THE IMPACT OF LABEL NOISE ON THE CLASSIFICATION MODELS FOR HYPERSPECTRAL IMAGES

MEIZHU LI, SHAOGUANG HUANG AND ALEKSANDRA PIŽURICA

Department of Telecommunications and Information Processing, TELIN-GAIM, Ghent University, Belgium, meizhu.li@ugent.be

Abstract. Supervised classification methods rely heavily on labeled training data. However, errors in the manually labeled data arise inevitably in practice, especially in applications where data labeling is a complex and expensive process, as is often the case in remote sensing. Erroneous labels affect the learning models, deteriorate the classification performances and hinder thereby subsequent image analysis and scene interpretation. In this paper, we analyze the effect of erroneous labels on spectral signatures of landcover classes in remotely sensed hyperspectral images (HSIs). We analyze also statistical distributions of the principal components of HSIs under label noise in order to interpret the deterioration of the classification performance. We compare the behaviour of different types of classifiers: spectral only and spectral-spatial classifiers based on different learning models including deep learning. Our analysis reveals which levels of label noise are acceptable for a given tolerance in the classification accuracy and how robust are different learning models in this respect.

Key words. Robust classification, hyperspectral images, remote sensing, label noise.

1 Introduction

Hyperspectral images (HSIs) are being extensively used in numerous applications in various domains, including geosciences [1], agriculture [2], defense and security [3] and environment monitoring [4]. Image classification, which assigns a class label to each image pixel, plays an essential role in the automatic analysis and interpretation of HSIs.

In the past decade, numerous supervised classification methods for HSIs have been proposed [1, 5] and have achieved satisfactory classification performance. Most of them are designed under the assumption that the training data does not contain erroneous labels. However, in practice imprecise labels are inevitable as labeling is often labor intensive and involves a lot of manual work [6, 7]. The erroneous labels falsely increase the feature variability within class and decrease the discrimination of features across classes. This affects thereby the training of classifiers towards making an incorrect recognition for the new samples, resulting in a degraded classification per-

formance. We shall refer to the erroneous data labels as label noise. Classification methods built on diverse techniques such as Naive Baysian model [8], k-nearest neighbours (k-NN) [9], support vector machine (SVM), sparse representation classification (SRC) [1, 10] and deep neural networks, will be influenced by the label noise differently [7, 11]. Thus, it is of great interest to investigate which levels of label noise can be tolerated in practice, for a given (user-defined or application-dependent) allowed drop in the classification accuracy and how does this depend on the particular classifier type.

Observing that research on this problem is very scarce, in this paper we study thoroughly the behaviour of several representative supervised classification approaches in the scenarios where different levels of label noise are present in the training data. We assume that the label noise is uniformly distributed in the training data of different classes. We characterise statistically its effect on the spectral signatures of landcover classes and the statistical distributions of features. Our empirical results explain from this perspective clearly the reason for the excellent robustness of Bayesian classifiers (and in particular the simple naive Bayesian classifier) compared to some more complex approaches, such as SVM [12], SRC [13], SRC-based classifier with spectral-spatial features (SJSRC) [10] and three spectral-spatial deep learning methods (SSUN, SSRN and CBSP) [14, 15, 16]. At the same time, the empirical results show how erroneous labels affect the model, resulting in a deteriorated classification performance. In addition, the comparison between spectral-based and spectral-spatial based methods demonstrates the benefit of using spatial information to improve the robustness to label noise. We also analyze the classifiers' tolerance to label noise given an acceptable OA degradation.

The rest of the paper is organized as follows. The representative classification methods for HSI that are used for analysis in this paper are briefly introduced in Section 2. In Section 3, we explain our simulation approach and we analyze the influence of label noise on different aspects. Experimental results and analysis are given in Section 4. We conclude the paper in Section 5.

2 Representative Classification Methods for HSI

Here, we review briefly the classifiers that we use for the analysis in this paper. We denote by $\mathbf{x} = (x_1, \cdots, x_m)$ a training sample and $\mathbf{y} = (y_1, \cdots, y_m)$ a test sample, where x_i and y_i are the corresponding i-th features. Both of these vectors are pixel values of a HSI at a given spatial position in m spectral bands. Let C denote the class variable that is assigned to these samples and that takes values c in a finite set $\mathcal{C}$.

2.1 Naive Bayes Classifiers (NBCs)

NBCs are simple Bayesian classifiers. For any given feature vector $\mathbf{x}$, an NBC returns the Maximum a Posteriori (MAP) estimate of the class variable C, assuming the conditional independence $P(\mathbf{x}|c) = \prod_{i=1}^{m} P(x_i|c)$. The estimated class is thus:

$$\hat{c} = \arg\max_{c \in \mathcal{C}} P(c|\mathbf{x}) = \arg\max_{c \in \mathcal{C}} P(c) \prod_{i=1}^{m} P(x_i|c). \tag{1}$$

2.2 K-nearest-neighbor classifier (k-NN)

In k-NN algorithm, the test sample $\mathbf{y}$ is classified by the majority voting of its k nearest neighbors, which are often measured by the Euclidean distance as follows:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum\nolimits_m (x_m - y_m)^2}. \tag{2}$$

Let $\mathcal{N}_y$ be the set of k nearest neighbors of $\mathbf{y}$ according to Equation (2). The test sample $\mathbf{y}$ is assigned to the class that is most common among $\mathcal{N}_y$.

2.3 Support vector machine (SVM)

SVM learns a separating hyperplane from a given set of training data with an optimal decision boundary to each class [17], and categorizes new data points by the learned hyperplane. Let $K(\mathbf{x}_i, \mathbf{x}_j)$ be a kernel function which defines an inner product in the feature space. The decision function implemented by SVM can be written as:

$$f(\mathbf{y}) = sgn(\sum_{i=1}^{N} c_i \alpha_i K(\mathbf{y}, \mathbf{x}_i) + b), \tag{3}$$

where c_i is the corresponding label of sample $\mathbf{x}_i$, b is a real number and the coefficients α_i are obtained by solving the convex Quadratic Programming (QP) problem [18].

2.4 Sparse Representation Classification (SRC)

SRC identifies the label of test data in two steps: sparse representation and classification. Sparse representation represents a test data $\mathbf{y}$ by a linear combination of a few atoms from a dictionary $\mathbf{D} \in \mathbb{R}^{m \times d}$, which in SRC is constructed specially by the training samples $\{\mathbf{x}_i\}_{i=1}^{d}$. We denote by $\mathbf{D}_i \in \mathbb{R}^{m \times d_i}$ the i-th subdictionary in $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_c]$ where each column of $\mathbf{D}_i$ is a training sample of i-th class. The resulting sparse coefficients vector $\boldsymbol{\alpha} \in \mathbb{R}^d$ of $\mathbf{y}$ can be obtained by solving the following optimization problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{y} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 \quad s.t. \quad \|\boldsymbol{\alpha}\|_0 \leq K, \tag{4}$$

where $\|\boldsymbol{\alpha}\|_0$ denotes the number of non-zero elements in $\boldsymbol{\alpha}$ and K is the sparsity level, i.e., the largest number of atoms in dictionary $\mathbf{D}$ needed to represent any input sample $\mathbf{y}$. The optimization problem in Eq. (4) is typically solved with a greedy algorithm, such as Orthogonal Matching Pursuit (OMP) [19]. Then, the class of the test sample is identified by calculating the class-specific residuals r_i [13]:

$$\begin{aligned} class(\mathbf{y}) &= \arg\min_{i=1,2,...,C} r_i(\mathbf{y}) \\ &= \arg\min_{i=1,2,...,C} \|\mathbf{y} - \mathbf{D}_i\boldsymbol{\alpha}_i\|_2, \end{aligned} \tag{5}$$

where $\boldsymbol{\alpha}_i$ are the sparse coefficients associated with class i.

2.5 SRC-based classifier with spectral-spatial features

We also consider a representative of SRC-based method where spatial information is included, and in particular we will use in our analysis the method of [10], called SJSRC, which employs super-pixel segmentation and encodes jointly all the pixels within one super-pixel. It assumes that similar pixels in local regions, which are defined by super-pixel segmentation, can be represented by a few common atoms in $\mathbf{D}$. This results in a row sparsity pattern on the coefficients matrix of the pixels within the same super-pixel. Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ represent a super-pixel composed of n pixels in m spectral bands and $\mathbf{A} \in \mathbb{R}^{d \times n}$ the corresponding coefficients matrix. SJSRC solves the following problem

$$\arg\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 \ \ s.t. \ \|\mathbf{A}\|_{row,0} \leq K_0, \tag{6}$$

where $\|\mathbf{A}\|_{row,0}$ denotes the number of non-zero rows of $\mathbf{A}$ and K_0 is the row-sparsity level. After finding $\mathbf{A}$, the class for the whole super-pixel $\mathbf{X}$ is decided as:

$$class(\mathbf{X}) = \arg\min_{i=1,2,...,C} \|\mathbf{X} - \mathbf{D}_i\mathbf{A}_i\|_F, \tag{7}$$

where $\mathbf{A}_i$ is the sub-matrix of $\mathbf{A}$ corresponding to class i.

2.6 Deep learning based spectral-spatial classifier

Deep learning methods have been increasingly used in HSI classification [20, 21, 22]. As representatives of these methods, we select three recent ones: SSUN [14], SSRN [15] and CBSP [16]. All the three combine spectral and spatial feature extraction.

2.6.1 Spectral-spatial unified network (SSUN)

The SSUN algorithm [14] integrates the spectral feature extraction, spatial feature extraction and classifier training into a unified neural network. It incorporates long short-term memory (LSTM) [23] network for band grouping and spectral feature extraction and the multiscale CNN (NSCNN) for spatial feature extraction. The loss function is defined as:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}^{joint} + \mathcal{L}^{spectral} + \mathcal{L}^{spatial} \\ &= -\frac{1}{d}\sum\nolimits_{i=1}^{d}[c_i log(\hat{c_i}^{joint}) + (1-c_i)log(1-\hat{c_i}^{joint})] \\ &- \frac{1}{d}\sum\nolimits_{i=1}^{d}[c_i log(\hat{c_i}^{spectral}) + (1-c_i)log(1-\hat{c_i}^{spectral})] \\ &- \frac{1}{d}\sum\nolimits_{i=1}^{d}[c_i log(\hat{c_i}^{spatial}) + (1-c_i)log(1-\hat{c_i}^{spatial})],\end{aligned} \tag{8}$$

where $\mathcal{L}^{joint}$ is the main loss function, $\mathcal{L}^{spectral}$ and $\mathcal{L}^{spatial}$ are two auxiliary loss functions, $\hat{c}_i^{joint}$, $\hat{c}_i^{spectral}$ and $\hat{c}_i^{spatial}$ are the corresponding predicted labels for the ith training sample, c_i is the true label, and d is the size of training set.

2.6.2 Spectral-spatial residual network (SSRN)

The SSRN algorithm [15] is an end-to-end spectral-spatial residual network that takes raw 3-D cubes as input data for hyperspectral image classification. In SSRN, the spectral and spatial residual blocks consecutively learn discriminative features from abundant spectral signatures and spatial contexts in hyperspectral imagery. Let $\mathbf{X}$ be the HSI data set, the spectral residual architecture is formulated as follows:

$$\begin{aligned}&\mathbf{X}^{p+2} = \mathbf{X}^p + F(\mathbf{X}^p;\theta), \\ &F(\mathbf{X}^p;\theta) = R(\hat{\mathbf{X}}^{p+1}) \times \mathbf{h}^{p+2} + \mathbf{b}^{p+2}, \\ &\mathbf{X} = R(\hat{\mathbf{X}}^p) \times \mathbf{h}^{p+1} + \mathbf{b}^{p+1},\end{aligned} \tag{9}$$

where $\theta = \{\mathbf{h}^{p+1}, \mathbf{h}^{p+2}, \mathbf{b}^{p+1}, \mathbf{b}^{p+2}\}$, $\mathbf{X}^{p+1}$ represents the n input 3-D feature cubes of $(p+1)$th layer, $\mathbf{h}^{p+1}$ and $\mathbf{b}^{p+1}$ denote the spectral convolutional kernels and bias in the $(p+1)$th layer, respectively, $\hat{\mathbf{X}}^p$ is the normalization result of batch feature cubes $\mathbf{X}^p$ in the pth layer, $R(\cdot)$ is the rectified linear unit activation function that sets elements with negative numbers to zero, $F(\mathbf{X}^p;\theta)$ is a residual function. The output tensor of the spectral residual block includes n 3-D feature cubes. The spatial residual block is defined similarly with the spectral block. The output of the spatial block is a 3-D feature volume. More details can be found in [21].

2.6.3 Convolution based spectral partitioning architecture (CBSP)

The CBSP algorithm [16] aims to develop a deep learning architecture using 3-D convolutional neural networks with spectral partitioning to extract features. It first performs a spatial transformation via 2-D convolution. The transformed image is partitioned on the spectral level and split into segments for efficient processing. 3-D convolution is then applied to each segment. Finally, convoluted segments are concatenated and fed to two fully-connected layers with dropout as regularization. The detailed description of CBSP can be found in [16].

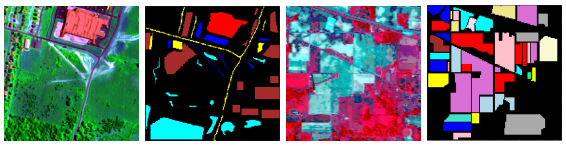
3 Model Uncertainty Caused by Label Noise

3.1 Data sets

We conduct our experiments on two real HSI data sets: *HYDICE Urban* and *Indian Pines*.

The *HYDICE Urban* data set, with 188 spectral bands, was captured by the HYDICE sensor over an urban eara. Its spatial size is 307×307 pixels and in our experiments, we use a part of this image with size 200×200 shown in Figure 1a. The ground truth classification is shown in Figure 1b.

The *Indian Pines* data set was gathered by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over Northwest Indiana in June 1992. After the removal



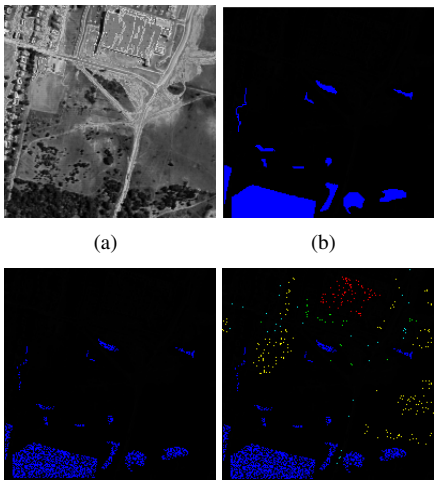
(a) (b) (c) (d)

Fig. 1: Two real hyperspectral data sets used in the experiments. (**a**) False color images of the selected part of *HYDICE Urban* and (**b**) the corresponding ground truth classification. (**c**) False color image of *Indian Pines* and (**d**) the corresponding ground truth classification.

of the water absorption bands, 200 bands remain, and the total data size is $145 \times 145 \times 200$ with 16 distinctive classes. The ground truth classification is shown in Figure 1d.

3.2 Model uncertainties analysis with noisy labels

We define the level of label noise ρ as the proportion of training samples that have wrong labels. The erroneous labels are chosen with equal probabilities in $\mathcal{C} \setminus \{c\}$, with $\mathcal{C}$ the set of class values and c the true class of label noise ρ as the proportion of training samples that have wrong labels. To reduce the data dimensionality, PCA is commonly applied on the original HSIs data. Fig. 2 shows an illustration of introducing label noise in the *HYDICE Urban* data set. The first PC is shown in Figure 2a. All the labelled samples in Class 1 are highlighted in Figure 2b. Next, we randomly select 50% of the highlighted samples as the training samples for Class 1 (Figure 2c). We also select at random a given portion ρ of the total training samples (from various classes) and flip each of them to one of the remaining classes at random. Figure 2d illustrates an instance of the resulting Class 1 labels for $\rho = 0.5$. Different colours denote different original classes of the training samples that were flipped to Class 1. Note that the choice of ρ is here merely for clearer illustration purposes; a situation with 50% of wrong labels is unlikely to be relevant in practice.



(a) (b) (c) (d)

Fig. 2: An illustration of introducing label noise. (**a**) the first PC of *HYDICE Urban*; (**b**) labelled samples in Class 1 (marked in blue); (**c**) training samples (50% of the labelled samples) in Class 1 and (**d**) an instance of the samples labelled as Class 1 when $\rho = 0.5$. Different colors denote samples from different classes that were erroneously flipped to Class 1.

Fig. 3 illustrates the effect of label noise on the average spectral signatures in *HYDICE Urban* data set. Without label noise the spectral signatures of different classes are rather different from each other. In the presence of label noise, they wrongly appear to be more similar to each other. Thus, label noise obviously trends to uniformise all the spectral signatures, which will affect inevitably the classification accuracy. In this case, label noise obviously tends to uniformise all the spectral signatures as expected, because now each of them is computed from a mixture of different classes.

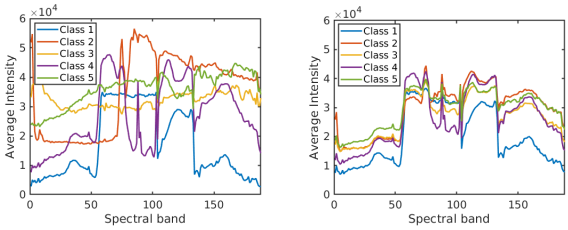


Fig. 3: Average spectral signatures for the *HYDICE Urban* data set with $\rho = 0$ (left) and with $\rho = 0.5$ (right).

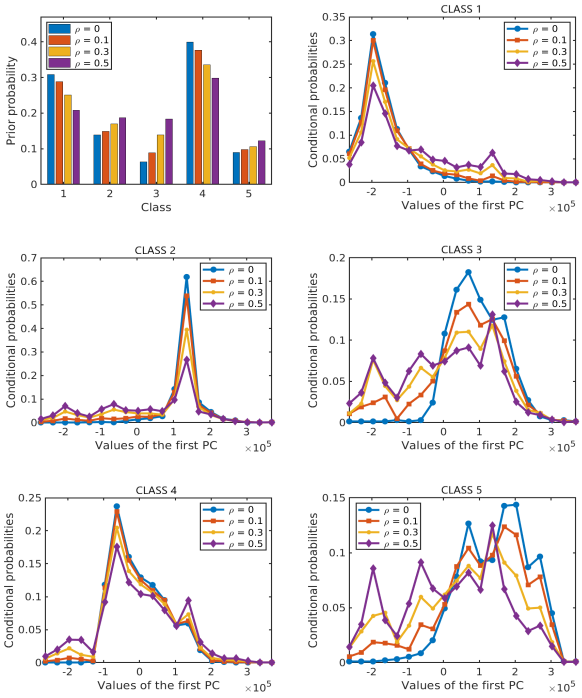


Fig. 4: The prior probabilities of classes (top left) and conditional probabilities of the first PC for different levels of label noise in *HYDICE Urban*.

Fig. 4 shows the effect of label noise on prior probabilities of classes (top left) and on conditional probabilities of the first PC. The PC values are uniformly discretized into twenty intervals. While the actual prior probabilities of different classes are significantly different from each other, these differences become smaller when label noise increases. The distributions conditioned on the class variable keep a similar shape when increasing ρ from 0 to 0.5, but the peak value decreases and the distribution shape gets more flattened compared to the distributions without label noise.

These results indicate that erroneous labels lead to model uncertainties, which will in their turn affect the classification performance. Baysian models, which are based on conditional probabilities like those in Fig. 4, are likely to be more robust to label noise than some other classifiers that rely more directly on spectral signatures (like those in Fig. 3). Conditional probability distributions do not change significantly until the label noise becomes very large.

In the following section, we will study the performance of the representative classifiers and explore which level of label noise can be tolerated depending on the acceptable drop in the classification accuracy.

4 Experimental Results

4.1 Experiments setting

The effect of erroneous labels is studied by evaluating the performance of the eight representative classification algorithms described in Section 2. Four of these (NBC, k-NN, SVM and SRC) are based on spectral features alone, and the remaining four (SJSRC, SSUN, SSRN and CBSP) make use of both spectral and spatial features. The SSUN, SSRN and CBSP methods are based on deep learning model. We detail their implementations by:

1. NBC with Gaussian distribution for the likelihood of the features where $P(x_i|c)$ in Equation (1) is defined as:

$$P(x_i|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left(-\frac{(x_i-\mu_c)^2}{2\sigma_c^2}\right) \qquad (10)$$

   30 and 55 PCs are extracted for *Indian Pines* and *HYDICE Urban* respectively, which represent more than 95% of the cumulative variance. The principle component analysis (PCA) is applied first to input HSI. Due to the decorrelating properties of PCA, conditional independence assumption that NBC relies on is well justified.

2. The number of neighbors for k-NN is obtained by five-fold cross validation over the training samples; we adopt the Radial Basic Function (RBF) kernel for SVM; the parameters in SRC and SJSRC are the same as those in [10].

3. For the three deep learning methods, the training epochs of SSUN are set as 200 with a learning rate of

0.001 and batch size of 64; For SSRN method, 10%, 10% and 80% of the labelled data are randomly assigned to training, validation and testing groups, respectively. The training epochs are 200 with batch size of 16; For CBSP method, 10%, 5% and 85% of the labelled data are randomly selected as training, validation and testing data. The training epochs are 600 with batch size of 50.

In the following experiments, 10 percent of samples are randomly selected for training and the rest are for testing. The reported results are averaged values over 10 runs with different training samples. We evaluate the classification performance by overall accuracy (OA), which is the ratio between correctly classified testing samples and the total number of testing samples.

4.2 Experiments on *Indian Pines*

Fig. 5 (left) shows the overall accuracy of the eight algorithms on *Indian Pines* with ρ ranging from 0 to 0.9, to see the behaviour of selected classifiers and to explore at which level of label noise their performance starts to drop. When there is low-to-moderate amounts of label noise ($\rho \geq 0$), the four spectral-spatial methods show much better performance than the four spectral-based methods. When there is no label noise ($\rho = 0$) the deep learning method SSRN yields the best OA, while the naive Bayesian classifier (NBC) is inferior to all other methods. This can partly be attributed to the fact that this particular NBC makes use of only spectral features while the other better performing methods (SJSRC, SSUN, SSRN and CBSP) incorporate spatial next to spectral features. With the increasing levels of label noise, spectral-based algorithms k-NN, SVM and SRC show similar behaviour, but SRC performs worse than the other two and shows approximately linear decrease. The performance of NBC is the most stable, which can be well understood by analyzing the shape of the involved conditional probabilities (see Fig. 4 and the accompanying discussion in Section 3). The performance of NBC drops suddenly when ρ exceeds 0.6. At this point, following further the flattening trend from Fig. 4, the conditional probability distributions become too flattened and the classifier can no longer reasonably operate. The overall accuracy of spectral-spatial methods SJSRC, SSUN, SSRN and CBSP deteriorate significantly with the increasing label noise and the three deep learning methods (SSUN, SSRN and CBSP) are especially vulnerable in this respect. The sparse coding method SJSRC achieves thus best performance over the whole range where $\rho > 0.1$.

Fig. 5 (right) shows the maximum level of label noise that a classifier can tolerate given a decreasing rate in the OA compared to the case with no label noise ($\rho = 0$). We analyze the tolerance of the eight classification models in the cases with OA decreasing in 5%, 10% and 15% compared to the OA of $\rho = 0$. We assume that the OAs between any two successive ρ (in steps of 0.1) decrease linearly as in Fig. 5. NBC shows the highest tolerance to label noise. E.g., if 5% decrease in OA can be tolerated, NBC allows 30% of erroneous labels. The three deep learning approaches (SSUN, SSRN and CBSP) exhibit very low tolerance to label noise, although they make use of both spectral and spatial features. The sparse coding approach based on spectral features alone (SRC) also shows low tolerance to label noise, but its version with spatial information (SJSRC) is much more robust, both compared to the basic SRC and to the deep learning methods.

4.3 Experiments on *HYDICE Urban*

Fig. 6 shows the performance of the eight algorithms on *HYDICE Urban*. Spectral-based algorithms k-NN, SVM and SRC show similar behaviour as in the other data set. NBC performs now better and even outperforms other algorithms for very large ρ. Also, NBC is again the most stable method. Its performance drops suddenly when ρ exceeds 0.6. The spectral-spatial methods

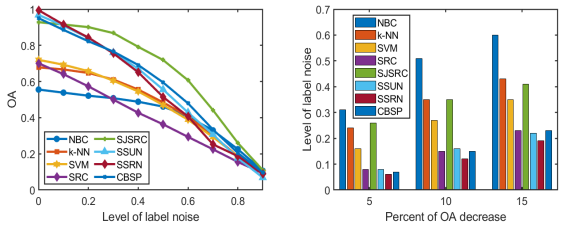


Fig. 5: Influence of label noise on OA (left) and the classifiers' tolerance of label noise at different drops in OA (right). Data set: *Indian Pines*.

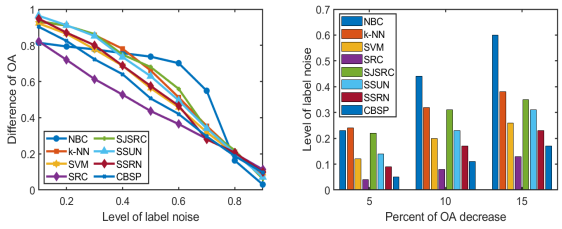


Fig. 6: Influence of label noise on OA (left) and the classifiers' tolerance of label noise at different drops in OA (right). Data set: *HYDICE Urban*.

(SJSRC, SSUN, SSRN and CBSP) also behave similarly as on the other data set and their overall accuracy deteriorates significantly with the increasing label noise.

Percentages of wrong labels that can be tolerated for a given decrease in OA, shown in the right of Fig. 6, show similar trends as in the first data set. NBC shows again the highest tolerance to label noise in the three cases. The sparse coding approach based on spectral alone (SRC) shows very low tolerance to label noise, but the version with spatial information (SJSRC) is much more robust to label noise, both compared to basic SRC and to the deep learning methods SSUN, SSRN and CBSP.

5 Conclusion

We analysed the effect of erroneous data labeling on supervised HSI classification from different aspects: the estimated spectral signatures of different classes, the estimated statistical distributions of features and the performance of different types of classification algorithms. The analysis reveals that Bayesian classifiers, even under the simplest naive Bayesian model (NBC) are more robust to label noise than methods based on support vector machines (SVM), sparse coding and deep learning. Deep learning approaches exhibited in all our experiments the biggest vulnerability to label noise. This agrees with recent studies that show susceptibility of deep learning to various other perturbations, such as noise in the data and adversarial attacks. We provided explanation for the robustness of the Bayesian approach by analyzing the effect of label noise on the probability distributions of the principal components conditioned on the class variable. These statistical distributions change gently with increasing the label noise (remaining peaked at the same positions and getting gradually flattened). This is the reason why the classification performance of NBC remains very stable until the label noise becomes excessively large. The k-NN method also demonstrated very robust performance, which can be attributed to its majority voting strategy.

Our analysis shows also clearly the importance of using spatial context not only to improve the classification accuracy in ideal settings but also to improve the robustness to label noise. Sparse coding methods that make use of both spectral and spatial information showed excellent performance and can be considered as a good choice of a classifier, which is not only highly accurate but also robust to non-ideal data labeling. It will be also of interest to explore Bayesian classifiers that combine both spectral and spatial features within a unified framework (e.g., as an extension of the NBC that we considered) and to compare those to the sparse coding approach.

Acknowledgment

This work was supported by the China Scholarship Council (CSC), by the Fonds voor Wetenschappelijk Onderzoek (FWO) project under Grant G.OA26.17N and the Flemish Government (AI Research Program).

References

[1] Shaoguang Huang, Hongyan Zhang, and Aleksandra Pižurica. Semisupervised sparse subspace clustering method with a joint sparsity constraint for hyperspectral remote sensing images. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, 12(3):989–999, 2019.

[2] Telmo Adão, Jonáš Hruška, Luís Pádua, José Bessa, Emanuel Peres, Raul Morais, and Joaquim João Sousa. Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry. *Remote Sens.*, 9(11):1110, 2017.

[3] Michael T Eismann, Alan D Stocker, and Nasser M Nasrabadi. Automated hyperspectral cueing for civilian search and rescue. *Proc. IEEE*, 97(6):1031–1055, 2009.

[4] Chen Wu, Liangpei Zhang, and Bo Du. Kernel slow feature analysis for scene change detection. *IEEE Trans. Geosci. Remote Sensing*, 55(4):2367–2384, 2017.

[5] Mingjing Wang and Huiling Chen. Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis. *Appl. Soft. Comput.*, 88:105946, 2020.

[6] Junjun Jiang, Jiayi Ma, Zheng Wang, Chen Chen, and Xianming Liu. Hyperspectral image classification in the presence of noisy labels. *IEEE Trans. Geosci. Remote Sensing*, 57(2):851–865, 2018.

[7] Meizhu Li, Shaoguang Huang, and Aleksandra Pi˘zurica. Robust dynamic classifier selection for remote sensing image classification. In *Proc. ICSIP*, pages 101–105. IEEE, 2019.

[8] Juan Mario Haut, Mercedes E Paoletti, Javier Plaza, Jun Li, and Antonio Plaza. Active learning with convolutional neural networks for hyperspectral image classification using a new bayesian approach. *IEEE Trans. Geosci. Remote Sensing*, 56(11):6440–6461, 2018.

[9] Mahdi Hasanlou and Farhad Samadzadegan. Comparative study of intrinsic dimensionality estimation and dimension reduction techniques on hyperspectral images using k-nn classifier. *IEEE Geosci. Remote Sens. Lett.*, 9(6):1046–1050, 2012.

[10] Shaoguang Huang, Hongyan Zhang, and Aleksandra Pižurica. A robust sparse representation model for hyperspectral image classification. *Sensors*, 17(9):2087, 2017.

[11] Meizhu Li, Shaoguang Huang, Jasper De Bock, Gert De Cooman, and Aleksandra Pižurica. A robust dynamic classifier selection approach for hyperspectral images with imprecise label information. *Sensors*, 20(18):5262, 2020.

[12] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *COLT92*, pages 144–152, 1992.

[13] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):210–227, 2008.

[14] Yonghao Xu, Liangpei Zhang, Bo Du, and Fan Zhang. Spectral–spatial unified networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sensing*, 56(10):5893–5909, 2018.

[15] Zilong Zhong, Jonathan Li, Zhiming Luo, and Michael Chapman. Spectral–spatial residual network for hyperspectral image classification: A 3-d deep learning framework. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2):847–858, 2017.

[16] Ringo SW Chu, Ho-Cheung Ng, Xiwei Wang, and Wayne Luk. Convolution based spectral partitioning architecture for hyperspectral image classification. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 3962–3965. IEEE, 2019.

[17] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE Trans. Neural Netw.*, 10(5):988–999, 1999.

[18] Sujun Hua and Zhirong Sun. Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, 17(8):721–728, 2001.

[19] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theory*, 53(12):4655–4666, 2007.

[20] Xian Li, Mingli Ding, and Aleksandra Pižurica. Deep feature fusion via two-stream convolutional neural network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sensing*, 58(4):2615–2629, 2019.

[21] Yushi Chen, Kaiqiang Zhu, Lin Zhu, Xin He, Pedram Ghamisi, and Jón Atli Benediktsson. Automatic design of convolutional neural network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):7048–7066, 2019.

[22] Yushi Chen, Ying Wang, Yanfeng Gu, Xin He, Pedram Ghamisi, and Xiuping Jia. Deep learning ensemble for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(6):1882–1897, 2019.

[23] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.