remote sensing

MDPI

Article

Spectral-Spatial Hyperspectral Image Classification via Robust Low-Rank Feature Extraction and Markov Random Field

Xiangyong Cao, Zongben Xu and Deyu Meng *

School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China; caoxiangyong@mail.xjtu.edu.cn (X.C.); zbxu@mail.xjtu.edu.cn (Z.X.)

* Correspondence: dymeng@mail.xjtu.edu.cn; Tel.: +86-181-0924-1418

Received: 8 May 2019; Accepted: 26 June 2019; Published: 2 July 2019

Abstract: In this paper, a new supervised classification algorithm which simultaneously considers spectral and spatial information of a hyperspectral image (HSI) is proposed. Since HSI always contains complex noise (such as mixture of Gaussian and sparse noise), the quality of the extracted feature inclines to be decreased. To tackle this issue, we utilize the low-rank property of local three-dimensional, patch and adopt complex noise strategy to model the noise embedded in each local patch. Specifically, we firstly use the mixture of Gaussian (MoG) based low-rank matrix factorization (LRMF) method to simultaneously extract the feature and remove noise from each local matrix unfolded from the local patch. Then, a classification map is obtained by applying some classifier to the extracted low-rank feature. Finally, the classification map is processed by Markov random field (MRF) in order to further utilize the smoothness property of the labels. To ease experimental comparison for different HSI classification methods, we built an open package to make the comparison fairly and efficiently. By using this package, the proposed classification method is verified to obtain better performance compared with other state-of-the-art methods.

Keywords: hyperspectral image classification; low-rank matrix factorization; Markov random field

1. Introduction

Hyperspectral remote sensors capture reflection light in several hundred narrow frequencies, which cover visible, near-infrared and shortwave infrared bands. Thus, the hyperspectral image (HSI) data obtained by the sensors contain rich spectral information and thus produce more accurate measures, which make it possible to distinguish the material of each pixel. Supervised classification has been an important tool in many HSI applications, such as land-use mapping [1–3], land-cover mapping [3], forest inventory [4], and urban-area monitoring [5]. Specifically, HSI classification aims to classify each spectral pixel into different classes.

In the last few decades, a variety of methods have been proposed for the task of HSI classification. Since HSI classification is usually conducted in the space of feature, it is thus very crucial to design effective feature representation method for this task. According to the methods of extracting features, the existing HSI classification approaches are roughly divided into three classes: unsupervised feature learning methods, supervised feature learning methods and semi-supervised methods. Due to the limited space, we mainly review the unsupervised methods and deep learning based method, which is a representative method of the supervised methods.

An unsupervised feature learning method plays an important role in the field of HSI classification, and the extracted feature by unsupervised feature learning methods is obtained by making use of two types of information from the raw HSI data, namely spectral information and spatial

information. While many early works utilize the spectral information only [6–10], spectral-spatial based methods have become a recent trend to extract features by simultaneously considering spatial and spectral knowledge [11–18]. Representative unsupervised feature learning methods used for HSI classification include principle component analysis (PCA), manifold learning, dictionary-based sparse representation, subspace projection, low-rank modeling and wavelet transform. Specifically, PCA is used to reduce the dimension of the HSI data [19]. Manifold learning is applied to dimensional reduction and feature extraction in HSI classification [20]. Dictionary-based sparse representation method is based on the assumption that each pixel vector in HSI can be expressed as a linear combination of a few training pixels [21–25]. Additionally, subspace projection methods are proposed [26–32] to address the issue of highly mixed pixels in the HSI. Wavelet transform is another popular tool to extract HSI features, such as 3-dimensional discrete wavelet transform (3DDWT) [17,33] and 3-dimensional Gabor wavelet (3DGabor) [34]. To be precise, a 3DDWT method can fuse spatial-contextual information into the spectral information for each spectral pixel and thus can extract spatial-spectral features. The 3DGabor method takes advantage of some Gabor wavelets to extract features in the joint spectral-spatial domains. Low-rank modeling methods, such as low-rank matrix factorization [35,36], robust principle component analysis [37] and low-rank representation subspace clustering [38], show another recent trend for unsupervised HSI classification [15,16]. These methods can not only discover the low-dimensional structure of every spectral pixel, but also incorporate the spatial-contextual information for each spectral pixel [15,16]. Recently, deep learning based methods have exhibited great feature representation ability for many applications [39,40], and thus have been extensively studied for the task of HSI classification, such as stacked auto-encoders (SAE) [41], deep belief networks (DBN) [42], deep Boltzmann machines (DBM) [43], convolutional neural networks (CNN) [44–47], and recurrent neural network (RNN) [48]. All of these deep learning based methods are implemented by utilizing the strong ability of network to extract spatial-spectral features in an end-to-end manner, and have been verified to be very effective and have excellent performance. Although the previous approaches perform well, these methods don't explicitly consider removing noise when extracting features. Thus, the quality of the extracted features tends to be easily influenced by the noise embedded in the HSI data. This paper proposes a new approach to deal with this problem based on the low-rank matrix factorization (LRMF) method since noise modeling can be easily embedded into the LRMF method. As aforementioned, low-rank modeling methods have been widely used in HSI classification, and LRMF is one of the most commonly utilized techniques. This technique aims to factorize a data matrix into two low-rank matrices. More specifically, for the HSI classification task, the LRMF method is applied to extracting compact low-rank features since local patch exhibits a low-rank property. However, traditional LRMF methods such as L_1-norm LRMF [49] and L_2-norm LRMF [50] are only optimal for Laplacian or Gaussian noise, and thus can not deal with complex noise well, while the noise of HSI data has been validated to be very complex [51] (e.g., mixture of Gaussian and sparse noise). In this paper, we adopt the mixture of Gaussian (MoG) distribution to model the complex noise embedded in the local patch of HSI data since MoG can approximate any continuous distributions in theory. The method of modeling complex noise by MoG is first proposed in [35]. Then, this idea was extended to the Bayesian framework of LRMF [52], and later applied to robust principle component analysis (RPCA) [53]. Moreover, the removal of complex noise in HSI based on a low-rank method has also been extensively studied recently, such as smooth rank approximation method [54], low-rank matrix factorization based on non-iid noise structure [55] and spatio-spectral total variation method [56,57]. Although all these methods achieve excellent performance in removing complex noise, they either deal with the noise in the whole image or don't use the strategy of mixture noise modeling.

In this paper, we thus adopt a robust LRMF method based on MoG noise modeling (MoGLRMF) to tackle the noise of each local patch. In addition, this method can extract compact low-rank features while removing complex noise. In this way, the new extracted features not only contain rich spectral-spatial information, but also alleviate the negative influence brought by noises. Additionally,

in order to utilize the smooth property of HSI labels that adjacent spectral pixel vectors are probable to have the same label, we further adopt Markov random field (MRF) to post-process the classification map. In many existing work [15,17,31,47], MRF has been verified to help boost the final classification accuracy since spatial smooth information is considered. In summary, the contributions of this work are shown as follows:

- The robust LRMF method based on modeling the noise as MoG is firstly adopted to tackle complex noise embedded in each local patch of HSI data. By using this method as well as considering the smooth property of labels by MRF, we propose a new method for HSI classification, which can simultaneously extract low-rank spectral-spatial features and remove some complex noise in the stage of feature extraction.
- Experimental results on four benchmark HSI datasets illustrate that the proposed method can obtain better performance compared with other state-of-the-art methods.

The rest of the paper is organized as follows: Section 2 introduces the proposed spectral-spatial feature extraction method. Section 3 presents the built package. Section 4 reports the experimental results. Section 5 makes a discussion on the proposed method and experimental results. Section 6 concludes this paper.

2. Spectral-Spatial Feature Extraction Using Robust Low-Rank Matrix Factorization

2.1. Notations

Before introducing the proposed method, some notations should be defined first. We denote the HSI dataset as $\mathcal{H} \in R^{h \times w \times d}$, where h, w and d are height, width and band number, respectively. We unfold $\mathcal{H}$ from the spectral dimension and denote it as $\mathbf{X} = \{\mathbf{x}_n \in R^d, n = 1, 2, \ldots, N, N = hw\}$. whose corresponding labels are denoted as $\mathbf{Y} = \{y_n \in \mathcal{K}, n = 1, 2, \ldots, N\}$, where $\mathcal{K} = \{1, 2, \ldots, Q\}$ is the label set. The training set for class q is defined as $\mathcal{D}_{l^{(q)}} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{l^{(q)}}, y_{l^{(q)}})\}$, and then the entire training set can be represented as $\mathcal{D}_l = \{\mathcal{D}_{l^{(1)}}, \mathcal{D}_{l^{(2)}}, \ldots, \mathcal{D}_{l^{(Q)}}\}$, where $l = \sum_{q=1}^{Q} l^{(q)}$ represents the sum of training samples. The aim of classification is to put a label $y_n \in \mathcal{K}$ on $\mathbf{x}_n$.

Additionally, we denote $\mathbf{H}_{ij}$ as the spectral pixel vector at spatial location (i, j). $\mathbf{S}_{ij} = \bigcup_{\{(u,v):|u-i| \le \lfloor \frac{s}{2} \rfloor, |v-j| \le \lfloor \frac{s}{2} \rfloor\}} \mathbf{H}_{uv} \in R^{d \times s^2}$ is the neighborhood of $\mathbf{H}_{ij}$ in the spatial dimension (including $\mathbf{H}_{ij}$) by using boxes of predefined spatial window with size $s \times s$, where s is an odd number ($s \ge 3$) and $\lfloor \frac{s}{2} \rfloor$ means that the number $\frac{s}{2}$ is rounded down to the nearest integer.

2.2. Spectral-Spatial Feature Extraction Using Robust Low-Rank Matrix Factorization

For each spatial neighborhood $\mathbf{S}$ (subscript ij is omitted for simplicity.), it mainly contains pixel vectors within the same class and also contains a few pixel vectors from other classes. Since spectral pixel vectors within the same class are similar, low-rank matrix factorization (LRMF) can thus be applied to investigating the spatial neighborhood information more precisely. Specifically, given a data matrix $\mathbf{S} \in R^{d \times s^2}$ with entries $S_{ij}s$, the LRMF model can be defined as

$$\min_{\mathbf{U},\mathbf{V}} ||\mathbf{S} - \mathbf{U}\mathbf{V}^T||, \tag{1}$$

where $\mathbf{U} = (\mathbf{u}_1^T; \mathbf{u}_2^T; \ldots; \mathbf{u}_d^T) \in \mathcal{R}^{d \times r}$ and $\mathbf{V} = (\mathbf{v}_1^T; \mathbf{v}_2^T; \ldots; \mathbf{v}_{s^2}^T) \in \mathcal{R}^{s^2 \times r}$ are both matrices with low-rank property ($r < \min(s^2, d)$). $||\cdot||$ represents a certain noise measure and is usually selected as L_1-norm or L_2-norm. After obtaining an optimal solution $(\mathbf{U}^*, \mathbf{V}^*)$, the spatial neighborhood $\mathbf{S}$ can be recovered by the product $\mathbf{U}^*\mathbf{V}^{*T}$, which is a low-rank matrix since $rank(\mathbf{U}^*\mathbf{V}^{*T}) \le rank(\mathbf{U}^*) \le r$.

However, HSI usually contains complex noise, such as mixture of sparse (stripe and deadline) and Gaussian noise [36,51]. This noise can affect the quality of the feature extracted by model (1) since L_1-norm or L_2-norm is optimal for Laplace and Gaussian noise, respectively. In order to deal with the complex noise scenario, many novel LRMF models have been proposed [35,36]. The key

idea is to assume the noise follows complicated mixture distribution, such as mixture of Gaussians (MoG) distribution [35] and mixture of Exponential Power (MoEP) distribution [36]. In this paper, we adopt the LRMF model with MoG noise assumption model (MoGLRMF) to recover each spatial neighborhood $\mathbf{S}$. Specifically, we denote the noise term as $\mathbf{E}$ with entries e_{ij}s, and each e_{ij} is assumed to follow MoG distribution, namely $e_{ij} \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(e_{ij}; 0, \sigma_k)$, where $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$, K denotes the number of mixed components, $\mathcal{N}(e_{ij}; 0, \sigma_k)$ is a Gaussian distribution with mean 0 and variance σ_k for the kth component. Then, in the maximum likelihood estimation (MLE) framework, the log-likelihood of noise term $\mathbf{E}$ can be written as

$$P(\mathbf{E}) = \sum_{i=1}^{d} \sum_{j=1}^{s^2} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(e_{ij}; 0, \sigma_k) \right), \tag{2}$$

where $e_{ij} = s_{ij} - \mathbf{u}_i^T \mathbf{v}_j$. Therefore, the MoGLRMF model is formulated as

$$\max_{\mathbf{U},\mathbf{V},\{\pi_k,\sigma_k\}_{k=1}^{K}} \sum_{i=1}^{d} \sum_{j=1}^{s^2} \log \left(\textstyle\sum_{k=1}^{K} \pi_k \mathcal{N}(s_{ij} - \mathbf{u}_i^T \mathbf{v}_j; 0, \sigma_k) \right). \tag{3}$$

The model (3) can be effectively solved by Expectation Maximization (EM) algorithm, which is a popular way to estimate the parameters of the model in the maximum-likelihood estimate (MLE) framework. The EM algorithm contains two iterative steps, namely Expectation step (E-step) and Maximization step (M-step). The E-step aims to calculate the posterior probability of the latent variable based on current estimated parameters. Then, based on the posterior distribution, we can construct the Q-function, which is a lower-bound of the original likelihood. Finally, M-step aims to re-estimate the parameters of this model by maximizing the Q-function. The two iterative steps stop until some convergence criteria are satisfied.

In order to design the EM algorithm, we first introduce an indicator variable $\mathbf{z}_{ij} = [z_{ij1}, z_{ij2}, \ldots, z_{ijK}]^T$ for each noise e_{ij}, where $z_{ijk} \in \{0, 1\}$ and $\sum_{k=1}^{K} z_{ijk} = 1$. $z_{ijk} = 1$ means that e_{ij} is generated from the k^{th} Gaussian distribution. To make $\mathbf{z}_{ij}$ contain the above property, we assume it follows multinomial distribution $\mathbf{z}_{ij} \sim \mathcal{M}(\boldsymbol{\pi})$, where $\boldsymbol{\pi}$ is the aforementioned mixing proportion parameter. Therefore, the complete log-likelihood function of noise term $\mathbf{E}$ and indicator variable $\mathbf{Z} = \{\mathbf{z}_{ij}\}_{i,j=1}^{s}$ can be expressed as:

$$P(\mathbf{E}, \mathbf{Z}) = \sum_{i=1}^{d} \sum_{j=1}^{s^2} \sum_{k=1}^{K} z_{ijk} [\log \pi_k + \log \mathcal{N}(s_{ij} - \mathbf{u}_i^T \mathbf{v}_j; 0, \sigma_k)]. \tag{4}$$

To optimize model (3) using the EM algorithm, we first conduct the E-step. In order to make the symbols simple, we denote all the parameters to be optimized as $\Theta = \{\mathbf{U}, \mathbf{V}, \{\pi_k, \sigma_k\}_{k=1}^{K}\}$. In addition, we assume that all the parameters $\Theta^{(t)}$ of the tth iteration are obtained and the goal is to estimate $\Theta^{(t+1)}$ of the $(t+1)$th iteration. Specifically, the E-step is shown as follows:

(1) E-step: In this step, we need to calculate the posterior probabilities $\gamma_{ijk} = P(z_{ijk} = 1 | \mathbf{E}; \Theta^{(t)})$ of the indicator variable z_{ij} based on the current parameters $\Theta^{(t)}$, which can be analytically calculated by the Bayes' formula as follows:

$$\gamma_{ijk} = \frac{\pi_k \mathcal{N}(s_{ij} - \mathbf{u}_i^{T(t)} \mathbf{v}_j^{(t)}; 0, \sigma_k^{(t)})}{\sum_{l=1}^{K} \pi_l^{(t)} \mathcal{N}(s_{ij} - \mathbf{u}_i^{T(t)} \mathbf{v}_j^{(t)}; 0, \sigma_l^{(t)})}. \tag{5}$$

After the posterior of the latent variable z_{ij} is obtained, we can construct the Q-function, which is defined as

$$\begin{aligned} Q(\Theta;\Theta^{(t)}) &= E_{\mathbf{Z}\sim P(\mathbf{Z}|\mathbf{E};\Theta^{(t)})}\left[\log P(\mathbf{E},\mathbf{Z};\Theta)\right], \\ &= \sum_{i=1}^{d}\sum_{j=1}^{s^2}\sum_{k=1}^{K} E_{z_{ijk}\sim z_{ijk}=1|\mathbf{E};\Theta^{(t)}}\left[z_{ijk}\right]\left[\log\pi_k + \log\mathcal{N}(s_{ij}-\mathbf{u}_i^T\mathbf{v}_j;0,\sigma_k)\right], \\ &= \sum_{i=1}^{d}\sum_{j=1}^{s^2}\sum_{k=1}^{K}\gamma_{ijk}[\log\pi_k + \log\mathcal{N}(s_{ij}-\mathbf{u}_i^T\mathbf{v}_j;0,\sigma_k)]. \end{aligned} \tag{6}$$

Therefore, the optimized objective function $Q(\Theta;\Theta^{(t)})$ can be expressed as:

$$\max_{\mathbf{U},\mathbf{V},\{\pi_k,\sigma_k\}_{k=1}^{K}} \sum_{i=1}^{d}\sum_{j=1}^{s^2}\sum_{k=1}^{K}\gamma_{ijk}[\log\pi_k + \log\mathcal{N}(s_{ij}-\mathbf{u}_i^T\mathbf{v}_j;0,\sigma_k)]. \tag{7}$$

(2) M-step: In this step, we need to optimize the Q-function $Q(\Theta;\Theta^{(t)})$ w.r.t. the parameters Θ based on current posterior probability. Specifically, the update of each parameter is shown as follows:

For $\{\pi_k\}_{k=1}^{K}$, we first write down the Lagrange function of $\{\pi_k\}_{k=1}^{K}$ by considering the constraint $\sum_{k=1}^{K}\pi_k = 1$, then we calculate the derivative of π_k of Lagrange function of $\{\pi_k\}_{k=1}^{K}$ and set it to be zero. Finally, the update equation of π_k is

$$\pi_k = \frac{1}{ds^2}\sum_{ij}\gamma_{ijk}. \tag{8}$$

For $\{\sigma_k\}_{k=1}^{K}$, we calculate the derivative of σ_k of optimized objective function (7) and set it to be zero. Then, the update equation of σ_k is

$$\sigma_k = \frac{1}{\sum_{ij}\gamma_{ijk}}\sum_{ij}\gamma_{ijk}(s_{ij}-\mathbf{u}_i^T\mathbf{v}_j)^2. \tag{9}$$

For $\mathbf{U}$ and $\mathbf{V}$, by reformulating objective function (7) with respect to $\mathbf{U}$ and $\mathbf{V}$, the following optimization problem needs to be solved

$$\min_{\mathbf{U},\mathbf{V}} ||\mathbf{W}\odot(\mathbf{S}-\mathbf{U}\mathbf{V}^T)||_2^2, \tag{10}$$

where the element w_{ij} of $\mathbf{W}$ is $\sum_{k=1}^{K}\frac{\gamma_{ijk}}{2\pi\sigma_k}$. To solve (10), many weighted L_2-norm LRMF methods can be used, such as the alternated least squares (ALS) [58] which updates the subspaces via minimizing the least square problem alternatively, and Damped Newton (DN) algorithm [59] which adopts Damped Newton algorithm to solve this model.

After we obtain an optimal solution $(\mathbf{U}^*,\mathbf{V}^*)$ of model (3) using the EM algorithm, the spatial neighborhood $\mathbf{S}$ can be recovered by $\hat{\mathbf{S}} = \mathbf{U}^*\mathbf{V}^{*T}$. Then, we regard the low-rank feature $\hat{\mathbf{H}}_{ij}$ in $\hat{\mathbf{S}}$ as the extracted feature (called MoGLRMF feature), and we use this feature in the following classification step instead of the original feature $\mathbf{H}_{ij}$. For simplicity, we denote the MoGLRMF feature of HSI as $\hat{\mathbf{H}} \in R^{h\times w\times d}$. Algorithm 1 summarizes the feature extraction algorithm, and Figure 1 shows the corresponding flowchart of this algorithm.

Algorithm 1 EM algorithm to extract the MoGLRMF feature.

Input: Spatial neighborhood $\mathbf{S}$ of original feature $\mathbf{H}_{ij}$.
Initialization: $\mathbf{U}$, $\mathbf{V}$ and low-rank feature $\hat{\mathbf{H}}_{ij}$.
while stopping criterion is not satisfied **do**
  Update $\{\gamma_{ijk}\}_{i=1,j=1,k=1}^{d,s^2,K}$ via Equation (5)
  Update $\{\pi_k, \sigma_k\}_{k=1}^{K}$ via Equations (8) and (9)
  Update $\mathbf{U}$, $\mathbf{V}$ via DN algorithm [59].
End while
output: $\mathbf{U}$, $\mathbf{V}$ and low-rank feature $\hat{\mathbf{H}}_{ij}$.

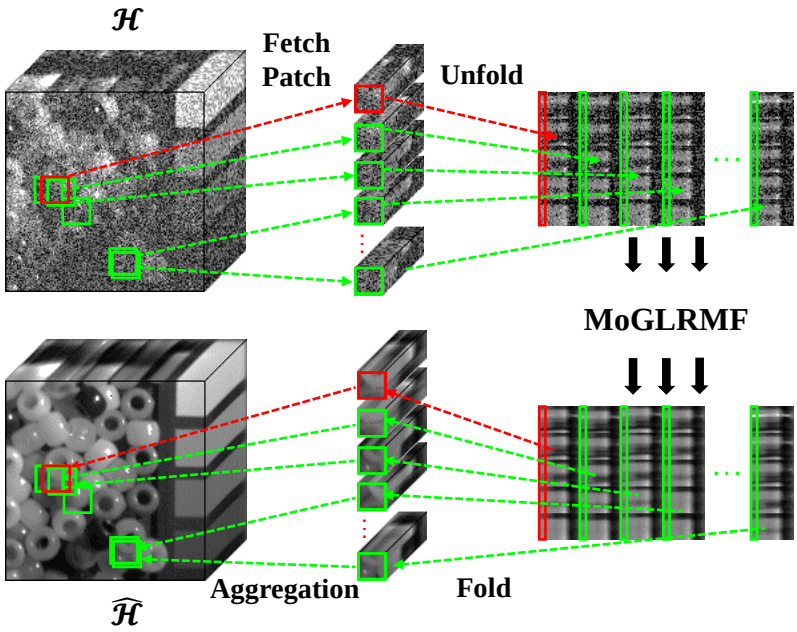

Figure 1. Flowchart of the proposed LRMF model with MoG noise (MoGLRMF) feature extraction algorithm. Firstly, we fetch each local patch and unfold it into matrix along the spectral dimension. Then, the MoGLRMF method is applied to each unfolded matrix, which is then folded back to local patch. Finally, each local patch is aggregated into the final feature tensor.

2.3. Classification and Post-Processing

After obtaining the MoGLRMF feature, we first divide the feature data into training set and testing set. Then, a classifier is trained by using the training set. Here, we don't explicitly point out which classifier is used in the classification stage and we will introduce how to select the best classifier in the experimental section. Finally, the trained classifier is adopted to classify each test sample and thus the classification map can be obtained.

To further boost the classification result, Markov Random Field (MRF) is a very common strategy to post-process the classification map in the HSI classification task [13,15,17,31]. More specifically, this strategy can be expressed as this optimization problem:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}\in\mathcal{K}^N} \left\{ \sum_{n=1}^{N} \log \mathbb{P}(y_n|\hat{\mathbf{h}}_n) + \mu \sum_{n=1}^{N} \sum_{m\in\mathcal{N}(n)} \delta(y_n - y_m) \right\}, \tag{11}$$

where $\hat{\mathbf{h}}_n$ is the nth extracted feature, μ is the smooth parameter, $\mathcal{N}(n)$ is the neighborhood of pixel feature vector $\hat{\mathbf{h}}_n$, $\delta(\cdot)$ is the unit impulse function (Impulse function is defined as: $\delta(0) = 1$ and $\delta(y) = 0$ for $y \neq 0$) and $\mathbf{y}^*$ is the final obtained classification map. Obviously, it should be noted that the second term in (11) reflects the spatial correlation between adjacent pixel vectors, i.e., adjacent pixel vectors are probable to have the same label. Additionally, since the second term of the objective

function in (11) is composed of many pairwise interaction terms, this problem is thus a combinatorial optimization problem, which is very challenging to optimize. Fortunately, many related works to approximate the optimal solution of this combinatorial optimization problem have been extensively studied, such as the graph cut based method [60,61], the belief propagation method [62] and the message passing method [63]. In this work, the α-expansion min-cut method [60] is used.

Based on the discussion, we know that the class probabilities $\mathbb{P}(y_n|\hat{\mathbf{h}}_n)$ play an important role in the MRF post-processing strategy. Therefore, we try to compute the predicted class probabilities of an input sample (namely $\mathbb{P}(y_n|\hat{\mathbf{h}}_n)$) in the classification stage. To illustrate this classification algorithm clearly, we summarize it in Algorithm 2 detailedly.

Algorithm 2 HSI Classification Algorithm.

Input: HSI dataset $\mathcal{H}$.
while stopping criterion is not satisfied **do**
  Extract low-rank features $\hat{\mathcal{H}}$ for $\mathcal{H}$ via Algorithm 1
  Train classifier using training set $\mathcal{D}_l$
  Compute class probabilities $\mathbb{P}(y_n|\hat{\mathbf{h}}_n), n = 1, 2, \ldots, N$
  Compute $\mathbf{y}^*$ via α-Expansion Algorithm
End while
output: Labels $\mathbf{y}^*$.

2.4. Complexity Analysis

In this subsection, we analyze the time complexity of the proposed MoGLRMF method in detail. Since these parameters γ, π and σ can be updated in closed-form, thus the corresponding complexities are $O(Kds^2r)$, $O(Kds^2)$ and $O(Kds^2r)$, respectively, where K, d, s^2 and r are the number of mixture component, spectral dimension, the number of spectral vector in a patch and the rank, respectively. For the update of $\mathbf{U}$ and $\mathbf{V}$, we adopt Damped Newton method, whose complexity is about $O(T_1(d+s^2)^3r^3)$ since we need to invert a $(m+n)r \times (m+n)r$ matrix, where T_1 is the iteration of the Damped Newton method. Thus, the complexity for dealing a local patch by MoGLRMF is $O(T_2(Kds^2r + T_1(d+s^2)^3r^3))$, where T_2 is the iteration of the EM algorithm. Additionally, since we need to deal with about $N = hw$ local patches where h and w are height and width of the HSI data, respectively, the total complexity is $O(NT_2(Kds^2r + T_1(d+s^2)^3r^3))$ in the feature extraction stage. As for the graph cut algorithm, the worst theoretical computational complexity is $O(N^3)$. Finally, the complexity of the proposed method is $O(NT_2(Kds^2r + T_1(d+s^2)^3r^3) + N^3)$. Although theoretical complexity is about $O(N^3)$, the computational time can be dramatically reduced by using parallel computation for each local patch since each patch is independent.

3. SHIP Package for Easy Comparison of HSI Classification Methods

To make an easy comparison of different HSI classification methods implemented on different datasets in our experiments (also to facilitate other researchers on the research), we have specifically built a package for supervised HSI classification, which we call *SHIP* (This code can be available at [64]) (abbreviation of Supervised HSI Classification Package). The details are introduced as follows.

In order to compare two different HSI features, an important rule to be obeyed is that the same classifier and post-processing strategy is needed. However, many existing works have not followed this rule and use different classifiers or post-processing strategies to compare two different features, which results in unfair comparison. To alleviate this issue, our proposed *SHIP* package is composed of three stages, namely feature extraction, classification and post-processing. Firstly, we consider six commonly used HSI classification features in the stage of feature extraction. More specifically, except the original raw data, this package contains four kinds of low-level features (e.g., PCA [19], Low-Rank [15], 3DDWT (This code can be available at [65]) [17,33], 3DGabor [34]), and one deep learning feature (e.g., SAE (This code can be available at [66]) [41]). Note that more features can be easily included in the package except the above ones. Secondly, nine types of representative classifiers

are included in the classification stage, such as k-nearest-neighbor (KNN), Gaussian Naive Bayes (GNB), linear discriminative analysis (LDA), logistic regression (LR), kernel support vector machine (KSVM), decision tree (DT), random forest (RF), gradient boosting (GB) and multiple layer perceptron (MLP). Thirdly, we embed the graph-cut algorithm [60,61] implemented in Python language into the package in the post-processing stage, which can efficiently post-process classification map and thus boost the final performance.

In summary, since each combination of feature and classifier is a HSI classification method, the proposed *SHIP* contains 54 methods (6 features $\times$ 9 classifiers) with post-processing. Many existing HSI classification methods can be regarded as special cases of this package, such as the method proposed in [17] (3DDWT feature with KSVM classifier and post-processing) and the method proposed in [15] (Low-rank feature with SVM classifier and post-processing). In addition, since many methods don't adopt the post-processing step, our package actually ameliorates them and thus tends to result in further performance improvement. Compared with several remote sensing clustering and semantic segmentation software packages, the proposed *SHIP* has some advantages, which is explicitly shown in Table 1. Next, all the experiments will be conducted by using this package.

Table 1. Comparison of different Hyperspectral image (HSI) analysis tool-boxes.

Toolbox	Open	Supervised Classification	Low-level Features (Num$\geq$3)	Deep Feature	Classifiers (Num$\geq$3)	Post-Processing
ENVI [67]	×	√	√	×	×	×
Orfeo [68]	√	√	√	×	√	×
GDAL [69]	√	×	×	×	×	×
Spectral [70]	√	√	×	×	√	×
RSGISLib [71]	√	√	×	×	×	×
EarchMapper [72]	√	√	×	√	×	√
SHIP	√	√	√	√	√	√

4. Experimental Results

To evaluate the performance of our proposed MoGLRMF feature in different scenarios, we conduct experiments on four benchmark datasets (These datasets can be available at http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes), which are summarized in Table 2, all of which have been included in our SHIP package. To make a comprehensive evaluation, four experiments are conducted by the *SHIP* package. These experiments are repeated 20 times. The overall accuracy and standard derivation are reported for each experiment.

Table 2. Summaries for datasets used for HSI classification.

Datasets	Material Class	Image Size	Bands	Spatial Resolution (m)	Acquired Sensor
Indian Pines	16	145 $\times$ 145	220	20	AVIRIS
Pavia University	9	610 $\times$ 340	103	1.3	ROSIS
Pavia Center	9	1096 $\times$ 715	102	1.3	ROSIS
Kennedy Space Center	13	512 $\times$ 614	176	18	AVIRIS

4.1. Selecting the Best Classifier Using SHIP

This experiment is conducted to verify the function of *SHIP* to select a suitable classifier for each competing feature. This experiment adopts two benchmark datasets, namely Indian Pines and Pavia University. Additionally, the specific experimental settings are shown as follows: for Indian Pines dataset, the training sample proportion of each class is selected as 1%. For Pavia University dataset, the number of training samples in each class is chosen as 50. The remaining samples are used to construct the test set. The final experimental results are illustrated in Table 3.

Table 3. Mean classification overall accuracy (OA) and standard deviation (std) (%) of all the combination of features and classifiers on two benchmark datasets using *SHIP* package. The best result for each feature is highlighted in bold.

Features	Raw	PCA [19]	Low-Rank [15]	3DDWT [33]	3DGabor [34]	SAE [41]	MoGLRMF
Classifiers	Indian Pines (1% samples each)						
KNN	57.96 (3.73)	53.27 (4.22)	63.70 (4.43)	61.58 (4.94)	59.97 (3.51)	61.32 (3.72)	66.10 (4.98)
GNB	50.52 (4.87)	54.05 (4.54)	50.06 (4.56)	46.27 (3.92)	55.26 (2.63)	53.26 (3.36)	50.07 (4.74)
LDA	58.00 (7.08)	45.18 (6.85)	59.56 (10.61)	57.70 (9.62)	52.55 (7.32)	47.82 (4.87)	62.78 (9.69)
LR	59.55 (4.93)	60.48 (6.59)	**77.16** (3.09)	**72.25** (3.53)	65.72 (3.57)	63.05 (5.30)	75.15 (4.94)
KSVM	52.07 (2.75)	51.76 (3.35)	58.10 (5.49)	56.21 (4.15)	56.52 (3.30)	56.24 (2.90)	57.50 (5.63)
DT	54.72 (4.74)	53.84 (5.73)	65.33 (4.56)	56.56 (4.05)	51.03 (2.77)	56.63 (3.12)	67.28 (3.10)
RF	60.90 (2.55)	58.62 (2.09)	72.01 (4.11)	64.72 (2.74)	57.48 (1.71)	63.35 (3.10)	**76.84** (2.78)
GB	55.40 (2.84)	55.74 (3.60)	70.09 (3.01)	64.37 (3.59)	53.33 (3.13)	60.52 (3.25)	70.53 (3.31)
MLP	**66.13** (4.02)	**63.78** (4.99)	75.68 (3.55)	71.11 (3.16)	**68.97** (3.82)	**69.81** (3.26)	74.58 (4.87)
Classifiers	Pavia University (50 samples each)						
KNN	87.48 (3.47)	67.04 (8.72)	92.39 (1.96)	90.45 (1.65)	90.90 (2.29)	86.67 (1.90)	91.22 (0.76)
GNB	70.32 (1.71)	89.89 (1.66)	63.10 (3.52)	61.80 (3.28)	70.35 (1.61)	73.32 (1.77)	63.62 (3.55)
LDA	83.12 (9.70)	83.93 (10.36)	80.70 (7.94)	65.16 (15.24)	76.35 (14.07)	63.76 (17.36)	83.92 (1.36)
LR	89.31 (2.59)	89.33 (2.07)	86.93 (4.02)	91.88 (1.28)	92.31 (2.12)	91.83 (1.95)	86.22 (1.63)
KSVM	88.94 (3.36)	91.92 (1.78)	91.28 (3.49)	92.26 (1.15)	**93.69** (1.63)	92.24 (2.19)	92.94 (0.90)
DT	79.01 (3.16)	83.28 (5.04)	84.63 (2.97)	87.47 (2.69)	85.14 (1.94)	84.10 (3.55)	89.95 (1.40)
RF	**90.0**0 (3.51)	**93.66** (2.01)	**94.40** (1.38)	91.84 (1.66)	90.74 (2.84)	**94.67** (2.26)	**94.72** (1.07)
GB	88.38 (3.61)	90.34 (1.87)	90.83 (2.28)	**93.72** (1.37)	89.61 (2.41)	92.05 (2.95)	92.07 (0.94)
MLP	75.78 (3.62)	88.82 (4.25)	75.35 (6.73)	74.95 (11.01)	80.94 (5.33)	84.83 (4.19)	71.85 (3.20)

Table 3 shows the mean overall accuracy and standard deviation (%) of all kinds of combination of features and classifiers on two benchmark datasets using *SHIP*. From Table 3, it can be easily observed that different classifiers can result in significantly different classification results for some specific feature, and thus it is necessary to find the best classifier for each feature. Let's take Indian Pines dataset as an example. For 3DDWT feature and SAE feature, the corresponding best classifiers are LR and MLP, respectively. However, the best classifier for each feature differs across different datasets. For example, as for Pavia University dataset, the corresponding best classifiers for 3DDWT feature and SAE feature are KSVM and RF, respectively, which is different from Indian Pines dataset. Therefore, this experiment implies that we should carefully select the best classifier for each feature and each dataset before conducting the experiment. The proposed *SHIP* has the advantage of helping choose the best classifier once the feature has been extracted. In addition, we can also evaluate the quality of each feature by comparing the results of each feature with the same classifier. We can easily observe from Table 3 that the proposed MoGLRMF feature performs best or second best in the cases with different classifiers. For example, the experimental result of Indian Pines dataset for all the features with RF classifier, the MoGLRMF feature achieves the best performance with overall accuracy 76.84%, while the second best is 72.01%. In addition, the experimental result of Pavia University dataset for all the features with KSVM classifier, and the MoGLRMF feature obtains the second best performance with overall accuracy 92.94%, which only falls behind the best performance 93.69%.

Therefore, these experimental results can not only verify that the classifier which can be easily selected by the *SHIP* package is an important factor to design a HSI classification method, but also demonstrate that the proposed MoGLRMF feature is state of the art due to the consideration of noise removal in extracting the spectral-spatial feature.

4.2. Synthetic Experiments

In this section, we conduct an experiment to verify the robustness of the proposed MoGLRMF feature. More specifically, we choose Pavia University as the dataset and then remove some noisy bands in this experiment. Thus, this dataset can be regarded as a clean one. For easy illustration, we give a clean band as well as a noisy removed band in Figure 2. Then, different types of noise are added into this dataset. Specifically, we add four kinds of noise.

(1) Gaussian noise: All the bands are contaminated by Gaussian noise $\mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 0.05$.

(2) Mixture of Gaussian and stripe noise: All the bands are firstly damaged by Gaussian noise $\mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 0.02$. Then, we randomly choose 40 bands to be added with stripe noise and the number of stripes for each band ranges from 20 to 40.

(3) Mixture of Gaussian and deadline noise: All the bands are firstly contaminated by Gaussian noise $\mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 0.02$. Then, we randomly select 40 bands to be added with deadline noise and the number for each band ranges from 5 to 15.

(4) Mixture of Gaussian and impulse noise: All the bands are firstly contaminated by Gaussian noise $\mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 0.02$. Then, we randomly select 40 bands to be added with impulse noise which contains different intensity, and the percentage of impulse ranges from 50% to 70%.

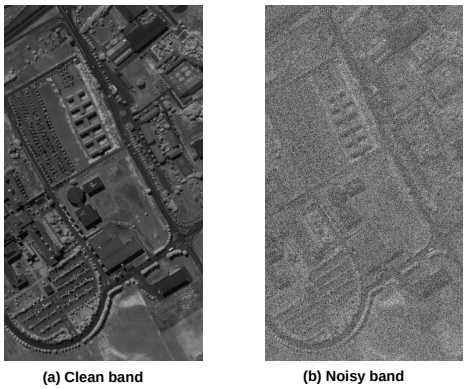

Figure 2. Pavia University bands. (**a**) clean band; (**b**) noisy band (This figure is better seen by zooming in on a computer screen).

After obtaining the noisy HSI dataset, we then adopt the proposed the MoGLRMF method as well as other HSI denoising methods to remove the noise, such as Low-rank method [15], group low-rank and spatial-spectral total variation (GLSSTV) [56] and low-rank matrix factorization based on non-iid noise structure (NMoGLRMF) [55]. The denoised HSI data are treated as the extracted feature. Finally, we compare the classification overall accuracy (OA) of all the methods using the *SHIP*. Specifically, we select 1% training sample proportions for each class in all the cases, and all the methods adopt the random forest as the classifier and use MRF to post-processing the classification map. The final experimental results are reported in Table 4.

Table 4. Overall accuracy (OA) of different methods on the synthetic experiment.

Methods	Clean Dataset	Noisy Dataset	Low-Rank	GLSSTV	NMoGLRMF	MoGLRMF
	Gaussian noise					
OA (%)	82.92 (0.75)	69.48 (0.96)	76.59 (0.84)	76.98 (0.68)	77.19 (0.61)	79.69 (0.77)
	Mixture of Gaussian and stripe noise					
OA (%)	82.92 (0.75)	71.50 (0.87)	75.48 (0.57)	77.12 (0.79)	77.36 (0.82)	79.75 (0.68)
	Mixture of Gaussian and deadline noise					
OA (%)	82.92 (0.75)	71.25 (0.64)	75.82 (0.74)	77.29 (0.90)	77.48 (0.54)	79.45 (1.07)
	Mixture of Gaussian and impulse noise					
OA (%)	82.92 (0.75)	70.98 (0.65)	74.34 (0.64)	76.20 (0.59)	76.36 (0.64)	78.48 (0.71)

From Table 4, it can be easily observed that noise can degrade the classification performance dramatically. For example, the Gaussian noise can make the overall accuracy reduce from 82.92% to 69.48%, about 13% off. The cause of this phenomenon is that noise could affect the quality of the intrinsic feature and make these features less discriminative. Additionally, we can also observe from Table 4 that the features extracted by all the comparing methods are capable of improving the accuracy. More specifically, the Low-rank method gets a 4% improvement, and both GLSSTV and NMoGLRMF make a 6% improvement while the proposed MoGLRMF method obtains a 8% improvement. The reason that MoGLRMF outperforms Low-rank and GLSSTV is due to the use of

MoG noise modeling strategy. As for NMoGLRMF, MoGLRMF outperforms it due to the utilization of patch-based method. In summary, these results also imply that MoGLRMF method is able to remove some complex noise and extract more robust discriminative features due to the consideration of MoG noise modeling strategy and the patch-based processing method.

4.3. Real Experiments

In this section, we conduct real experiments on four benchmark datasets which are displayed in Table 2. This experiment is designed to assess the sensitivity of the proposed method as the size of training set increases. More specifically, we choose 1%, 5% and 10% training sample proportions for each class on all the datasets. Since we encounter out-of-memory problem in dealing with KSC dataset and Pavia Center dataset, we thus crop them into the size of $270 \times 390 \times 100$ and $400 \times 300 \times 102$, respectively, by reshaping the spatial dimension or discarding some noisy bands. Additionally, the competing methods in this experiment are the state-of-the-art method included in the *SHIP*.

Firstly, we choose the best classifier for each feature according to the comprehensive results of the first experiment. The choice of classifier for each feature is shown in the first row of Table 5. For example, the RF classifier is selected for the raw feature, the low-rank feature, the SAE feature and the proposed MoGLRMF feature. In addition, Table 5 also demonstrates the mean classification overall accuracy and standard deviation (%) on four benchmark datasets using *SHIP* package.

From Table 5, we can observe that, with the number of training samples increasing, the classification accuracy keeps increasing, which is a reasonable result since more training samples can help train better classifiers. From Table 5, it can also be seen that the proposed MoGLRMF feature with RF classifier performs best in all the cases. For example, in the case of a 1% proportion on the Kennedy Space Center dataset, the MoGLRMF method achieves 86.59% OA, while the second best only obtains 84.67% OA. In addition, these results can also be used as baseline results for further research. In addition, we report the average cost time for each method in Table 5. From Table 5, it can be easily observed that, although we have utilized parallel strategy, the proposed method is the most time-consuming, which is attributed to the processing of a large number of local patches. Finally, to provide a better visualization for these results, we also show some classification maps (All the classification maps are the result of the 1st experiment in the 20 experiments) from Figures 3–5 by utilizing the *SHIP*. From the classification maps, it can also be easily seen that the MoGLRMF method can obtain more accurate and smooth classification maps compared with other competing methods.

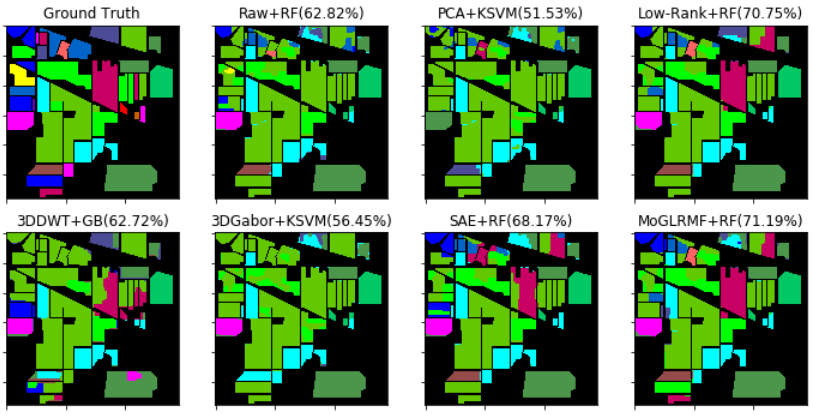

Figure 3. Classification maps of different methods on Indian Pines dataset (1% proportion training samples for each class).

Table 5. Mean classification overall accuracy (OA), standard deviation (std) (%) and time cost on four benchmark datasets using *SHIP* package. The best result for each setting is highlighted in bold.

Methods	Raw+RF	PCA+KSVM	Low-Rank+RF	3DDWT+GB	3DGabor+KSVM	SAE+RF	MoGLRMF+RF
Proportion/dataset	Indian Pines						
1%	61.85 (3.05)	51.74 (3.12)	72.36 (3.45)	64.75 (3.32)	56.52 (3.30)	64.00 (2.73)	**72.80** (3.24)
5%	81.85 (2.45)	72.76 (4.92)	92.44 (1.95)	89.09 (1.01)	85.80 (2.77)	82.75 (1.21)	**92.46** (2.34)
10%	88.57 (1.37)	84.34 (2.59)	95.70 (0.58)	94.64 (0.73)	94.86 (0.85)	90.55 (1.93)	**95.77** (0.73)
Average Time (s)	13.22	15.60	768.72	365.15	210.24	525.36	843.57
	Pavia University						
1%	84.16 (1.07)	85.81 (1.51)	89.70 (0.78)	90.17 (0.69)	90.31 (0.75)	89.00 (0.91)	**91.21** (0.95)
5%	92.39 (0.84)	93.23 (0.62)	95.63 (0.48)	95.20 (0.31)	95.13 (0.26)	94.46 (0.46)	**96.36** (0.51)
10%	96.00 (0.32)	94.75 (0.55)	97.60 (0.29)	98.20 (0.18)	98.01 (0.15)	96.30 (0.26)	**98.26** (0.22)
Average Time (s)	20.78	90.82	1235.42	752.38	482.80	689.36	1556.90
	Pavia Center						
1%	98.61 (0.49)	98.68 (0.42)	98.22 (0.56)	96.95 (0.04)	98.35 (0.30)	98.69 (0.32)	**98.92** (0.66)
5%	99.45 (0.20)	99.51 (0.15)	99.16 (0.32)	98.57 (0.10)	99.47 (0.23)	99.59 (0.14)	**99.60** (0.27)
10%	99.64 (0.14)	99.73 (0.08)	99.45 (0.16)	99.37 (0.06)	99.73 (0.17)	99.70 (0.07)	**99.76** (0.10)
Average Time (s)	7.24	8.56	525.29	487.54	296.30	566.23	805.76
	Kennedy Space Center						
1%	74.08 (6.54)	75.07 (7.04)	84.67 (4.85)	68.93 (2.52)	67.20 (6.75)	71.48 (2.44)	**86.59** (4.59)
5%	91.63 (2.95)	93.01 (2.55)	96.16 (1.41)	90.37 (3.31)	93.80 (1.47)	89.45 (2.30)	**96.84** (1.46)
10%	96.05 (1.07)	96.70 (1.48)	97.82 (1.03)	95.49 (1.90)	97.12 (1.22)	95.05 (1.34)	**98.36** (0.98)
Average Time (s)	8.30	8.47	495.60	438.11	275.42	510.34	788.50

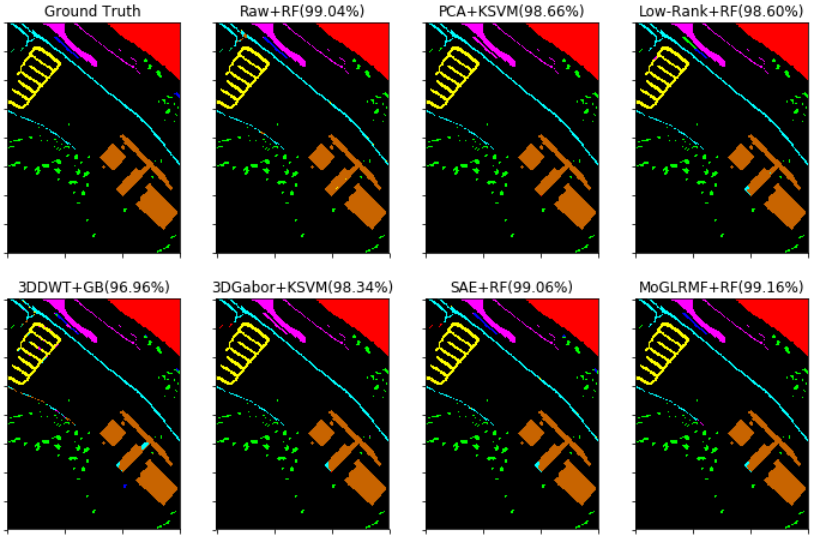

Figure 4. Classification maps of different methods on Pavia Center dataset (1% proportion training samples for each class).

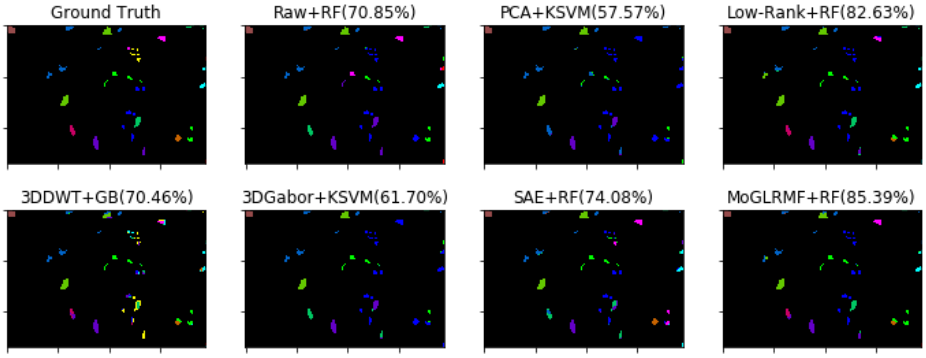

Figure 5. Classification maps of different methods on Kennedy Space Center (KSC) dataset (1% proportion training samples for each class).

Thus, to summarize, these experimental results on the real benchmark datasets can not only demonstrate that *SHIP* makes it easy to conduct HSI classification tasks, but also verify that the proposed classification method based on a MoGLRMF feature can achieve better performance in comparison with other competing methods since the proposed method is composed of a good feature (MoGLRMF), a suitable classifier (RF) and an efficient post-processing step, which function together to make our proposed method obtain state-of-the-art performance.

4.4. Impact of Parameter Settings

This experiment is conducted to assess the impact of the parameters involved in the classification method based on the MoGLRMF feature, such as patch size s, rank r, parameter of classifier and smooth parameter μ. Therefore, we conduct extensive experiments to try to find the best parameters and use overall accuracy (OA) as the measurement. The data used for the analysis is Pavia University and we randomly select 5% training sample proportion for each class.

Firstly, we evaluate the impact of different patch sizes $\{3, 5, 7, 9\}$ on the classification performance with other parameters fixed. Figure 6a illustrate the results. From this sub-figure, it can be observed that the proposed method tends to obtain better performance as patch size becomes larger since more spatial knowledge is considered in the training stage. More specifically, setting patch size as 7 can achieve the best classification accuracy and thus is adopted throughout all the experiments. Secondly, we assess the performance of the proposed method for different rank $r = \{2, 3, 4, 5\}$ with other parameters

fixed. The results are illustrated in Figure 6b. It can be easily seen from Figure 6b that $r = 2$ achieves the best classification accuracy and we use it throughout all the experiments. Thirdly, we make an assessment on the performance of different number of trees $N_{tree} = \{50, 100, 200, 500\}$ of the random forest classifier with other parameters fixed. The results are illustrated in Figure 6c. From this sub-figure, we can easily see that setting $N_{tree} = 100$ provides the best performance and thus we adopt $N_{tree} = 100$ throughout all the experiments. Finally, we evaluate the impact of different smooth parameter $\mu = \{10, 50, 100, 200\}$ on the classification accuracy. The results are illustrated in Figure 6d, where we can see that $\mu = 100$ obtains the best performance and thus this value is adopted throughout all the experiments.

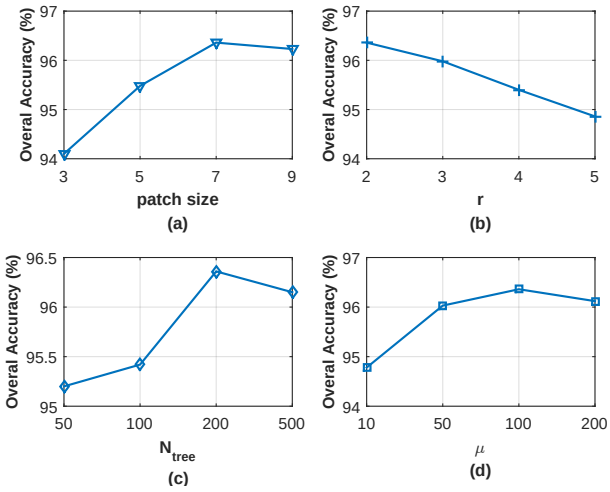

Figure 6. Sensitivity analysis of all the parameters. (**a**) Patch size. (**b**) Rank r. (**c**) Number of trees N_{tree}. (**d**) Smooth parameter μ.

Although the impact study of these parameters are conducted in a greedy manner (e.g., studying one parameter each time with other parameters fixed) and thus may not select the optimal parameters, this way is time-saving and has been verified to perform stably throughout all the experiments. Therefore, we adopt the settings of these parameters in our experiments.

5. Discussion

Firstly, these experimental results show that the selection of classifier is an important factor to design a state-of-the-art HSI classification method. In addition, noise removal is also important to extract qualified features. Thus, our proposed MoGLRMF method for HSI classification can obtain the best performance in comparison with other state-of-the-art methods on the benchmark datasets. However, there are still some limitations for the proposed method. More specifically, this method needs huge computation since it needs to deal with a large number of local patches when extracting the features. Although we have utilized the parallel strategy to speed up, there is still much space to accelerate this method, which will be a future research direction.

Additionally, the proposed method is very related to the method in [15], which adopts the robust principle component analysis (RPCA) [37] to extract the spectral-spatial feature. Since LRMF and RPCA are equivalent in some sense, the proposed MoGLRMF method can thus be regarded as an extension of the RPCA based method [15] in some sense. Compared with the two methods, there are still some differences. Firstly, the RPCA based method can only tackle the Gaussian noise while the MoGLRMF method can deal with the complex noise. Secondly, the RPCA based method utilizes the low-rank property of each segment produced by the segmentation algorithm while the MoGLRMF method makes use of the low-rank property of each local patch. Therefore, the strength of the proposed MoGLRMF method is that it can extract low-rank spectral-spatial features while removing complex noise simultaneously.

6. Conclusions and Future Work

This paper proposes a new feature extraction method for HSI classification. More specifically, the mixture of Gaussian (MoG) based LRMF method is adopted to simultaneously model the noise embedded in local patches of HSI and characterize the low-rank property of the local patch. Therefore, this method can not only extract low-rank spectral-spatial features, but can also alleviate the negative influence brought by noises. Additionally, we present an open package for supervised HSI classification, which can bring convenience to conduct experiments for other research to implement methods' comparison in a fair and efficient manner. Experiments conducted on four benchmark HSI datasets by using this package show that the proposed method can obtain better performance compared with other competing methods.

The future work of this research mainly focuses on replacing the matrix factorization method with a tensor factorization method in order to fully make use of the spectral and spatial information. More specifically, low-rank tucker tensor factorization and CP factorization can be attempted. In addition, we will further develop the *SHIP* package to make it include more state-of-the-art features. Additionally, the supervised HSI classification task will be extended and studied in the semi-supervised and unsupervised framework, which are more promising directions of this field.

Author Contributions: Formal analysis, X.C.; Funding acquisition, D.M.; Methodology, X.C.; Project administration, D.M.; Writing original draft, X.C.; Writing review and editing, D.M. and Z.X.

Funding: This research was supported by the China Postdoctoral Science Foundation funded project (2018M643655), the Fundamental Research Funds for the Central Universities, the National Key R&D Program of China (2018YFB1004300), and the China NSFC project under contracts 61661166011, 11690011, 61603292, 61721002, U1811461, 91860125.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Camps-Valls, G.; Bruzzone, L. Kernel-based methods for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 1351–1362. [CrossRef]
2. Petropoulos, G.P.; Kalaitzidis, C.; Vadrevu, K.P. Support vector machines and object-based classification for obtaining land-use/cover cartography from Hyperion hyperspectral imagery. *Comput. Geosci.* **2012**, *41*, 99–107. [CrossRef]
3. Kitada, K.; Fukuyama, K. Land-use and land-cover mapping using a gradable classification method. *Remote Sens.* **2012**, *4*, 1544–1558. [CrossRef]
4. Matsuki, T.; Yokoya, N.; Iwasaki, A. Hyperspectral tree species classification of Japanese complex mixed forest with the aid of LiDAR data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2177–2187. [CrossRef]
5. Shafri, H.Z.; Taherzadeh, E.; Mansor, S.; Ashurov, R. Hyperspectral remote sensing of urban areas: An overview of techniques and applications. *Res. J. Appl. Sci. Eng. Technol.* **2012**, *4*, 1557–1565.
6. Civco, D.L. Artificial neural networks for land-cover classification and mapping. *Int. J. Geogr. Inf. Sci.* **1993**, *7*, 173–186. [CrossRef]
7. Bischof, H.; Leonardis, A. Finding optimal neural networks for land use classification. *IEEE Trans. Geosci. Remote Sens.* **1998**, *36*, 337–341. [CrossRef]
8. Gualtieri, J.A.; Cromp, R.F. Support vector machines for hyperspectral remote sensing classification. *Proc. SPIE* **1998**, *3584*. [CrossRef]
9. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
10. Camps-Valls, G.; Gómez-Chova, L.; Calpe-Maravilla, J.; Martín-Guerrero, J.D.; Soria-Olivas, E.; Alonso-Chordá, L.; Moreno, J. Robust support vector method for hyperspectral data classification and knowledge discovery. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1530–1542. [CrossRef]
11. Benediktsson, J.A.; Palmason, J.A.; Sveinsson, J.R. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 480–491. [CrossRef]

12. Bandos, T.V.; Bruzzone, L.; Camps-Valls, G. Classification of hyperspectral images with regularized linear discriminant analysis. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 862–873. [CrossRef]
13. Tarabalka, Y.; Fauvel, M.; Chanussot, J.; Benediktsson, J.A. SVM-and MRF-based method for accurate classification of hyperspectral images. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 736–740. [CrossRef]
14. Villa, A.; Benediktsson, J.A.; Chanussot, J.; Jutten, C. Hyperspectral image classification with independent component discriminant analysis. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 4865–4876. [CrossRef]
15. Xu, Y.; Wu, Z.; Wei, Z. Spectral–Spatial Classification of Hyperspectral Image Based on Low-Rank Decomposition. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2370–2380. [CrossRef]
16. Jia, S.; Zhang, X.; Li, Q. Spectral–Spatial Hyperspectral Image Classification Using Regularized Low-Rank Representation and Sparse Representation-Based Graph Cuts. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2473–2484. [CrossRef]
17. Cao, X.; Xu, L.; Meng, D.; Zhao, Q.; Xu, Z. Integration of three-dimensional, discrete wavelet transform and Markov random field for hyperspectral image classification. *Neurocomputing* **2017**, *226*, 90–100. [CrossRef]
18. Wang, Q.; Meng, Z.; Li, X. Locality adaptive discriminant analysis for spectral–spatial classification of hyperspectral images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2077–2081. [CrossRef]
19. Rodarmel, C.; Shan, J. Principal component analysis for hyperspectral image classification. *Surv. Land Inf. Sci.* **2002**, *62*, 115.
20. Crawford, M.M.; Ma, L.; Kim, W. Exploring nonlinear manifold learning for classification of hyperspectral data. In *Optical Remote Sensing*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 207–234.
21. Chen, Y.; Nasrabadi, N.M.; Tran, T.D. Hyperspectral image classification using dictionary-based sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3973–3985. [CrossRef]
22. Chen, Y.; Nasrabadi, N.M.; Tran, T.D. Hyperspectral image classification via kernel sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 217–231. [CrossRef]
23. Soltani-Farani, A.; Rabiee, H.R.; Hosseini, S.A. Spatial-aware dictionary learning for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 527–541. [CrossRef]
24. Zhang, H.; Li, J.; Huang, Y.; Zhang, L. A nonlocal weighted joint sparse representation classification method for hyperspectral imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2056–2065. [CrossRef]
25. Liu, J.; Wu, Z.; Wei, Z.; Xiao, L.; Sun, L. Spatial-spectral kernel sparse representation for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 2462–2471. [CrossRef]
26. Harsanyi, J.C.; Chang, C.I. Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach. *IEEE Trans. Geosci. Remote Sens.* **1994**, *32*, 779–785. [CrossRef]
27. Chang, C.I.; Zhao, X.L.; Althouse, M.L.; Pan, J.J. Least squares subspace projection approach to mixed pixel classification for hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **1998**, *36*, 898–912. [CrossRef]
28. Chang, C.I. Orthogonal subspace projection (OSP) revisited: A comprehensive study and analysis. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 502–518. [CrossRef]
29. Bagan, H.; Yasuoka, Y.; Endo, T.; Wang, X.; Feng, Z. Classification of airborne hyperspectral data based on the average learning subspace method. *IEEE Geosci. Remote Sens. Lett.* **2008**, *5*, 368–372. [CrossRef]
30. Yang, J.M.; Kuo, B.C.; Yu, P.T.; Chuang, C.H. A dynamic subspace method for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 2840–2853. [CrossRef]
31. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 809–823. [CrossRef]
32. Peng, X.; Feng, J.; Xiao, S.; Yau, W.Y.; Zhou, J.T.; Yang, S. Structured autoencoders for subspace clustering. *IEEE Trans. Image Process.* **2018**, *27*, 5076–5086. [CrossRef] [PubMed]
33. Qian, Y.; Ye, M.; Zhou, J. Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 2276–2291. [CrossRef]
34. Shen, L.; Jia, S. Three-dimensional Gabor wavelets for pixel-based hyperspectral imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 5039–5046. [CrossRef]
35. Meng, D.; De La Torre, F. Robust matrix factorization with unknown noise. In Proceedings of the IEEE International Conference on Computer Vision, Tampa, FL, USA, 5–8 December 2013; pp. 1337–1344.

36. Cao, X.; Chen, Y.; Zhao, Q.; Meng, D.; Wang, Y.; Wang, D.; Xu, Z. Low-rank matrix factorization under general mixture noise distributions. In Proceedings of the IEEE International Conference on Computer Vision, Tampa, FL, USA, 5–8 December 2015; pp. 1493–1501.
37. Candès, E.J.; Li, X.; Ma, Y.; Wright, J. Robust principal component analysis? *JACM* **2011**, *58*, 11. [CrossRef]
38. Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; Ma, Y. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 171–184. [CrossRef] [PubMed]
39. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Lake Tahoe, NV, USA, 2012; pp. 1097–1105.
40. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
41. Lin, Z.; Chen, Y.; Zhao, X.; Wang, G. Spectral-spatial classification of hyperspectral image using autoencoders. In Proceedings of the IEEE 9th International Conference on Information, Communications and Signal Processing (ICICS), Tainan, Taiwan, 10–13 December 2013; pp. 1–5.
42. Chen, Y.; Zhao, X.; Jia, X. Spectral–spatial classification of hyperspectral data based on deep belief network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2381–2392. [CrossRef]
43. Midhun, M.; Nair, S.R.; Prabhakar, V.; Kumar, S.S. Deep model for classification of hyperspectral image using restricted boltzmann machine. In Proceedings of the 2014 International Conference on Interdisciplinary Advances in Applied Computing, Amritapuri, India, 10–11 October 2014; ACM: New York, NY, USA, 2014; p. 35.
44. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*, 1–12. [CrossRef]
45. Yue, J.; Zhao, W.; Mao, S.; Liu, H. Spectral–spatial classification of hyperspectral images using deep convolutional neural networks. *Remote Sens. Lett.* **2015**, *6*, 468–477. [CrossRef]
46. Li, W.; Wu, G.; Zhang, F.; Du, Q. Hyperspectral image classification using deep pixel-pair features. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 844–853. [CrossRef]
47. Cao, X.; Zhou, F.; Xu, L.; Meng, D.; Xu, Z.; Paisley, J. Hyperspectral Image Segmentation with Markov Random Fields and a Convolutional Neural Network. *arXiv* **2017**, arXiv:1705.00727.
48. Mou, L.; Ghamisi, P.; Zhu, X. Deep Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3639–3655. [CrossRef]
49. Kwak, N. Principal component analysis based on L1-norm maximization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 1672–1680. [CrossRef] [PubMed]
50. Mitra, K.; Sheorey, S.; Chellappa, R. Large-scale matrix factorization with missing data under additional constraints. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems, Inc.: San Diego, CA, USA, 2010; pp. 1651–1659.
51. Cao, X.; Zhao, Q.; Meng, D.; Chen, Y.; Xu, Z. Robust low-rank matrix factorization under general mixture noise distributions. *IEEE Trans. Image Process.* **2016**, *25*, 4677–4690. [CrossRef] [PubMed]
52. Chen, P.; Wang, N.; Zhang, N.L.; Yeung, D.Y. Bayesian Adaptive Matrix Factorization with Automatic Model Selection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1284–1292.
53. Zhao, Q.; Meng, D.; Xu, Z.; Zuo, W.; Zhang, L. Robust principal component analysis with complex noise. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 55–63.
54. Ye, H.; Li, H.; Yang, B.; Cao, F.; Tang, Y. A Novel Rank Approximation Method for Mixture Noise Removal of Hyperspectral Images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4457–4469. [CrossRef]
55. Chen, Y.; Cao, X.; Zhao, Q.; Meng, D.; Xu, Z. Denoising hyperspectral image with non-iid noise structure. *IEEE Trans. Cybern.* **2017**, *48*, 1054–1066. [CrossRef]
56. Ince, T. Hyperspectral Image Denoising Using Group Low-Rank and Spatial-Spectral Total Variation. *IEEE Access* **2019**, *7*, 52095–52109. [CrossRef]
57. Aggarwal, H.K.; Majumdar, A. Hyperspectral image denoising using spatio-spectral total variation. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 442–446. [CrossRef]
58. De la Torre, F.; Black, M.J. A framework for robust subspace learning. *Int. J. Comput. Vis.* **2003**, *54*, 117–142. [CrossRef]

59. Buchanan, A.M.; Fitzgibbon, A.W. Damped newton algorithms for matrix factorization with missing data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–26 June 2005; pp. 316–322.
60. Boykov, Y.; Veksler, O.; Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 1222–1239. [CrossRef]
61. Kolmogorov, V.; Zabin, R. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 147–159. [CrossRef] [PubMed]
62. Yedidia, J.S.; Freeman, W.T.; Weiss, Y. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Trans. Inf. Theory* **2005**, *51*, 2282–2312. [CrossRef]
63. Kolmogorov, V. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1568–1583. [CrossRef] [PubMed]
64. SHIP—Supervised HSI Classification Package. Available online: https://github.com/xiangyongcao/SHIP (accessed on 27 September 2018).
65. 3-Dimensional Discrete Wavelet Transform. Available online: https://github.com/xiangyongcao/3DDWT-SVM-GC (accessed on 2 July 2015).
66. Stack Auto-Encoder. Available online: https://github.com/hantek/deeplearn_hsi (accessed on 1 February 2014).
67. ENVI. Available online: https://www.harrisgeospatial.com/Software-Technology/ENVI (accessed on 2 March 2015).
68. Grizonnet, M.; Michel, J.; Poughon, V.; Inglada, J.; Savinaud, M.; Cresson, R. Orfeo ToolBox: Open source processing of remote sensing images. *Open Geospatial Data Softw. Stand.* **2017**, *2*, 15. [CrossRef]
69. GDAL—Geospatial Data Abstraction Library. Available online: https://www.gdal.org/ (accessed on 5 May 2019).
70. Boggs, T. Spectral Python. 2014. Available online: http://www.spectralpython.net (accessed on 17 January 2011).
71. Bunting, P.; Clewley, D.; Lucas, R.M.; Gillingham, S. The remote sensing and GIS software library (RSGISLib). *Comput. Geosci.* **2014**, *62*, 216–226. [CrossRef]
72. Kemker, R.; Gewali, U.B.; Kanan, C. EarthMapper: A Tool Box for the Semantic Segmentation of Remote Sensing Imagery. *arXiv* **2018**, arXiv:1804.00292.

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).