{1}^T \end{pmatrix} \mathbf{A} \tag{5}$$

From Eq. (2) we have

$$\text{rank}(\tilde{\mathbf{X}}_1) \geq \text{rank}(\mathbf{U}_1\mathbf{A}) = \text{rank}(\mathbf{X}_1) = d_1 + 1$$

But from Eq. (5) we have

$$\text{rank}(\tilde{\mathbf{X}}_1) \leq \text{rank}(\mathbf{A}) = d_1 + 1.$$

Thus $\text{rank}(\mathbf{Z}_1) = \text{rank}(\tilde{\mathbf{X}}_1) = d_1 + 1$.

Since $d_1 + 1 < n_1$, $\mathbf{Z}_1$ is low rank. Interestingly, this low-rank affine subspace presentation of Eqs. (1, 4) can be reformulated as a trace norm optimization problem:

$$\begin{aligned} &\min_{\mathbf{Z}_1} \|\mathbf{Z}_1\|_*, \\ &\text{s.t.} \;\; \mathbf{X}_1 = \mathbf{X}_1\mathbf{Z}_1, \; \mathbf{1}^T\mathbf{Z}_1 = \mathbf{1}^T \end{aligned} \tag{6}$$

where $\|\mathbf{Z}_1\|_*$ is the trace norm of $\mathbf{Z}_1$, *i.e.* the sum of singular values, or explicitly,

Lemma 2. $\mathbf{Z}_1$ *defined in Eq. (3) is an optimizer of the problem in Eq. (6).*

Due to the limited space, we omit the proof here[1].

In this paper, we hope to recover multiple $\mathbf{Z}$ which has diagonal block structure from $\mathbf{X}$ by which we solve the multi-subspace discovery problem.

Constructive Representation of K Subspaces

[1] One can also easily show that $\mathbf{Z}_1$ defined in Eq. (3) is one element in the subgradient of the Lagrangian $\mathcal{L}(\mathbf{Z}, \Lambda) = \|\mathbf{Z}\|_* - \mathbf{tr}(\tilde{\mathbf{X}}_1 - \tilde{\mathbf{X}}_1\mathbf{Z})^T\Lambda$,

Now consider the full case where the data points $\mathbf{X}$ belong exactly to K independent subspaces. Assume data points within a subspace are indexed sequentially, $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K]$. Repeat the above analysis for each subspace, we have

$$\mathbf{X} = [\mathbf{X}_1, \cdots, \mathbf{X}_K] = [\mathbf{X}_1\mathbf{Z}_1, \cdots, \mathbf{X}_K\mathbf{Z}_K] = \mathbf{XZ}, \tag{7}$$

where

$$\mathbf{Z} = \begin{pmatrix} \mathbf{Z}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{Z}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{Z}_K \end{pmatrix} \tag{8}$$

Thus by construction, we have the following,

Theorem 1. *If $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ belong exactly to K subspaces of rank d_k respectively, there exists $\mathbf{Z}$, such that*

$$\mathbf{X} = \mathbf{XZ}, \ \mathbf{1}^T\mathbf{Z} = \mathbf{1}^T. \tag{9}$$

where $\mathbf{Z}$ has the structure of Eq.(8) and rank$(\mathbf{Z}_k) = d_k + 1, 1 \leq k \leq K$.

Recovery of The Multiple Subspaces

Intuited by Lemma 2, and Theorem 1, one might hypothetically consider recovering the block structure by using the following optimization,

$$\begin{aligned} &\min_{\mathbf{Z}} \|\mathbf{Z}\|_*, \\ \text{s.t.} \ \ &\mathbf{X} = \mathbf{XZ}, \ \mathbf{1}^T\mathbf{Z} = \mathbf{1}^T, \end{aligned} \tag{10}$$

which is a convex problem since the objective function $\|\mathbf{Z}\|_*$ is a convex function *w.r.t* $\mathbf{Z}$ and the domain constraints $\mathbf{X} = \mathbf{XZ}$, $\mathbf{1}^T\mathbf{Z}_1 = \mathbf{1}^T$ is an affine space, which is a convex domain. This is desirable property: if a solution $\mathbf{Z}^*$ is a local solution, $\mathbf{Z}^*$ must be a global solution. However, a convex optimization could have multiple global solutions, *i.e.*, the global solution is not unique.

This optimization indeed has one optimal solution:

Theorem 2. *The optimization problem of Eq. (10) has the optimal solution*

$$\mathbf{Z}^* = \tilde{\mathbf{X}}^+\tilde{\mathbf{X}} \tag{11}$$

where

$$\tilde{\mathbf{X}} = \begin{pmatrix} \mathbf{X} \\ \mathbf{1}^T \end{pmatrix}. \tag{12}$$

In general, $\mathbf{Z}^*$ is not sparse and does not have the sparse block structure of $\mathbf{Z}$ in Eq. (8). Similar data representation model was represented in [5], which suffers from the same problem. Here we extend the model to solve the general multi-subspace problem and provide a proof of the uniqueness of the solution.

To recover a solution which has the sparse structure of Eq. (8), we add a ℓ_1 term to optimization Eq. (10) to promote sparsity of the solution, and optimize the following

$$\begin{aligned} \min_{\mathbf{Z}} \; & J_1(\mathbf{Z}) = \|\mathbf{Z}\|_* + \delta\|\mathbf{Z}\|_1 \\ \text{s.t.} \;\; & \mathbf{X} = \mathbf{X}\mathbf{Z}, \; \mathbf{1}^T\mathbf{Z}_1 = \mathbf{1}^T, \end{aligned} \tag{13}$$

where $\|\mathbf{Z}\|_1$ is the element-wise ℓ_1 norm: $\|\mathbf{Z}\|_1 = \sum_{ij} |Z_{ij}|$ and δ is model parameter which control the balance between low rank and sparsity. In our theoretical studies, we only require $\delta > 0$. Because the ℓ_1 norm is convex and the optimization problem (13) is strictly convex at the minimizer, it has unique solution.

And fortunately, for problem Eq.(13), we have the following theorem,

Proposition 1. *Assume $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K$ are independent affine subspaces. Let $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K]$, then all the minimizers of problem Eq.(13) have the form of Eq.(8). Further more, each block $\mathbf{Z}_k$ has only one connected component.*

The proof of Proposition 1 can be found in the supplementary materials of this paper.

Since each block $\mathbf{Z}_k$ has only one connected component and all the whole $\mathbf{Z}$ is block diagonal, the number of affine subspaces is trivial to recovered, which is the number of connected components of $\mathbf{Z}$. The membership of each data points to the affine spaces is also guaranteed to be recovered.

3 Multi-Subspace Representation with Noise

Typically data are drawn from multiple subspaces but with noise. Thus $\mathbf{X} = \mathbf{X}\mathbf{Z}$ does not hold anymore for any low rank $\mathbf{Z}$. On the other hand, we can combine the two constraints in Eq. (13) as,

$$\begin{pmatrix} \mathbf{X} \\ \mathbf{1}^T \end{pmatrix} = \begin{pmatrix} \mathbf{X} \\ \mathbf{1}^T \end{pmatrix} \mathbf{Z}. \tag{14}$$

With the notation of $\tilde{\mathbf{X}}$ in Eq. (12), we have $\tilde{\mathbf{X}} = \tilde{\mathbf{X}}\mathbf{Z}$. We may express the relationship as $\tilde{\mathbf{X}} = \tilde{\mathbf{X}}\mathbf{Z} + \mathbf{E}$, where $\mathbf{E}$ represents noise. To handle such noise case, in the optimization objective of Eq.(13), we add the term

$$\|\mathbf{E}\|_{\ell_2/\ell_1} = \sum_j \sqrt{\sum_i \mathbf{E}_{ij}^2} = \sum_{j=1}^n \left\| \begin{pmatrix} \mathbf{x}_j \\ 1 \end{pmatrix} - \begin{pmatrix} \mathbf{X} \\ \mathbf{1}^T \end{pmatrix} \mathbf{z}_j \right\|.$$

This is the ℓ_2/ℓ_1-norm of matrix of $\mathbf{E}$. This norm is more robust against outliers than the usual Frobenius norm. With this noise correction term, we solve,

$$\min_{\mathbf{Z}} \; \|\tilde{\mathbf{X}} - \tilde{\mathbf{X}}\mathbf{Z}\|_{\ell_2/\ell_1} + \lambda\|\mathbf{Z}\|_* + \delta\|\mathbf{Z}\|_1, \tag{15}$$

where λ and δ are parameters which control the importance of $\|\mathbf{Z}\|_*$ and $\mathbf{Z}_1$, respectively.

3.1 Multi-Subspace Representation

Notice that if the data contain noise and the constraints in Proposition 1 do not hold, we lose the guarantee of the block diagonal structure of $\mathbf{Z}$. However, since the low rank and sparsity regularizer of Eq. (15), the final solution $\mathbf{Z}$ can be interpreted as representation coefficient of $\mathbf{X}$. We call such representation as Multi-Subspace Representation (MSR).

In summary, MSR representation of data $\mathbf{X}$ is given by the following:
(1) From input data $\mathbf{X}$, solve the optimization Eq.(15) to obtain $\mathbf{Z}$;
(2) The MSR representation of $\mathbf{X}$ is $\mathbf{XZ}$, i.e., the representation of $\mathbf{x}_i$ is $\mathbf{X}\mathbf{z}_i$.

In §4, we develop an algorithm to solve Eq. (15) and in §5, some applications of our model in machine learning are given.

3.2 Relation to Previous Work

The MSR representation here is motivated by the affine subspace clustering problem. However, some properties of the representation have been investigated in previous work by other researchers. First notice that $\mathbf{Z}$ is sparse, the representation of $\mathbf{x}_i \approx \mathbf{Z}\mathbf{z}_i$ is similar to the one in sparse coding [6,7]. Interestingly, research in other communities suggests that in the natural process and even in human cognition, information is often organized in a sparse way, *e.g.* Vinge *et al.* discover that primary visual cortex (area *V1*) uses a sparse code to efficiently represent natural scenes [8].

In the sparse representation model, for each testing object, we seek a sparse representation of the testing object by all objects in training data set. Such learning mechanisms implicitly learn the structure, under the assumption that the sparse representation coefficients are imbalanced among groups. To be more specific, given a set of training data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ ($p \times n$ matrix, where p is the dimension of the data) and a testing data point $\mathbf{x}_t$, they solve the following optimization problem

$$\min_{\alpha_t} \|\mathbf{x}_t - \mathbf{X}\alpha_t\|^2 + \lambda\|\alpha_t\|_1, \tag{16}$$

where α_t ($n \times 1$ vector) has the reconstruction coefficients of x_t using all the training data objects $\mathbf{X}$, λ is the model parameter, and $\|\cdot\|_1$ is the ℓ_1 norm: $\|a\|_1 = \sum_{i=1} |a_i|$.

Wright *et al* introduce the Sparse Represented-Based Classification method [9], which uses the following strategy for class prediction,

$$\arg\min_k r_k = \|\mathbf{x}_t - \mathbf{X}\alpha_t^k\|, \tag{17}$$

where r_k is the representation error using the training samples in group k and α_t^k is obtained by setting the coefficients in α_t, corresponding to training samples not in class k, to zero, *i.e.*

$$\alpha_t^k(i) = \begin{cases} \alpha_t(i) & \text{if } i \in C_k, \\ 0 & \text{otherwise,} \end{cases}$$

where C_k is a set of all data points in class $k, k = 1, 2, \cdots, K$, and K is the number of classes.

On the other hand, $\mathbf{Z}$ in our model is also low rank, which is a natural requirement of most of data representation techniques, such as the low rank kernel methods [10] and robust Principle Component Analysis [11]. One can easily find literacy of the low rank representation in real world applications in various domains which indicates that low rank is one of the intrinsic properties of the data we observe, *e.g.* the missing value recover of DNA microarrays [12].

By combining the two basic properties (sparsity and low rank), our model naturally captures a proper representation of the data. We will demonstrate the quality of such representation using comprehensive empirical evidences in the experimental section.

4 An Efficient Algorithm and Analysis

4.1 Outline of the Algorithm

Assume we are solving a general problem of

$$J(\mathbf{x}) = f(\mathbf{x}) + \phi(\mathbf{x}), \tag{18}$$

where $f(\mathbf{x})$ is smooth and $\phi(\mathbf{x})$ is non-smooth and convex. If one of the elements in subgradient of $\phi(\mathbf{x})$ can be written as product of $g(\mathbf{x})$ and $h(\mathbf{x})$, *i.e.*,

$$g(\mathbf{x})h(\mathbf{x}) \in \partial\phi(\mathbf{x}),$$

where $h(\mathbf{x})$ is smooth and $\partial\phi(\mathbf{x})$ is the subgradient of $\phi(x)$, then instead of solving Eq. (18), we iteratively solve the following,

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}} \tilde{J}(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}^t) \int h(\mathbf{x}) d\mathbf{x}. \tag{19}$$

Notice that $\partial\tilde{J}(\mathbf{x})/\partial\mathbf{x} \in \partial J(\mathbf{x})$ when $\mathbf{x} = \mathbf{x}^t$. Hopefully, at convergence, $\mathbf{x}^{t+1} = \mathbf{x}^t$, then $\mathbf{0} \in \partial J(\mathbf{x})$ at $\mathbf{x}^t$, which means $\mathbf{x}^t$ is an optimizer of $J(\mathbf{x})$.

In general, the iterative steps in Eq. (19) cannot guarantee the convergence of $\mathbf{x}$ (*i.e.* $\mathbf{x}^{t+1} = \mathbf{x}^t$), and even the convergence of $J(\mathbf{x})$ (*i.e.* $J(\mathbf{x}^{t+1}) = J(\mathbf{x}^t)$). Fortunately, in our case of Eq. (15), our optimization technique guarantees both, and thus our algorithm guarantees to be an optimizer. Further more, in our algorithm, optimization problem in Eq. (19) has a close form solution, thus our algorithm is efficient.

4.2 Optimization Algorithm

Here we first present the optimization algorithm of Eq.(15), and then present theoretical analysis of the algorithm.

The algorithm is summarized in Algorithm 1. In the algorithm, $\mathbf{z}_i$ denotes the i-th column of $\mathbf{Z}$. The converged optimal solution is only weakly dependent on

Algorithm 1. $(\mathbf{X}, \lambda, \delta)$

```
Input: Data $\mathbf{X}$, model parameters $\lambda, \delta$
Output: $\mathbf{Z}$ which optimizes Eq.(15).
Initialization: Compute $\tilde{\mathbf{X}}$ using Eq. (12), $\mathbf{Z} = \mathbf{0}$.
while not converged do
  $\mathbf{B} = \left(\mathbf{Z}\mathbf{Z}^T + \epsilon\mathbf{I}\right)^{-1/2}$
  for $i = 1 : n$ do
    $\mathbf{d}_i = \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{X}}\mathbf{z}_i\|$,
    $\mathbf{D}_i = \mathbf{diag}\left(Z_{1i}^{-1}, Z_{2i}^{-1}, \cdots, Z_{ni}^{-1}\right)$,
    $\mathbf{z}_i = \left[\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} + \lambda\mathbf{d}_i\left(\mathbf{B} + \delta\mathbf{D}\right)\right]^{-1}\tilde{\mathbf{X}}^T\tilde{\mathbf{x}}_i$,
  end for
end while
Output: $\mathbf{Z}$
```

parameter. We set δ to $\delta = 1$. ϵ is an auxiliary constant for improving numerical stability in computing trace norm. We set $\epsilon = 10^{-8}$ in all experiments.

In the third line of the **for** loop, we are actually solving the problem in Eq. (19). In practice, we do not explicitly compute the inverse. Instead, we solve the following linear equation to obtain $\mathbf{z}_i$,

$$\left[\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} + \lambda\mathbf{d}_i\left(\mathbf{B} + \delta\mathbf{D}\right)\right]\mathbf{z}_i = \tilde{\mathbf{X}}^T\tilde{\mathbf{x}}_i. \tag{20}$$

The algorithm is simple which involves no other optimization procedures. The algorithm generally converges in about 10 iterations in our experiments.

We have developed theoretical analysis for this algorithm, convering three properties for this algorithm: convergence, objective function value decreasing monotonically, and converging to global solution.

4.3 Theoretical Analysis of Algorithm 1

Before presenting the main theories for Algorithm 1, we first introduce two useful lemmas here.

Lemma 3

$$\|\mathbf{Z}\|_* = \lim_{\epsilon \longrightarrow 0} \mathbf{tr}\left(\mathbf{Z}\mathbf{Z}^T + \epsilon\mathbf{I}\right)^{1/2}, \tag{21}$$

and

$$\lim_{\epsilon \longrightarrow 0}\left(\mathbf{Z}\mathbf{Z}^T + \epsilon\mathbf{I}\right)^{-1/2}\mathbf{Z} \in \partial\|\mathbf{Z}\|_*, \tag{22}$$

where $\partial\|\mathbf{Z}\|_*$ is the subgradient of trace norm.

Here $\epsilon\mathbf{I}$ is introduced for numerical stability.

Lemma 4. *Assume matrices $\mathbf{Z}$ and $\mathbf{Y}$ have the same size. Let $\mathbf{A} = \left(\mathbf{Y}\mathbf{Y}^T + \epsilon\mathbf{I}\right)^{1/2}$ and $\mathbf{B} = \left(\mathbf{Z}\mathbf{Z}^T + \epsilon\mathbf{I}\right)^{1/2}$. Then the following holds*

$$\mathbf{tr}\mathbf{A} - \mathbf{tr}\mathbf{B} + \frac{1}{2}\mathbf{tr}\mathbf{Z}^T\mathbf{B}^{-1}\mathbf{Z} - \frac{1}{2}\mathbf{tr}\mathbf{Y}^T\mathbf{B}^{-1}\mathbf{Y} \leq 0. \tag{23}$$

Proof

$$\begin{aligned}
&\mathbf{trA} - \mathbf{trB} + \frac{1}{2}\mathbf{tr}\mathbf{Z}^T\mathbf{B}^{-1}\mathbf{Z} - \frac{1}{2}\mathbf{tr}\mathbf{Y}^T\mathbf{B}^{-1}\mathbf{Y}\\
=&\mathbf{trA} - \mathbf{trB} + \frac{1}{2}\mathbf{trB}^{-1}\left(\mathbf{ZZ}^T - \mathbf{YY}^T\right)\\
=&\frac{1}{2}\mathbf{trB}^{-1}\left(2\mathbf{BA} - 2\mathbf{B}^2 + \mathbf{ZZ}^T - \mathbf{YY}^T\right)\\
=&\frac{1}{2}\mathbf{trB}^{-1}\left(2\mathbf{BA} - 2\mathbf{B}^2 + \mathbf{ZZ}^T + \epsilon\mathbf{I} - \mathbf{YY}^T - \epsilon\mathbf{I}\right)\\
=&\frac{1}{2}\mathbf{trB}^{-1}\left(2\mathbf{BA} - \mathbf{B}^2 - \mathbf{A}^2\right)\\
=&-\frac{1}{2}\mathbf{trB}^{-1/2}\left(\mathbf{A} - \mathbf{B}\right)^2\mathbf{B}^{-1/2} \leq 0.
\end{aligned}$$

One should notice that here **A** and **B** are symmetric full rank matrices.

Lemma 4 serves as a crucial part of our main theorem, which is stated as follows,

Theorem 3. *Algorithm 1 monotonically decreases the following objective,*

$$\min_{\mathbf{Z}} J(\mathbf{Z}) = \|\tilde{\mathbf{X}} - \tilde{\mathbf{X}}\mathbf{Z}\|_{\ell_2/\ell_1} + \lambda\mathbf{tr}\left(\mathbf{ZZ}^T + \epsilon\mathbf{I}\right)^{\frac{1}{2}} + \delta\|\mathbf{Z}\|_1, \tag{24}$$

i.e. $J(\mathbf{Z}_{t+1}) \leq J(\mathbf{Z}_t)$*, where* $\mathbf{Z}_t$ *is the solution of* **Z** *in the* t*-th iteration.*

Since the objective in Eq.(24) is lower bounded by 0, Theorem 3 guarantees the convergence of the objective value. Further more, we have

And according to Lemma 3, we know that the above solution is also the optimal solution of Eq.(15) when $\epsilon \to 0$.

We provide the proofs of all the theoretical analysis above in the supplementary materials.

5 Applications

5.1 Using Multi-Subspace Representation as Preprocessing

Since **Z** is low rank, **XZ** is also low rank. And since **Z** is sparse, **XZ** can be interpreted as a sparse coding representation of **X**. According to the analysis in §3.2, we hopefully improve the qualities of the data representation by using **XZ**. In our study, we replace **X** by **XZ** as a preprocessing step for various machine learning problems, where **Z** is the optimal solution of Eq. (15).

Notice that the learning of **Z** in Eq. (15) is unsupervised, which requires no further label information. Thus we can apply it as preprocessing for any machine learning tasks, as long as the data are represented in Euclidean space. In this paper, we employ MSR for clustering, semi-supervised learning, and classification. We will demonstrate the performance of the preprocessing in the experimental section.

5.2 Using Multi-Subspace Representation as Classifier

Here we try to directly make use of our MSR model as a standalone classifier. Assume we have n data points in the data set, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ and the first m data points have discrete class labels $y_1, y_2, \cdots, y_m$ in K classes, $y_i \in \{1, 2, \cdots, K\}$. The classification problem is to determine the class label of $\mathbf{x}_i, i = m+1, \cdots, n$. Let $\mathbf{Z}$ be the optimal solution of Eq.(15) for n data points. The MSR representation of each image is $\mathbf{X}\mathbf{z}_i,\ i = 1, \cdots, n$. The class prediction of our model for unlabeled data $\mathbf{x}_t,\ t = m+1, \cdots, n$, is

$$\arg\min_k r_k = \|\mathbf{X}\mathbf{z}_t - \hat{\mathbf{x}}_t^k\|,\ \hat{\mathbf{x}}_t^k = \sum_{i \in C_k} \mathbf{x}_i Z_{it}. \tag{25}$$

Here $\hat{\mathbf{x}}_t^k$ is the representation of testing object $\mathbf{x}_t$ using objects in class C_k, $k = 1, 2, \cdots, K$.

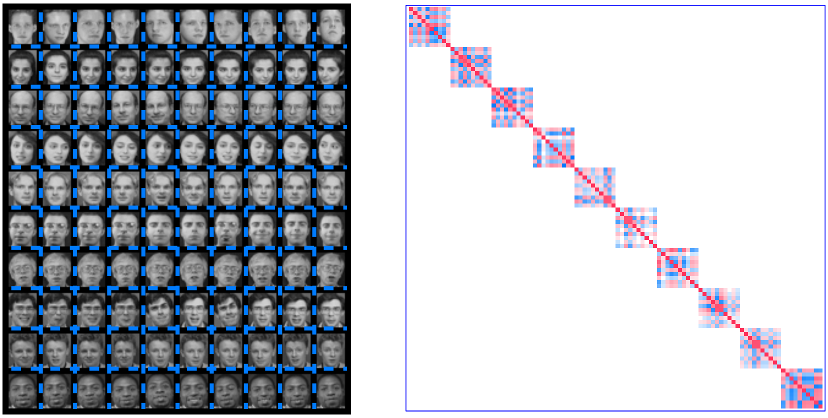
The classification strategy is similar with Wright *et al*'s approach [9]. We will compare the two models in the experimental section.

6 Experiment

6.1 A Toy Example

We demonstrate with toy example of the affine space recovering by our method in Figure 2. (**a**) shows 100 images from 10 groups used in this example, which are selected from the AT&T data set, details can be found in the experimental section. In order to obtain 10 affine subspaces which satisfy the constraints in Proposition 1, we remove the last principle component in each group of face images. To be more specific, for each group $\mathbf{X}_k$, we first subtract the data points by the group mean $\mathbf{m}_k : \bar{\mathbf{X}}_k = \mathbf{X}_k - \mathbf{m}_k \mathbf{1}^T$, then perform a PCA (Principle Component Analysis) on the zero-mean data and keep the first 8 principle components and get rid of the 9-th principle component. Then the data is projected back on to the original space and the mean $\mathbf{m}_k$ is added back. Assume the resulting PCA projection is $\mathbf{U}_k$ then the processed data $\mathbf{Y} = \mathbf{U}_k \mathbf{U}_k^T \bar{\mathbf{X}}_k + \mathbf{m}_k$ are used in our example, $k = 1, 2, \cdots, 10$. The images in which the last principle component have been removed are shown in Figure 2 (**a**). Notice that they are visually almost identical to the original image since the energy of the last component is close to zero. Then we solve Eq. (13) and the optimal solution is shown in Figure 2 (**b**), in which white color represents zeros, blue colors represent negative values, and red positive values. One can see that within each group, the values of the subgraph represented by $\mathbf{Z}_k$ (defined in Eq. (8)) is a single connected component and among the ten $\mathbf{Z}_k, k = 1, 2, \cdots, 10$ they are disconnected components.

As suggested in the previous section, our multi-subspace representation model has various potential real world applications. In the section, we will verify the quality of our model as a preprocessing method in three types of machine learning tasks, *i.e.* clustering, semi-supervised learning, and classification. We also evaluate our model as a standalone classifier.



(a)

(b)

Fig. 2. A toy example of multi-subspace discovery problem and our solution. **(a)**: 100 images in which the last component has been removed within each group. Each row is one group which has 10 images. Within each group, the data are rank deficient, which satisfy the conditions in Proposition 1. **(b)**: the optimal solution of $\mathbf{Z}$ in Eq. (13). White color represents zeros, blue colors represent negative values, and red positive values. Within each group, the values of the subgraph represented by $\mathbf{Z}_k$ (defined in Eq. (8)) is a single connected component and the among the 10 $\mathbf{Z}_k, k = 1, 2, \cdots, 10$ they are disconnected components.

6.2 Experimental Settings

Datasets

We evaluate the performance of our model on 5 real world datasets, including two face image data bases, **LFW** (Labeled Faces in the Wild)[2], **AT&T**[3], two UCI datasets **Austrian** and **Dermatology** [13], and one handwritten character data BinAlpha[4]. All the data sets are used with the original data, without any further preprocessing.

Compared Methods

For the usage of preprocessing of our model, we compare 3 clustering algorithms (Normalized Cut [14], Spectral Embedding Clustering [15] and K-means), two standard semi-supervised learning algorithms (Local and Global Constancy by [16] and Gaussian Fields and Harmonic Functions by [17]), and two standard classification algorithms (linear Support Vector Machines and k-Nearest Neighbor).

For the usage of standalone classifier, we compare our method with Wright *et. al*'s sparse representation based approach [9].

[2] http://www.itee.uq.edu.au/∼conrad/lfwcrop/

[3] http://people.cs.uchicago.edu/˜dinoj/vis/ORL.zip

[4] http://www.cs.toronto.edu/˜roweis/data.html

Validation Settings

All the clustering algorithms compared in our experiments require random initializations. Thus we run the algorithms for 50 random trials and report the averages. For semi-supervised learning, we randomly split the data into 30% and 70% where the 30% of the data points are used as labeled data and 70% are used as unlabeled data. We repeat the random splitting for 50 times, where the average result is reported. For classification, when comparing our method as a preprocessing algorithm, we use the same splitting strategy as in semi-supervised learning, but splitting in to 50% for training and the other half for testing. For classification, when comparing our method as a standalone classifier, we use 30% for training and the rest 70% for testing. The reason is that for some of the datasets, the data points are well separated and the classification accuracy is very high, then the difference between approaches is not obvious. Thus here we use fewer data samples as the training set to enlarge the differences.

Parameter settings

K-means has no parameters. For kNN we use $k = 1$, *i.e.* just use the nearest neighbor classifier. For the Normalized Cut (NCut), Spectral Embedding Clustering (SEC) in clustering, Local and Global Constancy (LGC), and Gaussian Fields and Harmonic Functions (GFHF) in semi-supervised learning, we establish the graph using Gaussian kernel: $W_{ij} = \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2\right)$, where γ is the parameter which is set to be $\gamma = [0.1, 0.5, 1, 2, \cdots, 30]$ and σ is the average of pairwise Euclidian distances among all data points.

For Wright *et. al*'s sparse representation (SR), we use LARS [18] to obtain the full LASSO path solution and use m top ranked coefficients according to the shrinking order in LARS solution path. We choose m from $m = 1, 2, \cdots \min(n, p)$ where n is the number of data points and p is the number of data dimension. The reason we use LARS is that it is more efficient than any other ℓ_1 solver in the sense that LARS computes all the possible solution with different parameters at once and for other solver, we need to retrain the model every time we change the parameter, which is time consuming for the purpose of highly parameter tuning. For our method, we choose λ from $[0.5, 0.6, \cdots, 2.5]$.

6.3 Experimental Results

For the usage of preprocessing our model, the results are shown in Figure 3. Here we show the average accuracies for both original data without processing (marked as **Orig** in the figure) and the corresponding method on the preprocessed data by our method (marked as **MSR**). We further plot the original accuracy values of all the 50 random trials for each methods to visualize the overall differences of the performance.

One-way ANOVA (Analysis of Variance) is performed to test how significantly our method is better than the original method, and corresponding p value is also shown in the figure. $p \leq \epsilon$ means p is less than any positive values in machine precision, *i.e.* the p value is very close to 0.

Out of the $5 \times 7 = 35$ comparisons, our method significantly outperforms the original methods in 33 comparisons, with $p \leq 0.03$. There is one case (SVM on

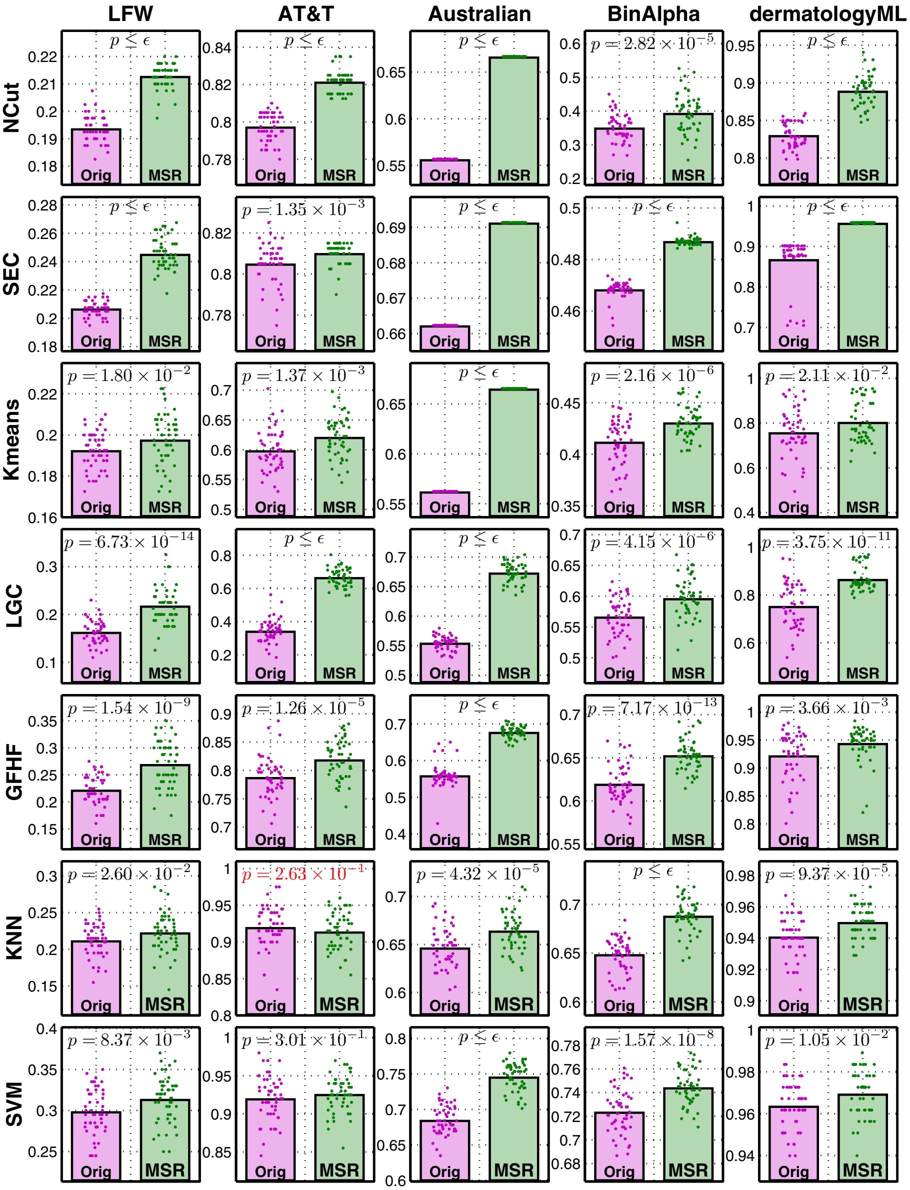


Fig. 3. Experimental results of our method as a preprocessing method on 7 learning methods and 5 data sets. The scattering dots represent the accuracy values of the methods and bars represent the averages. **Orig** and **MSR** denote the corresponding method on the original data and on the preprocessed by our method, respectively. The p stands for the significance of the one-way ANOVA test (for the hypothesis of "our method is better than the original method"). Out of 35 comparison, our method significantly outperforms the original methods in 33 cases, with $p \leq 0.03$. ϵ is the smallest positive values by machine precision.

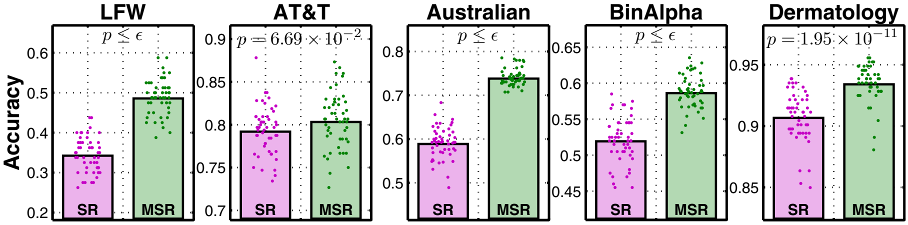


Fig. 4. A comparison of our model (**MSR**) and the Sparse Representation based method (**SR**) on 5 data sets. The p values represents the significance of one-way ANOVA test of the hypothesis "our method is better than SR".

AT&T data set) where our method is better but with no significant evidence. There is also another case in which our method is worse than the original method (kNN on **AT&T**), but the difference is not significant ($p = 0.263$).

For our model as a standalone classifier, the comparison results with Sparse Representation based method are shown in Figure 4. Out of 5 data sets, our method is significantly better than the Sparse Representation based method in four with $p \leq 0.01$.

7 Conclusions

In this paper, we present the multi-subspace representation and discovery model, which is motivated by the multi-subspace discovery problem. We solve the multi-subspace discovery problem by providing block diagonal representation matrix where the data points are connected in the same subspace and disconnected for different subspace. We then extend our approach to handle noisy real world data which leads to the Multi-Subspace Representation. We develop an efficient algorithm for the presented model and a global optimizer is guaranteed. Empirical studies suggest that our method improves the quality of the data by sparse and low rank representation and the induced standalong classifier outperforms standard sparse representation approach.

Acknowledgment. This research is partially supported by NSF-CCF-0830780, NSF-DMS-0915228, NSF-CCF-0917274.

References

1. Jenatton, R., Obozinski, G., Bach, F.: Structured sparse principal component analysis. In: Proc. AISTATS. Citeseer (2009)
2. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B 67, 301–320 (2005)
3. Beygelzimer, A., Kephart, J., Rish, I.: Evaluation of optimization methods for network bottleneck diagnosis. In: ICAC (2007)

4. Luo, D., Ding, C., Huang, H.: Towards structural sparsity: An explicit ℓ_2/ℓ_0 approach. In: 2010 IEEE International Conference on Data Mining, pp. 344–353. IEEE, Los Alamitos (2010)
5. Liu, G., Lin, Z., Yu, Y.: Robust subspace segmentation by low-rank representation. In: Proceedings of the 26th International Conference on Machine Learning. Citeseer, Haifa (2010)
6. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision research 37(23), 3311–3325 (1997)
7. Tibshirani, R.: Regression shrinkage and selection via the LASSO. J. Royal. Statist. Soc. B 58, 267–288 (1996)
8. Vinje, W.E., Gallant, J.L.: Sparse coding and decorrelation in primary visual cortex during natural vision. Science 287, 1273 (2000)
9. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 210–227 (2009)
10. Bach, F., Jordan, M.: Predictive low-rank decomposition for kernel methods. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 33–40. ACM, New York (2005)
11. Candes, E., Li, X., Ma, Y., Wright, J.: Robust principal component analysis (2009) (preprint)
12. Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.B.: Missing value estimation methods for DNA microarrays. Bioinformatics 17, 520 (2001)
13. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
14. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888–905 (2002)
15. Nie, F., Xu, D., Tsang, I., Zhang, C.: Spectral embedded clustering. In: Proceedings of the 21st International Joint Conference on Artifical intelligence, pp. 1181–1186. Morgan Kaufmann Publishers Inc., San Francisco (2009)
16. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Proc. Neural Info. Processing Systems (2003)
17. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proc. Int'l Conf. Machine Learning (2003)
18. Efron, B., Hastie, T., Johnstone, L., Tibshirani, R.: Least angle regression. Annals of Statistics 32, 407–499 (2004)