Joint Feature Selection and Subspace Learning

Quanquan Gu, Zhenhui Li and Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign
{qgu3,zli28,hanj}@illinois.edu

Abstract

Dimensionality reduction is a very important topic in machine learning. It can be generally classified into two categories: feature selection and subspace learning. In the past decades, many methods have been proposed for dimensionality reduction. However, most of these works study feature selection and subspace learning independently. In this paper, we present a framework for joint feature selection and subspace learning. We reformulate the subspace learning problem and use $L_{2,1}$-norm on the projection matrix to achieve row-sparsity, which leads to selecting relevant features and learning transformation simultaneously. We discuss two situations of the proposed framework, and present their optimization algorithms. Experiments on benchmark face recognition data sets illustrate that the proposed framework outperforms the state of the art methods overwhelmingly.

1 Introduction

High-dimensional data in the input space is usually not good for classification due to the *curse of dimensionality*. A common way to resolve this problem is dimensionality reduction, which has attracted much attention in machine learning community in the past decades. Generally speaking, dimensionality reduction techniques can be classified into two categories: (1) feature selection [Guyon and Elisseeff, 2003]: to select a subset of most representative or discriminative features from the input feature set, and (2) subspace learning [Belhumeur *et al.*, 1997][He and Niyogi, 2003][He *et al.*, 2005][Yan *et al.*, 2007] (a.k.a feature transformation): to transform the original input features to a lower dimensional subspace.

The most popular subspace learning methods include *Principal Component Analysis* (PCA) [Belhumeur *et al.*, 1997], *Linear Discriminant Analysis* (LDA) [Belhumeur *et al.*, 1997], *Locality Preserving Projection* (LPP) [He and Niyogi, 2003] and *Neighborhood Preserving Embedding* (NPE) [He *et al.*, 2005]. Despite different motivations of these methods, they can all be interpreted in a unified *Graph Embedding* framework [Yan *et al.*, 2007].

One major disadvantage of the above methods is that the learned projection is a linear combination of all the original features, thus it is often difficult to interpret the results. Sparse subspace learning methods attempted to solve this problem. For example, [Zou *et al.*, 2004] proposed a sparse PCA algorithm based on ℓ_2-norm and ℓ_1-norm regularization. [Moghaddam *et al.*, 2006] proposed both exact and greedy algorithms for binary class sparse LDA as well as its spectral bound. [Cai *et al.*, 2007] proposed a unified sparse subspace learning (SSL) framework based on ℓ_1-norm regularized *Spectral Regression*.

However, the selected features by sparse subspace methods are independent and generally different for each dimension of the subspace. See Figure 1 (a) for an illustrative toy example of the projection matrix learned by SSL. Each *row* of the projection matrix corresponds to a *feature*, while each *column* corresponds to a *dimension of the subspace*. We can see that for the first dimension of the subspace, the 3rd and 6th features are not selected, while for the second dimension of the subspace, the selected features are all except the 1st and 4th one. Hence it is still unclear which features are really useful as a whole. Our goal is to learn a projection matrix like Figure 1 (b), which has *row-sparsity* (elements in a row are all zero). Hence it is able to discard the irrelevant features (e.g., the 1st, 5th and 7th features) and transform the relevant ones simultaneously. One intuitive way is performing feature selection [Guyon and Elisseeff, 2003] before subspace learning. However, since these two sub-processes are conducted individually, the whole process is likely suboptimal.

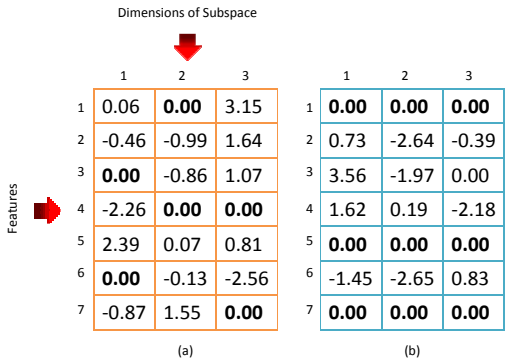


Figure 1: An illustrative toy example of the projection matrices learned by (a) Sparse subspace learning; and (b) feature selection and subspace learning

Based on the above motivation, in this paper, we aim to jointly perform feature selection and subspace learning. To achieve this goal, we reformulate subspace learning as solving a linear system equation, during which we use $L_{2,1}$-norm on the projection matrix, encouraging row-sparsity. It is worth noting that $L_{2,1}$-norm has already been successfully applied in Group Lasso [Yuan *et al.*, 2006], multi-task feature learning [Argyriou *et al.*, 2008], joint covariate selection and joint subspace selection [Obozinski *et al.*, 2010]. The resulted optimization problem includes two situations, for each of which we present a very simple algorithm, that is theoretically guaranteed to converge. Experiments on benchmark face recognition data sets demonstrate the effectiveness of the proposed framework.

The remainder of this paper is organized as follows. In Section 2, we briefly introduce the graph embedding view of subspace learning. In Section 3, we present a framework for joint feature selection and subspace learning. In Section 4, we review some related works. Experiments on benchmark face recognition data sets are demonstrated in Section 5. Finally, we draw a conclusion and point out future work in Section 6.

1.1 Notations

Given a data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, we aim to learn a projection matrix $\mathbf{A} \in \mathbb{R}^{d\times m}$, projecting the input data into an m-dimensional subspace. For a matrix $\mathbf{A} \in \mathbb{R}^{d\times m}$, we denote the ith row of $\mathbf{A}$ by $\mathbf{a}^i$, and the jth column of $\mathbf{A}$ by $\mathbf{a}_j$. The Frobenius norm of $\mathbf{A}$ is defined as $||\mathbf{A}||_F = \sqrt{\sum_i^d ||\mathbf{a}^i||_2^2}$, and the $L_{2,1}$-norm of $\mathbf{A}$ is defined as $||\mathbf{A}||_{2,1} = \sum_i^d ||\mathbf{a}^i||_2$.

2 Graph Embedding View of Subspace Learning

Many dimensionality reduction methods have been proposed to find low-dimensional representation of $\mathbf{x}_i$. Despite different motivations of these methods, they can be nicely interpreted in a general graph embedding framework [Yan *et al.*, 2007]. In graph embedding, we construct a data graph $\mathcal{G}$ whose vertices correspond to $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Let $\mathbf{W} \in \mathbb{R}^{n\times n}$ be a symmetric adjacency matrix with W_{ij} characterizes the favorite relationship among the data. The purpose of graph embedding is to find the optimal low-dimensional vector representation for the vertices of graph $\mathcal{G}$ that best preserves the relationship between the data points. In this paper, we focus on linear dimensionality reduction. That is, $\mathbf{X}^T\mathbf{A}$. The optimal $\mathbf{A}$ is given by the following optimization problem,

$$\begin{aligned} \min_{\mathbf{A}} \quad & \text{tr}(\mathbf{A}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{A}) \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{A} = \mathbf{I}, \end{aligned} \tag{1}$$

where $D_{ii} = \sum_j W_{ij}$ is a diagonal matrix, and $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is called *graph Laplacian* [Chung, 1997], $\mathbf{I}$ is the identity matrix with proper size. The above problem can be solved by generalized eigen-decomposition $\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{A} = \mathbf{\Lambda}\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{A}$, where $\mathbf{\Lambda}$ is a diagonal matrix whose diagonal elements are eigenvalues.

With different choices of $\mathbf{W}$, the linear graph embedding framework leads to many popular linear dimensionality reduction methods, e.g. PCA [Belhumeur *et al.*, 1997], LDA [Belhumeur *et al.*, 1997], LPP [He and Niyogi, 2003] and NPE [He *et al.*, 2005]. We briefly give two examples below.

LDA: Suppose we have c classes and the k-th class have n_k samples, $n_1 + \cdots + n_c = n$. Define

$$W_{ij} = \begin{cases} \frac{1}{n_k}, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the } k\text{-th class} \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

LPP [He and Niyogi, 2003]: Define

$$W_{ij} = \begin{cases} d(\mathbf{x}_i, \mathbf{x}_j), & \text{if } \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i) \text{ or } \mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_j) \\ 0, & \text{otherwise} \end{cases}, \tag{3}$$

where $\mathcal{N}_k(\mathbf{x}_i)$ denotes the set of k nearest neighbors of $\mathbf{x}_i$, $d(\mathbf{x}_i, \mathbf{x}_j)$ measures the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, which can be chosen as Gaussian kernel $e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}$ or cosine distance $\frac{\mathbf{x}_i^T\mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|}$. For more examples and other extensions of graph embedding, please refer to [Yan *et al.*, 2007].

3 Joint Feature Selection and Subspace Learning

Since each row of the projection matrix corresponds to a feature in the original space, in order to do feature selection, it is desirable to have some rows of the projection matrix be all zeros. This motivates us to use $L_{2,1}$-norm on the projection matrix, which leads to row-sparsity of the projection matrix. As a result, based on Eq. (1), we formulate joint feature selection and subspace learning as follows,

$$\begin{aligned} \min_{\mathbf{A}} \quad & ||\mathbf{A}||_{2,1} + \mu\text{tr}(\mathbf{A}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{A}) \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{A} = \mathbf{I}, \end{aligned} \tag{4}$$

where μ is a regularization parameter. Although the objective function is convex, the constraint is not. Hence it is difficult to optimize. In the following, we will reformulate the problem to make it easy to be solved.

Theorem 3.1. *Let $\mathbf{Y} \in \mathbb{R}^{n\times m}$ be a matrix of which each column is an eigenvector of eigen-problem $\mathbf{W}\mathbf{y} = \lambda\mathbf{D}\mathbf{y}$. If there exists a matrix $\mathbf{A} \in \mathbb{R}^{d\times m}$ such that $\mathbf{X}^T\mathbf{A} = \mathbf{Y}$, then each column of $\mathbf{A}$ is an eigenvector of eigen-problem $\mathbf{X}\mathbf{W}\mathbf{X}^T\mathbf{a} = \lambda\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{a}$ with the same eigenvalue λ.*

Proof. This is the corollary of Theorem 1 in [Cai *et al.*, 2007]. □

Theorem 3.1 shows that instead of solving the eigen-problem $\mathbf{X}\mathbf{W}\mathbf{X}^T\mathbf{A} = \mathbf{\Lambda}\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{A}$, $\mathbf{A}$ can be obtained by the following two steps:

1. Solve the eigen-problem $\mathbf{W}\mathbf{Y} = \mathbf{\Lambda}\mathbf{D}\mathbf{Y}$ to get $\mathbf{Y}$;
2. Find $\mathbf{A}$ which satisfies $\mathbf{X}^T\mathbf{A} = \mathbf{Y}$.

Note that only the second step involves $\mathbf{A}$. $\mathbf{X}^T\mathbf{A} = \mathbf{Y}$ is a linear system problem, which may behave in any one of three possible ways: (1) The system has infinitely many solutions; (2) The system has a single unique solution; and (3) The system has no solution. In the rest of this section, we will discuss (1) in one situation and (2) (3) in another situation.

3.1 Situation 1

When the linear system has infinitely many solutions, we formulate the proposed method as

$$\begin{aligned} \min_{\mathbf{A}} \quad & ||\mathbf{A}||_{2,1} \\ \text{s.t.} \quad & \mathbf{X}^T\mathbf{A} = \mathbf{Y}. \end{aligned} \tag{5}$$

The optimization problem in Eq. (5) is similar to the problem appeared in Multiple Measurement Vector model in signal processing [Sun *et al.*, 2009]. We derive a very simple algorithm in the sequel.

The lagrangian function of the problem in Eq. (5) is

$$L(\mathbf{A}) = ||\mathbf{A}||_{2,1} - \text{tr}(\mathbf{\Gamma}^T(\mathbf{X}^T\mathbf{A} - \mathbf{Y})). \tag{6}$$

Taking the derivative of $L(\mathbf{A})$ with respect to $\mathbf{A}$[1], and setting the derivative to zero, we get

$$\frac{\partial L(\mathbf{A})}{\partial \mathbf{A}} = \mathbf{G}\mathbf{A} - \mathbf{X}\mathbf{\Gamma}, \tag{7}$$

where $\mathbf{G}$ is a diagonal matrix with the i-th diagonal element equal to

$$g_{ii} = \begin{cases} \mathbf{0}, & \text{if } \mathbf{a}^i = \mathbf{0} \\ \frac{1}{||\mathbf{a}^i||_2}, & \text{otherwise} \end{cases}. \tag{8}$$

Left multiplying the two sides of Eq. (7) by $\mathbf{X}^T\mathbf{G}^{-1}$, and using the constraint $\mathbf{X}^T\mathbf{A} = \mathbf{Y}$, we have

$$\mathbf{\Gamma} = (\mathbf{X}^T\mathbf{G}^{-1}\mathbf{X})^{-1}\mathbf{Y} = 0. \tag{9}$$

Note that when $g_{ii} = 0$, $\mathbf{G}^{-1}$ cannot be computed. This is handled by adding a smoothing term ϵ to g_{ii}. It can be justified that the algorithm converges to the global solution if decreasing value of ϵ is used.

Substituting Eq. (9) into Eq. (7), we get

$$\mathbf{A} = \mathbf{G}^{-1}\mathbf{X}(\mathbf{X}^T\mathbf{G}^{-1}\mathbf{X})^{-1}\mathbf{Y}. \tag{10}$$

In summary, we present the algorithm for optimizing Eq.(5) in Algorithm 1.

Algorithm 1 Joint Feature Selection and Subspace Learning (Situation 1)

Initialize: $\mathbf{G}_0 = \mathbf{I}, t = 0$;
Compute $\mathbf{Y}$ based on $\mathbf{W}\mathbf{Y} = \mathbf{\Lambda}\mathbf{D}\mathbf{Y}$;
repeat
  Compute $\mathbf{A}_{t+1} = \mathbf{G}_t^{-1}\mathbf{X}(\mathbf{X}^T\mathbf{G}_t^{-1}\mathbf{X})^{-1}\mathbf{Y}$;
  Compute $\mathbf{G}_{t+1}$ based on $\mathbf{A}_{t+1}$;
  $t = t + 1$;
until convergence

The convergence of this algorithm was proved in [Nie *et al.*, 2010]. In addition, there is no additional parameter introduced besides the parameters that are needed to construct the graph as in traditional linear graph embedding. This is a very appealing property.

[1]For $||\mathbf{A}||_{2,1}$, we compute its sub-gradient because it is not smooth.

3.2 Situation 2

When the linear system has a single unique solution, then it is also the solution of Eq. (5). However, it does not have row sparsity. When the linear system has no solution, then the formulation in Eq. (5) will have no solution either. In both of these cases, we turn to solve the constrained problem as follows,

$$\begin{aligned} \min_{\mathbf{A}} \quad & ||\mathbf{A}||_{2,1} \\ \text{s.t.} \quad & ||\mathbf{X}^T\mathbf{A} - \mathbf{Y}||_F^2 \leq \delta. \end{aligned} \tag{11}$$

Or equivalently the regularized problem,

$$\min_{\mathbf{A}} ||\mathbf{A}||_{2,1} + \mu||\mathbf{X}^T\mathbf{A} - \mathbf{Y}||_F^2. \tag{12}$$

This is a Group Lasso problem [Yuan *et al.*, 2006]. When $\mu = \infty$, the optimization problem in Eq. (12) degenerates to that in Eq. (5). Note that it is difficult to give an analytical relationship between δ and μ. However, such a relationship is not crucial for our problem. The objective function in Eq. (12) is a non-smooth but convex function. In the following, we will derive an algorithm which is similar to Algorithm 1 for solving Eq. (12).

Taking the derivative of Eq. (12) with respect to $\mathbf{A}$, and setting the derivative to zero, we get

$$\frac{\partial L(\mathbf{A})}{\partial \mathbf{A}} = \mathbf{G}\mathbf{A} + 2\mu(\mathbf{X}\mathbf{X}^T\mathbf{A} - \mathbf{X}\mathbf{Y}) = 0, \tag{13}$$

which leads to

$$\mathbf{A} = 2\mu(\mathbf{G} + 2\mu\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{Y}, \tag{14}$$

where $\mathbf{G}$ is defined in Eq. (8). Since $\mathbf{G}$ also depends on $\mathbf{A}$, the above closed-form expression of the optimal $\mathbf{A}$ is fundamentally a fixed-point iteration.

According to the Woodbury matrix identity [Golub and Loan, 1996]

$$(\mathbf{A}+\mathbf{U}\mathbf{C}\mathbf{V})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1}, \tag{15}$$

we can further get

$$\begin{aligned} \mathbf{A} &= 2\mu\mathbf{G}^{-1}\mathbf{X}\mathbf{Y} - 2\mu\mathbf{G}^{-1}\mathbf{X}(\mathbf{I} - (\mathbf{X}^T\mathbf{G}^{-1}\mathbf{X} + \frac{1}{2\mu}\mathbf{I})^{-1})\mathbf{Y} \\ &= \mathbf{G}^{-1}\mathbf{X}(\mathbf{X}^T\mathbf{G}^{-1}\mathbf{X} + \frac{1}{2\mu}\mathbf{I})^{-1}\mathbf{Y}. \end{aligned} \tag{16}$$

It is worth noting that if $\mu = \infty$, Eq. (16) reduces to Eq. (10). This shows the relation between Eq. (5) and Eq. (12) again.

In summary, we present the algorithm for optimizing Eq.(12) in Algorithm 2. The convergence of this algorithm can be proved similarly to [Nie *et al.*, 2010].

4 Related Work

In this section, we discuss some approaches which are closely related to our method.

In order to pursue interpretability, [Cai *et al.*, 2007] proposed *Sparse Subspace Learning* (SSL), which is based on ℓ_1-norm regularization on each column of the projection, i.e., $\mathbf{a}$,

$$\min_{\mathbf{a}} ||\mathbf{X}^T\mathbf{a} - \mathbf{y}||_2^2 + \mu||\mathbf{a}||_1, \tag{17}$$

Algorithm 2 Joint Feature Selection and Subspace Learning (Situation 2)

Initialize: $\mathbf{G}_0 = \mathbf{I}, t = 0$ and μ;
Compute $\mathbf{Y}$ based on $\mathbf{WY} = \mathbf{\Lambda DY}$;
repeat
  Compute $\mathbf{A}_{t+1} = \mathbf{G}_t^{-1}\mathbf{X}(\mathbf{X}^T\mathbf{G}_t^{-1}\mathbf{X} + \frac{1}{2\mu}\mathbf{I})^{-1}\mathbf{Y}$;
  Compute $\mathbf{G}_{t+1}$ based on $\mathbf{A}_{t+1}$;
  $t = t + 1$;
until convergence

where $\mathbf{y}$ is the eigenvector of $\mathbf{Wy} = \lambda\mathbf{Dy}$. Due to the nature of the ℓ_1 penalty, some entries in $\mathbf{a}$ will be shrunk to exact zero if μ is large enough, which results in a sparse projection. However, SSL does not lead to feature selection, because each column of the projection matrix is optimized one by one, and their sparsity patterns are independent.

Most recently, [Masaeli *et al.*, 2010] proposed *Linear Discriminant Feature Selection*, which modifies LDA to admit feature selection as follows,

$$\min_{\mathbf{A}} \text{tr}((\mathbf{A}^T\mathbf{S}_w\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{S}_b\mathbf{A})) + \mu\sum_{i=1}^{d}||\mathbf{a}^i||_\infty, \qquad (18)$$

where $\mathbf{S}_b$ is the between-class scatter matrix, and $\mathbf{S}_w$ is the with-in class scatter matrix [Belhumeur *et al.*, 1997], $\sum_{i=1}^{d}||\mathbf{a}^i||_\infty$ is the ℓ_1/ℓ_∞ norm of $\mathbf{A}$. Note that ℓ_1/ℓ_∞ has the similar nature of $L_{2,1}$-norm, which leads to row-sparsity. The optimization problem is convex and solved by quasi-Newton method [Boyd and Vandenberghe, 2004]. The main disadvantage of this method is that the evaluation of the gradient of $\text{tr}((\mathbf{A}^T\mathbf{S}_w\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{S}_b\mathbf{A}))$ is computationally very expensive. Hence it is limited to small-scale data.

5 Experiments

In this section, we evaluate two instances of our framework, FSSL with LDA-type adjacency matrix defined in Eq. (2), referred to as FSSL(LDA), FSSL with LPP-type adjacency matrix defined in Eq. (3), referred to as FSSL(LPP), and compare them with the state of the art subspace learning methods, e.g. PCA, LDA, LPP [He and Niyogi, 2003], and sparse subspace learning approaches, e.g. SSL with LDA-type adjacency matrix, denoted by SSL(LDA) [Cai *et al.*, 2007], SSL with LPP-type adjacency matrix, denoted by SSL(LPP) [Cai *et al.*, 2007]. We also compare it with a feature selection method, e.g., Fisher score (FS), and feature selection followed with subspace learning (FS+SL). In detail, we use Fisher score to do feature selection and LDA (or LPP) to do subspace learning, which are referred to as FS+SL(LDA) and FS+SL(LPP) respectively. We use *1-Nearest Neighbor* classifier as baseline. All the experiments were performed in Matlab on a Intel Core2 Duo 2.8GHz Windows 7 machine.

5.1 Data Sets

We use two standard face recognition databases which are used in [Cai *et al.*, 2007].

Extended Yale-B database contains 16128 face images of 38 human subjects under 9 pose and 64 illumination conditions. In our experiment, we choose the frontal pose and use all the images under different illumination, thus we get 2414 image in total. All the face images are manually aligned and cropped. They are resized to 32×32 pixels, with 256 gray levels per pixel. Thus each face image is represented as a 1024-dimensional vector.

CMU PIE face database [Sim *et al.*, 2003] contains 68 individuals with 41368 face images as a whole. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination and expression. In our experiment, one near frontal poses ($C27$) are selected under different illuminations, lighting and expressions which leaves us 49 near frontal face images for each individual.

5.2 Parameter Settings

For both data sets, $p = 10, 20, 30$ images were randomly selected as training samples for each person, and the rest images were used for testing. The training set was used to learn a subspace, and the recognition was performed in the subspace by 1-Nearest Neighbor classifier. Since the training set was randomly chosen, we repeated each experiment 20 times and calculated the average recognition accuracy. In general, the recognition rate varies with the dimensionality of the subspace. The best average performance obtained as well as the corresponding dimensionality is reported in Table 1 and Table 2. We also report the computational time (in second) of subspace learning[2].

For LDA, as in [Belhumeur *et al.*, 1997], we first use PCA to reduce the dimensionality to $n - c$ and then perform LDA to reduce the dimensionality to $c - 1$. This is also known as *Fisher Face* [Belhumeur *et al.*, 1997]. For LPP, we use the cosine distance to compute the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. For FS+LDA and FS+LPP, we first use Fisher Score to select 50% features and then perform LDA (or LPP) to reduce the dimensionality. For SSL(LDA), SSL(LPP), we tune μ by searching the grid $\{10, 20, \ldots, 100\}$ according to [Cai *et al.*, 2007]. For FSSL(LDA) and FSSL(LPP), when $p = 10$, the linear system has infinite solution, we run Algorithm 1. And when $p = 20, 30$, we run Algorithm 2, where we simply set $\mu = 0.1$. The smoothing term ϵ is set to 0.01.

5.3 Results

The experimental results are shown in Table 1 and Table 2. We can observe that (1) SSL is better than the corresponding linear subspace learning method (e.g., SSL(LDA) $>$ LDA), which implies sparse subspace learning is able to improve the classification performance of subspace learning method; (2) FSSL outperforms the corresponding SSL method overwhelmingly (e.g., FSSL(LDA) $>$ SSL(LDA)), which indicates feature selection can improve the corresponding linear subspace learning method greatly; (3) Feature selection before subspace learning (FS+SL) sometimes achieves better results than subspace learning and even better than SSL. This implies the potential performance gain of doing feature selection during subspace learning. However, at more cases, FS+SL(LDA) and FS+SL(LPP) are worse than LDA and LPP. This is because feature selection and subspace learning are

[2]It does not include the testing time of Nearest Neighbor classifier.

Table 1: Face recognition accuracy on the Yale-B data set

Data set	10 training			20 training			30 training		
	Acc	Dim	Time	Acc	Dim	Time	Acc	Dim	Time
Baseline	53.44±0.82	–	–	69.24±1.19	–	–	77.39±0.98	–	–
PCA	52.41±0.89	200	0.53	67.04±1.18	200	2.13	74.57±1.07	200	3.74
FS	64.34±1.40	200	2.92	76.53±1.19	200	3.02	82.15±1.14	200	3.59
LDA	78.33±1.31	37	0.38	85.75±0.84	37	2.44	81.19±2.05	37	1.18
LPP	79.70±2.96	76	0.66	80.24±5.49	75	4.7	86.40±1.45	78	11.13
FS+SL(LDA)	77.89±1.82	37	2.79	87.89±0.88	37	3.65	93.91±0.69	37	3.71
FS+SL(LPP)	66.15±5.63	77	2.87	88.43±1.11	74	3.03	93.85±0.69	38	3.08
SSL(LDA)	81.56±1.38	37	17.38	89.68±0.85	37	28.99	92.88±0.68	37	36.53
SSL(LPP)	80.73±1.27	43	75.57	89.69±0.82	37	123.76	92.97±0.66	37	175.67
FSSL(LDA)	86.64±1.04	37	7.96	**96.66±0.75**	37	7.14	**98.77±0.33**	36	12.17
FSSL(LPP)	**87.97±1.02**	74	3.23	95.57±0.66	74	7.99	97.97±0.40	37	14.26

Table 2: Face recognition accuracy on the PIE data set

Data set	10 training			20 training			30 training		
	Acc	Dim	Time	Acc	Dim	Time	Acc	Dim	Time
Baseline	75.84±1.21	–	–	90.35±1.18	–	–	91.68±0.95	–	–
PCA	75.34±1.25	198	1.78	89.99±1.07	200	3.76	95.07±0.69	197	3.58
FS	82.66±1.00	170	3.78	91.74±1.07	200	5.23	94.66±0.63	197	4.84
LDA	90.80±0.87	67	1.70	94.14±0.54	67	1.68	96.52±0.53	67	1.87
LPP	92.35±0.47	171	3.26	94.42±0.51	145	10.61	96.51±0.53	67	11.29
FS+SL(LDA)	87.64±0.96	67	4.77	94.45±0.60	67	5.31	96.08±0.53	66	5.21
FS+SL(LPP)	88.75±0.81	150	4.94	94.49±0.60	67	4.03	96.18±0.57	94	5.14
SSL(LDA)	93.33±0.47	67	58.51	96.83±0.56	67	67.63	97.85±0.38	66	95.15
SSL(LPP)	92.90±0.47	67	288.34	96.63±0.62	67	632.30	97.94±0.39	70	644.26
FSSL(LDA)	94.31±0.42	67	6.46	**98.02±0.33**	64	27.60	**98.44±0.27**	64	52.60
FSSL(LPP)	**96.41±0.39**	67	11.44	97.84±0.38	67	26.86	98.38±0.39	100	42.15

conducted individually in FS+SL. So the selected features are not necessary helpful for subspace learning. In contrast, FSSL perform feature selection and subspace learning simultaneously, the selected features are generally beneficial for subspace learning. Hence the proposed framework improves subspace learning consistently; and (4) The computational time of FSSL is much less than that of SSL.

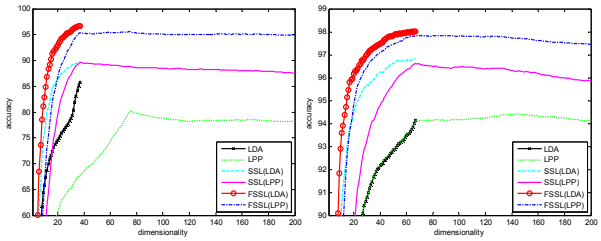
5.4 Study on the Dimensionality of the Subspace

In this subsection, we get a closer look at the recognition accuracy with respect to the dimensionality of the learned subspace. Figure 2 shows the performance of all the subspace learning methods on the two databases with 20 training samples respectively, where the horizontal axis represents the dimensionality of the subspace, and the vertical axis denotes the average recognition accuracy of 20 independent runs.

It is shown that the improvement of our methods over sparse subspace learning methods and subspace space learning methods are consistent over a wide range of the dimensionality of the subspace, which strengthens the superiority of our approaches. Similar phenomenon can be observed when we use 10 and 30 training samples. For the space limit, we do not show them.

5.5 Projection Matrices & Selected Features

To get a better understanding of our approach, we plot the projection matrices of our method and related methods in Figure 3 (a),(b) and (c). Clearly, the projection matrix of LDA is not sparse, while each column of the projection matrix of

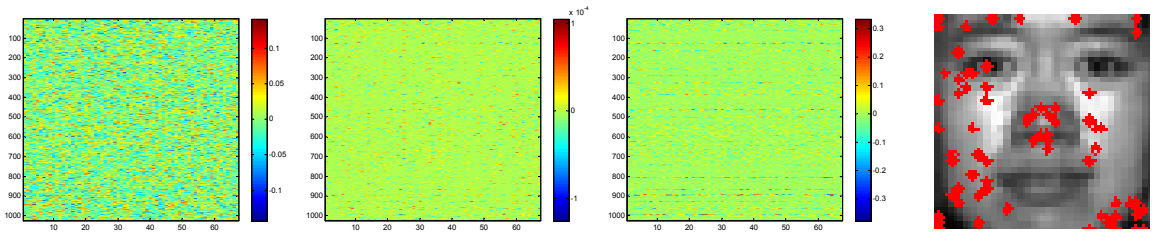


(a) YaleB (b) PIE

Figure 2: Face recognition with 20 selected images for training on (a) YaleB; and (b) PIE database. For better viewing, pleas see it in color pdf file.

SSL(LDA) is sparse. However, the sparsity patterns of each column are not coherent. In contrast, each row of the projection matrix of FSSL(LDA) tends to be zero or nonzero simultaneously, which benefits from the nature of $L_{2,1}$-norm, and leads to feature selection.

From Figure 3 (d), we can see that the selected features (pixels) are asymmetric. In other word, if one pixel is selected, its axis symmetrical one will not be selected. This is because the face image is roughly axis symmetric, so one in a pair of axis symmetrical pixels is redundant given the other one is selected. Moreover, the selected pixels are mostly around the eyebrow, the corner of eyes, nose and cheek, which are discriminative for distinguishing face images of different people. This is consistent with our real-life expe-



(a) LDA (b) SSL(LDA) (c) FSSL(LDA) (d) Selected features

Figure 3: (a)(b)(c)shows the projection matrix learned by LDA, SSL(LDA) and FSSL (LDA) with 10 training samples per person and $\mu = 0.1$ for SSL(LDA), (d) shows the pixels selected by the proposed method, which are marked by red points

rience.

6 Conclusion and Future Work

In this paper, we propose to do feature selection and subspace learning simultaneously in a joint framework, which is based on using $L_{2,1}$-norm on the projection matrix, that achieves the goal of feature selection. Experiments on benchmark face recognition data sets illustrate the efficacy of the proposed framework. In our future work, we will study joint feature selection and nonlinear subspace learning in kernel space.

Acknowledgments

The work was supported in part by NSF IIS-09-05215, U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265, and the U.S. Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA).

References

[Argyriou *et al.*, 2008] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[Belhumeur *et al.*, 1997] Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–720, 1997.

[Boyd and Vandenberghe, 2004] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, 2004.

[Cai *et al.*, 2007] Deng Cai, Xiaofei He, and Jiawei Han. Spectral regression: A unified approach for sparse subspace learning. In *ICDM*, pages 73–82, 2007.

[Chung, 1997] Fan R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, February 1997.

[Golub and Loan, 1996] Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[Guyon and Elisseeff, 2003] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[He and Niyogi, 2003] Xiaofei He and Partha Niyogi. Locality preserving projections. In *NIPS*, 2003.

[He *et al.*, 2005] Xiaofei He, Deng Cai, Shuicheng Yan, and HongJiang Zhang. Neighborhood preserving embedding. In *ICCV*, pages 1208–1213, 2005.

[Masaeli *et al.*, 2010] Mahdokht Masaeli, Glenn Fung, and Jennifer G. Dy. From transformation-based dimensionality reduction to feature selection. In *ICML*, pages 751–758, 2010.

[Moghaddam *et al.*, 2006] Baback Moghaddam, Yair Weiss, and Shai Avidan. Generalized spectral bounds for sparse lda. In *ICML*, pages 641–648, 2006.

[Nie *et al.*, 2010] Feiping Nie, Heng Huang, Xiao Cai, and Chris Ding. Efficient and robust feature selection via joint $l_{2,1}$ norms minimization. In *Advances in Neural Information Processing Systems 23*, pages 1813–1821, 2010.

[Obozinski *et al.*, 2010] Guillaume Obozinski, Ben Taskar, and Michael I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20:231–252, April 2010.

[Sim *et al.*, 2003] Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination, and expression database. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(12):1615–1618, 2003.

[Sun *et al.*, 2009] Liang Sun, Jun Liu, Jianhui Chen, and Jieping Ye. Efficient recovery of jointly sparse vectors. In *Advances in Neural Information Processing Systems 22*, pages 1812–1820, 2009.

[Yan *et al.*, 2007] Shuicheng Yan, Dong Xu, Benyu Zhang, HongJiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):40–51, 2007.

[Yuan *et al.*, 2006] Ming Yuan, Ming Yuan, Yi Lin, and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

[Zou *et al.*, 2004] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15:2006, 2004.