Chapter 8
Sparsity-Based Methods for Classification

Zebin Wu, Yang Xu and Jianjun Liu

Abstract Sparsity is an important prior for various signals, and sparsity-based methods have been widely used in hyperspectral image classification. This chapter introduces the sparse representation methodology and its related techniques for hyperspectral image classification. To start with, we provide a brief review on the mechanism, models, and algorithms of sparse representation classification (SRC). We then introduce several advanced SRC methods that can improve hyperspectral image classification accuracy by incorporating spatial–spectral information into SRC models. As a case study, a hyperspectral image SRC method based on adaptive spatial context is discussed in detail to demonstrate the performance of SRC methods in hyperspectral image classification.

8.1 Introduction

In the last few decades, sparsity has become one of the most important concepts in the field of signal processing. Sparsity concept has been widely employed in a variety of fields, e.g., source separation, restoration, and compression. Sparse representation was originally derived from compressed sensing [1–3], suggesting that if a signal is sparse or compressive, the original signal can be reconstructed with a few number of samplings. By introducing sparsity in sampling, compressed sensing has achieved great success in information theory, image acquisition, image processing, medical imaging, remote sensing, etc. Compressed sensing has also motivated many researches on sparse representation. As a matter of fact, signals in real world may not be sparse in the original space, but they can be sparse in an appropriate basis.

Z. Wu (✉) · Y. Xu
Nanjing University of Science and Technology, Nanjing, China
e-mail: wuzb@njust.edu.cn

J. Liu
Jiangnan University, Wuxi, China

S. Prasad and J. Chanussot (eds.), *Hyperspectral Image Analysis*,
Advances in Computer Vision and Pattern Recognition,
https://doi.org/10.1007/978-3-030-38617-7_8

Hyperspectral imaging sensors record reflected light in hundreds of narrow frequencies covering the visible, near-infrared, and shortwave infrared bands. This abundant spectral information yields more precise measures and makes it possible to gain insight into the material at each pixel in the image. Supervised classification plays a central role in hyperspectral image (HSI) analysis, such as land-use or land-cover mapping, forest inventory, or urban-area monitoring [4]. Many methods have been proposed for solving the HSI classification problem, such as logistic regression [5], support vector machines (SVM) [6], artificial neural networks [7], and k-nearest neighbor (KNN) classifier [8]. These methods can serve the purpose of generating acceptable classification results. However, the high dimensionality of hyperspectral data remains a challenge for HSI classification.

To address this problem, sparse representation [9, 10] has been employed for classifying high-dimensional signals. A sparse representation classification (SRC) method [10] has been first proposed for face recognition. A test signal is sparsely represented by an over-complete dictionary composed of labeled training samples. At the decision level, the label of each test sample is set as the class whose corresponding atoms maximally represent the original test sample. Since then, SRC has been widely used in face recognition [10, 11], speech recognition [12], and image super-resolution [13]. Chen et al. [14] proposed an SRC framework for solving the HSI classification problem, in which each sample is a pixel's spectral responses. Inspired by this work, many improved SRC methods have been proposed for HSI classification.

In this chapter, we investigate the SRC methods and present several advanced models of sparse representation for HSI classification. More specifically, we will give a case study of SRC method that improves the classification accuracy by incorporating the spectral–spatial information of HSI into the SRC framework.

8.2 Sparse Representation-Based HSI Classification

In the theory of sparse representation, given a dictionary, each signal can be linearly represented by a set of atoms in the dictionary. Designing an over-complete dictionary and obtaining the sparse representation vector through sparse coding are the two main goals of sparse representation.

In HSI classification, SRC assumes that the features belonging to the same class approximately lie in the same low-dimensional subspace spanned by dictionary atoms from the same class. Suppose we have M distinct classes and $N_i (i = 1, 2, \ldots, M)$ training samples for each class. Each class has a sub-dictionary $\mathbf{D}_i = [\mathbf{d}_{i,1}, \mathbf{d}_{i,2}, \ldots, \mathbf{d}_{i,N_i}] \in \mathbb{R}^{B \times N_i}$ in which the columns represent training samples and B is the number of spectral bands. A test pixel $\mathbf{x} \in \mathbb{R}^B$ can be represented by a sparse linear combination of the training pixels as

$$\mathbf{x} = \mathbf{D}\,\boldsymbol{\alpha} \tag{8.1}$$

where $\mathbf{D} = [\mathbf{D}_1\, \mathbf{D}_2 \ldots \mathbf{D}_M] \in \mathbb{R}^{B\times N}$ with $N = \sum_{i=1}^{M} N_i$ is the dictionary constructed by combining all sub-dictionaries $\{\mathbf{D}_i\}_{i=1,\ldots,M}$. $\boldsymbol{\alpha} \in \mathbb{R}^N$ is an unknown sparse vector with K nonzero entries. Here, we denote $K = \|\boldsymbol{\alpha}\|_0$. The sparse coefficient vector $\boldsymbol{\alpha}$ is obtained by solving the following problem

$$\min_{\boldsymbol{\alpha}} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2 \quad \text{s.t}\, \|\boldsymbol{\alpha}\|_0 \leq K_0 \tag{8.2}$$

where K_0 is a pre-specified upper bound of K. The class label of $\mathbf{x}$ is determined by the minimal residual between $\mathbf{x}$ and its approximation from each class sub-dictionary, i.e.,

$$\text{class}(\mathbf{x}) = \arg \min_{i=1,2,\ldots,M} \|\mathbf{x} - \mathbf{D}_i\boldsymbol{\alpha}_i\|_2 \tag{8.3}$$

where $\boldsymbol{\alpha}_i$ is the sub-vector corresponding to the i-th class, and $\mathbf{D}_i$ denotes the sub-dictionary.

Problem (2) is NP-hard, and can be approximately solved by greedy algorithms, such as orthogonal match pursuit (OMP) and subspace pursuit (SP).

In OMP algorithm, we select one atom from the dictionary that is most correlated with the residual. The algorithmic flow of the OMP algorithm is described in Algorithm 8.1.

Algorithm 1 Orthogonal Matching Pursuit

Input: Dictionary $\mathbf{D} = [\mathbf{d}_1\ \mathbf{d}_2\ \ldots\ \mathbf{d}_N]$, test samples $\mathbf{x}$, normalize the dictionary $\mathbf{D}$ and $\mathbf{x}$.

Initialize: Set the residual $\mathbf{r}_0 = \mathbf{x}$, set the index set $\Lambda_0 = \varnothing$, set iteration count $k = 1$;

While termination criterion not not satisfied **do**

1) Compute $\lambda_k = \arg\max_{i=1,\ldots,N} \mathbf{r}_{k-1}^T \mathbf{d}_i$, and find the atom that matches with residual most;
2) Update the index set $\Lambda_k = \Lambda_{k-1} \cup \lambda_k$;
3) Compute $\mathbf{P}_k = (\mathbf{D}_{\Lambda_k}^T \mathbf{D}_{\Lambda_k})^{-1} \mathbf{D}_{\Lambda_k}^T \mathbf{x} \in \mathbb{R}^k$, where $\mathbf{D}_{\Lambda_k}$ is the sub-dictionary composed of the atoms from the index set;
4) Compute the residual $\mathbf{r}_k = \mathbf{x} - \mathbf{D}_{\Lambda_k}\mathbf{P}_k$;
5) $k = k + 1$;

Output: the index set $\Lambda = \Lambda_{k+1}$, the sparse vector $\boldsymbol{\alpha}$, where the non-zero elements are $(\mathbf{D}_{\Lambda_k}^T \mathbf{D}_{\Lambda_k})^{-1} \mathbf{D}_{\Lambda_k}^T \mathbf{x}$, and the support is the determined by the index set.

The procedure of SP algorithm is similar to that of OMP algorithm. The difference is that SP finds all the K atoms that satisfy (8.2) during one iteration. The complete procedure of SP algorithm is provided in Algorithm 8.2.

Algorithm 2 Subspace Pursuit

Input: Dictionary $\mathbf{D}=[\mathbf{d}_1\ \mathbf{d}_2\ ...\ \mathbf{d}_N]$, test samples $\mathbf{x}$, sparsity K, normalize the dictionary $\mathbf{D}$ and $\mathbf{x}$.

Initialize: Set the index set Λ_0, where the element of Λ_0 are determined by the K largest elements in $\mathbf{x}^T\mathbf{D}$, set the residual $\mathbf{r}_0=\mathbf{x}-\mathbf{D}_{\Lambda_0}(\mathbf{D}_{\Lambda_0}^T\mathbf{D}_{\Lambda_0})^{-1}\mathbf{D}_{\Lambda_0}^T\mathbf{x}$, set iteration count $k=1$;

While stopping criterion not satisfied **do**

1) Find the indices of K atoms according to the K largest elements in $\mathbf{r}_{k-1}^T\mathbf{d}_i$, denoted as l
2) Update the index set $\Lambda_k=\Lambda_{k-1}\cup l$;
3) Compute $\mathbf{P}_k=(\mathbf{D}_{\Lambda_k}^T\mathbf{D}_{\Lambda_k})^{-1}\mathbf{D}_{\Lambda_k}^T\mathbf{x}\in\mathbb{R}^{2K}$;
4) Fin the K largest elements in $\mathbf{P}_k$, and update the index set Λ_k;
5) Compute the residual $\mathbf{r}_k=\mathbf{x}-\mathbf{D}_{\Lambda_k}(\mathbf{D}_{\Lambda_k}^T\mathbf{D}_{\Lambda_k})^{-1}\mathbf{D}_{\Lambda_k}^T\mathbf{x}$;
6) $k=k+1$;

Output: the index set $\Lambda=\Lambda_{k+1}$, the sparse vector $\boldsymbol{\alpha}$, where the non-zero elements are $(\mathbf{D}_{\Lambda_k}^T\mathbf{D}_{\Lambda_k})^{-1}\mathbf{D}_{\Lambda_k}^T\mathbf{x}$, and the support is the determined by the index set.

8.3 Advanced Models of Sparse Representation for Hyperspectral Image Classification

Many advanced methods based on SRC have been proposed for HSI classification.

In HSI, pixels within a small neighborhood usually consist of similar materials. Therefore, these pixels tend to have high spatial correlation [14]. The corresponding sparse coefficient vectors share a common sparsity pattern as follows.

Let $\{\mathbf{x}_t\}_{t=1,\ldots,T}$ be T pixels in a fixed window centered at $\mathbf{x}_1$. These pixels can be represented by

$$\begin{aligned}\mathbf{X}&=[\mathbf{x}_1\mathbf{x}_2\ldots\mathbf{x}_T]=[\mathbf{D}\,\boldsymbol{\alpha}_1\,\mathbf{D}\,\boldsymbol{\alpha}_2\ldots\mathbf{D}\,\boldsymbol{\alpha}_T]\\&=\mathbf{D}\underbrace{[\boldsymbol{\alpha}_1\,\boldsymbol{\alpha}_2\ldots\boldsymbol{\alpha}_T]}_{\mathbf{S}}=\mathbf{DS}\end{aligned}\tag{8.4}$$

In the joint sparsity model (JSM), the sparse vectors $\{\boldsymbol{\alpha}_t\}_{t=1,\ldots,T}$ share the same support Ω. $\mathbf{S}$ is a sparse matrix with $|\Omega|$ nonzero rows, which can be obtained by solving the following optimization problem,

$$\min_{\mathbf{S}}\|\mathbf{X}-\mathbf{DS}\|_F\quad\text{s.t}\quad\|\mathbf{S}\|_{\text{row},0}\le K_0\tag{8.5}$$

where $\|\mathbf{S}\|_{\text{row},0}$ denotes the number of nonzero rows of $\mathbf{S}$, and $\|\cdot\|_F$ denotes the Frobenius norm. The problem in (8.5) can be approximately solved by the simultaneous version of OMP (SOMP). The label of the central pixel $\mathbf{x}_1$ can be determined minimizing the total residual

$$\text{class}(\mathbf{x}_1) = \arg \min_{i=1,\ldots,M} \|\mathbf{X} - \mathbf{D}_i \mathbf{S}_i\|_F \tag{8.6}$$

where $\mathbf{S}_i$ is the sub-sparse coefficient matrix corresponding to the i-th class.

Note that, the optimization models (8.2) and (8.5) are non-convex, and can be converted into convex versions by relaxing the norm constraints:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{8.7}$$

$$\min_{\mathbf{S}} \frac{1}{2}\|\mathbf{X} - \mathbf{D}\mathbf{S}\|_F^2 + \lambda\|\mathbf{S}\|_{1,2} \tag{8.8}$$

where $\|\boldsymbol{\alpha}\|_1 = \sum_{i=1}^{N} |\alpha_i|$ is the ℓ_1 norm, $\|\mathbf{S}\|_{1,2} = \sum_{i=1}^{N} \|\mathbf{s}^i\|_2$ is the $\ell_{1,2}$ norm, and $\mathbf{s}^i$ represents the i-th row of $\mathbf{S}$.

The JSM model enforces that the pixels in the neighborhood of the test sample are represented by the same atoms. However, if the neighboring pixels are on the boundary of several homogeneous regions, they would be classified into different classes. In this scenario, different sub-dictionaries should be used. Laplacian sparsity promotes sparse coefficients of neighboring pixels belonging to different clusters to be different from each other. For this reason, a weight matrix $\mathbf{W}$ is introduced, where w_{ij} represents the similarity between a pair of pixels $\mathbf{x}_i$ and $\mathbf{x}_j$ in the neighborhood of the text sample. As reported in [15], the optimization problem with additional Laplacian sparsity prior can be described as

$$\min_{\mathbf{S}} \frac{1}{2}\|\mathbf{X} - \mathbf{D}\mathbf{S}\|_F^2 + \lambda_1\|\mathbf{S}\|_1 + \lambda_2 \sum_{i,j} w_{ij} \|\mathbf{s}_i - \mathbf{s}_j\|_2^2 \tag{8.9}$$

where λ_1 and λ_2 are regularization parameters. $\mathbf{s}_i$ is the i-th column of matrix $\mathbf{S}$. Weight matrix $\mathbf{W}$ can characterize the similarity among neighboring pixels in the spectral space. If two pixels are similar, the weight value will be large. As a result, their corresponding sparse codes will be similar. On the other hand, if two pixels are less similar, the weight value will be small, allowing a large difference between their sparse codes. Laplacian sparsity prior is more flexible than the joint sparsity prior. In fundamental, the joint sparsity prior can be regarded as a special case of Laplacian sparsity. Laplacian sparsity prior can well characterize more pixels in the image, since the sparse codes of the neighboring pixels are not limited to have the same supports. Suppose $\mathbf{L} = \mathbf{I} - \mathbf{H}^{-1/2}\mathbf{W}\mathbf{H}^{-1/2}$ is the normalized symmetric Laplacian matrix and, $\mathbf{H}$ is the degree matrix computed from $\mathbf{W}$. We can have the following

new optimization problem:

$$\min_{\mathbf{S}} \frac{1}{2}\|\mathbf{X} - \mathbf{DS}\|_F^2 + \lambda_1\|\mathbf{S}\|_1 + \lambda_2 tr(\mathbf{SLS}^T) \tag{8.10}$$

In JSM model, each pixel is represented by the atoms in the dictionary, and is classified according to the residual between the sparse codes multiplying the sub-dictionary. It is a reasonable assumption that each pixel can only be represented by one sub-dictionary. This condition can be achieved by enforcing the sparse codes corresponding to one sub-dictionary to be active and other ones to be inactive. Group Lasso sums up the Euclidean norm of the sparse codes corresponding to all sub-dictionaries as the sparsity prior. In [15], group Lasso is introduced as the new regularization in the optimization problem, i.e.,

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\,\boldsymbol{\alpha}\|_2^2 + \lambda \sum_{g\in G} \omega_g \left\|\boldsymbol{\alpha}_g\right\|_2 \tag{8.11}$$

where $g \subset \{G_1, G_2, \cdots G_M\}$, $\sum_{g\in G} \left\|\boldsymbol{\alpha}_g\right\|_2$ represents the group sparse prior defined in terms of M groups, ω_g is the weight and is set to the square root of the cardinality of the corresponding group. Note here that $\boldsymbol{\alpha}_g$ represents the coefficients of different groups. In a similar way, the group sparsity [15] can be employed in the JSM model as follows:

$$\min_{\mathbf{S}} \frac{1}{2}\|\mathbf{X} - \mathbf{DS}\|_F^2 + \lambda \sum_{g\in G} \omega_g \left\|\mathbf{S}_g\right\|_F \tag{8.12}$$

where $\sum_{g\in G} \left\|\mathbf{S}_g\right\|_F$ refers to the collaborative group Lasso regularization defined in terms of groups, and $\mathbf{S}_g$ is the sub-matrix corresponding to the g-th sub-dictionary.

In models (8.11) and (8.12) only group sparsity is introduced, and the sparsity of the sparse code corresponding to sub-dictionary is not taken into consideration. When the sub-dictionary is over-complete, it is important to introduce the sparsity within each group [15]. The ℓ_1-norm regularization can be incorporated into the objective function of (8.11) as follows:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\,\boldsymbol{\alpha}\|_2^2 + \lambda_1 \sum_{g\in G} \omega_g \left\|\boldsymbol{\alpha}_g\right\|_2 + \lambda_1\|\boldsymbol{\alpha}\|_1 \tag{8.13}$$

Similarly, the problem in (8.13) can be extended to JSM as follows:

$$\min_{\mathbf{S}} \frac{1}{2}\|\mathbf{X} - \mathbf{DS}\|_F^2 + \lambda_1 \sum_{g\in G} \omega_g \left\|\mathbf{S}_g\right\|_F + \lambda_1 \sum_{g\in G} \omega_g \left\|\mathbf{S}_g\right\|_1 \tag{8.14}$$

Another effective method is to introduce the correlation coefficient (CC) [16]. Traditionally, CC value is used to measure the correlation between different variables. In HSI classification, we can use CCs to determine whether pixels represent the same class. In general, CC can be calculated as follows:

$$\rho = \frac{\mathrm{cov}(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{\mathrm{var}(\mathbf{x}_i)} \cdot \sqrt{\mathrm{var}(\mathbf{x}_j)}} = \frac{\sum_{z=1}^{B} (\mathbf{x}_{iz} - u_{\mathbf{x}_i})(\mathbf{x}_{jz} - u_{\mathbf{x}_j})}{\sqrt{\sum_{z=1}^{B} (\mathbf{x}_{iz} - u_{\mathbf{x}_i})^2} \cdot \sqrt{\sum_{z=1}^{B} (\mathbf{x}_{jz} - u_{\mathbf{x}_j})^2}} \tag{8.15}$$

where $\mathrm{var}(\mathbf{x}_i)$ and $\mathrm{var}(\mathbf{x}_j)$ are the variance of $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. $\mathbf{x}_{iz}$ refers to the z-th element in $\mathbf{x}_i$.$u_{\mathbf{x}_i} = (1/B) \sum_{z=1}^{B} \mathbf{x}_{iz}$, and $u_{\mathbf{x}_j} = (1/B) \sum_{z=1}^{B} \mathbf{x}_{jz}$ represents the mean values of the corresponding vectors. According to the definition of CC, we have $|\rho| \leq 1$. Stronger correlation indicates that ρ is close to 1.

Following the method in [16], CCs among the training samples and test samples are first calculated. Given a test sample $\mathbf{x}$ and any training sample $\mathbf{d}_j^i$, where $\mathbf{d}_j^i$ represents the j-th atom in the i-th sub-dictionary. The CC between $\mathbf{x}$ and $\mathbf{d}_j^i$ can be calculated as follows:

$$\rho_j^i = \frac{\mathrm{cov}(\mathbf{d}_j^i, \mathbf{x})}{\sqrt{\mathrm{var}(\mathbf{d}_j^i)} \cdot \sqrt{\mathrm{var}(\mathbf{x})}} = \frac{\sum_{z=1}^{B} [(\mathbf{d}_j^i)_z - u_{\mathbf{d}_j^i}][(\mathbf{x})_z - u_{\mathbf{x}}]}{\sqrt{\sum_{z=1}^{B} [(\mathbf{d}_j^i)_z - u_{\mathbf{d}_j^i}]^2} \cdot \sqrt{\sum_{z=1}^{B} [(\mathbf{x})_z - u_{\mathbf{x}}]^2}}. \tag{8.16}$$

We define a matrix $\boldsymbol{\rho}^i = \{\rho_1^i, \rho_2^i, \ldots, \rho_{N_i}^i\}$. This matrix is sorted in descending order according to CCs among different training samples. Subsequently, the mean of L largest $\boldsymbol{\rho}^i$ is calculated as the CC cor^i. Assuming that the L largest $\boldsymbol{\rho}^i$ consists of $\{\rho_1^i, \rho_2^i, \ldots, \rho_L^i\}$, the CC cor^i can be calculated as

$$cor^i = \frac{1}{L}(\rho_1^i + \rho_2^i + \cdots + \rho_L^i). \tag{8.17}$$

Finally, the CC is combined with the JSM at the decision level to exploit the CCs among training and test samples as well as the representation residuals.

$$\mathrm{class}(\mathbf{x}_1) = \arg \min_{i=1,\ldots,M} \|\mathbf{X} - \mathbf{D}_i \mathbf{S}_i\|_F + \lambda(1 - cor^i(\mathbf{x}_1)) \tag{8.18}$$

where $cor^i \in [0, 1]$ represents the CCs among pixels, and λ is the regularization parameter.

One more approach to improve SRC is kernel trick. As an extension of SRC, kernel SRC (KSRC) uses the kernel trick to project data into a feature space, in which the projected data are linearly separable.

Suppose the feature mapping function $\phi : \mathbb{R}^B \to \mathbb{R}^K, (B \leq K)$ maps the features and also the dictionary to a high-dimensional feature space, $\mathbf{x} \to \phi(\mathbf{x})$, $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N] \to \phi(\mathbf{D}) = [\phi(\mathbf{d}_1), \phi(\mathbf{d}_2), \ldots, \phi(\mathbf{d}_N)]$. By replacing the

mapped features and dictionary in (8.7), we have the KSRC model,

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\phi(\mathbf{x}) - \phi(\mathbf{D})\boldsymbol{\alpha}\|_2 + \lambda\|\boldsymbol{\alpha}\|_1. \tag{8.19}$$

Similarly, the class label of $\mathbf{x}$ is determined as

$$\text{class}(\mathbf{x}) = \arg \min_{i=1,2,\ldots,M} \|\phi(\mathbf{x}) - \phi(\mathbf{D}_i)\boldsymbol{\alpha}_i\|_2. \tag{8.20}$$

It is worth mentioning that all ϕ mappings used in KSRC occur in the form of inner products, allowing us to define a kernel function $\mathbf{k}$ for any samples $\mathbf{x}_i \in \mathbb{R}^B$.

$$\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle \tag{8.21}$$

In this way, KSRC can be constructed using only the kernel function, without considering the mapping ϕ explicitly. Then, the optimization problem can be rewritten as

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{Q}\,\boldsymbol{\alpha} - \boldsymbol{\alpha}\,\mathbf{p} + \lambda\|\boldsymbol{\alpha}\|_1 + C \tag{8.22}$$

where $C = \frac{1}{2}\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)$ is a constant, $\mathbf{Q}$ is a $B \times B$ matrix with $\mathbf{Q}_{ij} = \mathbf{k}(\mathbf{d}_i, \mathbf{d}_j)$, and $\mathbf{p}$ is a $B \times 1$ vector with $\mathbf{p}_i = \mathbf{k}(\mathbf{d}_i, \mathbf{x})$. Analogously, the classification criterion can be rewritten as

$$\text{class}(\mathbf{x}) = \arg \min_{i=1,2,\ldots,M} \delta_i^T(\boldsymbol{\alpha})\mathbf{Q}\delta(\boldsymbol{\alpha}) - 2\delta_i^T(\boldsymbol{\alpha})\mathbf{p} \tag{8.23}$$

where $\delta_i(\cdot)$ is the characteristic function that selects coefficients within the i-th class and sets all other coefficients to zero.

Valid kernels are only those satisfying the Mercer's condition [17, 18]. Some commonly used kernels in kernel methods include linear kernel, polynomial kernel, and Gaussian radial basis function kernel. Assuming $\mathbf{k}_1$ and $\mathbf{k}_2$ are two valid Mercer's kernels over $\mathcal{X} \times \mathcal{X}$ with $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^B$ and $z > 0$, the direct sum $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{k}_1(\mathbf{x}_i, \mathbf{x}_j) + \mathbf{k}_2(\mathbf{x}_i, \mathbf{x}_j)$, tensor product $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{k}_1(\mathbf{x}_i, \mathbf{x}_j) \cdot \mathbf{k}_2(\mathbf{x}_i, \mathbf{x}_j)$, or scaling $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = z\mathbf{k}_1(\mathbf{x}_i, \mathbf{x}_j)$ are valid Mercer's kernels [19].

A suitable kernel is a kernel whose structure reflects data relations. To properly define such a kernel, unlabeled information and geometrical relationships between labeled and unlabeled samples are very useful. The spatial–spectral kernel sparse representation is proposed [20], in which the neighboring filtering kernel is presented and the corresponding optimization algorithm is developed.

A full family of composite kernels (CKs) for the combination of spectral and spatial contextual information have been presented in SVM [21, 22]. These kernels are valid and are all suitable for KSRC. Although one can improve the performance of KSRC by CK, it is worth noting that the kernel should learn all high-order similarities between neighboring samples directly, and should reflect the data lying in complex

manifolds. For these purposes, the neighbor filtering (NF) kernel would be a good choice, which computes the spatial similarity between neighboring samples in the feature space.

Given $\mathbf{x}^m \in \Omega, m = 1, 2, \ldots, \omega^2$, with Ω being the spatial window ω around pixel. Let $\phi(\mathbf{x}^m)$ be the image of $\mathbf{x}^m$ under the mapping ϕ. In order to describe $\phi(\mathbf{x})$, a straightforward way is to use the average of spatially neighboring pixels in the kernel space. This method is similar to the mean filtering. The estimated vector is given by

$$\mathrm{MF}(\phi(\mathbf{x})) = \frac{1}{\omega^2} \sum_{m=1}^{\omega^2} \phi(\mathbf{x}^m). \tag{8.24}$$

However, the mean filtering rarely reflects relative contributions (which treats every neighboring pixel equally). To address this issue, the neighboring filtering is defined as

$$\mathrm{NF}(\phi(\mathbf{x})) = \frac{1}{\sum_m \mathbf{w}^m} \sum_{m=1}^{\omega^2} \mathbf{w}^m \phi(\mathbf{x}^m) \tag{8.25}$$

where $\mathbf{w}^m = \exp(-\gamma_0 ||\mathbf{x} - \mathbf{x}^m||_2^2)$ and parameter $\gamma_0 > 0$ acts as a degree of filtering.

Let us consider two different pixels $\mathbf{x}_i$ and $\mathbf{x}_j$. We are interested in defining a similarity function that estimates the proximity between them in a sufficiently rich feature space. A straightforward kernel function reflecting the similarity between them is obtained by evaluating the kernel function between the estimated vectors

$$\begin{aligned}
\mathbf{k}_{\mathrm{NF}}(\mathbf{x}_i, \mathbf{x}_j) &= \left\langle \mathrm{NF}(\phi(\mathbf{x}_i)), \mathrm{NF}(\phi(\mathbf{x}_j)) \right\rangle \\
&= \left\langle \frac{\sum_{m=1}^{\omega^2} \mathbf{w}_i^m \phi(\mathbf{x}_i^m)}{\sum_m \mathbf{w}_i^m}, \frac{\sum_{n=1}^{\omega^2} \mathbf{w}_i^n \phi(\mathbf{x}_j^n)}{\sum_n \mathbf{w}_i^n} \right\rangle \\
&= \frac{\sum_{m=1}^{\omega^2} \sum_{n=1}^{\omega^2} \mathbf{w}_i^m \mathbf{w}_j^n \mathbf{k}(\mathbf{x}_i^m, \mathbf{x}_j^n)}{\sum_m \mathbf{w}_i^m \sum_n \mathbf{w}_i^n},
\end{aligned} \tag{8.26}$$

which is referred to as neighbor filtering (NF) kernel. Similarly, we can define mean filtering (MF) kernel as follows:

$$\begin{aligned}
\mathbf{k}_{\mathrm{MF}}(\mathbf{x}_i, \mathbf{x}_j) &= \left\langle \mathrm{MF}(\phi(\mathbf{x}_i)), \mathrm{MF}(\phi(\mathbf{x}_j)) \right\rangle \\
&= \left\langle \frac{1}{\omega^2} \sum_{m=1}^{\omega^2} \phi(\mathbf{x}_j^m), \frac{1}{\omega^2} \sum_{n=1}^{\omega^2} \phi(\mathbf{x}_j^n) \right\rangle \\
&= \frac{1}{\omega^4} \sum_{m=1}^{\omega^2} \sum_{n=1}^{\omega^2} \mathbf{k}(\mathbf{x}_i^m, \mathbf{x}_j^n),
\end{aligned} \tag{8.27}$$

which computes the spatial similarity between neighboring samples, whereas the cluster similarity is computed in the mean map kernel.

Since $\mathbf{Q}$ is a valid kernel, the objective function of (8.22) is convex, which is the same as the objective function of (8.19) except for the definition of $\mathbf{Q}$ and $\mathbf{p}$. Therefore, alternating direction method of multipliers (ADMM) [23] can be used to solve this problem. By introducing a new variable $\mathbf{u} \in \mathbb{R}^B$, the objective function can be rewritten as

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{Q}\,\boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{p} + \lambda\|\boldsymbol{\alpha}\|_1 \\ &\text{s.t.} \quad \mathbf{u} = \boldsymbol{\alpha}. \end{aligned} \tag{8.28}$$

ADMM imposes the constraint $\mathbf{u} = \mathbf{a}$ which can be defined as

$$\begin{cases} (\boldsymbol{\alpha}^{(t+1)}, \mathbf{u}^{(t+1)}) = \arg\min\limits_{\boldsymbol{\alpha},\mathbf{u}} \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{Q}\,\boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{p} + \lambda\|\boldsymbol{\alpha}\|_1 + \frac{\mu}{2}\left\|\boldsymbol{\alpha} - \mathbf{u} - \mathbf{d}^{(t)}\right\|_2^2 \\ \mathbf{d}^{(t+1)} = \mathbf{d}^{(t)} - (\boldsymbol{\alpha}^{(t+1)} - \mathbf{u}^{(t+1)}) \end{cases} \tag{8.29}$$

where $t \geq 0$ and $\mu > 0$. The minimizing solution $\boldsymbol{\alpha}^{(t+1)}$ is simply determined as

$$\boldsymbol{\alpha}^{(t+1)} \leftarrow (\mathbf{Q} + \mu\mathbf{I})^{-1}(\mathbf{p} + \mu(\mathbf{u}^{(t)} + \mathbf{d}^{(t)})), \tag{8.30}$$

where $\mathbf{I}$ is the identity matrix. The minimizing solution $\mathbf{u}^{(t+1)}$ is the soft threshold [24],

$$\mathbf{u}^{(t+1)} \leftarrow \text{soft}(\boldsymbol{\alpha}^{(t+1)} - \mathbf{d}^{(t)}, \lambda/\mu), \tag{8.31}$$

where $\text{soft}(\cdot, \tau)$ denotes the component-wise application of the soft-threshold function $y \leftarrow \text{sign}(y)\max\{|y| - \tau, 0\}$.

The optimization algorithm for KSRC is summarized in Algorithm 8.3.

Algorithm 3 Spatial-Spectral Kernel Sparse Representation Classification

Input: A training dictionary $\mathbf{D} \in \mathbb{R}^{B\times N}$, and a test sample $\mathbf{x} \in \mathbb{R}^B$

1) Select the Mercer kernel $\mathbf{k}_{NF}$ (or others) and its parameters.
2) Compute the matrix $\mathbf{Q}$, and the vector $\mathbf{p}$.
3) Set $t = 0$, choose $\mu > 0$, $\mathbf{s}^{(0)}$, $\mathbf{u}^{(0)}$, $\mathbf{d}^{(0)}$.
4) repeat.
5) Compute $\mathbf{s}^{(t+1)}$, $\mathbf{u}^{(t+1)}$ and $\mathbf{d}^{(t+1)}$ using (29)
6) $t \leftarrow t+1$
7) until some stopping criterion is satisfied.
8) compute the M residuals $r_i(\mathbf{x}) = \delta_i^T(\mathbf{s})\mathbf{Q}\delta_i(\mathbf{s}) - 2\delta_i^T(\mathbf{s})\mathbf{p}, i = 1, 2, ..., M$.

Output: The estimated label of $\mathbf{x}$ according to (23)

8.4 A Case Study of Hyperspectral Image Sparse Representation Classification Based on Adaptive Spatial Context

8.4.1 Model and Algorithm

In model (8.5), pixels in a fixed window centered at the test pixel are selected to be simultaneously sparse represented. All pixels in the fixed window have the same correlation with the center pixel. However, this condition does not always hold, especially for pixels located on the edge which can be seen as class boundary. It is obvious that pixels on the same side of the edge will have stronger correlation. Since different pixels have different spatial context, the definition of local structure for the adaptive spatial context is essential to HSI classification.

In the field of image recovery, steering kernel (SK) [25] is a popular local method, which can effectively express the adaptive local structure. This method starts with making an initial estimate of the image gradients using a gradient estimator, and then uses the estimate to measure the dominant orientation of the local gradients in the image [26]. The obtained orientation information is then used to adaptively "steer" the local kernel, resulting in elongated, elliptical contours spread along the directions of the local edge structure.

Taking into consideration that HSI generally contains hundreds of sub-images, a high-dimensional steering kernel (HDSK) [27] is defined where the gradient estimator contains every sub-image's gradients. The gradients in vertical and horizontal directions are written as follows:

$$(\nabla \mathbf{x}_i^v, \nabla \mathbf{x}_i^h) = \left(\frac{\left\|\mathbf{x}_i - \mathbf{x}_{i+1}^v\right\|_1}{B}, \frac{\left\|\mathbf{x}_i - \mathbf{x}_{i+1}^h\right\|_1}{B}\right) \tag{8.32}$$

where $\mathbf{x}_{i+1}^v$ and $\mathbf{x}_{i+1}^h$ represent the neighboring pixels of $\mathbf{x}_i$ in vertical and horizontal directions. HDSK for pixel $\mathbf{x}_i$ is defined as

$$w_{ij} = \frac{\sqrt{\det(\mathbf{C}_i)}}{2\pi h^2} \exp\left(-\frac{(\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{C}_i (\mathbf{e}_i - \mathbf{e}_j)}{2h^2}\right) \tag{8.33}$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ represent the coordinates of pixel $\mathbf{x}_i$ and pixel $\mathbf{x}_j$, respectively, h is the smoothing parameter used for controlling the supporting range of the steering kernel, and $\mathbf{C}_i$ is the symmetric gradient covariance in vertical and horizontal directions in a $M \times M$ window centered at $\mathbf{x}_i$. A naïve estimate of this covariance matrix can be obtained by $\mathbf{C}_i = \mathbf{J}_i^T \mathbf{J}_i$, where

$$\mathbf{J}_i = \begin{bmatrix} \nabla \mathbf{x}_1^v & \nabla \mathbf{x}_1^h \\ \vdots & \vdots \\ \nabla \mathbf{x}_{M\times M}^v & \nabla \mathbf{x}_{M\times M}^h \end{bmatrix} \tag{8.34}$$

Here, $\mathbf{x}_1, \cdots, \mathbf{x}_{M\times M}$ are the $M \times M$ neighboring pixels in the local window centered at $\mathbf{x}_i$. The resulting w_{ij} can be explained as the correlation between pixels $\mathbf{x}_i$ and $\mathbf{x}_j$. Since a large weight in steering kernel mean two pixels have strong correlation, HDSK could be an effective way to represent the local structure. For example, Fig. 8.1 shows the 10-th band image in the University of Pavia HSI and the calculated HDSKs for different pixels. It can be observed that when pixels are in a homogeneous region, the shape of HDSK is cycles without any directional preference. When the pixels are in the intersection or the boundary of different classes, the shape of HDSKs is oval and exhibits clear directional preference. The direction of the long axis of the oval indicates that similar pixels may appear in this direction.

Once having determined the local structure of a test pixel x_i using (8.20), we select P pixels whose weights are larger than the others. These pixels can be stacked as $\mathbf{X}^P = [\mathbf{x}_{i1}\, \mathbf{x}_{i2} \ldots \mathbf{x}_{iP}] \in \mathbb{R}^{B\times P}$, and $\mathbf{w}^P = [w_1\, w_2 \ldots w_P]^T$ is the corresponding

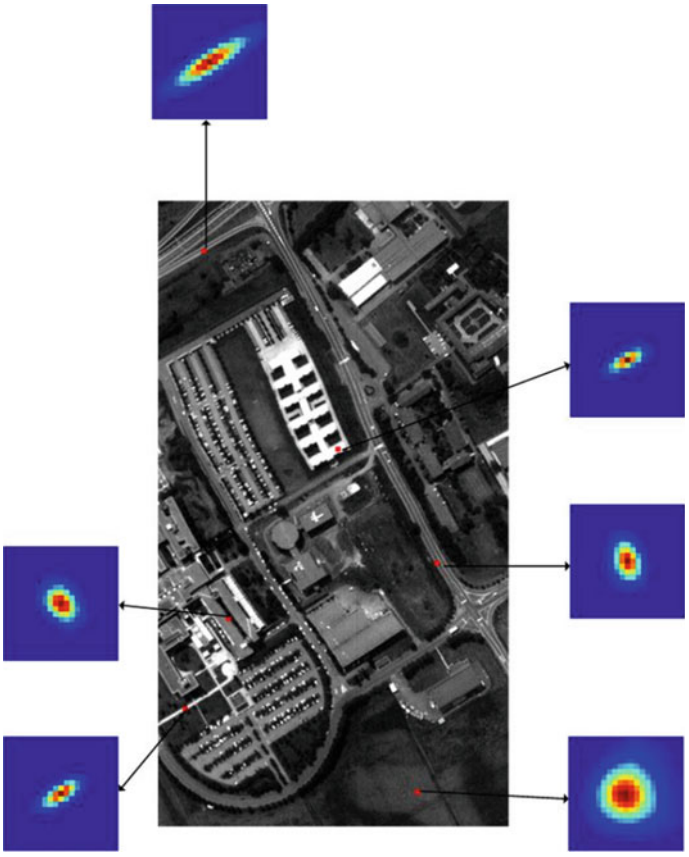

Fig. 8.1 Examples of HDSKs

weight vector. It is believed that these selected P pixels have more compact inner patterns than those in a fixed window do. The adaptive spatial contextual information is introduced by the following problem:

$$\begin{aligned} \mathbf{S}^P &= \arg\min_{\mathbf{S}^P} \left\| \mathbf{X}^P - \mathbf{D}\mathbf{S}^P \right\|_F \\ &\text{s.t} \quad \left\| \mathbf{S}^P \right\|_{\text{row},0} \le K_0 \end{aligned} \tag{8.35}$$

Algorithm 4 ASC-SOMP Algorithm

Input: Dictionary $\mathbf{D} = [\mathbf{d}_1\ \mathbf{d}_2\ ...\ \mathbf{d}_N]$, test samples $\{\mathbf{x}_i\}_{i=1,2,...,L}$, window size M, and the number of selected pixels P

1) **Pre-calculate**: First compute gradients as in (34), then compute the gradient covariance $\mathbf{C}_i, \{i = 1, 2, ..., L\}$.

While $i \le L$ **do**

2) Compute steering kernel of x_i according to (33)
3) Sort the pixels in the window as their weights from large to small, select the first P pixels and stack them as and record their weights
4) Initialization, residual $\mathbf{R}_0 = \mathbf{X}^P$, index set $\mathbf{\Lambda}_0 = \varnothing$, iteration counter $k = 1$

   While stopping criterion has not been met **do**

   a) Find the index of the atom that best approximates all residuals, $\lambda_k = \arg\max_{i=1,...,N} \| \mathbf{R}_{k-1}^T \mathbf{d}_i \|_\infty$

   b) Update the index set $\mathbf{\Lambda}_k = \mathbf{\Lambda}_{k-1} \cup \{\lambda_k\}$

   c) Compute $\mathbf{M}_k = (\mathbf{D}_{\Lambda_k}^T \mathbf{D}_{\Lambda_k})^{-1} \mathbf{D}_{\Lambda_k}^T \mathbf{X}^P \in \mathbb{R}^{k \times P}$, $\mathbf{D}_{\Lambda_k} \in \mathbb{R}^{B \times k}$ consists of the k atoms in $\mathbf{D}$ indexed in $\mathbf{\Lambda}_k$

   d) Determine the residual

   e) $k \leftarrow k + 1$

   Output: the sparse representation , its nonzero rows indexed by $\mathbf{\Lambda}$ which are the K rows of the matrix $(\mathbf{D}_\Lambda^T \mathbf{D}_\Lambda)^{-1} \mathbf{D}_\Lambda^T \mathbf{X}^P$ where $\mathbf{\Lambda} = \mathbf{\Lambda}_{k-1}$

5) Determine the label of $\mathbf{X}_i$ according to (36)

end while

Once the coefficient matrix $\mathbf{S}^P$ is obtained, a new classifier is designed based on the HDSK. As the weights in the HDSK reflect the influence of neighboring pixels on the test pixel, the original decision rule (8.6) is replaced by

$$class(\mathbf{x}_i) = \arg\min_{j=1,...,M} \left\| (\mathbf{X}^P - \mathbf{D}_j \mathbf{S}_j^P) \mathbf{w}^P \right\|_2 \tag{8.36}$$

The joint sparse HSI classification method based on adaptive spatial context is named adaptive spatial context SOMP (ASC-SOMP), of which the general flow is summarized in Algorithm 8.4.

8.4.2 Experimental Results and Discussion

This section uses two real hyperspectral datasets to verify the effectiveness of ASC-SOMP algorithm. For each image, the pixel-wise SVM, SVM with composite kernel (SVM-CK) [19], OMP [14], SOMP [14] are compared with ASC-SOMP both visually and quantitatively. We select Gaussian radial basis function (RBF) for the pixel-wise SVM and SVM-CK methods, since RBF has proved its capability handling complex nonlinear class distributions. The parameters in SVM-based methods are obtained by fivefold cross-validation. For methods involved with composite kernels, the spatial kernels were built by using the mean and standard deviation of the neighboring pixels in a window per spectral channel. Each value of the results is obtained after performing ten Monte Carlo runs.

The training and test samples are randomly selected from the available ground truth map. The classification accuracy is evaluated by the overall accuracy (OA) which is defined as the ratio of the number of accurately classified samples to the number of test samples, the coefficient of agreement (κ) which is the ratio of the amount of corrected agreement to the amount of expected agreement, and the average accuracy (AA). To be specific, OA is calculated by

$$OA = \sum_{i=1}^{C} \mathbf{E}_{ij}/N \tag{8.37}$$

where N is the total number of samples, and $\mathbf{E}_{ij}$ represents the number of samples in class i which are miss-classified to class j.

AA is calculated by

$$AA = \left(\sum_{i=1}^{C} \left(\mathbf{E}_{ij} \Big/ \sum_{j=1}^{C} \mathbf{E}_{ij} \right) \right) \Big/ C \tag{8.38}$$

The κ statistic is calculated by weighting the measured accuracies. This metric incorporates the diagonal and off-diagonal entries of the confusion matrix and is given by

$$\kappa = \left(N \left(\sum_{i=1}^{C} \mathbf{E}_{ii} \right) - \sum_{i=1}^{C} \left(\sum_{j=1}^{C} \mathbf{E}_{ij} \sum_{j=1}^{C} \mathbf{E}_{ji} \right) \right) \Big/ \left(N^2 - \sum_{i=1}^{C} \left(\sum_{j=1}^{C} \mathbf{E}_{ij} \sum_{j=1}^{C} \mathbf{E}_{ji} \right) \right) \tag{8.39}$$

8.4.2.1 Hyperspectral Dataset of AVIRIS Indian Pines

The Indian Pines image contains 145 × 145 pixels and 200 spectral reflectance bands, among which 24 water absorption bands have been removed. The ground truth contains 16 land cover classes and a total of 10366 labeled pixels. We randomly choose 10% of labeled samples for training, and use the rest 90% for testing. The false color image and ground truth are shown in Fig. 8.2a, b.

The parameters for ASC-SOMP algorithm are set to $P = 120$, $K_0 = 25$, $h = 25$, and $M = 21$. The window size of SOMP algorithm is empirically set to 9×9. The classification results, in terms of overall accuracy (OA), average accuracy (AA), κ

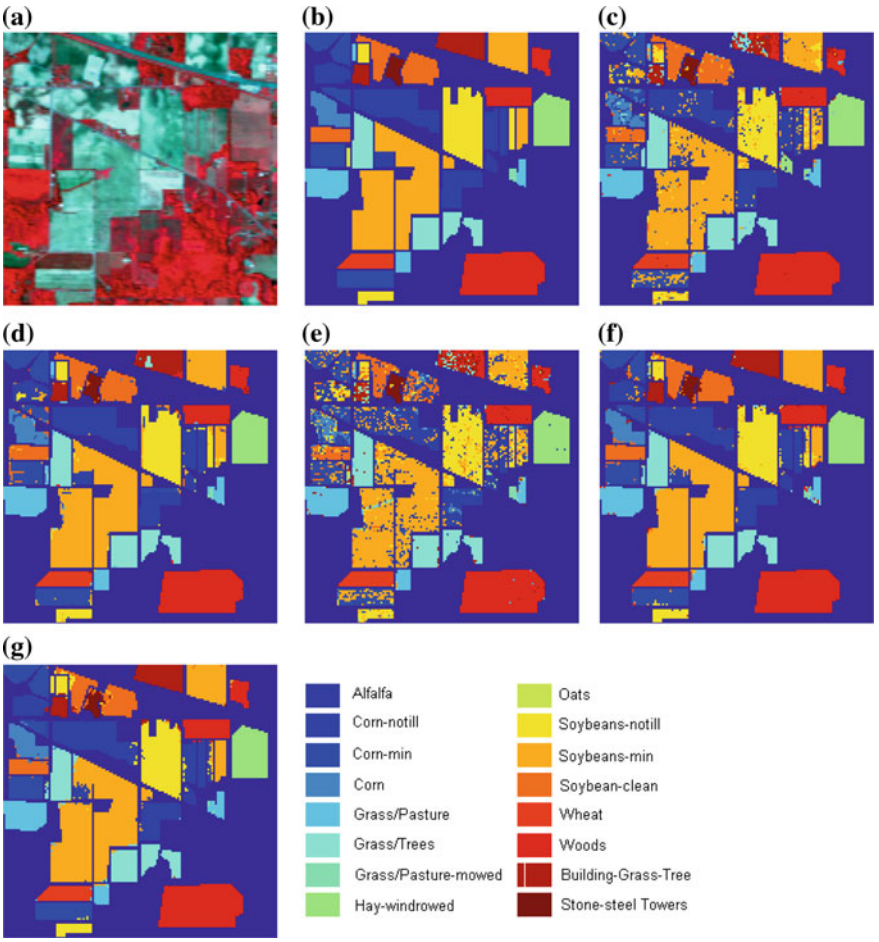

Fig. 8.2 Classification results of Indian Pines image, **a** false color image (R, 57 G, 27 B, 17), **b** ground truth, **c** SVM (OA, 85.24%), **d** SVM-CK (OA, 93.60%), **e** OMP (OA, 75.67%), **f** SOMP (OA, 95.28%), **g** ASC-SOMP (96.79%)

Table 8.1 Classification accuracy (%) For the Indian Pines image on the test set

Class	#train samples	#test samples	SVM	SVM-CK	OMP	SOMP	ASC-SOMP
Alfalfa	6	48	31.25	62.08	65.62	85.42	**91.67**
Corn-no till	144	1290	82.80	92.71	64.58	94.88	**95.74**
Corn-min till	84	750	75.01	91.29	61.36	94.93	**96.27**
Corn	24	210	64.42	79.71	44.80	91.43	**95.24**
Grass/Pasture	50	447	93.08	**95.59**	91.09	89.49	93.96
Grass/Trees	75	672	95.46	98.09	94.04	98.51	**99.70**
Grass/Pasture-mowed	3	23	4.35	49.56	84.78	**91.30**	56.20
Hay-windrowed	49	440	98.81	98.47	97.97	95.55	**100**
Oats	2	18	0.00	0.00	**43.33**	0.00	22.22
Soybeans-no till	97	871	76.76	89.97	70.76	89.44	**92.31**
Soybeans-min till	247	2221	87.76	96.13	76.22	97.34	**98.42**
Soybean-clean till	62	552	85.25	89.49	57.91	88.22	**92.39**
Wheat	22	190	98.53	96.63	97.73	**100**	99.47
Woods	130	1164	97.62	98.04	94.09	99.14	**100**
Building-Grass-Trees-Drives	38	342	56.11	89.29	44.26	99.12	**100**
Stone-steel Towers	10	85	81.17	88.11	90.47	**96.47**	95.29
OA (%)			85.24	93.60	75.67	95.28	**96.79**
AA (%)			70.52	**92.70**	72.22	88.45	89.33
κ			83.11	82.20	73.69	94.60	**96.34**

statistic, and class individual accuracies, are shown in Table 8.1. The final maps are illustrated in Fig. 8.2c–g. It can be observed that ASC-SOMP algorithm achieves the highest OA of 96.79%, which is 1.5% higher than the second-highest OA. Classification results using different percentages of labeled samples for training are shown in Fig. 8.3. In this figure and the following, error bars indicate the standard deviation by random sampling. From Fig. 8.3, both numerical and statistical differences can be observed.

Next, we demonstrate the impact of the number of selected neighboring pixels P upon the performance of ASC-SOMP algorithm. We use 10% of data in each class as training samples. The number of selected pixels P ranges from $P = 80$ to $P = 140$, and the sparsity level K_0 ranges from $K_0 = 5$ to $K_0 = 45$. The plots of overall accuracy evaluated on the entire test set are shown in Fig. 8.4. When $K_0 \geq 25$ and $P \geq 110$, a relatively high classification accuracy can be achieved. Compared with SOMP algorithm, ASC-SOMP leads to the same optimal K_0 value, but the optimal P value is significantly larger. As pixels are selected according to their spatial correlation to the center pixel, it is reasonable to select more pixels that can be sparsely represented simultaneously.

To investigate the effect of the introduced adaptive spatial context, we compare ASC-SOMP with traditional joint sparsity method in detail. It is obvious that SOMP is not able to identify any samples belonging to oats class. This observation is because oat pixels cover a very narrow region of size 10×2 located in the middle-left of

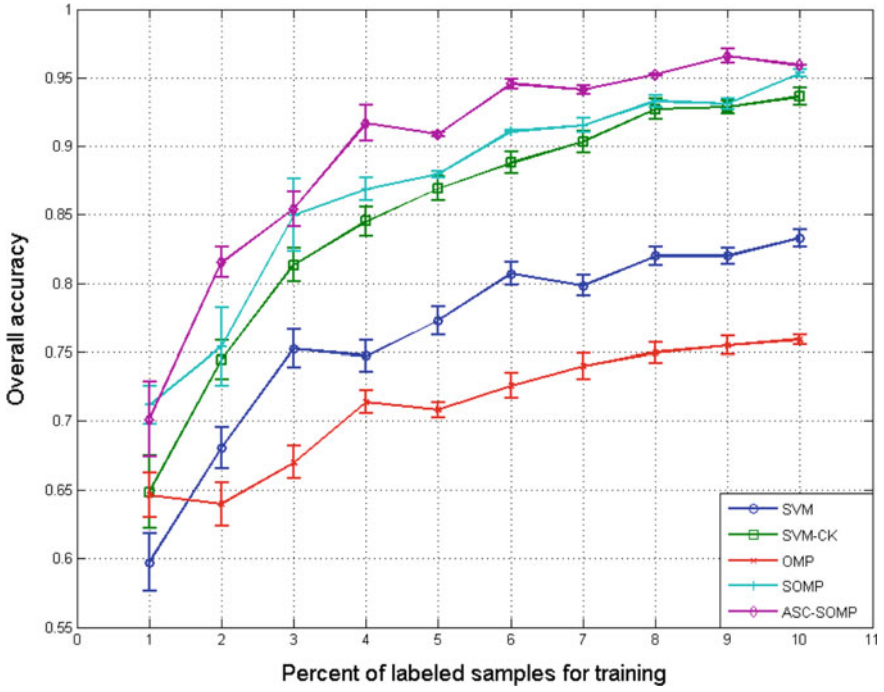

Fig. 8.3 The overall accuracy of Indian Pines for different numbers of training samples

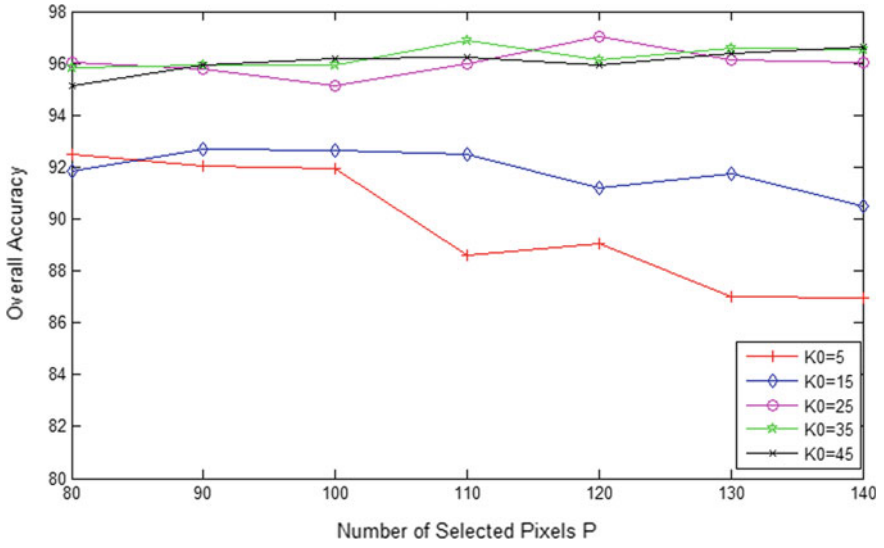

Fig. 8.4 Effects of the sparsity level K_0 and number of selected pixels P for Indian Pines

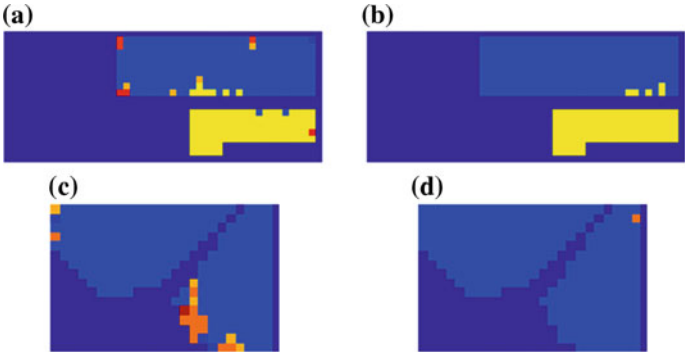

Fig. 8.5 Amplified map in two regions, **a** and **c** are results of SOMP, **b** and **d** are results of ASC-SOMP

the image. In SOMP, the optimal 9×9 local window centered at each oat pixel is dominated by pixels belonging to the other two adjacent classes. In contrast, ASC-SOMP achieves a 22.22% classification accuracy for oat class. By introducing adaptive spatial context, pixels distributed along the direction of the narrow region are selected as they have large correlation with the test pixel. On the other hand, pixels belonging to the other two classes whose weights are small have less impact upon our decision rule. Thus, better results can be obtained. However, the classification accuracy for oat class is still very low, because the total number of oat class is much less than the selected pixels to be sparsely represented simultaneously, and most of the selected pixels do not belong to oat class oat.

Taking into consideration that the effect of adaptive spatial context is clearer in the class boundary, more attention should be paid on the edge. We amplify the region of SOMP result and the region of ASC-SOMP result to verify the effect of adaptive spatial context. Figure 8.5 shows that our classification result has less wrong-classified pixels in the class boundary, demonstrating the advantages of the adaptive spatial context.

8.4.2.2 Hyperspectral Dataset of ROSIS Pavia University

The second hyperspectral data set was collected by the ROSIS optical sensor over the urban area of the Pavia University, Italy. The image size in pixels is 610×340, with a very high spatial resolution of 1.3 m per pixel. The number of data channels in the acquired image is 103 (with the spectral range from 0.43 to 0.86 μm). Nine classes of interest were considered, including tree, asphalt, bitumen, gravel, metal sheet, shadow, bricks, meadow, and soil. Figure 8.6a, b shows the three-band false color image and the ground truth map, respectively. We randomly sampled 60 pixels for each class as the training samples and use the remainder as test samples. The optimal parameter settings for the ASC-SOMP method are $P = 100$ and $K_0 = 5$. In SOMP, the window size was set to 9×9, and the sparsity level was set to $K_0 =$

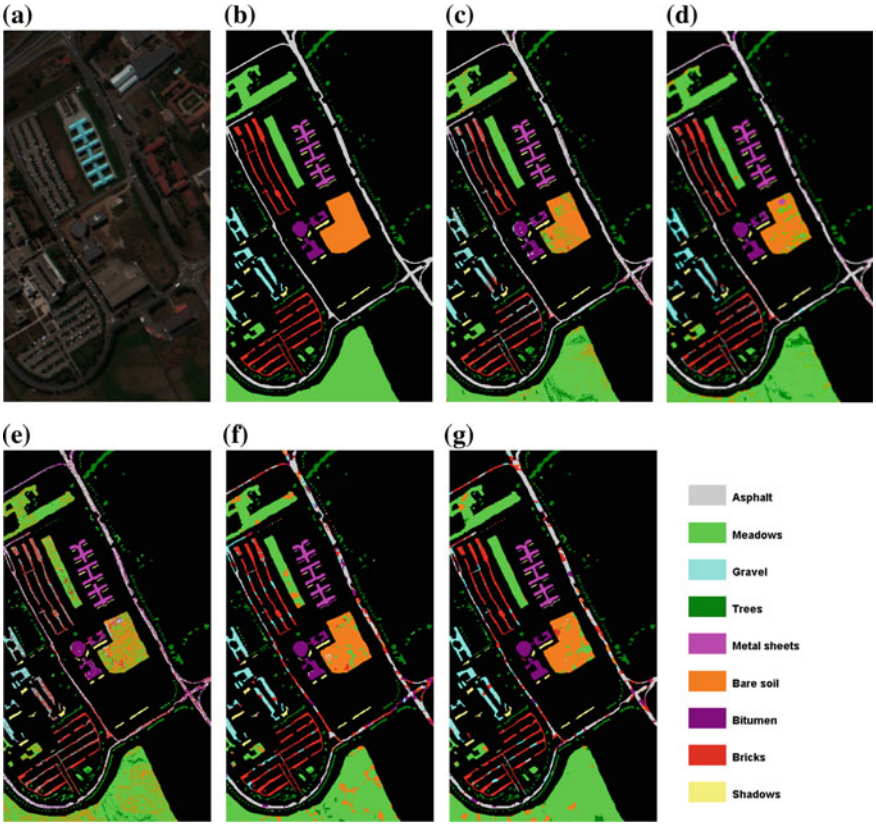

Fig. 8.6 Classification results of University of Pavia image, **a** false color image (R, 57 G, 27 B, 17), **b** ground truth, **c** SVM (OA, 84.26%), **d** SVM-CK (OA, 91.60%), **e** OMP (OA, 71.12%), **f** SOMP (OA, 83.60%), **g** ASC-SOMP (85.07%)

15. We set $h = 25$ and $M = 21$ as in the previous set of experiments. The final classification maps are illustrated in Fig. 8.6c–g. The classification results, in term of overall accuracy (OA), average accuracy (AA), k statistic, and class individual accuracies, are provided in Table 8.2. The ASC-SOMP method outperforms other methods except for SVM-CK. SVM-CK achieves the best results since it is a spectral–spatial nonlinear kernel method. Figure 8.7 illustrates the classification accuracies by using different number of training samples. This result justifies the robustness of ASC-SOMP method. Figure 8.8 shows the performance in terms of overall accuracy with different numbers of selected pixels P at sparsity level $K_0 = 5$ and $K_0 = 10$, respectively. The number of selected pixels P ranges from 50 to 110. Figure 8.8 also shows that the overall accuracy improves as P value increases. This conclusion isconsistent with the conclusion drawn on the dataset of AVIRIS Indian Pines.

Table 8.2 Classification accuracy (%) for University of Pavia on the test set

Class	#train samples	#test samples	SVM	SVM-CK	OMP	SOMP	ASC-SOMP
Asphalt	60	6571	77.92	**88.98**	57.62	47.87	52.01
Bare soil	60	18589	81.67	**93.09**	71.96	91.59	91.36
Bitumen	60	2039	82.13	87.65	65.85	92.15	**93.52**
Bricks	60	3004	95.33	**97.52**	89.83	89.34	95.97
Gravel	60	1285	99.15	99.47	99.75	**100**	99.24
Meadows	60	4969	87.92	**89.66**	63.38	87.74	86.76
Metal sheets	60	1270	93.59	94.55	85.85	95.98	**97.92**
Shadows	60	3622	83.70	83.03	68.30	84.40	**87.00**
Trees	60	887	**99.96**	99.14	94.61	73.95	85.49
OA (%)			84.26	**91.60**	71.12	83.60	85.07
AA (%)			79.75	**88.95**	63.17	78.56	80.50
κ			89.04	**92.57**	77.46	84.78	87.70

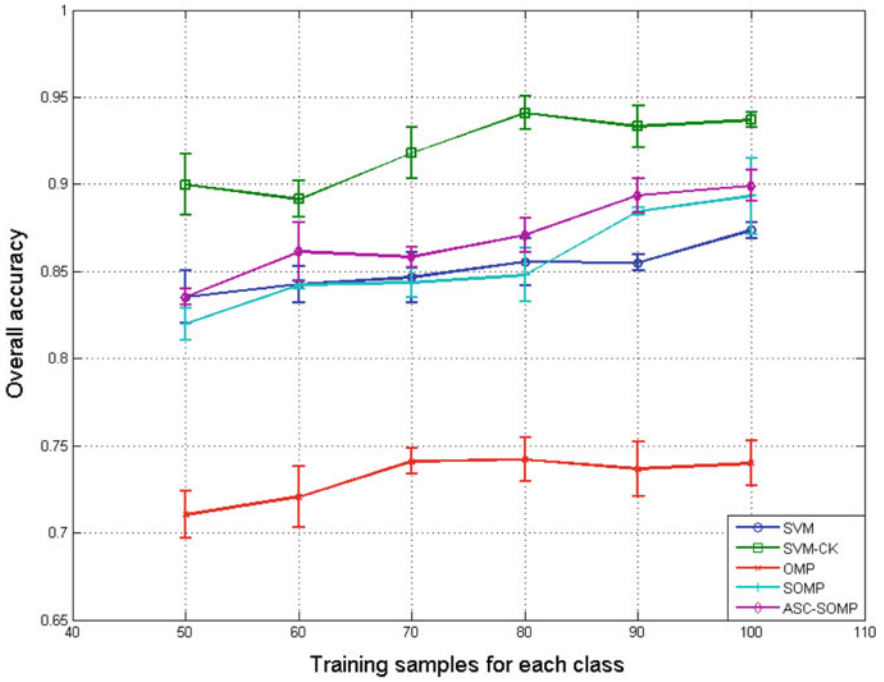

Fig. 8.7 The overall accuracy of University of Pavia for different numbers of training samples

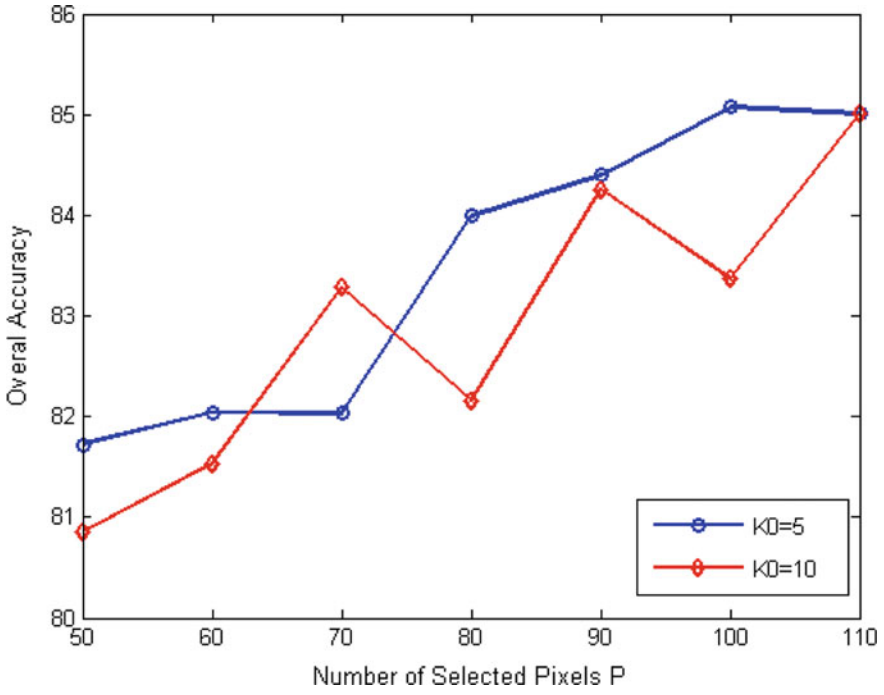

Fig. 8.8 Effect of different numbers of selected pixels P for University of Pavia

8.4.2.3 Discussion

The ASC-SOMP method and the nonlocal-weighted version of SOMP (NLW-JSRC) [28] both were developed for improving the original SOMP method. The weights for the neighboring pixels are calculated in both methods. We compared our method with NLW-JSRC. All experiments were performed using the same experimental setup as in the work of NLW-JSRC, where 9% of the labeled data are randomly sampled as the training samples, and the remainder of the data are used as test samples. Tables 8.3 and 8.4 present the comparisons of results by both methods. We can observe that the ASC-SOMP method outperforms the NLW-JSRC method, indicating that the steering kernel can better describe the spatial context than the nonlocal weights can.

h and M are two important parameters that control the supporting range of the steering kernel and determine the contributions of the selected pixels to the classification of test pixel. We further evaluate the classification accuracy on the two images for different h and M values. We use the same training samples as in previous experiments. h ranges from 1 to 45, and the window size M ranges from 13×13 to 29×29. Figure 8.9a indicates that the classification accuracy is relatively high when h is between 10 and 35. If h is too small, the variance of the weights is large, resulting in the outcome that a few pixels with large weights dominate the classification decision. If h is too large, on the other hand, the gap between different pixels' weights

Table 8.3 Numerical comparison with NLW-JSRC for Indian Pines

Class	#train samples	#test samples	NLW-JSRC	ASC-SOMP
1	6	46	**95.00**	81.25
2	129	1299	92.99	**94.25**
3	83	747	87.82	**95.33**
4	24	213	85.45	**96.66**
5	48	435	93.33	**93.76**
6	73	657	**100**	99.25
7	5	23	73.91	**95.23**
8	48	430	**100**	**100**
9	4	16	31.25	**50.00**
10	97	875	90.51	**92.30**
11	196	2259	96.90	**98.85**
12	59	534	**96.82**	86.12
13	21	184	**100**	**100**
14	114	1151	**99.91**	**99.91**
15	39	347	96.25	**98.82**
16	12	81	97.53	100
OA (%)			95.19	**96.35**
κ			94.50	**95.83**

Table 8.4 Numerical comparison with NLW-JSRC for University of Pavia

Class	#train samples	#test samples	NLW-JSRC	ASC-SOMP
1	579	6034	87.67	**96.56**
2	932	17717	98.91	**99.90**
3	189	1910	79.42	**98.69**
4	276	2788	92.90	**96.77**
5	269	1076	**100**	**100**
6	453	4576	77.69	**99.08**
7	266	1064	96.43	**99.62**
8	331	3351	85.68	**97.01**
9	189	758	**98.81**	88.39
OA (%)			92.98	**98.54**
κ			90.46	**98.03**

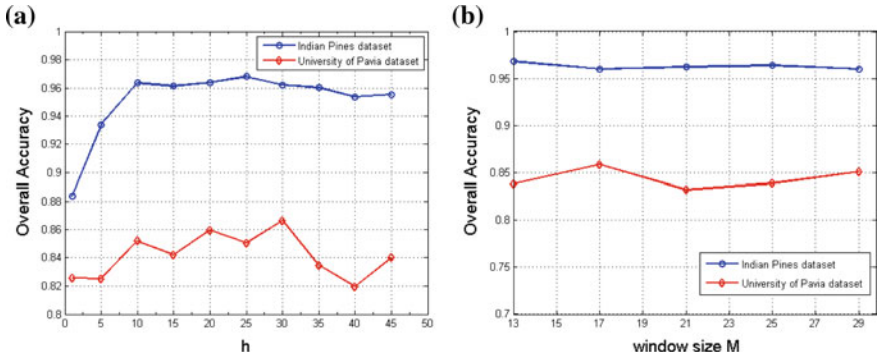

Fig. 8.9 **a** The classification accuracy for different h. **b** The classification accuracy for different window size M

is not clear enough, as the adaptive spatial context information is not used as much as possible. We can also observe from Fig. 8.9b that the classification accuracy is robust to the window size M as long as there are enough pixels to be selected.

8.5 Conclusions

Sparsity-based methods play an important role in HSI classification. Taking into consideration that the spectrum of a pixel lies in the low-dimensional subspace spanned by the training samples of the same class, sparse representation classification (SRC) is widely employed in HSI classification. Many advanced SRC models are presented to improve the classification accuracy, based on the structural sparsity priors, spectral–spatial information, kernel tricks, etc. This chapter reviews the structural sparsity priors and explains how the spectral–spatial information of HSI is incorporated into the SRC method. More specifically, a case study of HSI sparse representation classification based on adaptive spatial context is presented in detail. Experimental results demonstrate that, by combining SRC and adaptive spectral–spatial information, the performances of SRC can be significantly improved. Future work can be directed toward tensor sparse representation which can take full advantage of the high-order correlation in HSI and can preserve the spectral–spatial structure of HSI.

References

1. Donoho DL (2006) Compressed sensing. IEEE Trans Inf Theory 52(4):1289–1306
2. Baraniuk RG (2007) Compressive sensing. IEEE Signal Process Mag 24(4):118–121
3. Candès EJ, Romberg J, Tao T (2006) Robust uncertainty principles, Exact signal reconstruction from highly incomplete frequency information. IEEE Trans Inf Theory 52(2):489–509

4. Sun L, Zebin W, L Jianjun, X Liang, Wei Z (2015) Supervised spectral-spatial hyperspectral image classification with weighted markov random fields. IEEE Trans Geosci Remote Sens 53(3):1490–1503
5. Li J, Bioucas-Dias JM, Plaza A (2010) Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. IEEE Trans Geosci Remote Sens 48(11):4085–4098
6. Melgani F, Bruzzone L (2004) Classification of hyperspectral remote sensing images with support vector machines. IEEE Trans Geosci Remote Sens 42(8):1778–1790
7. Pan B, Shi Z, Xu X (2018) MugNet, deep learning for hyperspectral image classification using limited samples. ISPRS J Photogramm Remote Sens 145:108–119
8. Ma L, Crawford MM, Tian J (2010) Local manifold learning-based k-nearest-neighbor for hyperspectral image classification. IEEE Trans Geosci Remote Sens 48(11):4099–4109
9. Aharon M, Elad M, Bruckstein A (2006) K-SVD, an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans Signal Process 54(11):4311
10. Wright J, Yang AY, Ganesh A et al (2009) Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intell 31(2):210–227
11. Wagner A, Wright J, Ganesh A et al (2012) Toward a practical face recognition system, robust alignment and illumination by sparse representation. IEEE Trans Pattern Anal Mach Intell 34(2):372–386
12. Gemmeke JF, Virtanen T, Hurmalainen A (2011) Exemplar-based sparse representations for noise robust automatic speech recognition. IEEE Trans Audio Speech Lang Process 19(7):2067–2080
13. Yang J, Wright J, Huang TS et al (2010) Image super-resolution via sparse representation. IEEE Trans Image Process 19(11):2861–2873
14. Chen Y, Nasrabadi NM, Tran TD (2011) Hyperspectral image classification using dictionary-based sparse representation. IEEE Trans Geosci Remote Sens 49(10):3973–3985
15. Sun X, Qu Q, Nasrabadi NM et al (2014) Structured priors for sparse-representation-based hyperspectral image classification. IEEE Geosci Remote Sens Lett 11(7):1235–1239
16. Tu B, Zhang X, Kang X et al (2018) Hyperspectral image classification via fusing correlation coefficient and joint sparse representation. IEEE Geosci Remote Sens Lett 15(3):340–344
17. Liu J, Wu Z, Xiao Z, Yang J (2017) Hyperspectral image classification via kernel fully constrained least squares. In: 2017 IEEE international geoscience and remote sensing symposium, Fort Worth, 23–28 July 2017, pp 2219–2222
18. Aizerman A, Braverman E, Rozoner L (1964) Theoretical foundations of the potential function method in pattern recognition learning. Autom Remote Control 25:821–837
19. Camps-Valls G, Gomez-Chova L, Muñoz-Mari' J, Vila-Francés J, Calpe-Maravilla J (2006) Composite kernels for hyperspectral image classification. IEEE Geosci Remote Sens Lett 3(1):93–97
20. Liu J, Wu Z, Wei Z et al (2013) Spatial-spectral kernel sparse representation for hyperspectral image classification. IEEE J Sel Top Appl Earth Obs Remote Sens 6(6):2462–2471
21. Tuia D, Camps-Valls G (2001) Urban image classification with semisupervised multiscale cluster kernels. IEEE J Sel Top Appl Earth Obs Remote Sens 4(1):65–74
22. Gomez-Chova L, Camps-Valls G, Bruzzone L, Calpe-Maravilla J (2010) Mean map kernel methods for semisupervised cloud classification. IEEE Trans Geosci Remote Sens 48(1):207–220
23. Bioucas-Dias J, Figueiredo M (2010) Alternating direction algorithms for constrained sparse regression, Application to hyperspectral unmixing. In: Proceedings of WHISPERS, Reykjavik, Iceland, June 2010, pp 1–4. IEEE
24. Combettes P et al (2006) Signal recovery by proximal forward-backward splitting. Multiscale Model Simul 4(4):1168–1200
25. Takeda H, Farsiu S, Milanfar P (2007) Kernel regression for image processing and reconstruction. IEEE Trans Image Process 16(2):349–366
26. Feng X, Milanfar P (2002) Multiscale principal components analysis for image local orientation estimation. In: 36th Asilomar conference signals, systems and computers, Pacific Grove, CA

27. Xu Y, Wu Z, Wei ZH (2014) Joint sparse hyperspectral image classification based on adaptive spatial context. J Appl Remote Sens 8(1):083552
28. Zhang H et al (2013) A nonlocal weighted joint sparse representation classification method for hyperspectral imagery. IEEE J Sel Top Appl Earth Obs Remote Sens 7(6):2056–2065
29. Takeda H, Farsiu S, Milanfar P (2007) Kernel regression for image processing and reconstruction. IEEE Trans Image Process 16:349–366