Joint & Progressive Learning from High-Dimensional Data for Multi-Label Classification

Anonymous ECCV submission

Paper ID ***

Abstract. Despite the fact that nonlinear subspace learning techniques (e.g. manifold learning) have successfully applied to data representation, there is still room for improvement in explainability (explicit mapping), generalization (out-of-samples), and cost-effectiveness (linearization). To this end, a novel linearized subspace learning technique is developed in a joint and progressive way, called **j**oint and **p**rogressive **l**earning str**a**teg**y** (J-Play), with its application to multi-label classification. The J-Play learns high-level and semantically meaningful feature representation from high-dimensional data by 1) jointly performing multiple subspace learning and classification to find a latent subspace where samples are expected to be better classified; 2) progressively learning multi-coupled projections to linearly approach the optimal mapping bridging the original space with the most discriminative subspace; 3) locally embedding manifold structure in each learnable latent subspace. Extensive experiments are performed to demonstrate the superiority and effectiveness of the proposed method in comparison with previous state-of-the-art methods.

Keywords: Alternating direction method of multipliers, high-dimensional data, manifold regularization, multi-label classification, joint learning, progressive learning

1 Introduction

High-dimensional data are often characterized by very rich and diverse information, which enables us to classify or recognize the targets more effectively and analyze data attributes more easily, but inevitably introduces some drawbacks (e.g. information redundancy, complex noise effects, high storage-consuming, etc.) due to *the curve of dimensionality*. A general way to address this problem is to learn a low-dimensional and high-discriminative feature representation. In general, it is also called as dimensionality reduction or subspace learning. In the past decades, a large number of subspace learning techniques have been developed in the machine learning community, with successful applications to biometrics [1][2], image/video analysis [3], visualization [4], hyperspectral data

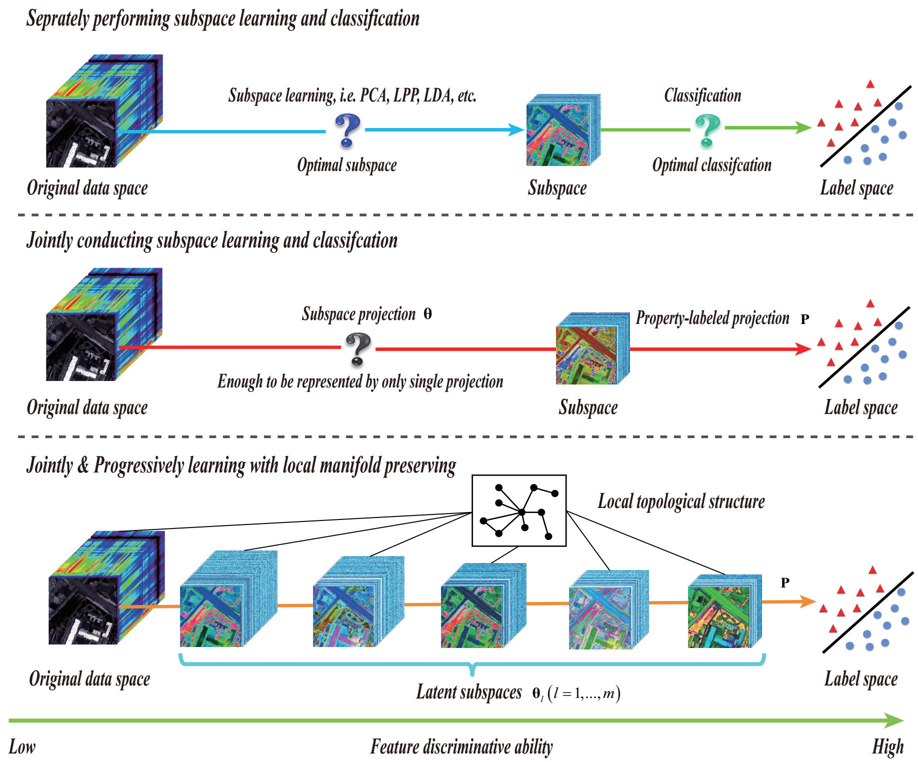


Fig. 1. The motivation interpolation from separately performing subspace learning and classification to joint learning to joint & progressive learning again. The subspaces learned from our model indicates the higher feature discriminative ability as explained by the green bottom line.

dimensionality reduction and classification [5]. These subspace learning techniques are generally categorized into linear or nonlinear methods. Theoretically, nonlinear approaches are capable of curving the data structure in a more effective way. There is, however, no explicit mapping function (poor explainability), and meanwhile it is relatively hard to embed the out-of-samples into the learned subspace (weak generalization) as well as high computational cost (lack of cost-effectiveness). Additionally, for a task of multi-label classification, these classic subspace learning techniques, such as principal component analysis (PCA) [6], local discriminant analysis (LDA) [1], local fisher discriminant analysis (LFDA) [7], manifold learning (e.g. Laplacian eigenmaps (LE) [8], locally linear embedding (LLE) [9]) and their linearized methods (e.g. locality preserving projection (LPP)[10], neighborhood preserving embedding (NPE)[11]), are commonly applied as a disjunct feature learning step before classification, whose limitation mainly lies in a weak connection between features by subspace learning and label space (see the top panel of Fig. 1). It is unknown which learned features (or subspace) can improve the classification.

Recently, a feasible solution to the above problems can be generalized as a joint learning framework [12] that simultaneously considers linearized subspace learning and classification, as illustrated in the middle panel of Fig. 1. Following it, more advanced methods have been proposed and applied in various fields, including supervised dimensionality reduction (e.g. least-squares dimensionality reduction (LSDR) [13] and its variants: least-squares quadratic mutual information derivative (LSQMID) [14]), multi-modal data matching and retrieval [15, 16], and heterogeneous features learning for activity recognition [17, 18]. In these work, the learned features (or subspace) and label information are effectively connected by regression techniques (e.g. linear regression) to adaptively estimate a latent and discriminative subspace. Despite this, they still fail to find an optimal subspace, as single linear projection is hardly enough to represent the complex transformation from the original data space to the potential optimal subspace.

Motivated by the aforementioned studies, we propose a novel **j**oint and **p**rogressive **l**earning str**a**teg**y** (J-Play) to linearly find an optimal subspace for general multi-label classification, illustrated in the bottom panel of Fig. 1. We practically extend the existing joint learning framework by learning a series of subspaces instead of single subspace, aiming at progressively converting the original data space to a potentially optimal subspace through multi-coupled intermediate transformations [19]. Theoretically, by increasing the number of subspaces, coupled subspace variations are gradually narrowed down to a very small range that can be represented effectively via a *linear transformation*. This renders us to find a good solution easier, especially when the model is complex and non-convex. We also contribute to structure learning in each latent subspace by locally embedding manifold structure.

The main highlights of our work can be summarized as follows:

- A linearized progressive learning strategy is proposed to describe the variations from the original data space to potentially optimal subspace, tending to find a better solution. A joint learning framework that simultaneously estimates subspace projections (connect the original space and the latent subspaces) and a property-labeled projection (connect the learned latent subspaces and label space) is considered to find a discriminative subspace where samples are expected to be better classified.
- Structure learning with local manifold regularization is performed in each latent subspace.
- Based on the above techniques, a novel joint and progressive learning strategy (J-Play) is developed for multi-label classification.
- An iterative optimization algorithm based on the alternating direction method of multipliers (ADMM) is designed to solve the proposed model.

2 Joint & Progressive Learning Strategy (J-Play)

2.1 Notations

Let $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_k, ..., \mathbf{x}_N] \in \mathbb{R}^{d_0 \times N}$ be a data matrix with d_0 dimensions and N samples, and the matrix of corresponding class labels be $\mathbf{Y} \in \{0, 1\}^{L \times N}$. The

kth column of $\mathbf{Y}$ is $\mathbf{y}_k = [\mathbf{y}_{k1}, ..., \mathbf{y}_{kt}, ..., \mathbf{y}_{kL}]^T \in \mathbb{R}^{L\times 1}$ whose each element can be defined as follows:

$$\mathbf{y}_{kt} = \begin{cases} 1, & \text{if } \mathbf{y}_k \text{ belongs to the } t\text{-th class;} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

In our task, we aim to learn a set of coupled projections $\{\mathbf{\Theta}_l\}_{l=1}^m \in \mathbb{R}^{d_l \times d_{l-1}}$ and a property-labeled projection $\mathbf{P} \in \mathbb{R}^{L\times d_m}$, where m stands for the number of subspace projections and $\{d_l\}_{l=1}^m$ are defined as the dimensions of those latent subspaces respectively, while d_0 is specified as the dimension of $\mathbf{X}$.

2.2 Basic Framework of J-Play from the View of Subspace Learning

Subspace learning is to find a low-dimensional space where we expect to maximize certain properties of the original data, e.g. variance (PCA), discriminative ability (LDA), and graph structure (manifold learning). Yan et al. [20] summarized these subspace learning methods in a general graph embedding framework.

Given an undirected similarity graph $G = \{\mathbf{X}, \mathbf{W}\}$ with the vertices $\mathbf{X} \in \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ and the adjacency matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$, we can intuitively measure the similarities among the data. By preserving the similarities relationship, the high-dimensional data can be well embedded into the low-dimensional space, which can be formulated by denoting the low-dimensional data representation as $\mathbf{Z} \in \mathbb{R}^{d\times N}$ $(d \ll d_0)$ in the following

$$\min_{\mathbf{Z}} \mathrm{tr}(\mathbf{Z}\mathbf{L}\mathbf{Z}^{\mathrm{T}}), \quad \text{s.t.} \quad \mathbf{Z}\mathbf{D}\mathbf{Z}^{\mathrm{T}} = \mathbf{I}, \tag{2}$$

where $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$ is a diagonal matrix, $\mathbf{L}$ is a Laplacian matrix defined by $\mathbf{L} = \mathbf{D} - \mathbf{W}$ [21], and $\mathbf{I}$ is the identity matrix. In our case, we aim at learning multi-coupled linear projections to find optimal mapping, therefore a linearized subspace learning problem can be reformulated on the basis of Eq. (2) by substituting $\mathbf{\Theta}\mathbf{X}$ for $\mathbf{Z}$

$$\min_{\mathbf{\Theta}} \mathrm{tr}(\mathbf{\Theta}\mathbf{X}\mathbf{L}\mathbf{X}^{\mathrm{T}}\mathbf{\Theta}^{\mathrm{T}}), \quad \text{s.t.} \quad \mathbf{\Theta}\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}\mathbf{\Theta}^{\mathrm{T}} = \mathbf{I}, \tag{3}$$

which can be solved by generalized eigenvalue decomposition.

Different from the previously mentioned subspace learning methods, a regression-based joint learning model [12] can explicitly bridge the learned latent subspace and labels, which can be formulated in a general form:

$$\min_{\mathbf{P},\mathbf{\Theta}} \frac{1}{2}\mathbf{E}(\mathbf{P}, \mathbf{\Theta}) + \frac{\beta}{2}\mathbf{\Phi}(\mathbf{\Theta}) + \frac{\gamma}{2}\mathbf{\Psi}(\mathbf{P}), \tag{4}$$

where $\mathbf{E}(\mathbf{P}, \mathbf{\Theta})$ is the error term defined as $\|\mathbf{Y} - \mathbf{P}\mathbf{\Theta}\mathbf{X}\|_{\mathrm{F}}^2$, $\|\bullet\|_{\mathrm{F}}$ represents a Frobenius norm, β and γ are the corresponding penalty parameters. $\mathbf{\Phi}$ and $\mathbf{\Psi}$ denote regularization functions, which might be l_1 norm, l_2 norm, $l_{2,1}$ norm or manifold regularization. Herein, the variable $\mathbf{\Theta}$ is called intermediate transformation and the corresponding subspace generated by $\mathbf{\Theta}$ is called latent subspace

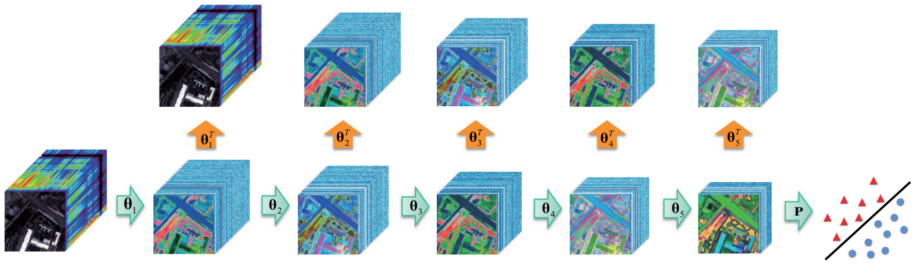


Fig. 2. The illustration of the proposed J-Play framework.

where the feature can be further structurally learned and represented in a more suitable way [18].

On the basis of Eq. (4), we further extend the framework by following a progressive learning strategy:

$$\min_{\mathbf{P},\{\mathbf{\Theta}_l\}_{l=1}^m} \frac{1}{2}\mathbf{E}(\mathbf{P},\{\mathbf{\Theta}_l\}_{l=1}^m) + \frac{\beta}{2}\mathbf{\Phi}(\{\mathbf{\Theta}_l\}_{l=1}^m) + \frac{\gamma}{2}\mathbf{\Psi}(\mathbf{P}), \tag{5}$$

where $\mathbf{E}(\mathbf{P},\{\mathbf{\Theta}_l\}_{l=1}^m)$ is specified as $\|\mathbf{Y}-\mathbf{P}\mathbf{\Theta}_m...\mathbf{\Theta}_l...\mathbf{\Theta}_1\mathbf{X}\|_{\mathrm{F}}^2$ and $\{\mathbf{\Theta}_l\}_{l=1}^m$ represent a set of intermediate transformations.

2.3 Problem Formulation

Following the general framework given in Eq.(5), the proposed J-Play can be formulated as the following constrained optimization problem:

$$\begin{aligned} &\min_{\mathbf{P},\{\mathbf{\Theta}_l\}_{l=1}^m} \frac{1}{2}\mathbf{\Upsilon}(\{\mathbf{\Theta}_l\}_{l=1}^m) + \frac{\alpha}{2}\mathbf{E}(\mathbf{P},\{\mathbf{\Theta}_l\}_{l=1}^m) + \frac{\beta}{2}\mathbf{\Phi}(\{\mathbf{\Theta}_l\}_{l=1}^m) + \frac{\gamma}{2}\mathbf{\Psi}(\mathbf{P}) \\ &\text{s.t.} \quad \mathbf{X}_l = \mathbf{\Theta}_l\mathbf{X}_{l-1}, \quad \mathbf{X}_l \succeq 0, \quad \|\mathbf{x}_{lk}\|_2 \preceq 1, \quad \forall l = 1,2,...,m, \end{aligned} \tag{6}$$

where $\mathbf{X}$ is assigned to $\mathbf{X}_0$, while α, β, and γ are three penalty parameters corresponding to the different terms, which aim at balancing the importance between the terms. Fig. 2 illustrates the J-Play framework. Since Eq. (6) is a typically ill-posed problem, reasonable assumptions or priors need to be introduced to search a solution in a narrowed range effectively. More specifically, we cast Eq.(6) as a least-square regression problem with reconstruction loss term ($\mathbf{\Upsilon}(\bullet)$), prediction loss term ($\mathbf{E}(\bullet)$) and two regularization terms ($\mathbf{\Phi}(\bullet)$ and $\mathbf{\Psi}(\bullet)$). We detail these terms one by one as follows.

1) Reconstruction Loss Term $\mathbf{\Upsilon}(\{\mathbf{\Theta}_l\}_{l=1}^m)$: Without any constraints or prior, directly estimating multi-coupled projections in J-Play is hardly performed with the increase of the number of estimated projections. This can be reasonably explained by gradient missing between the two neighboring variables estimated in the process of optimization. That is, the variations between these neighboring projections are made to be tiny and even zero. In particular, when the number of projections increases to a certain extent, most of learned projections tend to

Algorithm 1: Joint & Progressive Learning Strategy (J-Play)

Input: $\mathbf{Y}, \mathbf{X}, \mathbf{L}$, and parameters α, β, γ and $maxIter$.
Output: $\{\mathbf{\Theta}_l\}_{l=1}^m$.

```
Initialization Step:
Greedily initialize Θ_l corresponding to each latent subspace:
for l = 1 : m do
    Θ_l^0 ← LPP(X_{l-1})
    Θ_l ← AutoRULe(X_{l-1}, Θ_l^0, L)
    X_l ← Θ_l X_{l-1}
end
Fine-tuning Step:
t = 0, ζ = 1e − 4;
while not converged or t > maxIter do
    Fix other variables to update P by solving a subproblem of P;
    for i = 1 : m do
        Fix other variables to update Θ_l^{t+1} by solving a subproblem of Θ_l;
    end
    Compute the objective function value Obj^{t+1} and check the convergence condition: if
     |(Obj^{t+1} − Obj^t) / Obj^t| < ζ then
        Stop iteration;
    else
        t ← t + 1;
    end
end
```

be zero and become meaningless. To this end, we adopt a kind of autoencoder-like scheme to make the learned subspace projected back to the original space as much as possible. The benefits of the scheme are, on one hand, to prevent the data over-fitting to some extent, especially avoiding overmuch noises from being considered; on the other hand, to establish an effective link between the original space and the subspace, making the learned subspace more meaningful. Therefore, the resulting expression is

$$\mathbf{\Upsilon}(\{\mathbf{\Theta}_l\}_{l=1}^m) = \sum\nolimits_{l=1}^{m} \|\mathbf{X}_{l-1} - \mathbf{\Theta}_l^T \mathbf{\Theta}_l \mathbf{X}_{l-1}\|_{\mathrm{F}}^2. \tag{7}$$

In our case, to fully utilize the advantages of this term, we consider it in each latent subspace as shown in Eq.(7).

2) Predication Loss Term $\mathbf{E}(\mathbf{P}, \{\mathbf{\Theta}_l\}_{l=1}^m)$: This term is to minimize the empirical risk between the original data and the corresponding labels through multi-coupled projections in a progressive way, which can be formulated as

$$\mathbf{E}(\mathbf{P}, \{\mathbf{\Theta}_l\}_{l=1}^m) = \|\mathbf{Y} - \mathbf{P}\mathbf{\Theta}_m ... \mathbf{\Theta}_l ... \mathbf{\Theta}_1 \mathbf{X}\|_{\mathrm{F}}^2. \tag{8}$$

3) Local Manifold Regularization $\mathbf{\Phi}(\{\mathbf{\Theta}_l\}_{l=1}^m)$: As introduced in [16], a manifold structure is an important prior for subspace learning. Superior to vector-based feature learning, such as artificial neural network (ANN), a manifold structure can effectively capture the intrinsic structure between samples. To facilitate structure learning in J-Play, we perform the local manifold regularization to each latent subspace. Specifically, this term can be expressed by

$$\mathbf{\Phi}(\{\mathbf{\Theta}_l\}_{l=1}^m) = \sum\nolimits_{l=1}^{m} \mathrm{tr}(\mathbf{\Theta}_l \mathbf{X}_{l-1} \mathbf{L} \mathbf{X}_{l-1}^{\mathrm{T}} \mathbf{\Theta}_l^{\mathrm{T}}). \tag{9}$$

4) Regression Coefficient Regularization $\mathbf{\Psi}(\mathbf{P})$: The regularization term can promote us to derive a more reasonable solution with a reliable generalization to our model, which can be written as

$$\mathbf{\Psi}(\mathbf{P}) = \|\mathbf{P}\|_{\mathrm{F}}^2. \tag{10}$$

Moreover, the non-negativity constraint with respect to each learned dimension-reduced feature (e.g. $\{\mathbf{X}_l\}_{l=1}^{m} \succeq 0$) is considered since we aim to obtain a meaningful low-dimensional feature representation similar to original image data acquired in a non-negative unit. In addition to the non-negativity constraint, we also impose a norm constraint [1] for sample-based of each subspace: $\|\mathbf{x}_{lk}\|_2 \preceq 1, \forall k = 1, ..., N$ and $l = 1, ..., m$.

2.4 Model Optimization

Considering the complexity and the non-convexity of our model, we pretrain our model to have an initial approximation of subspace projections $\{\mathbf{\Theta}_l\}_{l=1}^{m}$ as this can greatly reduce the model's training time and also help finding an optimal solution easier. This is a common tactic that has been successfully employed in deep autoencoders [23]. Inspired by this trick, we propose a pre-training model with respect to $\mathbf{\Theta}_l, \forall l = 1, ..., m$ by simplifying Eq.(6) as

$$\min_{\mathbf{\Theta}_l} \frac{1}{2}\mathbf{\Upsilon}(\mathbf{\Theta}_l) + \frac{\eta}{2}\mathbf{\Phi}(\mathbf{\Theta}_l) \quad \text{s.t.} \quad \mathbf{X}_l \succeq 0, \quad \|\mathbf{x}_{lk}\|_2 \preceq 1, \tag{11}$$

which is named as **auto-r**econstructing **u**nsupervised **le**arning (AutoRULe). Given the outputs of AutoRULe, the problem of Eq. (6) can be more effectively solved by an alternatively minimizing strategy that separately solves two subproblems with respect to $\{\mathbf{\Theta}_l\}_{l=1}^{m}$ and $\mathbf{P}$. Therefore, the global algorithm of J-Play can be summarized in **Algorithm 1**,where AutoRULe is initialized by LPP.

The pre-training method (AutoRULe) can be effectively solved via the ADMM-based framework. Following this, we consider an equivalent form of Eq. (11) by introducing multiple auxiliary variables $\mathbf{H}$, $\mathbf{G}$, $\mathbf{Q}$ and $\mathbf{S}$ to replace $\mathbf{X}_l$, $\mathbf{\Theta}_l$, $\mathbf{X}_l^{+}$ and $\mathbf{X}_l^{\sim}$, respectively, where $()^{+}$ denotes an operator that converts each component of the matrix to its absolute value and $()^{\sim}$ is a proximal operator for solving the constraint of $\|\mathbf{x}_{lk}\|_2 \preceq 1$ [24], written as follows

$$\begin{aligned}
\min_{\mathbf{\Theta}_l,\mathbf{H},\mathbf{G},\mathbf{Q},\mathbf{S}} & \frac{1}{2}\mathbf{\Upsilon}(\mathbf{G},\mathbf{H}) + \frac{\eta}{2}\mathbf{\Phi}(\mathbf{\Theta}_l) = \frac{1}{2}\|\mathbf{X}_{l-1} - \mathbf{G}^{\mathrm{T}}\mathbf{H}\|_{\mathrm{F}}^2 + \frac{\eta}{2}\,\mathrm{tr}(\mathbf{X}_l\mathbf{L}\mathbf{X}_l^{\mathrm{T}}) \\
\text{s.t.} \quad & \mathbf{Q} \succeq 0, \quad \|\mathbf{s}_k\|_2 \preceq 1, \quad \mathbf{X}_l = \mathbf{\Theta}_l\mathbf{X}_{l-1}, \\
& \mathbf{X}_l = \mathbf{H}, \quad \mathbf{\Theta}_l = \mathbf{G}, \quad \mathbf{X}_l = \mathbf{Q}, \quad \mathbf{X}_l = \mathbf{S}.
\end{aligned} \tag{12}$$

[1] Regarding this constraint,please refer to [22] for more details.

The augmented Lagrangian version of Eq. (12) is

$$\begin{aligned}
&\mathscr{L}_{\mu}\left(\mathbf{\Theta}_l, \mathbf{H}, \mathbf{G}, \mathbf{Q}, \mathbf{S}, \{\mathbf{\Lambda}_n\}_{n=1}^{4}\right) \\
&\quad= \frac{1}{2}\|\mathbf{X}_{l-1} - \mathbf{G}^{\mathrm{T}}\mathbf{H}\|_{\mathrm{F}}^2 + \frac{\eta}{2}\operatorname{tr}(\mathbf{\Theta}_l\mathbf{X}_{l-1}\mathbf{L}\mathbf{X}_{l-1}^{\mathrm{T}}\mathbf{\Theta}_l^{\mathrm{T}}) + \mathbf{\Lambda}_1^{\mathrm{T}}(\mathbf{H} - \mathbf{\Theta}_l\mathbf{X}_{l-1}) \\
&\quad+ \mathbf{\Lambda}_2^{\mathrm{T}}(\mathbf{G} - \mathbf{\Theta}_l) + \mathbf{\Lambda}_3^{\mathrm{T}}(\mathbf{Q} - \mathbf{\Theta}_l\mathbf{X}_{l-1}) + \mathbf{\Lambda}_4^{\mathrm{T}}(\mathbf{S} - \mathbf{\Theta}_l\mathbf{X}_{l-1}) + \frac{\mu}{2}\|\mathbf{H} - \mathbf{\Theta}_l\mathbf{X}_{l-1}\|_{\mathrm{F}}^2 \\
&\quad+ \frac{\mu}{2}\|\mathbf{G} - \mathbf{\Theta}_l\|_{\mathrm{F}}^2 + \frac{\mu}{2}\|\mathbf{Q} - \mathbf{\Theta}_l\mathbf{X}_{l-1}\|_{\mathrm{F}}^2 + \frac{\mu}{2}\|\mathbf{S} - \mathbf{\Theta}_l\mathbf{X}_{l-1}\|_{\mathrm{F}}^2 + l_R^{+}(\mathbf{Q}) + l_R^{\sim}(\mathbf{S}),
\end{aligned} \tag{13}$$

where $\{\mathbf{\Lambda}_n\}_{n=1}^{4}$ are Lagrange multipliers and μ is the penalty parameter. The two terms $l_R^{+}(\bullet)$ and $l_R^{\sim}(\bullet)$ represent two kinds of projection operators, respectively. That is, $l_R^{+}(\bullet)$ is defined as

$$max(\bullet) = \begin{cases} \bullet\ , & \bullet \succ 0 \\ 0\ , & \bullet \preceq 0, \end{cases} \tag{14}$$

while $l_R^{\sim}(\bullet_k)$ is a vector-based operator defined by

$$prox_f(\bullet_k) = \begin{cases} \frac{\bullet_k}{\|\bullet_k\|_2}\ , & \|\bullet_k\|_2 \succ 1 \\ \bullet_k\ , & \|\bullet_k\|_2 \preceq 1, \end{cases} \tag{15}$$

where $\bullet_k$ is the kth column of matrix $\bullet$. **Algorithm 2** details the procedures of AutoRULe.

The two subproblems in **Algorithm 1** can be optimized alternatively as follows:

Optimization with respect to $\mathbf{P}$: This is a typical least square regression problem, which can be written as

$$\min_{\mathbf{P}} \frac{\alpha}{2}\mathbf{E}(\mathbf{P}) + \frac{\gamma}{2}\mathbf{\Psi}(\mathbf{P}) = \frac{\alpha}{2}\|\mathbf{Y} - \mathbf{P}\mathbf{\Theta}_m ... \mathbf{\Theta}_l ... \mathbf{\Theta}_1\mathbf{X}\|_{\mathrm{F}}^2 + \frac{\gamma}{2}\|\mathbf{P}\|_{\mathrm{F}}^2, \tag{16}$$

which has a closed-form solution

$$\mathbf{P} \leftarrow (\alpha\mathbf{Y}\mathbf{V}^{\mathrm{T}})(\alpha\mathbf{V}\mathbf{V}^{\mathrm{T}} + \gamma\mathbf{I})^{-1}, \tag{17}$$

where $\mathbf{V} = \mathbf{\Theta}_m ... \mathbf{\Theta}_l ... \mathbf{\Theta}_1, \forall l = 1, ..., m$.

Optimization with respect to $\{\mathbf{\Theta}_l\}_{l=1}^{m}$: The variables $\{\mathbf{\Theta}_l\}_{l=1}^{m}$ can be individually optimized, and hence the optimization problem of each $\mathbf{\Theta}_l$ can be generally formulated by

$$\begin{aligned}
\min_{\mathbf{\Theta}_l} \frac{1}{2}\mathbf{\Upsilon}(\mathbf{\Theta}_l) + \frac{\alpha}{2}\mathbf{E}(\mathbf{\Theta}_l) + \frac{\beta}{2}\mathbf{\Phi}(\mathbf{\Theta}_l) &= \frac{1}{2}\|\mathbf{X}_{l-1} - \mathbf{\Theta}_l^{\mathrm{T}}\mathbf{\Theta}_l\mathbf{X}_{l-1}\|_{\mathrm{F}}^2 \\
&+ \frac{\alpha}{2}\|\mathbf{Y} - \mathbf{P}\mathbf{\Theta}_m ... \mathbf{\Theta}_l ... \mathbf{\Theta}_1\mathbf{X}\|_{\mathrm{F}}^2 + \frac{\beta}{2}\operatorname{tr}(\mathbf{\Theta}_l\mathbf{X}_{l-1}\mathbf{L}\mathbf{X}_{l-1}^{\mathrm{T}}\mathbf{\Theta}_l^{\mathrm{T}}) \\
\text{s.t.}\quad \mathbf{X}_l = \mathbf{\Theta}_l\mathbf{X}_{l-1}, \quad \mathbf{X}_l \succeq 0, &\quad \|\mathbf{x}_{lk}\|_2 \preceq 1,
\end{aligned} \tag{18}$$

Algorithm 2: Auto-reconstructing unsupervised learning (AutoRULe)

Input: $\mathbf{X}_{l-1}, \mathbf{\Theta}_l^0, \mathbf{L}$, and parameters η and $maxIter$.

Output: $\mathbf{\Theta}_l$.

Initialization: $\mathbf{H}^0 = \mathbf{\Theta}_l^0 \mathbf{X}_{l-1}, \mathbf{G}^0 = \mathbf{0}, \mathbf{Q}^0 = \mathbf{P}^0 = \mathbf{0}, \mathbf{\Lambda}_2^0 = \mathbf{0}, \mathbf{\Lambda}_1^0 = \mathbf{\Lambda}_3^0 = \mathbf{\Lambda}_4^0 = \mathbf{0}, \mu^0 = 1e-3, \mu_{max} = 1e6, \rho = 2, \varepsilon = 1e-6, t = 0.$

while *not converged* or $t > maxIter$ **do**

Fix $\mathbf{H}^t, \mathbf{G}^t, \mathbf{Q}^t, \mathbf{P}^t$ to update $\mathbf{\Theta}_l^{t+1}$ by

$$\begin{aligned}\mathbf{\Theta}_l =&(\mu \mathbf{H}\mathbf{X}_{l-1}^T + \mathbf{\Lambda}_1 \mathbf{X}_{l-1}^T + \mu \mathbf{G} + \mathbf{\Lambda}_2 + \mu \mathbf{Q}\mathbf{X}_{l-1}^T + \mathbf{\Lambda}_3 \mathbf{X}_{l-1}^T \\ &+ \mu \mathbf{P}\mathbf{X}_{l-1}^T + \mathbf{\Lambda}_4 \mathbf{X}_{l-1}^T)(\eta(\mathbf{X}_{l-1}\mathbf{L}\mathbf{X}_{l-1}^T) + 3\mu(\mathbf{X}_{l-1}\mathbf{X}_{l-1}^T) + \mu \mathbf{I})^{-1}.\end{aligned}$$

Fix $\mathbf{\Theta}_l^{t+1}, \mathbf{G}^t, \mathbf{Q}^t, \mathbf{P}^t$ to update $\mathbf{H}^{t+1}$ by

$$\mathbf{H} = (\mathbf{G}\mathbf{G}^T + \mu \mathbf{I})^{-1}(\mathbf{G}\mathbf{X}_{l-1} + \mu \mathbf{\Theta}_l \mathbf{X}_{l-1} - \mathbf{\Lambda}_1).$$

Fix $\mathbf{H}^{t+1}, \mathbf{\Theta}_l^{t+1}, \mathbf{Q}^t, \mathbf{P}^t$ to update $\mathbf{G}^{t+1}$ by

$$\mathbf{G} = (\mathbf{H}\mathbf{H}^T + \mu \mathbf{I})^{-1}(\mathbf{H}\mathbf{X}_i + \mu \mathbf{\Theta}_l - \mathbf{\Lambda}_2).$$

Fix $\mathbf{H}^{t+1}, \mathbf{G}^{t+1}, \mathbf{\Theta}_l^{t+1}, \mathbf{P}^t$ to update $\mathbf{Q}^{t+1}$ by

$$\mathbf{Q} = max(\mathbf{\Theta}_l \mathbf{X}_{l-1} - \mathbf{\Lambda}_3/\mu, 0).$$

Fix $\mathbf{H}^{t+1}, \mathbf{G}^{t+1}, \mathbf{\Theta}_l^{t+1}, \mathbf{Q}^{t+1}$ to update $\mathbf{P}^{t+1}$ by

$$\mathbf{P} = prox_f(\mathbf{\Theta}_l \mathbf{X}_{l-1} - \mathbf{\Lambda}_4/\mu).$$

Update Lagrange multipliers by

$$\mathbf{\Lambda}_1^{t+1} = \mathbf{\Lambda}_1^t + \mu^t(\mathbf{H}^{t+1} - \mathbf{\Theta}_i^{t+1}\mathbf{X}_{l-1}), \mathbf{\Lambda}_2^{t+1} = \mathbf{\Lambda}_2^t + \mu^t(\mathbf{G}^{t+1} - \mathbf{\Theta}_i^{t+1}),$$

$$\mathbf{\Lambda}_3^{t+1} = \mathbf{\Lambda}_3^t + \mu^t(\mathbf{Q}^{t+1} - \mathbf{\Theta}_i^{t+1}\mathbf{X}_{l-1}), \mathbf{\Lambda}_4^{t+1} = \mathbf{\Lambda}_4^t + \mu^t(\mathbf{P}^{t+1} - \mathbf{\Theta}_i^{t+1}\mathbf{X}_{l-1}).$$

Update penalty parameter by

$$\mu^{t+1} = min(\rho \mu^t, \mu_{max}).$$

Check the convergence conditions: **if** $\|\mathbf{H}^{t+1} - \mathbf{\Theta}_l^{t+1}\mathbf{X}_{l-1}\|_F < \varepsilon$ *and* $\|\mathbf{G}^{t+1} - \mathbf{\Theta}_l^{t+1}\|_F < \varepsilon$ *and* $\|\mathbf{Q}^{t+1} - \mathbf{\Theta}_l^{t+1}\mathbf{X}_{l-1}\|_F < \varepsilon$ *and* $\|\mathbf{P}^{t+1} - \mathbf{\Theta}_l^{t+1}\mathbf{X}_{l-1}\|_F < \varepsilon$ **then**

Stop iteration;

else

$t \leftarrow t + 1$;

end

end

which can be basically deduced by following the framework of **Algorithm 2**. The only difference lies in the optimization subproblem with respect to $\mathbf{H}$ whose solution can be collected by solving the following problem:

$$\begin{aligned}\min_{\mathbf{H}} \frac{1}{2}\|\mathbf{X}_{l-1} - \mathbf{G}^{\mathrm{T}}\mathbf{H}\|_{\mathrm{F}}^2 + \frac{\alpha}{2}\|\mathbf{Y} - \mathbf{P}_l\mathbf{H}\|_{\mathrm{F}}^2 + \mathbf{\Lambda}_1^{\mathrm{T}}(\mathbf{H} - \mathbf{\Theta}_l\mathbf{X}_{l-1}) \\ + \frac{\mu}{2}\|\mathbf{H} - \mathbf{\Theta}_l\mathbf{X}_{l-1}\|_{\mathrm{F}}^2 \quad \text{s.t.} \quad \mathbf{P}_l = \mathbf{P}_{l-1}\mathbf{\Theta}_{l+1}, \quad \mathbf{P}_0 = \mathbf{P}.\end{aligned} \tag{19}$$

The analytical solution of Eq. (19) is given by

$$\mathbf{H} \leftarrow (\alpha \mathbf{P}_l^{\mathrm{T}}\mathbf{P}_l + \mathbf{G}\mathbf{G}^{\mathrm{T}} + \mu \mathbf{I})^{-1}(\alpha \mathbf{P}_l^{\mathrm{T}}\mathbf{Y} + \mathbf{G}\mathbf{X}_{l-1} + \mu \mathbf{\Theta}_l\mathbf{X}_{l-1} - \mathbf{\Lambda}_1). \tag{20}$$

Finally, we repeat these optimization procedures until a stopping criterion is satisfied. Please refer to **Algorithm 1** and **Algorithm 2** for more explicit steps.

3 Experiments

In this section, we conduct the classification to quantitatively evaluate the performance of the proposed method (J-Play) using three popular and advanced classifiers, namely the nearest neighbor (NN) based on the Euclidean distance, kernel support vector machines (KSVM) and canonical correlation forest (CCF), in comparison with previous state-of-the-art methods. Overall accuracy (OA) is given to quantify the classification performance.

3.1 Data Description

The experiments are performed on two different types of datasets: hyperspectral datasets and face datasets, as both of them easily suffer from the information redundancy and need to improve the representative ability of features. We have used the following two hyperspectral datasets and two face datasets:

1) *Indian Pines AVIRIS Image:* The first hyperspectral cube was acquired by the AVIRIS sensor with the size of $145 \times 145 \times 220$, which consists of 16 class of vegetation. More specific classes and the arrangement of training and test samples can be found in [25]. The first image of Fig. 3 shows a false color image of Indian Pines data.

2) *University of Houston Image:* The second hyperspectral cube was provided for the 2013 IEEE GRSS data fusion contest acquired by ITRES-CASI sensor with size of $349 \times 1905 \times 144$. The information regarding classes and corresponding train and test samples can be found in [5]. A false color image of the study scene is shown in the first image of Fig. 4.

3) *Extended Yale-B Dataset:* We only choose a subset of the mentioned dataset with the frontal pose and the different illuminations of 38 subjects (2414 images in total), which can widely used in evaluating the performance of subspace learning [26][27]. These images were aligned and cropped to the size of 32×32, that is, 1024-dimensional vector-based representation. Each individual has 64 near frontal images under different illuminations.

4) *AR Dataset:* Similar to [28], we choose a subset of AR under the conditions of illumination and expressions, which comprises of 100 subjects. Each person has 14 images with seven ones from Session 1 as training set and others from Session 2 as testing samples. The images are resized to 60×43.

3.2 Experimental Steup

As the fixed training and testing samples are given for the hyperspectral datasets, subspace learning techniques can directly be performed on training set to learn an optimal subspace where the testing set can be simply classified by NN, KSVM, and CCF. For the face datasets, since there is no standard training and testing sets, ten replications are performed for randomly selecting training and testing samples. A random subset with 10 facial images per individual is chosen with labels as the training set and the rest of it is considered to be the testing set. Furthermore, we compare the performance of the proposed method (J-Play) with the baseline (original features without dimensionality reduction) and six popular and advanced methods (PCA, LPP, LDA, LFDA, LSDR, and LSQMID). With learning the different number of coupled projections, the proposed method can be successively specified as J-Play$_1$,...,J-Play$_l$,...,J-Play$_m$, $\forall l = 1, ..., m$. To investigate the trend of OAs, m are uniformly set up to 7 on the four datasets.

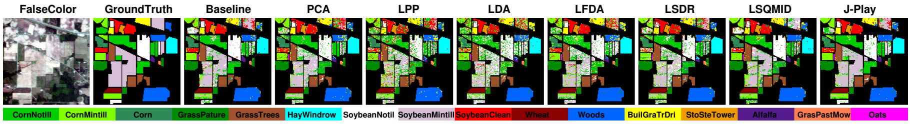


Fig. 3. A false color image, ground truth and classification maps of the different algorithms obtained using CCF on the Indian Pines dataset.

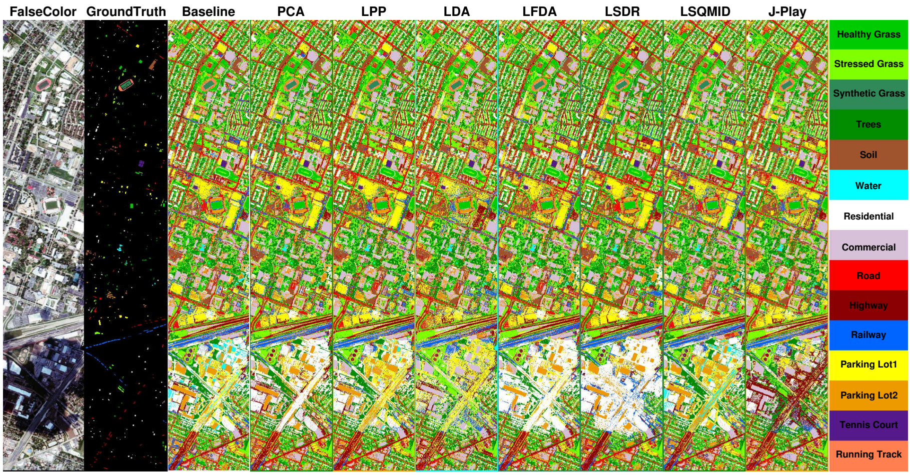


Fig. 4. A false color image, ground truth and classification maps of the different algorithms obtained using CCF on the Houston dataset.

3.3 Results of Hyperspectral Data

Initially, we conduct a 10-fold cross-validation for the different algorithms on the training set in order to estimate the optimal parameters which can be selected from $\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$. Table 1 lists classification performances of the different methods with the optimal subspace dimensions obtained by cross-validation using three different classifiers. Correspondingly, the classification maps are given in Figs. 3 and 4 to intuitively highlight the difference.

Overall, PCA performs basically similar performance with the baseline using the three different classifiers on the two datasets. For LPP, due to its sensitivity to noise, it yields a poor performance on the first dataset, while on the relatively high-quality second dataset, LPP steadily outperforms the baseline and PCA. In the supervised algorithms, owing to the limitation of training samples and discriminative power, the classification accuracies of classic LDA is holistically lower than those previously mentioned. With a more powerful discriminative criterion, LFDA obtains more competitive results by locally focusing on discriminative information, which are generally better than those of the baseline, PCA, LPP, and LDA. However, the features learned by LFDA is sensitive to noise and the number of neighbors, resulting in the unstable performance particularly for the different classifiers. For LSDR and LSQMID, they aim to find a linear projection by maximizing the mutual information between input and out-

Table 1. Quantitative performance comparisons on two hyperspectral datasets. The best results for the different classifiers are shown in red.

Methods	Indian Pines dataset			Houston dataset		
	NN	KSVM	CCF	NN	KSVM	CCF
Baseline (220/144)	65.89%	66.56%	81.71%	72.78%	80.19%	83.06%
PCA (20/20)	65.99%	77.73%	77.95%	72.75%	79.54%	83.17%
LPP (20/30)	64.86%	63.02%	68.75%	75.31%	80.86%	83.23%
LDA (15/14)	64.66%	62.71%	66.34%	75.81%	76.66%	79.51%
LFDA (15/14)	72.47%	72.48%	75.34%	72.95%	82.02%	82.64%
LSDR (50/40)	73.67%	76.84%	77.38%	76.80%	80.39%	81.64%
LSQMID (60/80)	66.94%	78.90%	79.32%	76.31%	80.23%	81.69%
J-Play$_1$ (20/30)	78.81%	82.04%	82.42%	78.22%	83.32%	85.32%
J-Play$_2$ (20/30)	80.87%	83.75%	83.83%	79.16%	84.41%	86.75%
J-Play$_3$ (20/30)	83.59%	85.08%	84.52%	80.13%	83.68%	88.11%
J-Play$_4$ (20/30)	83.92%	85.21%	84.92%	79.64%	83.25%	85.64%
J-Play$_5$ (20/30)	83.76%	85.30%	84.71%	80.00%	82.21%	85.19%
J-Play$_6$ (20/30)	83.56%	84.79%	83.68%	79.69%	82.45%	84.60%
J-Play$_7$ (20/30)	82.70%	83.82%	83.04%	77.81%	81.03%	84.05%

put from the view of statistics. With fully considering the mutual information, they achieve the good performance on the two given hyperspectral datasets.

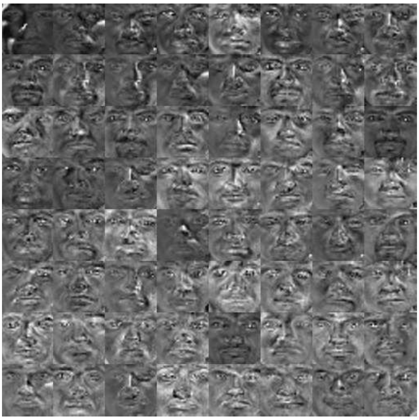
Remarkably, the performance of the proposed method (J-Play) is superior to the other methods on the two hyperspectral datasets. This indicates that J-Play is prone to learn a better feature representation and robust against noise. On the other hand, with the increase of m, the performance of J-Play steadily increases to the best with around 4 or 5 layers for the first dataset and 2 or 3 layers for the second one, and then gradually decreases with a slight perturbation since our model is only trained on the training set.

3.4 Results of Face Images

As J-Play is proposed as a general subspace learning framework for multi-label classiciation, we additionally used two popular face datasets to further assess its generalization capability. Similarly, cross-validation on training set is conducted for estimating the optimal parameter combination on the extended Yale-B and AR datasets. Considering the high-dimensional vector-based face images, we first perform the PCA for face images in order to roughly reduce the feature redundancy, whose results are further explored to the dimensionality reduction methods by following the previous work on face recognition (e.g. LDA (Fisher-faces) [1] and LPP (Laplacianfaces) [2]). Table 2 gives the corresponding OAs using the different methods on the two face datasets respectively.

By comparison, the performance of PCA and LPP is steadily superior to that of baseline, while PCA is even better than LPP. For supervised approaches, LDA performs better than baseline, PCA, LPP and even LFDA, showing an

(a) Extended Yale-B dataset



(b) AR dataset

Fig. 5. Visualization of partial facial features learned by the proposed J-Play on two face datasets.

Table 2. Quantitative performance comparisons on two face datasets. The best results for the different classifiers are shown in red.

Methods	Extended Yale-B dataset			AR dataset		
	NN	KSVM	CCF	NN	KSVM	CCF
Baseline (1024/2580)	45.77%	45.87%	76.99%	71.71%	72.29%	80.29%
PCA (120/80)	41.05%	81.47%	83.53%	68.43%	80.29%	81.43%
LPP (170/70)	70.75%	76.55%	77.48%	70.86%	74.00%	79.86%
LDA (37/99)	80.88%	78.37%	83.68%	81.43%	82.29%	85.38%
LFDA (37/99)	81.02%	80.88%	83.58%	71.29%	75.71%	80.38%
LSDR (60/80)	71.29%	76.40%	78.66%	75.14%	79.00%	80.14%
LSQMID (60/80)	71.48%	77.09%	78.37%	73.29%	74.29%	79.29%
J-Play$_1$ (170/210)	73.01%	79.30%	80.29%	73.57%	79.86%	77.86%
J-Play$_2$ (170/210)	81.17%	84.27%	85.22%	82.29%	86.00%	84.57%
J-Play$_3$ (170/210)	83.43%	85.50%	85.76%	85.43%	88.71%	87.43%
J-Play$_4$ (170/210)	84.07%	86.09%	86.55%	85.29%	87.71%	87.71%
J-Play$_5$ (170/210)	84.56%	86.14%	86.20%	85.71%	87.29%	88.86%
J-Play$_6$ (170/210)	85.35%	85.64%	86.53%	85.14%	87.29%	88.29%
J-Play$_7$ (170/210)	85.74%	85.45%	86.20%	86.57%	86.86%	88.71%

impressive result. Due to the less number of training samples from face datasets, LSDR and LSQMID are limited to effectively estimate the mutual information between the training samples and labels, resulting in the performance degradation compared to the hyperspectral data. The proposed method outperforms other algorithms, which indicates that this method can effectively learn an optimal mapping from original space to label space, further improving the classification accuracy. Likewise, there is a similar trend for the proposed method with the increase of m that J-Play can basically obtain the optimal OAs with around 4 or 5 layers and more layers would lead to the performance degradation. We also characterize and visualize each column of the learned projection, as shown in Fig. 5 where those high-level or semantically meaningful features, i.e. face features under the different pose and illumination, can be learned well, making the faces identified easier.

4 Conclusions

To effectively find an optimal subspace where the samples can be semantically represented and thereby be better classified or recognized, we proposed a novel linearized subspace learning framework (J-Play) which aims at learning the feature representation from the high-dimensional data in a joint and progressive way. Extensive experiments of multi-label classification are conducted on two types of datasets: hyperspectral images and face images, in comparison with some previously proposed state-of-the-art methods. The promising results using J-Play demonstrate its superiority and effectiveness. In the future, we will further build an unified framework based on J-Play by extending it to semi-supervised learning, transfer learning, or multi-task learning.

References

1. Martnez, A.M., Avinash, C.K.: Pca versus lda. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **23**(2) (2001) 228–233
2. He, X., Hu, S., Niyogi, P., Zhang, H.J.: Face recognition using laplacianfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **27**(3) (2005) 328–340
3. Tosato, D., Farenzena, M., Spera, M., Murino, V., Cristani, M.: Multi-class classification on riemannian manifolds for video surveillance. In: Europe Conference on Computer Vision (ECCV). (2010) 378–391
4. Saul, S.L., Roweis, S.T.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. Journal of Machine Learning Research (JMLR) **4** (2003) 119–155
5. Hong, D., Yokoya, N., Zhu, X.: Learning a robust local manifold representation for hyperspectral dimensionality reduction. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS) **10**(6) (2017) 2960–2975
6. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemometrics and Intelligent Laboratory Systems **2**(1) (1987) 37–52
7. Sugiyama, M.: Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. Journal of Machine Learning Research (JMLR) **8** (2007) 1027–1061
8. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation **15**(6) (2003) 1373–1396
9. Roweis, S.T., Lawrence, K.S.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500) (2000) 2323–2326
10. He, X., Niyogi, P.: Locality preserving projections. In: Advances in Neural Information Processing Systems (NIPS). (2004) 153–160
11. He, X., Cai, D., Yan, S., Zhang, H.J.: Neighborhood preserving embedding. In: International Conference on Computer Vision (ICCV). Volume 2. (2005) 1208–1213
12. Ji, S., Ye, J.: Linear dimensionality reduction for multi-label classification. In: International Joint Conference on Artifical Intelligence (IJCAI). Volume 9. (2009) 1077–1082
13. Suzuki, T., Sugiyama, M.: Sufficient dimension reduction via squared-loss mutual information estimation. Neural Computation **25**(3) (2013) 725–758
14. Tangkaratt, V., Sasaki, H., Sugiyama, M.: Direct estimation of the derivative of quadratic mutual information with application in supervised dimension reduction. Neural Computation **29**(8) (2017) 2076–2122
15. Wang, K., He, R., Wang, W., Wang, L., Tan, T.: Learning coupled feature spaces for cross-modal matching. In: International Conference on Computer Vision (ICCV). (2013) 2088–2095
16. Wang, K., He, R., Wang, L., Wang, W., Tan, T.: Joint feature selection and subspace learning for cross-modal retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **38**(10) (2016) 2010–2023
17. Hu, J., Zheng, W., Lai, J., Zhang, J.: Jointly learning heterogeneous features for rgb-d activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 5344–5352
18. Hu, J., Zheng, W., Lai, J., Zhang, J.: Jointly learning heterogeneous features for rgb-d activity recognition. (2016)

19. Kan, M., Shan, S., Chang, H., Chen, X.: Stacked progressive auto-encoders (spae) for face recognition across poses. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2014) 1883–1890
20. Yan, S., Xu, D., Zhang, B., Zhang, H.J., Yang, Q., Lin, S.: Graph embedding and extensions: A general framework for dimensionality reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **29**(1) (2007) 40–51
21. Chung, F.R.K.: Spectral graph theory. American Mathematical Society (1997)
22. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: Advances in Neural Information Processing Systems (NIPS). (2007) 801–808
23. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786) (2006) 504–507
24. Heide, F., Heidrich, W., Wetzstein, G.: Fast and flexible convolutional sparse coding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 5135–5143
25. Ghamisi, P., Benediktsson, J.A., Ulfarsson, M.O.: Spectralspatial classification of hyperspectral images based on hidden markov random fields. IEEE Transactions on Geoscience and Remote Sensing **52**(5) (2014) 2565–2574
26. Zhang, L., Yang, M., Feng, X.: Sparse representation or collaborative representation: which helps face recognition?. In: International Conference on Computer Vision (ICCV). (2011) 471–478
27. Cai, D., He, X., Han, J.: Spectral regression: A unified approach for sparse subspace learning. In: International Conference on Data Mining (ICDM). (2007) 73–82
28. Yang, M., Zhang, L., Yang, J., Zhang, D.: Robust sparse coding for face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2011) 625–632