A New Sparse Dimension Reduction Method for Hyperspectral Images

Jing-zun ZHANG1,2,*, Li-ping LIAO1,3, Guang-mei XU1, Rui-zhe ZHANG1 and Ning HE1

1Smart City College, Beijing Union University, Beijing, CO 100101 China
2Key Laboratory of Digital Earth Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing, CO 10094 China
3Beijing Jiaotong University, Beijing, CO 10044 China

*Corresponding author

Keywords: Hyperspectral image, Dimension reduction, Double sparse& unsupervised.

Abstract. Motivated by the development of sparse representation, many sparsity-based methods have been successfully applied in Hyperspectral images (HSI) domain. These methods are beneficial to find the intrinsic structure from the high dimension data. Sparse dimension reduction is becoming one of the research hotspots, more efforts are needed to be further. In this paper, we present a new unsupervised dimension reduction (DR) method, which is called unsupervised double sparse learning method (UDSDL). UDSDL inherits the merits of sparse representation, it can support the possibility of data compressing while preserving more discriminative features. Experiments on a real HSI data set collected by the Airborne visible/Infrared Imaging Spectrometer are performed to demonstrate the effectiveness of the proposed UDSDL method.

Introduction

With its numerous and contiguous spectral bands, Hyperspectral Images (HSI) can provide detailed spectral information of the observed materials. It is nowadays considered as one of the most powerful remote sensing tools for acquiring precise information of the Earth’s surface. However, the increased volume of data contained in the HSI entails inconvenience and inefficiency in information storage, display, transmission, and processing for researchers [1][2]. Chang found that up to 90% of the spectral band is not necessary and can be neglected, without affecting the accuracy of the overall information interpretation [3]. It is desirable to develop a DR technique that can automatically find a discriminative and physically meaningful representative of the whole image cube without loss of information.

Sparsity-based methods [4] are the new data mining algorithms that can overcome the classical DR ones’ insufficiency under very high scenario but with very few applications in HSI domain. Therefore, we make a further study of sparsity-based models, present a new unsupervised HSI DR method by introducing $l_1$ norm regularization on loadings and elastic net constraint on the learned features. This method can provides additional data compression mechanism while preserving the discriminative information. The reminder of this paper is organized as follows: Section 2 presents the notations and briefly reviews the sparsity-based DR methods from a matrix factorization point of view and introduces two representatives of them. By analyzing the pros and cons of the above two, we give details of our model in Section 3. To demonstrate the effectiveness of the proposed model, extensive experimental results on real HSI dataset are reported and analyzed in Section 4. Section 5 concludes this paper and discusses paths for future research.

Related Work

Notation: In this paper, vectors are denoted by bold lower-case letters, matrices by bold upper-case ones, scalars by lower-case. All vectors are treated as column vectors. For instance, we consider in the rest of this paper a vector $x$ in $\mathbb{R}^n$ and a matrix $X$ in $\mathbb{R}^{m \times n}$. The columns of $X$ are represented by indexed vectors $x_1, \ldots, x_n$ such that we can write $X = [x_1, \cdots, x_n]$. The i-th entry is denoted by $x[i]$. 
and the i-th entry of j-th column of X is represented by x[i,j]. Same as the other literature, we define the $l_0 - norm$ of vector x as $\|x\|_0 = \# \{i, s.t. x[i] \neq 0 \}$, the $l_1 - norm$ of vector x is: $\|x\|_1 = \sum_{i=1}^{m}|x_i|$, the Frobenius norm of matrix X as $\|X\|_F = (\sum_{i=1}^{m} \sum_{j=1}^{n} |x_{i,j}|^2)^{\frac{1}{2}}$.

For a given matrix $X \in \mathbb{R}^{m \times n}$ with $r \operatorname{rank}(X) = k$, classical principal component analysis (PCA) seeks to find two smaller matrices $D$ and $A$ that satisfy the following formulation.

$$X = DA$$

It can be rewritten as the following formulation,

$$\min_{D \in \mathbb{R}^{m \times k}, A \in \mathbb{R}^{k \times n}} \|X - DA\|_F^2$$

Matrix D with lower dimension contains the latent structure of matrix X and it can be seen as a representative of X.

With potential sparsity requirements on D and A of model (2), we can get sparse solutions, which is very common in statistics and signal processing domain. Formulated as follows:

$$\min_{D \in \mathbb{R}^{m \times k}, A \in \mathbb{R}^{k \times n}} \frac{1}{2} \|X - DA\|_F^2 + \lambda \psi(A)$$

where, matrix A carries the decomposition coefficients of the signals X, each signal $x_i$ form X can be represented as $x_i \approx Da_i = \sum_{j=1}^{k} a_{i,j}d_j$. $\psi$ is sparsity-inducing regularization function, columns of A will become sparse under its control. And $C$ is the constraints on D, for example, $C = \{D \in \mathbb{R}^{m \times k} : \forall j \|d_j\|_2 \leq 1\}$. Parameter $\lambda$ controls the trade-off between data fitting and sparsity.

The above sparse model can be seen as a matrix factorization problem with sparse regularization, which has been proven very effective for feature extraction [29]. It is also flexible enough to accommodate a large set of constraints and regularizations, and has gained significant attention in scientific domains where dimension reduction is a key aspect [30]. Two representative of it are as following.

Zou and Hastie [5] imposed a method called SPCA, formulating PCA as a regression-type optimization problem and using the elastic-net regulation to produce modified principal components with sparse loadings.

$$\min_{A \in \mathbb{R}^{m \times k}, B \in \mathbb{R}^{n \times k}} \sum_{i=1}^{m} \|x_i - AB^T x_i\|^2 + \lambda \sum_{j=1}^{k} \|a_j\|^2 + \sum_{j=1}^{k} \lambda_1,j \|a_j\|_1 \quad s.t. A^T A = I_{k \times k}$$

An alternate optimization strategy was used in SPCA to get the solutions of sparse loading matrix $B$ and modified principal components matrix $A$. Four main procedures were included in their algorithm:

1) Apply conventional PCA on $X$ and initialize $A$ with the first k loadings,
2) Solve an elastic-net problem to get solution $B$ with respect to fixed $A$
3) Compute the SVD of $X^T XB$
4) Repeat 2) and 3) under a pre-defined convergence criteria.

SPCA has proved to be a very efficient dimension reduction method for synthetic dataset and gene expression dataset. However, as we can find that SVD decomposition on the whole dataset was needed in every iteration, making more complexity and limited information extracted from the original dataset. This can be found in the later experiment section.

Haipeng Shen et al. [6] proposed sPCA-rSVD method as another choice for sparse dimension reduction. They used the connection of PCA with SVD to extract the PCs through solving a low rank matrix approximation problem. Their model formulation is (5).

$$\|X - \tilde{u}\tilde{v}^T\|_F^2 + P_\lambda(\tilde{v})$$

sPCA-rSVD can accommodate different form of sparse regularizations ($P_\lambda$) on the targeted feature ($\tilde{v}$). Different from SPCA, it didn’t obtain the desired number of principal feature simultaneously, but one by one. Concretely, they get the first $\tilde{u}_1$ and $\tilde{v}_i$ via the rank-one
approximate of $X$, and then get the second from another rank-one approximate of the residual matrix $X_1 = X - \hat{u}_1 \hat{v}_1$.

**Proposed Method**

In this section we formulate our method also form the matrix factorization view and introduce different regularization strategies to illustrate a new way of feature extraction from HSI. Because the dimension reduction procedure doesn’t need the label information of the HSI image and can support sparsity tuning possibility on both the loadings and the features, we name it UDSDL(Unsupervised Double Sparsity Dimension Learning). Details of UDSDL are as follows.

Performing conventional PCA on the given dataset $X \in R^{m \times n}$, we can get its first $k$ largest principal components by solving the following formulation:

$$
\min_{U \in R^{m \times k}, V \in R^{n \times k}} \|X - UV\|_F^2 \quad s.t. \ U^T U = I_k, \ (k < n)
$$

Reform it to the same as the sparse model:

$$
\min_{D \in R^{m \times k}, A \in R^{k \times n}} \|X - DA\|_F^2, \ C = \{D \in R^{m \times k} : \forall i \neq j \ d_i^T d_j = 0\}
$$

Here, $D = U$ and $A = V^T$, conventional PCA with orthogonal requirement transforms to a matrix factorization model with constraint $C$.

Sparse solutions can be obtained by enforcing sparsity-inducing regularizations on $A$ or $D$ or even on both. Mairal[7] proofed that these sparse matrix $D$ is or is equivalent to the latent features of $X$. Then, dimension reduction with sparse constraint can be cast to the following sparse model.

$$
\min_{D \in R^{m \times k}, A \in R^{k \times n}} \|X - DA\|_F^2 + \lambda \varphi(A) \quad C = \{D \in R^{m \times k} : \forall i \neq j \ d_i^T d_j = 0\}
$$

It can be performed by minimizing a quadratic data-fitting term, with constraints and/or penalties over the loadings $\alpha$ and the features $D$.

$$
\min_{\alpha \in R^{m \times k}, D \in R^{m \times k}} \sum_{i=1}^{n} \frac{1}{2} \|x_i - D\|_2^2 + \lambda \varphi(\alpha_i)
$$

where $\|\|_2$ is the Euclidean norm, $C$ is a convex set of $R^{m \times k}$, and $\varphi : R^k \rightarrow R$ is a penalty(equal to the above mentioned sparsity-inducing regularization) over the loadings to enforce sparsity.

In general, the constraint set $C$ is traditionally a technical constraint ensuring that the coefficients $\alpha_i$ do not vanish, making the effect of the penalty disappear. Choice of constraint set and penalties typically offers some flexibility for a specific problem; for instance, the most used $l_1 - norm$ ($\varphi = \|\|_1$), leading to sparse coefficients $\alpha$ and $l_2 - norm(\varphi = \|\|_2)$ constraint on $D$ can ensure the reasonability of solution $A$.

Following the above analysis, we use $l_1 - norm$ on $A$ and elastic-net (used in Zou and Hastie 2006 to get sparse loadings, with the form $\|d\|_2^2 + \gamma d_1 \leq 1$ ) constraints on $D$, leading to sparse coefficients $\alpha$ and sparse features $d$ as the same time. Our model can be formulated as follows:

$$
\min_{\alpha \in R^{m \times k}, D \in R^{m \times k}} \frac{1}{2} \|X - DA\|_2^2 + \lambda \sum_{i=1}^{n} \|\alpha_i\|_1, \ C = \{D \in R^{m \times k} : \forall j \ (d_j^2 + \gamma d_1) \leq 1\}, \ k < n
$$

Obviously, this model can introduce sparsity both on $A$ and $D$ controlled by the tuning parameter $\lambda$ and $\gamma$. This model isn’t convex with both $D$ and $A$ simultaneously, but is convex with $A$ when keeping $D$ fixed and vice versa. Thus, an alternative–optimization strategy is the optimal choice to solve it, as proposed by Lee et al [8].

For fixed $D$, $A = [\alpha_1, \ldots, \alpha_n]$ can be obtained by solving the typical $l_1 - normal$ sparse coding problem:

$$
\alpha_i = \arg \min_{\alpha \in R^k} \frac{1}{2} \|x_i - D\|_2^2 + \lambda \|\alpha\|_1
$$

There exit three efficient methods to the above optimization problem: coordinate descent [9][10][11], proximal gradient [12][13], LARS (Least angle Regression) [14][15]. Details of each
method can be found by relevant reference, due to the limitation of space we omit the analysis of them. For computing complexity consideration we choose the LARS method.

Once a sparse solution of A is found via LARS, the next step is to solve the optimization with respect to D. For fixed A, finding the optimal D is a classical dictionary learning problem also with a few approaches, such as stochastic gradient descent [16][17], MOD[8], K-SVD[18], online dictionary learning [7].

Based on stochastic approximations, online dictionary learning method proposed by Mairal et al. has two obvious advantages: scaling up gracefully to large datasets and faster performance than the others. These two characters just satisfy the high performance requirement of HSI processing under the context of big data. That is the exact reason we choose it as our foundation and make adaptive improvement to find the sparse features by changing the $l_2 - norm$ constraint to elastic-net with corresponding solving algorithm named LARS-EN [19]. This improvement can not only enforce sparsity, but also ensure stability, even if it seems simple. As the new coefficient matrix A has been obtained, we can update each columns of matrix D iteratively and get the new D satisfying the following formulation.

$$D_t = \arg\min_{D \in C} \frac{1}{T} \sum_{i=1}^{T} \left( \frac{1}{2} \|x_i - Dx_i\|_2^2 + \lambda \|\alpha_i\|_1 \right), C = \left\{ D \in \mathbb{R}^{m \times k} : \forall j \|d_j\|_2^2 + \gamma \|d_j\|_1 \leq 1 \right\} \quad (12)$$

By summarizing the above analyses, we can give the UDSDL algorithm as follows.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
No. & Step \\
\hline
1 & Initialization: $D^0 \in \mathbb{R}^{m \times k}$ with random values, $A^0 \in \mathbb{R}^{k \times k}$ ← 0, $B^0 \in \mathbb{R}^{m \times k}$ ← 0 \\
2 & Input: $k, T, A^0, B^0, \lambda, X \in \mathbb{R}^{m \times n}$ (each column with zero mean and $l_2 - norm$ unit), $\gamma$ \\
3 & for $t=1$ to $T$ do \\
4 & Random draw a column vector $x_i$ from $X$, solve following sparse coding by LARS $\alpha_t = \arg\min_{\alpha \in \mathbb{R}^n} \frac{1}{2} \|x_i - D^{t-1} \alpha\|_2^2 + \lambda \|\alpha\|_1$ \\
5 & Compute: $A_t \leftarrow A^{t-1} + \alpha_t \alpha_t^T$, $B_t \leftarrow B^{t-1} + x_i \alpha_t^T$ \\
6 & Compute $D_t$ by updating each column of $D^{t-1}$ as follows: \\
7 & for $j=1$ to $k$ do \\
8 & $u_j \leftarrow -\frac{1}{A_{jj}} (b_j - D\alpha_j) + d_j$ \\
9 & $d_j \leftarrow \arg\min_{d \in \mathbb{R}^k} \|u_j - d\|_2^2$ s.t. $\|d\|_2^2 + \gamma \|d\|_1 \leq 1$ (Solved by LARS-EN) \\
10 & end for \\
11 & end for \\
12 & Output: $D \in \mathbb{R}^{m \times k}$ \\
\hline
\end{tabular}
\end{table}

X $\in \mathbb{R}^{m \times n}$ is the original HSI data with high dimensionality and each column of it is first centered and normalized to have unit $l_2 - norm$, $D \in \mathbb{R}^{m \times k}$ is the desired low dimensional matrix, scalar T is the pre-defined iteration number for optimization, parameter $\lambda$ is used to tune the sparsity of matrix A and $\gamma$ is used to tune the sparsity of matrix D. Two temporary matrices $A_t = \sum_{i=1}^{t} \alpha_i \alpha_i^T, B_t = \sum_{i=1}^{t} x_i \alpha_i^T$ store the vectors produced in each iteration and they are initialized to zero in the beginning. Each column of the desired matrix D will be updated one by one by the new sparse coefficient vector $\alpha$ in every iteration.

**Experiments**

**Competitors and Evaluation Method**

In this section, we validate our proposed method with three typical feature extraction methods as competitors: first is PCA, the de facto technique with no sparsity regularization or constraint, the other two are sparse-oriented: SPCA[5] and sPCA-rSVD [6] respectively.

Without loss of universality, we evaluate the performance of these DR (dimension reduction) techniques by the quantized classification results conducted on the low dimensional data.
Support vector machine (SVM) [20] was selected as the classifier, as it performs well even with a limited number of training samples. Three quantitative measurements are used to evaluate the DR results: OA (overall accuracy), AA (average accuracy), and the Kappa coefficient (kappa).

**HSI Data Set**

One HSI benchmark dataset named Indian Pines (https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html) was used in our experiment. False color composite image of the Indian Pines dataset and the ground truth can be seen in Figure 3.

**Experimental Setup**

In each experiment, we first use different DR methods on the HSI data, then, the support vector machine (SVM) classifier with polynomial kernels is employed for classification on the result low dimensions of each method. We use the MATLAB SVM Toolbox, LIBSVM [20], in our experiments. The SVM parameters (the degree d and the penalty factor C) are selected by grid search method within the same given set \{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3 \}. We use the implementation of LARS and LARS-EN, provided by another MATLAB Toolbox SpaSM (http://spams-devel.gforge.inria.fr/).

We use the regularization tuning parameter $\lambda = 1.2/\sqrt{m}$ in all of our experiments. The $1/\sqrt{m}$ is a classical normalization factor [21] and the constant 1.2 has shown to yield about 20 nonzero coefficients for Indian Pines Set. For faire comparison, we set the SPCA and the sPCA-rSVD have the same number of nonzero loadings.

The second tuning parameter is responsible for the controlling of the number of zero entries of columns in the learned feature matrix D. We will experimentally explore its effect on real HSI dataset.

Parameter $\gamma$ is responsible for controlling the number of zero entries of the desired feature matrix D. We will experimentally explore its effect on real HSI dataset.

In each experiment, we randomly choose 10% of the related samples for training and the remaining 90% for testing. Specific classes and the selected number of training and testing data in each class are reported in Table 2.

**Experimental Results and Analysis**

First we evaluate the applicability of the parse based methods on AVIRIS Indian Pines.

As neither SPCA nor sPCA-rSVD has the sparsity tuning ability in components (or features), we first set the UDSDL parameter $\gamma = 0$ to vanish the sparse constraint on D. Different number of dimensions (range in [5 10 15 20 25 30 35 40 45 50 55 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200]) was extracted from the original dataset by four DR methods: PCA, SPCA, sPCA-rSVD and UDSDL.

Changing curve of overall accuracy (in percent) corresponding to different dimensions extracted by DR method is shown in Figure 1 and the partially enlarged one is in Figure 2. Different colors are used to represent the result of each method. The curve with black color represent the classification result from the original data without dimension reduction, which is used as a baseline. Corresponding legend of color and method can be found on the right bottom of each figure.
From the above two figures we can see:

1) The overall trend of each method’s performance is upward, but the growth rate is very slow after reaching a certain dimension.

2) PCA method (blue curve with asterisk) presents optimal performance in most dimensions but without any sparse tuning strategy.

3) UDSDL (red curve with hollow circle) can obtain better OA than baseline after 20 dimensions and exceeds both SPCA (green curve with short vertical line) and sPCA-rSVD (yellow dot dash curve) under any dimension.

We tabulate the quantized AA, OA, Kappa and accuracy for each classes in Table 2 according to their peak performance. Numbers in bold font represent the optimal result under different methods.

Two observations can be found from the quantized results:

1) UDSDL obtains the sub-optimal OA and Kappa but the best AA among the four DR methods.

2) For the classes with very few training samples (see classes of 1, 7, 9), classification accuracy of UDSDL exceeds that of PCA.

Table 2. Classification accuracy for Indian Pines Scene (optimal dimensions in bracket).

<table>
<thead>
<tr>
<th>No</th>
<th>Name</th>
<th>Train</th>
<th>Test</th>
<th>original</th>
<th>PCA(20)</th>
<th>SPCA(20)</th>
<th>sPCA-rSVD(20)</th>
<th>UDSDL(20)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alfalfa</td>
<td>6</td>
<td>54</td>
<td>66.6667</td>
<td>68.7500</td>
<td>67.3750</td>
<td>72.9167</td>
<td>72.9167</td>
</tr>
<tr>
<td>2</td>
<td>Corn-no till</td>
<td>144</td>
<td>1434</td>
<td>77.7519</td>
<td>80.0775</td>
<td>80.0713</td>
<td>76.7442</td>
<td>78.9147</td>
</tr>
<tr>
<td>3</td>
<td>Corn-min till</td>
<td>84</td>
<td>834</td>
<td>79.2000</td>
<td>82.6667</td>
<td>77.4853</td>
<td>68.4000</td>
<td>73.6000</td>
</tr>
<tr>
<td>4</td>
<td>Corn</td>
<td>24</td>
<td>234</td>
<td>82.8517</td>
<td>70.9524</td>
<td>76.5333</td>
<td>76.6667</td>
<td>79.5238</td>
</tr>
<tr>
<td>5</td>
<td>Grass/pasture</td>
<td>50</td>
<td>497</td>
<td>95.0783</td>
<td>87.2483</td>
<td>94.0537</td>
<td>93.5123</td>
<td>94.4072</td>
</tr>
<tr>
<td>6</td>
<td>Grass/tree</td>
<td>75</td>
<td>747</td>
<td>97.1726</td>
<td>94.1964</td>
<td>91.7292</td>
<td>89.8810</td>
<td>93.7500</td>
</tr>
<tr>
<td>7</td>
<td>Grass/pasture-mowed</td>
<td>3</td>
<td>26</td>
<td>95.6522</td>
<td>95.6522</td>
<td>93.7391</td>
<td>95.6522</td>
<td>95.6522</td>
</tr>
<tr>
<td>8</td>
<td>Hay-windrowed</td>
<td>49</td>
<td>489</td>
<td>96.3636</td>
<td>97.9545</td>
<td>95.3273</td>
<td>94.7727</td>
<td>96.5909</td>
</tr>
<tr>
<td>9</td>
<td>Oats</td>
<td>2</td>
<td>20</td>
<td>50.0000</td>
<td>33.3333</td>
<td>49.0000</td>
<td>50.0000</td>
<td>61.1111</td>
</tr>
<tr>
<td>10</td>
<td>Soybeans-no till</td>
<td>97</td>
<td>968</td>
<td>73.1343</td>
<td>82.8932</td>
<td>76.6223</td>
<td>69.6800</td>
<td>68.4271</td>
</tr>
<tr>
<td>11</td>
<td>Soybeans-min till</td>
<td>247</td>
<td>2468</td>
<td>79.7839</td>
<td>84.1063</td>
<td>81.3652</td>
<td>81.9451</td>
<td>84.4665</td>
</tr>
<tr>
<td>12</td>
<td>Soybeans-clean till</td>
<td>62</td>
<td>614</td>
<td>84.9638</td>
<td>85.1449</td>
<td>83.2645</td>
<td>78.4420</td>
<td>87.6812</td>
</tr>
<tr>
<td>14</td>
<td>Woods</td>
<td>130</td>
<td>1294</td>
<td>94.5017</td>
<td>93.2990</td>
<td>93.1168</td>
<td>96.7354</td>
<td>94.2440</td>
</tr>
<tr>
<td>15</td>
<td>Bldg-grass-tree-drives</td>
<td>38</td>
<td>380</td>
<td><strong>57.6023</strong></td>
<td>50.5848</td>
<td>50.7193</td>
<td>40.6433</td>
<td>51.4620</td>
</tr>
<tr>
<td>16</td>
<td>Stone-steel towers</td>
<td>10</td>
<td>95</td>
<td><strong>91.7647</strong></td>
<td>84.7059</td>
<td>89.2994</td>
<td>90.5882</td>
<td>90.5882</td>
</tr>
<tr>
<td></td>
<td>OA</td>
<td></td>
<td></td>
<td>83.4281</td>
<td><strong>84.7045</strong></td>
<td>83.0945</td>
<td>81.0898</td>
<td>83.4174</td>
</tr>
<tr>
<td></td>
<td>AA</td>
<td></td>
<td></td>
<td>82.6229</td>
<td>80.6899</td>
<td>81.1135</td>
<td>79.7540</td>
<td><strong>82.6756</strong></td>
</tr>
<tr>
<td></td>
<td>kappa</td>
<td></td>
<td></td>
<td>0.8115</td>
<td><strong>0.8259</strong></td>
<td>0.8102</td>
<td>0.7837</td>
<td>0.8106</td>
</tr>
</tbody>
</table>
Alignment the above curves, quantized and visual results of the four methods we can find that although the UDSDL method proposed in this paper is slightly lower than PCA in OA and Kappa indexed, the average accuracy of all classes and accuracy of some classes, especially those with very small number of training set are higher than it. This fully proves the ability of UDSDL to extract discriminant features from small training sets, which is very important for HSI. At the same time, different from the SPCA and sPCA-rSVD, UDSDL introduce the random approximation strategy which can avoid high time-consuming operations such as eigenvalue solution and SVD decomposition and is more applicable in the context of large amounts of data.

Further, we explore the possibility of enforcing sparsity on features (according to matrix D in model 11) and preserving the sparsity of A at the same time. Here we tune the parameter $\gamma$ range from 0 to 0.5 with 0.02 step increasement of each test, use the proportion of zero entries in feature matrix as a measure of sparsity. Figure 4 presents the curve of OA changing with the sparsity of features.

We can see that UDSDL can achieve the highest OA with no sparsity constraint and with a slightly fallen according to less sparsity and keeps a relatively stable performance in a fixed sparsity range between [0.05, 0.5]. This demonstrates that we can have the privileged choice of balancing among the accuracy and storage demanding. In some situation, on-board processing of HSI for example, fast processing requirements under limited storage space are more important than the precise classification results and UDSDL provides the possibility that none of the other method can.

Conclusions

In this paper, we present a sparse dimension reduction method UDSDL for HSI. This method can leaning the latent sparse structure from high dimension data in an unsupervised mode. Comparing to this kind of two typical sparse methods SPCA and sPCA-rSVD, our method can learn more discriminative features with the same sparsity and can enforce double sparsity when necessary. Comparing to the de facto DR technique, PCA, the experimental results on real HSI dataset indicate that UDSDL can achieve the very close performance with much more sparse and less volume of
data in terms of classification accuracy. Future work will involve exploring the parallel version of this method for further speedup.

Figure 4. SVM classification of Indian Pines (OA versus different sparsity).

Acknowledgements
This work was supported by the National Natural Science Foundation of China (61872042); the Key Project of the Beijing Education Commission (KZ201911417048). supported in part by the 2018 visiting scholar project and 2019 scientific research project of Beijing Union University (ZK50201903).

References


