Infrared ultraspectral signature classification based on a restricted Boltzmann machine with sparse and prior constraints

Xiaoguang Mei, Yong Ma, Fan Fan, Chang Li, Chengyin Liu, Jun Huang & Jiayi Ma

To cite this article: Xiaoguang Mei, Yong Ma, Fan Fan, Chang Li, Chengyin Liu, Jun Huang & Jiayi Ma (2015) Infrared ultraspectral signature classification based on a restricted Boltzmann machine with sparse and prior constraints, International Journal of Remote Sensing, 36:18, 4724-4747

To link to this article: http://dx.doi.org/10.1080/01431161.2015.1079664

Published online: 08 Oct 2015.

Submit your article to this journal

Article views: 29

View related articles

View Crossmark data
Infrared ultraspectral signature classification based on a restricted Boltzmann machine with sparse and prior constraints

Xiaoguang Mei\textsuperscript{a,b}, Yong Ma\textsuperscript{a,*}, Fan Fan\textsuperscript{a}, Chang Li\textsuperscript{b}, Chengyin Liu\textsuperscript{b}, Jun Huang\textsuperscript{a}, and Jiayi Ma\textsuperscript{a}

\textsuperscript{a}Electronic Information School, Wuhan University, Wuhan, 430072 P. R. China; \textsuperscript{b}School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, 430074 P. R. China

(Received 9 January 2015; accepted 28 July 2015)

The state-of-the-art ultraspectral technology brings a new hope for the high precision applications due to its high spectral resolution. However, it comes with new challenges brought by the improvement of spectral resolution such as the Hughes phenomenon and over-fitting issue, and our work is aimed at addressing these problems. As new Markov random field (MRF) models, the restricted Boltzmann machines (RBMs) have been used as generative models for many different pattern recognition and artificial intelligence applications showing promising and outstanding performance. In this article, we propose a new method for infrared ultraspectral signature classification based on the RBMs, which adopt the regularization-based techniques to improve the classification accuracy and robustness to noise compared to traditional RBMs. First, we add an arctan-like term to the objective function as a sparse constraint to improve the classification accuracy. Second, we utilize a Gaussian prior to avoid the over-fitting problem. Third, to further improve the classification performance, a multi-layer RBM model, a deep belief network (DBN), is adopted for infrared ultraspectral signature classification. Experiments using different spectral libraries provided by the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) and the Environmental Protection Agency (EPA) were performed to evaluate the performance of the proposed method by comparing it with other traditional methods, including spectral coding-based classifiers (binary coding (BC), spectral feature-based binary coding (SFBC), and spectral derivative feature coding (SDFC) matching methods), a novel feature extraction method termed crosscut feature extraction matching (CF), and three machine learning methods (artificial deoxyribonucleic acid (DNA)-based spectral matching (ADSM), DBN, and sparse deep belief network (SparseDBN)). Experimental results demonstrate that the proposed method is superior to the other methods with which it was compared and can simultaneously improve the accuracy and robustness of classification.

1. Introduction

Spectral signatures define the characteristics of objects based on their absorptance, reflectance, and transmittance of electromagnetic radiation (Shanmugam and Srinivasaperumal 2014). According to the research of Hunt (1977), for earth materials, the typical half depth of absorption peaks ranges from 20 to 40 nm. In hyperspectral remote-sensing applications, spectral resolution has achieved less than 10 nm, which enables precise classification among different materials. However, it requires even higher spectral resolution in some applications such as gas spectral signature analysis (Schäfer
et al. 2003; Schürmann et al. 2007), e.g. the spectral resolution can reach up to 1 cm\(^{-1}\) (‘cm\(^{-1}\)’ is the unit of the wavenumber resolution), and the corresponding wavelength resolution differs for each band. For example, at the typical mid-infrared waveband 3 \(\mu\)m, the corresponding wavelength resolution is 0.9 nm. Fortunately, the state-of-the-art ultraspectral technologies (Ehlers, Gähler, and Janowsky 2003), e.g. the tropospheric emission spectrometer (TES) (Beer 2006) and the infrared atmospheric sounding interferometer (IASI) (Bioucas-Dias et al. 2013), have achieved a spectral resolution of less than 0.5 cm\(^{-1}\), which brings a new hope for those high precision applications. Nevertheless, the breakthrough of spectral resolution leads to some new problems, which makes the spectral signature analysis become more challenging.

Higher spectral resolution usually brings higher dimension of data. With regard to the learning-based classification method, while the spectral resolution gets higher, for limited training samples, the complexity of fitting the hyperplane of classifier gets higher, which may lead to the fitted hyperplane deviating from the expected one, and hence the accuracy of classification decreases. This issue is caused by the growth of dimension and also known as the Hughes (1968) phenomenon. Although ultraspectral technologies can acquire higher precision spectral information than that of hyperspectral, the high dimension issue probably causes a decrease of the classification accuracy. This is a critical issue which limits the ultraspectral technologies from being applied to remote sensing.

Moreover, in hyper/ultraspectral detectors, also well known as spectroradiometer systems, the relationships among signal-to-noise ratio (SNR), resolution, and measurement time have been called the spectroscopic trading rules. Specifically, SNR is in direct proportion to the square root of the scan time \(t\), the spectral resolution \(\Delta \nu\) and the radiation flux \(R\), i.e. SNR \(\propto \sqrt{t \Delta \nu R}\) (Griffiths and De Haseth 2007). When \(R\) and \(t\) are fixed, the SNR decreases with the improvement of the spectral resolution. In low SNR situation, the Euclidean distance between detected spectra and real spectra increases, which probably degrades the performance of matching-based classification algorithms. Furthermore, noise may produce over-fitting and yield poor results. Therefore, how to address the classification accuracy decrease problem caused by the increase of spectral resolution is another critical issue in ultraspectral signature classification.

To address the above-mentioned issues, we adopt restricted Boltzmann machines (RBMs) (Hinton and Salakhutdinov 2006) for infrared ultraspectral signature classification. RBMs have been used as generative models for many different applications including object recognition (Nair and Hinton 2009), phone recognition (Dahl, Mohamed, and Hinton et al. 2010), and human motion modelling (Taylor, Hinton, and Roweis 2006). To the best of our knowledge, it has not yet been used in the hyper/ultraspectral field. The traditional RBM is based on nonlinear neural network and statistical mechanics, which can automatically extract data features and reduce data dimension. It can also be stacked as multi-layer neural network to enhance the performance of convergence affected by error spreading. However, the traditional RBM also has the over-fitting issues caused by noise.

In this article, to avoid the over-fitting issues, we use a zero-mean Gaussian regularization term in the objective function, which can be interpreted as assuming a prior on the model’s parameters. Then, a max-likelihood objective function can be transferred into a max-posterior fashion which could improve the inference ability of the model. Nair and Hinton (2009) claimed that classification performance can be improved by using features that are only rarely activated, thus adding the sparsity regularization term is a direct way for improvement. Concerning sparsity regularizations, \(\ell_0\) norm is intractable and needs
combinatorial search, while $\ell_1$ norm is well achievable but has great approximation error. To obtain better classification accuracy, we propose to use an arctan-like constraint which is able to achieve better approximation to the $\ell_0$ norm compared with the $\ell_1$ norm. Finally, according to Salakhutdinov and Murray (2008)’s research, multi-layer RBM networks can improve the variational lower bound of the model and better approximate the global minimum. In this article, we use several RBMs to establish the deep belief network (DBN) (Hinton 2009) which is a typical multi-layer RBM network.

2. Related work
Over the past years, two general approaches have been investigated for spectral signature characterization (Chang et al. 2009). The first is a coding-based approach, which encodes spectral signatures as code words. Then, spectral analysis is conducted by using the Hamming distance as a spectral similarity measure. A typical example is the binary coding (BC) (Jia and Richards 1993). It compares the detected value of each waveband with a threshold, and then transforms the comparison results into binary digits. The algorithm has very low complexity. However, due to the significant error in the quantization process, it may lose some important spectral information. Besides, it cannot classify spectra within the same class, due to the homogeneity of spectral signatures in the same class (Jiao, Zhong, and Zhang 2012). To address these problems, other algorithms were proposed, e.g. spectral feature-based binary coding (SFBC) (Qian et al. 1996) uses extra bits to encode the slope of two neighbouring bands and the amplitude variation of each band about their mean deviation. Spectral derivative feature coding (SDFC) (Chang et al. 2009) defines four types of the successive gradient changes in spectral values among three consecutive adjacent bands. It also uses extra bits to encode the four types of gradient changes. All of these algorithms use extra bits to encode spectral features such as spectral derivatives and neighbouring waveband differences, which can take more spectral information into consideration. The second general approach is a signature estimation-based approach which estimates spectral profiles for signatures, and spectral analysis is then carried out by using the commonly used least squares error as a criterion for optimality, such as methods based on the wavelet and Kalman filter (Chang et al. 2009; Ghiyamat et al. 2015).

The above-mentioned two kinds of methods have shown their good performance in multispectral or even hyperspectral applications. But, for ultraspectral technology, with its higher spectral resolution and data dimension, the traditional methods cannot well meet the needs in high precision and robust detection such as greenhouse and target detection applications. Recently, Fang et al. (2013) proposed a crosscut feature extraction matching method (CF), which can rapidly fulfil the matching process and be implemented in real-time with excellent classification accuracy. However, it uses the crosscut intersections as the feature and adopts the Euclidean distance for matching, which could make the accuracy decrease due to its sensitivity to noise. Jiao, Zhong, and Zhang (2012) proposed artificial deoxyribonucleic acid (DNA)-based spectral matching (ADSM), which combines SDFC with a genetic method and shows excellent processing efficiency. Besides, some feature matching methods (Ma et al. 2014; Ma, Qiu, et al. 2015; Ma, Zhao, Ma, et al. 2015; Ma, Zhou, Zhao, et al. 2015) in the computer vision area can also be generalized for spectral signature classification, but they have a prerequisite that the spectral features should be extracted in advance.
3. Preliminary

An RBM is a Markov random field (MRF) associated with an undirected graph (Hinton 2012) as shown in Figure 1. It comprises of \( m \) visible logistic units \( V = (V_1, \ldots, V_m) \) denoting the observable data, and \( n \) hidden logistic units \( H = (H_1, \ldots, H_n) \) to capture the dependencies between the observed variables. The random variables \((V, H)\) take values \( (v, h) \in \{0, 1\}^{m+n} \) and the joint probability distribution in this model is characterized by the Gibbs distribution

\[
p(v, h) = \frac{1}{Z} \exp\left(-\sum_{v, h} E(v, h)\right) \quad (\text{where } Z = \sum_{v, h} E(v, h) \text{ is the normalization constant})
\]

with the energy function

\[
E(v, h) = -\sum_{i=1}^{n} \sum_{j=1}^{m} w_{i,j} v_i h_j - \sum_{i=1}^{n} b_i v_i - \sum_{i=1}^{n} c_i h_i. \quad (1)
\]

For all \( i \in \{1, \ldots, n\} \) and \( j \in \{1, \ldots, m\} \), \( w_{i,j} \) is a weighting variable associated with the edge between units \( V_j \) and \( H_i \). \( v_j \) is the value of \( V_j \) and \( h_i \) is that of \( H_i \). \( b_i \) and \( c_i \) are bias terms associated with the \( i \)th visible and the \( i \)th hidden node variable, respectively.

Different from the Boltzmann machines (BMs) (Aarts and Korst 1988), the graph of an RBM has only connections between the layer of hidden and visible variables. In terms of probability, this means that the hidden variables are independent given the state of the visible variables and vice versa:

\[
p(h|v) = \prod_{i=1}^{n} p(h_i|v) \quad \text{and} \quad p(v|h) = \prod_{j=1}^{m} p(v_j|h). \quad (2)
\]

The marginal distribution of the visible variables can be calculated by

\[
p(v) = \frac{1}{2} \sum_h p(v, h) = \frac{1}{2} \sum_h \exp(-E(v, h)) = \frac{1}{2} \prod_{j=1}^{m} \exp(b_j v_j) \prod_{i=1}^{n} \left(1 + \exp\left(c_i + \sum_{j=1}^{m} w_{i,j} v_j\right)\right). \quad (3)
\]

The RBM can be interpreted as a stochastic neural network, where nodes and edges correspond to neurons and synaptic connections, respectively. The conditional probability of a single variable being one can be interpreted as the firing rate of a (stochastic) neuron with the activation function \( \text{sigmoid}(x) = 1/(1 + \exp(-x)) \), because it implies (Fischer and Igel 2012)
\[ p(H_i = 1|v) = \text{sigmoid} \left( \sum_{j=1}^{m} w_{i,j}v_j + c_i \right) \]  

(4)

and

\[ p(V_j = 1|h) = \text{sigmoid} \left( \sum_{i=1}^{n} w_{i,j}h_i + b_j \right). \]  

(5)

For data such as natural images or spectra, logistic units are a very poor representation. One strategy is to replace the binary visible units by linear units with independent Gaussian noise, which guarantee the value of \( v_j \) to be continuous and remain in \([0, 1]\). The energy function and the conditional probabilities then become (Hinton 2012)

\[
E(v, h) = -\sum_{i,j} \frac{v_i}{\sigma_j} h_i w_{ij} + \sum_{j \in \text{vis}} \frac{(v_j - b_j)^2}{2\sigma_j^2} - \sum_{i \in \text{hid}} c_i h_i, 
\]

(6)

\[
p(V_j|h) = \mathcal{N} \left( \sum_{i \in \text{hid}} w_{i,j}h_i + b_j, \sigma_j^2 \right),
\]

(7)

\[
p(H_i = 1|v) = \text{sigmoid} \left( \frac{1}{\sigma_j^2} \sum_{j \in \text{vis}} w_{i,j}v_j + c_i \right),
\]

(8)

where \( \sigma_j^2 \) is the variance of the Gaussian white noise for visible unit \( j \), and \( \mathcal{N}(\cdot) \) represents the normal distribution. ‘vis’ and ‘hid’ denote the sets of the indices of visible and hidden units, respectively.

The learning of the RBM network is based on the maximum likelihood estimation (MLE) with the given training set. The objective function of the training process is given by

\[
\max_{\theta} \left\{ \mathcal{L} (\theta|v) \right\} = \max_{\theta} \left\{ p (v|\theta) \right\} = \max_{\theta} \left\{ \prod_h p(v, h|\theta) \right\}. 
\]

(9)

The likelihood \( \ldots : \theta \to \mathbb{R} \) of the data set maps parameter vector \( \theta \) from a parameter space \( \theta \) to \( \prod_h p(v, h|\theta) \).

Usually the problem formulation expressed by Equation (9) can be rewritten in a log-likelihood form. For training set \( S = \{v_1, v_2, \ldots, v_L\} \), the objective function becomes

\[
\max_{\theta} \ln \mathcal{L} (\theta|v_l) = \max_{\theta} \left\{ \ln \left( \sum_{h_l} p(v_l, h_l|\theta) \right) \right\}, 
\]

(10)

where \( L \) is the number of training samples, \( v_l \) is the \( l \)th sample in \( S \), and \( h_l \) is the hidden layer corresponding to \( v_l \).

The model of the form Equation (10) can be solved by the gradient descent algorithms. For a single training example \( v_l \), the log-likelihood becomes
\[
\ln \mathcal{L}(\theta|v_l) = \ln p(v_l|\theta) = \ln \frac{1}{Z} \sum_{h_l} \exp\left(-E(v_l, h_l)\right) = \ln \sum_{h_l} \exp\left(-E(v_l, h_l)\right) - \sum_{v_l, h_l} \exp\left(-E(v_l, h_l)\right)
\]

(11)

and for the gradient we get

\[
\frac{\partial \ln \mathcal{L}(\theta|v_l)}{\partial \theta} = \frac{\partial}{\partial \theta} \left( \ln \sum_{h_l} \exp\left(-E(v_l, h_l)\right) \right) - \frac{\partial}{\partial \theta} \left( \ln \sum_{v_l, h_l} \exp\left(-E(v_l, h_l)\right) \right)
\]

\[
= - \frac{1}{\sum_{h_l} \exp\left(-E(v_l, h_l)\right)} \sum_{h_l} \exp\left(-E(v_l, h_l)\right) \frac{\partial E(v_l, h_l)}{\partial \theta}
\]

\[
+ \frac{1}{\sum_{v_l, h_l} \exp\left(-E(v_l, h_l)\right)} \sum_{v_l, h_l} \exp\left(-E(v_l, h_l)\right) \frac{\partial E(v_l, h_l)}{\partial \theta}
\]

\[
= - \sum_{h_l} p(h_l|v_l) \frac{\partial E(v_l, h_l)}{\partial \theta} + \sum_{v_l, h_l} p(v_l, h_l) \frac{\partial E(v_l, h_l)}{\partial \theta}.
\]

(12)

Note that the last expression of equality Equation (12) is the difference of two expectations: the expected values of the energy function under the model distribution and under the conditional distribution of the hidden variables given the training sample \(v_l\). Directly calculating the two expectations, which run over all values of the respective variables, leads to a computational complexity which is in general exponential with the number of variables of the MRF. To avoid this computational burden, the expectations can be approximated by samples drawn from the corresponding distributions based on Markov chain Monte Carlo (MCMC) techniques such as Gibbs sampling and \(k\)-step contrastive divergence (CD-\(k\), Hinton 2002).

4. The proposed RBM with sparse and prior constraints

4.1. Sparse constraint on activated nodes of hidden layers

As shown in Figure 1, the hidden layer nodes take values in \(\{0, 1\}^m\), and they are refereed as being activated when they take value 1. Nair and Hinton (2009) claimed that the classification performance can be improved by using features that are only rarely active. A straightforward way is to use sparse constraint in the hidden layer nodes. Lee, Ekanadham, and Ng (2008) proposed a sparse variant of DBN called the sparse deep belief net (SparseDBN) algorithm, which held promise for modelling higher-order features. It uses a regularization term to sparsify the activated nodes of hidden layers, and the regularization term makes the sparseness of the hidden layer nodes maintain at a fixed level of \(p\). This algorithm can effectively make the activated nodes of the hidden layer sparse. However, the selection of \(p\) has a great impact on the convergence of the RBM, and a bad selection of \(p\) may lead to the RBM that is hard to converge.

\(\ell_0\) norm \(\|x\|_0, x \in \mathbb{R}^m\) has been proved to be the best measurement for the level of sparseness (Donoho 2006). It is a common sense approach which simply seeks the simplest explanation fitting the data. However, optimization problems with \(\ell_0\) norm are a non-convex optimization problems, which often need complicated combinatorial search
(Candes, Wakin, and Boyd 2008). Alternatively, these problems are converted into convex optimization problems by using the $\ell_1$ norm ($||x||_1, x \in \mathbb{R}^n$). In this article, to achieve better approximation to the $\ell_0$ norm and simultaneously preserve the convexity, we propose to use the penalty function for a scalar $x$ as follows:

$$f_{\text{arctan}, \varepsilon}(x) = \arctan\left(\frac{|x|}{\varepsilon}\right),$$

(13)

where $\varepsilon$ is a constant. $f_{\text{arctan}, \varepsilon}$ tends to the $\ell_0$ norm as $\varepsilon \to 0$. The comparison among the penalty functions of the $\ell_0$ norm, $\ell_1$ norm, and the arctan-like function is shown in Figure 2. It is obvious that Equation (13) can better approximate to $\ell_0$ than $\ell_1$ norm.

According to Equation (13), the average activated node rate can be constrained by

$$L_{\text{arctan}} = \sum_{l=1}^{L} \sum_{i=1}^{n} \arctan\left(\frac{|E(h_{i,l}|v_l)|}{\varepsilon}\right),$$

(14)

where $h_{i,l}$ is the value of the $i$th hidden layer node of $h_l$. $E(\cdot|\cdot)$ denotes the conditional expectation and $E(h_{i,l}|v_l)$ represents the average rate of the activated nodes in the hidden layer, which can be calculated by

$$E(h_{i,l}|v_l) = \sum_{h_{i,l}} p(h_{i,l}|v_l) h_{i,l} = p(h_{i,l} = 1|v_l) 1 + p(h_{i,l} = 0|v_l) 0$$

$$= p(h_{i,l} = 1|v_l).$$

(15)

Hence, the objective function (see Equation (10)) becomes

$$\max_{\theta} \left\{ \sum_{l=1}^{L} \ln \left( \sum_{h_l} p(v_l, h_l|\theta) \right) - \lambda L_{\text{arctan}} \right\},$$

(16)

Figure 2. The penalty functions of the three different types of norms for sparse constraint.
where \( \lambda \) is the Lagrange coefficient. According to Equation (15), for the gradient of \( L_{\text{arctan}} \) we get

\[
\frac{\partial L_{\text{arctan}}}{\partial w_{i,j}} = \frac{\partial}{\partial w_{i,j}} \sum_{l=1}^{L} \sum_{i=1}^{n} \arctan(\frac{|\mathbb{E}(h_{i,l}|v_i)|}{\varepsilon}) \]

\[
= \sum_{l=1}^{L} \sum_{i=1}^{n} \frac{\partial}{\partial w_{i,j}} \arctan(p(h_{i,l} = 1|v_i)/\varepsilon) \]

\[
= \frac{1}{\varepsilon} \sum_{l=1}^{L} \sum_{i=1}^{n} \frac{p(h_{i,l} = 1|v_i) [1 - p(h_{i,l} = 1|v_i)] v_{i,l}}{1 + (p(h_{i,l} = 1|v_i)/\varepsilon)^2}.
\]

Similarly, we have

\[
\frac{\partial L_{\text{arctan}}}{\partial b_j} = \frac{1}{\varepsilon} \sum_{l=1}^{L} \sum_{i=1}^{n} \frac{p(h_{i,l} = 1|v_i) [1 - p(h_{i,l} = 1|v_i)]}{1 + (p(h_{i,l} = 1|v_i)/\varepsilon)^2} \quad \text{and} \quad \frac{\partial L_{\text{arctan}}}{\partial c_i} = 0,
\]

where \( p(h_{i,l} = 1|v_i) = \text{sigmoid}(\frac{1}{\sigma_f}(\sum_{i=1}^{m} w_{i,j}v_i + b_j)) \).

4.2. Prior constraint on parameters of RBM

As previously mentioned, in the spectroradiometer systems, the higher spectral resolution could degrade the signal-to-noise ratio. We have analysed the objective function of RBM, and the optimal solution may be too loose, which may lead to over-fitting caused by noise. The regularization strategy is often adopted to avoid the over-fitting phenomenon, which adds a penalty term to the objective function in order to discourage the coefficients from reaching large values.

We design the regularization term in the Bayesian framework to avoid the problem of over-fitting, and the framework can be stated in words ‘posteriori \( \propto \) likelihood \( \times \) prior’. Thus, the MLE problem (see Equation (9)) can be transformed to a maximum a posteriori (MAP) problem

\[
\text{MAP:} \quad \max_{\theta} \{p(v|\theta)p(\theta)\} = \max_{\theta} \left\{ \prod_{v \in S} \prod_{h} p(v, h|\theta) p(\theta) \right\}. \tag{19}
\]

Assume that the parameter vector \( \theta = \{w_{i,j}, b_j, c_i\} \) follows normal standard distribution, i.e. \( p(\theta) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\theta^2}{2}\right) \). By omitting the constant of \( p(\theta) \) and using log-MAP expression, the optimization problem (see Equation (10)) becomes

\[
\max_{\theta} \left\{ \sum_{l=1}^{L} \ln \left( \sum_{h_l} p(v_l, h_l|\theta) \right) - \eta \frac{\|\theta\|^2}{2} \right\}, \tag{20}
\]

where \( \eta \) represents the weight constant of the prior term.
4.3. The proposed RBM with sparse and prior constraints

According to Equations (16) and (20), we proposed the infrared ultraspectral signature classification method based on RBM with sparse and prior constraints. The problem formulation of the proposed method is given by

\[
\gamma(v_l, h_l, \theta) \propto \max_\theta \left( \sum_{l=1}^L \ln \left( \sum_{h_l} p(v_l, h_l | \theta) \right) - \lambda L_{\text{arctan}} - \eta \frac{\| \theta \|^2}{2} \right) \tag{21}
\]

Training of RBMs is based on gradient-descent optimization. According to Equation (12), we get the gradient

\[
\frac{\partial \gamma(v_l, h_l, \theta)}{\partial w_{i,j}} = - \sum_{h_l} p(h_l | v_l) \frac{\partial E(v_l, h_l)}{\partial w_{i,j}} + \sum_{v_i, h_l} p(v_l, h_l) \frac{\partial E(v_l, h_l)}{\partial w_{i,j}} - \lambda \frac{\partial L_{\text{arctan}}}{\partial w_{i,j}} - \eta \frac{\partial \| \theta \|^2}{2}
\]

\[
= \sum_{h_l} p(h_l | v_l) h_{l,i} v_{l,j} - \sum_{v_i, h_l} p(v_l, h_l) h_{l,i} v_{l,j} - \lambda \frac{\partial L_{\text{arctan}}}{\partial w_{i,j}} - \eta w_{i,j} \tag{22}
\]

To compute Equation (22), we need to compute \( \sum_{h_l} p(h_l | v_l) h_{l,i} v_{l,j} \) at first. Let \( h_{-i,l} \) denote the state of all hidden units except the \( i \)th one in \( h_l \). Then, we have

\[
\sum_{h_l} p(h_l | v_l) h_{l,i} v_{l,j} = \sum_{h_l} p(h_{k,l} | v_l) h_{l,i} v_{l,j} = \sum_{h_{-i,l}} \sum_{h_{i,l}} p(h_{i,l} | v_l) p(h_{-i,l} | v_l) = \sum_{h_{-i,l}} p(h_{-i,l} | v_l) \sum_{h_{i,l}} p(h_{i,l} | v_l) h_{i,l} v_{l,j} \tag{23}
\]

\[
= \sum_{h_{-i,l}} p(h_{-i,l} | v_l) \sum_{h_{i,l}} p(h_{i,l} | v_l) h_{i,l} v_{l,j} = \sum_{h_{i,l}} p(h_{i,l} | v_l) v_{l,j} = p(h_{i,l} = 1 | v_l) v_{l,j}.
\]

Based on Equations (22) and (23), we obtain

\[
\frac{\partial \gamma(v_l, h_l, \theta)}{\partial w_{i,j}} = p(h_{i,l} = 1 | v_l) v_{l,j} - \sum_{v_i} p(v_l) p(h_{i,l} = 1 | v_l) v_{l,j} - \lambda \frac{\partial L_{\text{arctan}}}{\partial w_{i,j}} - \eta w_{i,j}
\]

\[
= p(h_{i,l} = 1 | v_l) v_{l,j} - \mathbb{E}_{p(v_l)}(p(h_{i,l} = 1 | v_l) v_{l,j}) - \lambda \frac{\partial L_{\text{arctan}}}{\partial w_{i,j}} - \eta w_{i,j}, \tag{24}
\]

where \( \mathbb{E}_{p(v_l)}(\cdot) \) is an expectation over \( p(v_l) \). It is intractable because it needs to run over \( 2^m \) (\( m \) is the number of visible units) states of the visible layer and \( m \) is usually very large for
ultraspectral data. To avoid the exponential complexity of summing over all values of the visible variables, we can approximate this expectation by sampling from the model distribution. It can be implemented by Gibbs sampling. However, it requires running the Markov chain ‘long enough’ to ensure convergence to stationarity. Since the computational costs of such an MCMC approach are still too large, the CD-$k$ algorithm has been proposed by Hinton (2002) to solve the problem. The idea of CD-$k$ is quite simple: instead of approximating the second term in the log-likelihood gradient by a sample from the RBM-distribution, a Gibbs chain runs for only $k$ steps (and usually $k = 1$). The Gibbs chain is initialized with a training example $v^{(0)}$ of the training set and yields the sample $v^{(k)}$ after $k$ steps. Each step $t$ consists of sampling $h^{(t)}$ from $p(h|v^{(t)})$ and sampling $v^{(t+1)}$ from $p(v|h^{(t)})$ subsequently. Specifically, suppose the visible nodes $V$ take the value of $v(v \in \mathbb{R}^m)$, the CD-$k$ algorithm is implemented as follows:

$$
\begin{align*}
    v^{(0)} & \leftarrow v; \\
    h^{(0)} & \sim p(h|v^{(0)}) \\
    v^{(1)} & \sim p(v|h^{(0)}); \\
    h^{(1)} & \sim p(h|v^{(1)}); \\
    \ldots \ldots \\
    h^{(k)} & \sim p(h|v^{(k)}) \\
    v^{(k+1)} & \sim p(v|h^{(k)}),
\end{align*}
$$

where the symbol ‘~’ denotes the sampling operation. When the sampling process is done, the expectation in Equation (24) can be obtained by

$$
\mathbb{E}_{p(w)}(p(h_{i,l} = 1|v_{j,l}) \approx p(h_{i,l} = 1|v^{(k)}_{j,l}).
$$

The detailed proof of Equation (26) is shown in the work of Fischer and Igel 2014. Thus, Equation (24) becomes

$$
\frac{\partial y(v_i, h_i; \theta)}{\partial w_{i,j}} = p(h_{i,l} = 1|v^{(0)}_{j,l})v^{(0)}_{j,l} - p(h_{i,l} = 1|v^{(k)}_{j,l})v^{(k)}_{j,l} - \lambda \frac{\partial \text{arctan}}{\partial w_{i,j}} - \eta w_{i,j}.
$$

According to Equation (17), we then have

$$
\frac{\partial y(v_i, h_i; \theta)}{\partial w_{i,j}} = p(h_{i,l} = 1|v^{(0)}_{j,l})v^{(0)}_{j,l} - p(h_{i,l} = 1|v^{(k)}_{j,l})v^{(k)}_{j,l}
- \lambda \sum_{i=1}^{L} \sum_{j=1}^{n} p(h_{i,l} = 1|v^{(0)}_{i,j})\left[1 - p(h_{i,l} = 1|v^{(0)}_{i,j})]\right]v^{(0)}_{j,l} + \left(p(h_{i,l} = 1|v^{(0)}_{i,j})/\epsilon\right)^2 - \eta w_{i,j}.
$$

Similarly, we have
\[
\frac{\partial f(v_{t}, h_{t}, \theta)}{\partial b_{j}} = v_{j}^{(t)} - v_{j}^{(k)} - \eta w_{i,j}
\]
\[
\frac{\partial (v_{i}, h_{i}, \theta)}{\partial c_{i}} = p(h_{i,l} = 1 | v_{i}^{(t)}) - p(h_{i,l} = 1 | v_{i}^{(k)}) - \eta w_{i,j}
\]

The detailed learning procedures for single one training sample are shown in Algorithm 1. Note that the sampling operation in Algorithm 1 is based on the inverse cumulative distribution function (CDF) (Bengio et al. 2007). Specifically, for a given distribution with CDF, \(F(\cdot)\), a scalar \(U\) is sampled uniformly from \([0, 1]\), then the inverse CDF, \(F^{-1}(U)\), is calculated and it is taken as the sampling result. Sampling \(h_{i}\) from distribution \(p(h_{i} = 1 | v)\) can be given by

\[
h_{i} = F^{-1}(U) = \frac{\ln(1 - U(1 - \exp(\alpha(v))))}{\alpha(v)}
\]

where \(\alpha(v) = \frac{1}{\sigma} \sum_{j} w_{i,j} v_{j} + c_{i}\). Similarly, we can sample \(v_{i}\) from distribution \(p(v_{i} | h)\).

**Algorithm 1:** The proposed RBM learning procedures for one training sample.

**Function updateRBM(\(v, W\))**

1: initialize \(w_{i,j} = 0, b_{j} = 0, c_{i} = 0, \Delta w_{i,j} = 0, \Delta b_{j} = 0, \Delta c_{i} = 0\);
2: \(v^{(0)} \leftarrow v\);
3: for \(t = 0, \ldots, k - 1\) do
4:      for \(i = 1, \ldots, n; j = 1, \ldots, m\) do
5:         sample \(h_{i}^{(t)} \sim p(h_{i} = 1 | v^{(t)})\);
6:         sample \(v_{j}^{(t+1)} \sim p(v_{j} | h^{(t)})\);
7:      end for
8:  end for
9: for \(i = 1, \ldots, n, j = 1, \ldots, m\) do
10: \(\Delta w_{i,j} \leftarrow \Delta w_{i,j} + p(h_{i} = 1 | v^{(0)}) v_{j}^{(0)} - p(h_{i} = 1 | v^{(k)}) v_{j}^{(k)} - \frac{\eta}{\sigma} \sum_{j} w_{i,j} v_{j} + c_{i}\);
11: \(\Delta b_{j} \leftarrow \Delta b_{j} + v_{j}^{(0)} - v_{j}^{(k)} - \frac{\eta}{\sigma} \sum_{i} p(h_{i} = 1 | v^{(0)}) [1 - p(h_{i} = 1 | v^{(k)})] - \eta b_{j}\);
12: \(\Delta c_{i} \leftarrow \Delta c_{i} + p(h_{i} = 1 | v^{(0)}) - p(h_{i} = 1 | v^{(k)}) - \eta c_{i}\);
13: \(w_{i,j} \leftarrow w_{i,j} + \Delta w_{i,j}\);
14: \(b_{j} \leftarrow b_{j} + \Delta b_{j}\);
15: \(c_{i} \leftarrow c_{i} + \Delta c_{i}\);
16: end for

*\(v\) denotes the single training sample and the weight matrix is denoted by \(W = (w_{i,j}, b_{j}, c_{i})\), for \(i = 1, \ldots, n, j = 1, \ldots, m\).*
5. The learning and testing of the proposed RBM

Salakhutdinov and Murray (2008) demonstrated that the multi-layer structure of RBM can improve the variational lower bound of the logarithmic likelihood function, which approaches the global optimum as close as possible. In order to improve the accuracy of the classification method, we use the DBN model described in the work of Hinton and Salakhutdinov (2006).

A DBN can be viewed as a composition of RBMs (Hinton 2009), which models the joint distribution between observed vector $V$ and $\ell$ hidden layers which are described by the vectors $H_k$ ($k = 1, \ldots, \ell$). The generative model formulation of DBN is given by

$$p(V, H_1, \ldots, H_\ell) = \left( \prod_{k=1}^{\ell-2} p(H_k|H_{k+1}) \right) p(H_{\ell-1}, H_\ell), \quad (32)$$

where $V = H_0, p(H_{k-1}|H_k)$ is a visible–given–hidden conditional distribution for visible units in an RBM associated with level $k$ of the DBN, and $p(H_{\ell-1}, H_\ell)$ is the visible–hidden joint distribution in the top-level RBM. The DBN with three hidden layer nodes is illustrated in Figure 3. Note that, in the proposed method, the RBMs of the traditional DBN are replaced by the RBMs we proposed in Section 4. Learning of this model involves two processes: the pre-training process and the fine-tuning process.

- Pre-training
  1. Train the first layer as an RBM that models the raw input $v = H_0$ as its visible layer.
  2. As outlined above, use that first layer to obtain a representation of the input data that will be used as data for the second layer, e.g. take hidden layer samples from $p(H_1|H_0)$ for this representation.
  3. Train the second layer as an RBM, taking the sampled data of (2) as the training example (for the visible layer of that RBM).

![Figure 3. The network graph of DBN+softmax classifiers.](#)
Iterate (2) and (3)) for the desired number of layers, each time propagating upward samples.

• Fine-tuning
  1. Use the softmax classifier to calculate the output vector $z$ as

  $$z = \frac{\exp(H_3W_c)}{\text{sum}(\exp(H_3W_c))},$$  

  where $z$ is a vector, each element is the probability of a certain class for the given input pattern $v$, i.e. $z_i = p(\text{class} = i | v)$. Then, the class can be decided by

  $$\text{class} = \max(z).$$

  2. Use the conjugated gradient algorithm to optimize the parameter matrices $W_1, W_2, W_2, W_2,$ and $W_c$. For the detailed training process, please refer to Hinton and Salakhutdinov (2006) and Hinton (2012).

Testing of DBN is just propagating upward samples. The detailed DBN learning and testing procedures are given in Algorithms 2 and 3, respectively.

Algorithm 2: The learning procedure of DBN.

Procedure

Input: training set $S = \{v_1, v_2, \ldots, v_L\}$, and its corresponding label set $T = \{t_1, \ldots, t_L\}$, where $t_i$ is represented by a $K$(the number of classes)-dimensional vector in which one of the elements is equal to 1, and all the remaining elements are equal to 0.

Output: the $j$th layer parameter matrix $W_j$ and the classifier parameter matrix $W_c$.

1: • Step 1: pre-training DBN
2: randomly initialize the order of samples in $S$;
3: initialize $H_1, H_2, H_3$ randomly;
4: for all $v \in S$ do
5:   for $j = 1, \ldots, 3$ do
6:     initialize $W_j = 0$;
7:     $H_0 = v$;
8:     for $k = 1$ to $j - 1$ do
9:       for $i = 1, \ldots, n$ do
10:          sample $H_{i,k} \sim p(H_{i,k} = 1 | H_{k-1})$;
11:       end for
12:     end for
13:     updateRBM($H_{j-1}, W_j$);
14:   end for
15: end for
16: • Step 2: fine-tuning DBN
17: randomly initialize the order of samples in $S$ again;
18: initialize the parameters of classifier $W_c$ randomly;
19: for all $v \in S$ do
20: calculate $z$ and find the maximum element of it;
21: optimize the parameter matrices ($W_1, \ldots, W_2, W_{c}$) through the conjugated gradient algorithm;
22: end for
end Procedure
3. $H_i (i = 1, 2, 3)$ denotes the $i$th hidden layer nodes in Figure 3.
10. $H_{i,k}$ denotes the $i$th node in $H_k$, referred to Equation (4).
13. Refer to Algorithm 1.
20. Refer to Equations (33) and (34).

Algorithm 3: The testing procedure of DBN.

Procedure
Input: testing set $S = \{v_1, \ldots, v_L\}$, and the corresponding label set $T = \{t_1, \ldots, t_L\}$.
Output: error rate.
1: $errcnt = 0$
2: for all $v \in S$ do
3:   $H_0 \leftarrow v$
4:   for $j = 1, \ldots, 3$ do
5:     for $i = 1, \ldots, n$ do
6:       sample $H_{i,j} \sim p(H_{i,j}|H_{j-1})$
7:     end for
8:   end for
9:   propagate upwards $H_3$ to the top layer classifier to obtain the label $t_{result}$;
10:  if $t_{result} \neq t(v)$ then
11:    $errcnt \leftarrow errcnt + 1$
12:  end if
13: end for
14: error rate = $\frac{errcnt}{L} \times 100\%$
end Procedure

1. $errcnt$ denotes the error count.
6. Refer to Equation (4).
9. Refer to Equations (33) and (34).
10. $t(v)$ is the corresponding label of $v$.

6. Experiments
The proposed method was coded in Matlab and performed on a computer with an Intel core i5 central processing unit at 3.5 GHz and 8 GB random access memory. Tests are on two spectral libraries which are provided by the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) (available at http://speclib.jpl.nasa.gov) and the United States Environmental Protection Agency (EPA) (available at http://www.epa.gov/ttn/emc/ftir/refnam.html). Consistent comparisons between the proposed method and BC (Jia and Richards 1993), SFBC (Qian et al. 1996), SDFC (Chang et al. 2009), CF (Fang et al. 2013), ADSM (Jiao, Zhong, and Zhang 2012), DBN (Hinton and Salakhutdinov 2006), and SparseDBN (Lee, Ekanadham, and Ng 2008) were completed.

Note that, for simplicity, we only consider the classification problem with five classes of spectra, i.e. the number of classes is set to 5 for the remainder of the article.

6.1. Data sets and settings
The proposed method is based on semi-supervised learning. Therefore, training, testing sets, and initial parameter settings for the model are necessary.
6.1.1. Training and testing sets

As mentioned above, we conducted experiments on two libraries provided by ASTER and EPA. In order to investigate the classification performance of the algorithms under challenging situations, five classes of spectra in each library are chosen for classification, which are in the same category with high similarity. The similarity measure is based on the spectral angle mapper (SAM) (Van der Meer 2006) which is defined by

$$\text{SAM}(s_i, s_j) = \cos^{-1}\left(\frac{\sum_{l=1}^{L} s_{i,l} s_{j,l}}{(\sum_{l=1}^{L} s_{i,l}^2)^{1/2} (\sum_{l=1}^{L} s_{j,l}^2)^{1/2}}\right),$$  (35)

where $s_i$ and $s_j$ denote two different spectral vectors. $s_{i,l}$ and $s_{j,l}$ are the $l$th entries of $s_i$ and $s_j$, respectively. The value of SAM ranges from 0 to 1, and the smaller value of SAM represents higher similarity between two spectra. The five selected spectra from the two libraries are shown in Figures 4 and 5, and the spectral similarities among the five different spectra are shown in Figure 6, which are calculated by Equation (35).

Specifically, for the spectral library provided by ASTER, we select five classes of spectra in the manmade category with attributes as follows: ‘General Construction Materials; Type: Manmade materials; Class: the General Construction Materials; Subclass: Paints; Particle Size: Solid’. The spectra with sample labels 0403uuu, 0405uuu, 0406uuu, 0407uuu, and 0408uuu are marked with spec1 to spec5, respectively. For the spectral library provided by EPA, the five selected classes of spectra consist of four samples called ‘1,1-Dimethyl hydrazine’ at 99.8, 99.9, 494.4, and 500 ppm, and one sample called ‘1,2-Propylenimine’ at 99.9 ppm. For convenience, we also mark them with spec1 to spec5, respectively. The five selected spectra of the two libraries with marks are shown in Figures 4 and 5, respectively.

The training set for the ASTER provided spectral library is generated by adding different levels of Gaussian white noise to the five spectra. The SNR from 30 to 50 dB are divided into 500 parts and the corresponding SNR of each part is set for each spectrum. Therefore, each spectrum is added with 500 different levels of noise. Finally, the training set with 2500 samples is created and we call it the ‘ASTER data set’.

![Figure 4](image-url)  
Figure 4. The five selected spectra from the spectral library provided by ASTER. For convenience, these spectra named by the sample labels 0403uuu, 0405uuu, 0406uuu, 0407uuu, and 0408uuu are marked with spec1 to spec5, respectively.
Figure 5. The five selected spectra from the spectral library provided by EPA. It consists of four samples called ‘1,1-Dimethyl hydrazine’ at different concentrations (99.8, 99.9, 494.4, and 500 ppm) and one sample called ‘1,2-Propylenimine’ at 99.9 ppm, which are also marked with spec1 to spec5, respectively.

Figure 6. The similarity of the selected spectra for the two data sets. (a) The spectral similarity between the five selected spectra from the spectral library provided by ASTER. (b) The spectral similarity between the five selected spectra from the spectral library provided by EPA.

Figure 7. The five representative spectra obtained from the ASTER data set. They are obtained by adding Gaussian white noise (SNR = 40 dB) to the original spectra in Figure 4.
Similarly, the EPA data set can be created. The five representative spectra of ASTER and EPA data sets with noise are shown in Figures 7 and 8.

The way of creating testing sets is similar to that of the training sets. The difference between them is just the noise levels which are cut down to 100, i.e. the five spectra are turned into 500 samples for ASTER and EPA data set experiments, respectively.

6.1.2. Parameter settings

The DBN and the softmax classifier are adopted in the experiments, which are shown in Figure 3. The number of nodes from the visible layer to the top hidden layer for the ASTER experiment and the EPA experiment are 4287, 3000, 3000, 1000 and 33,185, 20,000, 5000, 1000, respectively. The output for both ASTER and EPA experiments are five nodes. The detailed procedures of the method are given in Algorithms 2 and 3. Parameter settings are listed as follows:

- $\lambda$: it is selected according to the classification accuracy. Through the experiments, we find that higher classification accuracy can be achieved when $\lambda$ is between 0.1 and 0.001. Here we set $\lambda = 0.02$.
- $\gamma$: it is also selected according to the classification accuracy. This parameter is primarily to avoid over-fitting caused by the noise. Here we set $\gamma = 0.0002$.
- $\varepsilon$: it is used to adjust the degree of approximation to the $\ell_0$ norm by the arctan-like constraint. We tested the value from 0.1 to 0.0001. The closer the $\varepsilon$ approaches to 0, the more the sparse $H_1$ is. However, smaller $\varepsilon$ may not allow the network to easier converge, and the sparseness of $H_1$ rapidly decreases to 0 in a few iterations. Here, we set $\varepsilon = 0.005$.
- $\sigma_j$: denotes the standard deviation of the Gaussian white noise adding to the $j$th node of the visible layer. For simplicity, we set $\sigma_j = 1$ for $j=1, \ldots, m$.
- $K$: The total number of classes is set to 5.

Figure 8. The five representative spectra obtained from the EPA data set. They are obtained by adding Gaussian white noise (SNR = 45 dB) to the original spectra in Figure 5.
6.2. Experimental results

To validate the performance of our method, two comparison experiments are conducted. One focuses on the sparseness and convergence performance, and the other on classification accuracy.

6.2.1. Quantitative comparison of sparseness and convergence

The comparison of sparseness and convergence performance is conducted during the training phase among DBN, SparseDBN, and the proposed method. First, we only consider the sparseness degree of $H_1$ because it is the critical layer for feature extraction. Hoyer (2004)’s sparseness

$$\text{sparseness}(x) = \frac{\sqrt{n} - \left(\sum_{i=1}^{n} |x_i|\right) / \sqrt{\sum_{i=1}^{n} x_i^2}}{\sqrt{n} - 1}$$  \hspace{1cm} (36)$$

is adopted as the similarity measure, where $x$ denotes a spectrum vector, $n$ denotes the cardinality of $x$, and $x_i$ denotes the $i$th entry of $x$. The larger value of Equation (36) implies less activated nodes.

The results are shown in Table 1. It should be noted that there is no optimization on the sparseness of the hidden layer nodes for the traditional DBN, but the sparseness is still high, in other words, very few nodes of $H_1$ are activated. This may be due to the fact that the features of spectra are mainly the peaks and valleys, and the spectra are generally smooth, which makes the sparseness of DBN high even without sparse constraint. SparseDBN has better performance of sparseness, which increases about 2% and 3% compared with DBN on the two training sets. In contrast, the sparseness of our proposed method has the best sparseness on the two training sets compared with DBN and SparseDBN, which increases at least 2% and 6%, respectively. Moreover, experiments on the EPA data set have lower sparseness than that on the ASTER data set, which is because that the EPA training set has higher spectral resolution, and then it has more features and needs more activated nodes in $H_1$ to carry the information. This experiment validates the effectiveness of our proposed arctan-like constraint for sparseness in the hidden layers of DBN, and it has achieved better performance compared with the traditional DBN and SparseDBN.

Second, we analyse the convergence of the network including DBN, SparseDBN, and the proposed algorithm. As mentioned before, noise may bring over-fitting issues for the networks, and hence the classification accuracy significantly decreases, which may seriously affect the robustness of the algorithm. We fix the number of iterations to 60 for the ASTER data set and 200 for the EPA data set, and the convergence performances of the three algorithms on two data sets are shown in Figures 9 and 10, respectively. From the results, we see that at the beginning, the accuracies of different algorithms are

<table>
<thead>
<tr>
<th>Training set</th>
<th>DBN</th>
<th>SparseDBN</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASTER</td>
<td>0.9003</td>
<td>0.9214</td>
<td>0.9442</td>
</tr>
<tr>
<td>EPA</td>
<td>0.7749</td>
<td>0.8023</td>
<td>0.8635</td>
</tr>
</tbody>
</table>
different. The performances of SparseDBN and the proposed algorithm are obviously better than that of DBN on the two data sets, which proves that the sparse constraint of the proposed method contributes to the feature extraction. In Figures 9 and 10, the curves of the proposed method converge faster than the others, and they illustrate the lowest error rate. These results demonstrate that the proposed method has better convergence performance than the others.

Finally, we test the performance of the Gaussian prior constraint. Experiments using the proposed method both with and without the prior constraint are conducted on the two training sets. The results are shown in Figures 11 and 12. We see that the error rate...
decreases more quickly by adding the prior constraint, as it provides the prior knowledge of the parameters, which effectively avoids the over-fitting caused by the noise.

6.2.2. Quantitative comparison of classification performance

The proposed method is compared with other seven algorithms: BC, SFBC, SDFC, CF, ADSM, DBN, and SparseDBN at different SNRs. Each spectrum in a spectral library is picked out, and we add noise ranging from 30 to 50 dB to it for one classification test. Every test is performed for 20 times and then the average error $e_{\text{ASTER}}^{\text{av}}$ and $e_{\text{EPA}}^{\text{av}}$, for the two spectral libraries, respectively, is calculated. Note that the proposed model is trained beforehand according to Algorithm 2.

Tables 2–4 illustrate the results of the experiments for the accuracy comparisons. The BC method shows good accuracy performance at different noise levels. Here, the BC method we used is the traditional binary coding algorithm with just one threshold, which is set to the mean of the entire spectral radiation value. The accuracy performance of BC has obvious differences between the spectral libraries provided by ASTER and EPA. The results with the ASTER provided spectral library are better than that of the EPA provided library. This is because the profiles of ASTER’s spectra are relatively smoother than that of EPA’s spectra. If the profiles of spectra have too many steep and large slopes at low SNRs, the noise could affect the coding process, which may cause the problem that a

Table 2. Accuracy comparisons of different algorithms on ASTER and EPA spectral libraries for an SNR of 45.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$e_{\text{ASTER}}^{\text{av}}$</th>
<th>$e_{\text{EPA}}^{\text{av}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC</td>
<td>0.083</td>
<td>0.251</td>
</tr>
<tr>
<td>SFBC</td>
<td>0.088</td>
<td>0.113</td>
</tr>
<tr>
<td>SDFC</td>
<td>58.100</td>
<td>21.879</td>
</tr>
<tr>
<td>CF</td>
<td>0.008</td>
<td>0.097</td>
</tr>
<tr>
<td>ADSM</td>
<td>0.023</td>
<td>0.163</td>
</tr>
<tr>
<td>DBN</td>
<td>0.057</td>
<td>0.301</td>
</tr>
<tr>
<td>SparseDBN</td>
<td>0.010</td>
<td>0.323</td>
</tr>
<tr>
<td>Proposed method</td>
<td><strong>0.003</strong></td>
<td><strong>0.081</strong></td>
</tr>
</tbody>
</table>
correct binary bit could be turned into a false one. The SFBC method generally has better accuracy performance than the BC. This is because the SFBC includes the BC coding binary bits while uses extra binary bits to describe the inter-band information. The extra bits include the slope bits and the mean derivation bits, which describe the slopes between bands and the variations of amplitude about mean derivations, respectively. The SDFC method has the poorest accuracy performance in the experiments. This is because the SDFC includes the BC method while uses extra binary bits to describe the derivative information. The noise we added is additive white Gaussian noise, which could make the derivate bits fluctuate and disable the extraction of the band-wise information. The CF shows good results but without good robustness. It uses a number of equally spaced horizontal lines to intersect with the spectral curve and takes the total of intersections for each line. Finally, the total intersections over each line compose a vector which is taken as the feature of the spectrum. The spectral discrimination between two spectra can be executed by measuring the Euclidean distance between the feature vectors of the two spectra. Due to the dimensionality reduction via feature extraction process, this method has low complexity which can be used in high real-time applications, and it also has good classification accuracy according to the experiments. However, it has some critical flaws. First, the distance measured by the Euclidean distance is sensitive to noise and the
threshold needs sophisticated design. Second, if the spectrum is smooth and relatively monotonous, there may be few intersections which could decrease the robustness of the method. The ADSM algorithm combines the SDFC with the genetic method. It is a good combination that decreases the negative effect on SDFC caused by Gaussian noise. This method shows good results at different noise levels.

The above-mentioned ones are good algorithms which can be used in ultraspectral signature classification. However, they do not show good performance of robustness to noise. From the results, we can see that there are obvious accuracy gaps between different SNRs for the algorithms. DBN, sparseDBN, and the proposed method show the stabllest performance at different SNRs because they used the learning way to train the models at different noise levels. The results also demonstrate that the proposed method can improve the accuracy performance by adding the sparse and Gaussian prior constraints.

7. Conclusion

In this article, we analysed the high dimension and low SNR problems caused by the high spectral resolution in ultraspectral technology, which probably decrease the classification accuracy and cause over-fitting problems. To address these issues, we proposed a new method for infrared ultraspectral signature classification, which is based on the RBM with an arctan-like sparse constraint and a Gaussian prior regularization term. Overall, our proposed method indicates good potential for infrared ultraspectral signature classification. The comparison experiments are conducted on two spectral libraries provided by ASTER and EPA. Compared with the traditional spectral signature classification algorithms, our method rarely shows big fluctuations at three typical SNR levels (SNR = 40, 45, 50 dB), which implies that the proposed method is more robust to noise. The analyses of the sparse and prior constraints are shown and the results illustrate that the proposed method can improve the sparseness and converge faster than the traditional RBM-based methods. More importantly, the accuracy of the proposed method is superior to those of other compared methods.

Our future work will mainly cover the RBM-based ultraspectral data classification. The following points still need to be addressed in the future: (1) The training complexity. Due to the complexity of the training phase, the proposed method is not very suitable for real-time applications. More RBM training strategies should be compared and adopted; (2) The optimization issue. The proposed method uses the conjugated gradient optimization in the fine-tuning phase which is faster than the traditional gradient descent algorithms. However, it is also likely to be trapped in the local minimum. The simulated annealing algorithm will be analysed and adopted; (3) Training data issue. The training data may increase a little bit to adapt to new environment. When training data sets change, models should be trained again. It is unwise and time consuming for large data sets. Therefore, this issue should be taken into consideration.

Acknowledgements

We would like to thank the anonymous reviewers and the editor for their constructive comments that helped improve the manuscript.

Disclosure statement

No potential conflict of interest was reported by the authors.
Funding
Financial support for this study was provided by the National Natural Science Foundation of China under Grant [61275098]; the PhD Programs Foundation of the Ministry of Education of China under Grant [20120142110088]; the Natural Science Foundation of Hubei Province of China under Grants [2011CDB027], [2014CFB268]; the China Postdoctoral Science Foundation under Grants [2015M572194], [2015M570665].

References


