2012

Sparse Coding for Hyperspectral Images Using Random Dictionary and Soft Thresholding

Ender Oguslu
Old Dominion University

Khan Iftekharuddin
Old Dominion University, kiftekha@odu.edu

Jiang Li
Old Dominion University, jli@odu.edu

Mark Allen Neifeld (Ed.)

Amit Ashok (Ed.)

Original Publication Citation

Oguslu, E., Iftekharuddin, K., & Li, J. (2012) Sparse coding for hyperspectral images using random dictionary and soft thresholding. In M.A. Neifeld and A. Ashok (Eds.), *Visual Information Processing XXI, Proceedings of SPIE 8399* (83990A). SPIE of Bellingham, WA. https://doi.org/10.1117/12.919162

Sparse Coding for Hyperspectral Images using Random Dictionary and Soft Thresholding

Ender Oguslu, Khan Iftekharuddin and Jiang Li
Department of Electrical and Computer Engineering
Old Dominion University, VA, USA 23529

ABSTRACT

Many techniques have been recently developed for classification of hyperspectral images (HSI) including support vector machines (SVMs), neural networks and graph-based methods. To achieve good performances for the classification, a good feature representation of the HSI is essential. A great deal of feature extraction algorithms have been developed such as principal component analysis (PCA) and independent component analysis (ICA). Sparse coding has recently shown state-of-the-art performances in many applications including image classification. In this paper, we present a feature extraction method for HSI data motivated by a recently developed sparse coding based image representation technique. Sparse coding consists of a dictionary learning step and an encoding step. In the learning step, we compared two different methods, L_1-penalized sparse coding and random selection for the dictionary learning. In the encoding step, we utilized a soft threshold activation function to obtain feature representations for HSI. We applied the proposed algorithm to a HSI dataset collected at the Kennedy Space Center (KSC) and compared our results with those obtained by a recently proposed method, supervised locally linear embedding weighted *k*-nearest-neighbor (SLLE-W*k*NN) classifier. We have achieved better performances on this dataset in terms of the overall accuracy with a random dictionary. We conclude that this simple feature extraction framework might lead to more efficient HSI classification systems.

Keywords: Sparse coding, sparse representation, dictionary learning, feature extraction, remote sensing, hyperspectral imagery, hyperspectral classification.

1. INTRODUCTION

Hyperspectral sensors are the devices that obtain ground reflectance measurements across hundreds of very narrow spectral bands throughout the visible, near-infrared and mid-infrared portions of the electromagnetic spectrum. By measuring radiation over several small wavelength ranges, the sensor builds up a hyperspectral image (HSI) with a very high spectral resolution, making it a useful modality for fine differentiation between ground objects [1]. It also has the potential for more accurate and detailed information extraction than any other types of remote sensing data for determining terrain properties such as material classification, geological feature identification and environmental monitoring [2].

The spectral features in hyperspectral data contain significant structures and a proper characterization of these structures may result in improved data analysis. HSI data contains hundreds of bands and they are often highly correlated. It is usually assumed that low-dimensional manifolds are embedded in the high dimensional space. To extract the low-dimensional manifolds for further processing, both linear and nonlinear methods were investigated in the literature [3-9]. Linear techniques such as principal component analysis (PCA) [3] and locality preserving projections (LPP) [4] seek to reveal linear hidden low-dimensional manifolds in a high-dimensional space globally and locally, respectively. Nonlinear manifold such as isometric feature mapping [5], locally linear embedding (LLE) [6], local tangent space alignment (LTSA) [7] and Laplacian eigenmaps (LE) [8] try to unfold nonlinear manifolds embedded in high-dimensional space. Traditional manifold learning is usually performed in an unsupervised manner but recent work demonstrated excellent performances by incorporating supervised learning into the framework of manifold learning [9].

To achieve good performances for HSI classification, a good feature representation of the HSI is essential. Recent research on sparse coding shows that it can improve classification accuracies significantly and achieved state-of-the-art performances in many computer vision applications [10-15]. In sparse coding, the original data is converted to a new representation by projecting it onto a set of over-completed basis functions. Each basis function contains local or low level features and is a building block for data. Traditional feature extraction techniques, like frequency analysis and

Visual Information Processing XXI, edited by Mark Allen Neifeld, Amit Ashok, Proc. of SPIE
Vol. 8399, 83990A · © 2012 SPIE · CCC code: 0277-786X/12/$18 · doi: 10.1117/12.919162

Gabor Filters, extract features by projecting data onto a set of predefined basis functions without adaptation to the data. In contrast, basis functions in sparse coding are either learned or selected from data, making the feature extraction process adaptive. Since basis functions are over-complete, the achieved representations for data samples in sparse coding are usually sparse where a linear classifier is often sufficient for classification.

In this paper, we have applied the framework of sparse coding to HSI data collected at the Kennedy Space Center (KSC) for land cover classification. We compared two dictionary learning methods: random selection and sparse coding solved with the coordinate descent algorithm [16]. Both methods achieved similar results. Results from sparse coding were also compared to those obtained by a recently proposed method, supervised locally linear embedding weighted *k*-nearest-neighbor (SLLE-WkNN) [9] classifier. We have observed that dictionary learned even from randomly selected dictionary is able to achieve much better performances on the KSC data. To best of our knowledge, this is the first time to apply the sparse coding framework to this hyperspectral dataset for land cover classification and experimentally proved that a randomly selected dictionary can achieve better results than those from many recent algorithms such as SLLE-WkNN. A randomly selected dictionary does not require much computational resources thus making a simple and efficient HSI data classification system possible.

The remaining of this paper is organized as follows. In Section 2, we reviewed the HSI model and related work of sparse coding. The proposed algorithm is presented in Section 3 followed by experimental results in Section 4. Section 5 presents concluding remarks.

2. RELATED WORK

2.1 Model of hyperspectral imagery

HSI data, which is obtained by airborne or spaceborne sensors, consists of hundreds of images captured in different spectral channels. A typical structure of HSI is shown in Figure 1. Every pixel in the image, $x_i \in R^B$, is represented by a *B*-dimensional feature vector throughout relevant portions of the electromagnetic spectrum, where *B* is the number of spectral bands. This feature vector is called the spectrum of the material in this pixel. Though the abundant information in each pixel increases the capability of distinguishing different materials, it is often difficult to exploit HSI data due to the particular challenges in the remote sensing environment such as noise of measurement devices, energy interaction between the targeted area and the spectrometer, and spectral mixing where each pixel is composed of a combination of different materials [2]. All those challenges will jeopardize a precise material identification in HSI.

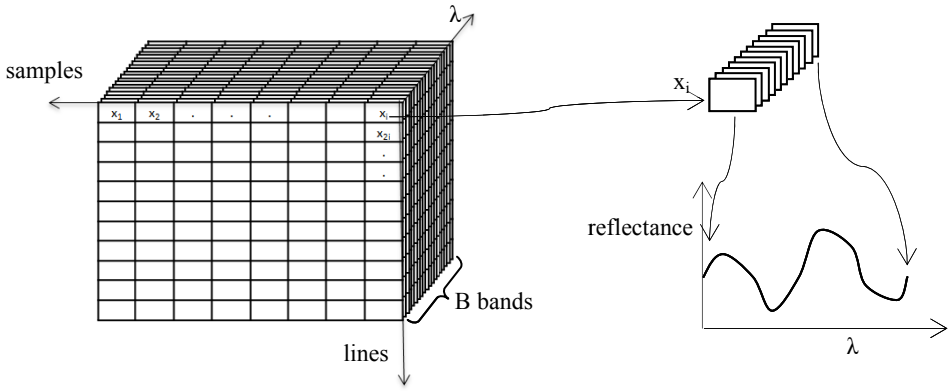


Figure 1. A typical data structure of HSI

To model HSI data, it is often assumed that the measured energy is proportional to the covered ground and the reflectivity of its environment [17]. The most common approach is to use the following linear mixture model,

$$\mathbf{X} = \mathbf{DA} + \mathbf{N} \tag{1}$$

where **X** is the HSI image consists of M pixels with B bands, $\mathbf{X} \in R^{BxM}$, **D** is the dictionary consists of K basis functions (atoms of materials) with a length of B, $\mathbf{D} \in R^{BxK}$, **A** is the coefficient matrix representing the mixture of dictionary functions, $\mathrm{A} \in R^{KxM}$ and **N** is additive noise.

2.2 Sparse coding

As a new feature extraction method, sparse coding attracted a great deal of attentions in image classification [15, 18 - 21]. In this section, we will briefly review the two steps involved in sparse coding: basis functions learning and encoding.

2.2.1 Basis function learning

The goal of sparse coding is to represent raw data (i.e., an image) as a linear combination of few basis functions from a dictionary learned from data. Given a training dataset, $X = \{x^{(i)}\}_{i=1}^{M}$, a dictionary, $D = \{d^{(i)}\}_{i=1}^{K}$, consisting of a set of basis functions, $d^{(i)}$, can be learned based on a L_1-penalized sparse coding formulation by optimizing the following cost function,

$$\min_{D,a^{(i)}} \sum_i \left\| Da^{(i)} - x^{(i)} \right\|_2^2 + \lambda \left\| a^{(i)} \right\|_1$$

$$\text{subject to } \left\| d^{(i)} \right\|_2^2 = 1, \forall i \tag{2}$$

where $x^{(i)}$ represents the i-th data sample in **X**, $a^{(i)}$ denotes the reconstruction weight for $x^{(i)}$ using the basis function in ***D*** and λ is a trade-off parameter. Because the L_1 norm penalty is involved, the resulting weights $a^{(i)}$ will be sparse meaning that most of them are zeros. The solution of equation (2) can be obtained using alternating minimization over the sparse codes and dictionary while holding the other fixed [15]. Equation (2) seems to be consistent to the HSI data model due to its high spatial resolution of latest imagery technology, i.e., only a few land cover materials may be present in one pixel and thus each pixel can be approximated by a few basis functions. Initial research on HSI data in this direction shows promising results [17, 22 - 24].

Dictionary learning plays an important role in the sparse coding framework because it will identify those building blocks from data. However, the learning process is difficult and time consuming. Fortunately, recent research showed that randomly selected dictionaries can also perform well. For example, Jarrett et al. [18] has found that features from a one-layer convolution pooling architecture with a random dictionary could achieve a sufficient recognition rate for image classification. In [15], Coates et al. have experimentally proved that the choice of basis functions does not affect much on classification performances as long as it is over complete.

2.2.2 Encoding

Once a dictionary is learned, an encoding step is performed to transform the input data samples into desirable representations based on the learned dictionary. For a data sample $x^{(i)}$, its representation $a^{(i)}$ can be obtained either by solving equation (2) with ***D*** fixed using the orthogonal matching pursuit (OMP-k) [25] or by the soft thresholding method. The soft thresholding technique [15, 26-28] achieves the sparse representation by applying the following operation,

$$a^{(i)} = sign(z^{(i)}) \max(0, \left| z^{(i)} \right| - t) \tag{3}$$

where t is an adjustable threshold and $z^{(i)} = D^T x^{(i)}$. This simple and efficient method was utilized for encoding in this paper.

3. PROPOSED METHOD

The proposed method for HSI data classification consists of three steps: (i) patch constructing, (ii) dictionary learning and (iii) encoding and classification.

3.1 Patch constructing from HSI data

The first step of learning dictionaries is to extract patches from unlabeled HSI data as illustrated in Figure 2. We constructed overlapping patches as *b*-dimensional (bands) vectors with a step size of 1 band along the spectral direction for each randomly selected pixel and we called *b* as the receptive field length. If *m* is the number of generated patches, the patches can be denoted as $\mathbf{X}=\{x^{(1)}, x^{(2)}, ..., x^{(m)}\}$, where $x^{(i)} \in R^b$.

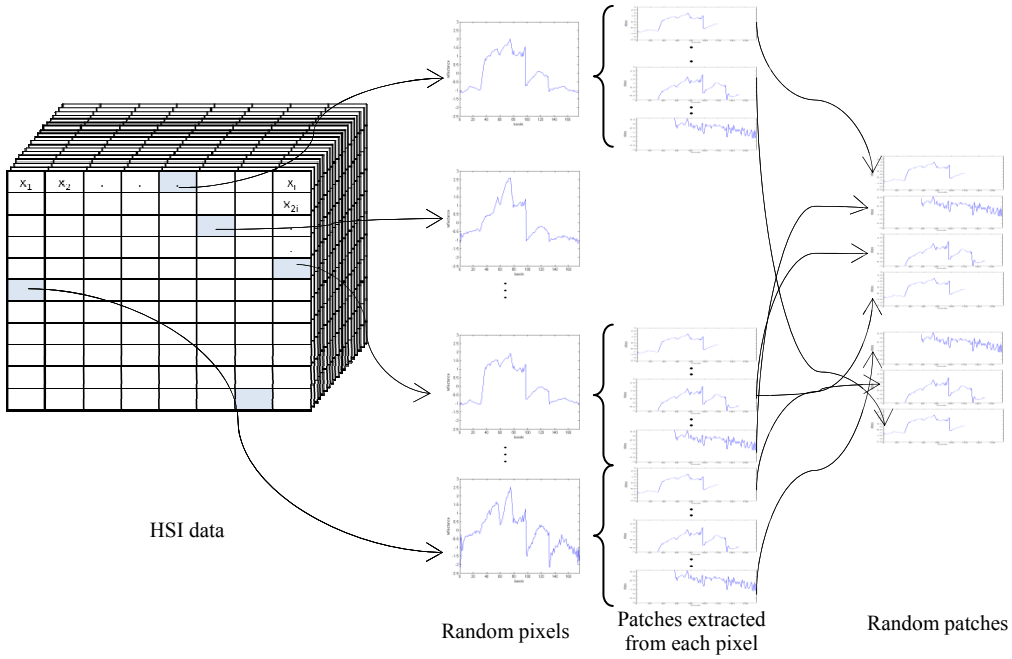


Figure 2. Extraction of patch blocks.

3.2 Dictionary learning based on constructed patches

We utilized two methods for the dictionary learning. The first method is to solve equation (2) where **X** denotes the constructed patches using the coordinate descent algorithm [16], which is often termed as "sparse coding". The second method for dictionary learning is relatively simple where a portion of constructed patches are randomly selected to compose the dictionary.

In sparse coding, data patches are preprocessed before the dictionary learning. First, each patch $x^{(i)}$ is normalized to be zero mean and unit variance. Then, the zero-phase components analysis (ZCA) whitening process [29], which is commonly used in deep learning, is applied to each patch. According to [19], this process is decisive for the quality of the learned feature representations. In this paper, we compared the two approaches for dictionary learning, sparse coding [16] and random selection [15]. This step yields the dictionary **D** for the HSI data.

3.3 Encoding and classification

With the learned dictionary, *D*, we define the feature representation for each of the HSI pixels as follows,

1. Divide the pixel bands into patches the same way as we did in dictionary learning,
2. Normalize and whiten the patches as described above,
3. Obtain a new representation by applying the soft thresholding technique,
4. Sum those new representations from all patches from the pixel to form the final representation.

The final step is called feature pooling. For each HSI pixel, we have extracted multiple patches and the feature pooling step can reduce the dimensionality of the final representation. Note that different pooling methods have been investigated for different applications [21] and any of them can be applied here. We applied sum pooling method which we split the featured patches into *k* equal-sized blocks and compute the sum of the vectors in each block. Once we obtain

the final representation vector for each pixel, we apply a linear support vector machine (SVM) classifier to classify the HSI data to different land cover categories.

4. EXPERIMENTAL RESULTS

4.1 Data description

A hyperspectral data set collected by NASA Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) over Kennedy Space Center (KSC) in March 1996 [30] was used in this paper. One single band image from this data set is shown in Figure 3. This sensor can acquire 224 bands with an 18-m spatial resolution and a 10-nm spectral resolution over the range of 400-2500 nm. There are 5211 out of 314,368 (512x614) pixels labeled. Details of the land cover classes are given in Table 1. After removing the water absorption and noisy bands, there are 176 bands left for this study.

Figure 3. A sample band of Kennedy Space Center HSI data.

Table 1. Class names and number of labeled KSC data.

Class	Class	Number/Percentage of Labeled
1	Scrub	761 (14.6%)
2	Willow swamp	243 (4.66%)
3	Cabbage palm hammock	256 (4.92%)
4	Cabbage/oak hammock	252 (4.84%)
5	Slash pine	161 (3.07%)
6	Oak/broadleaf hammock	229 (4.38%)
7	Hardwood swamp	105 (2.0%)
8	Graminoid marsh	431 (8.27%)
9	Spartina marsh	520 (9.9%)
10	Cattail marsh	404 (7.76%)
11	Salt marsh	419 (8.04%)
12	Mud flats	503 (9.66%)
13	Water	927 (17.8%)
	TOTAL	5211

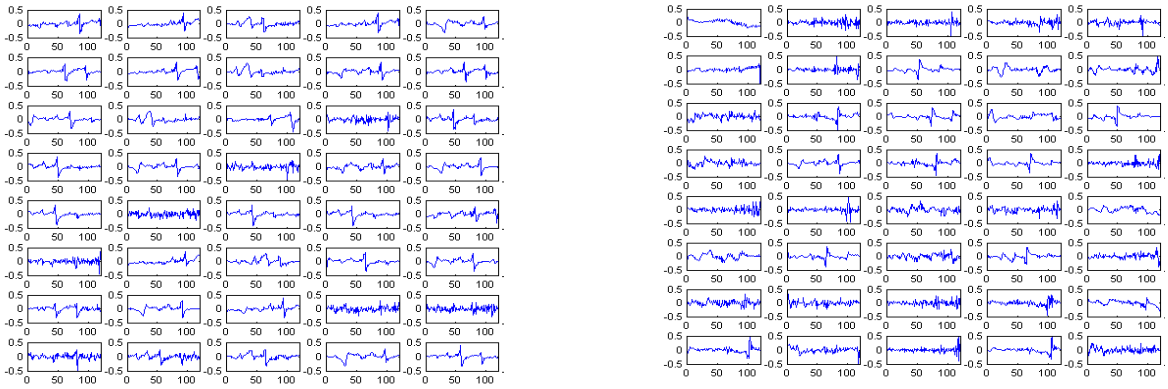
4.2 Experiment setup

For a fair comparison, we utilized the same configuration as that in [9] for our experiments. The parameters involved in our algorithm were tuned based on three-fold cross-validation using 50% of the labeled data. There were four parameters in the proposed method: (i) the number of basis functions, K, (ii) split number, k, (iii) receptive field length, b, and (iv) threshold, t, for encoding. After parameter selection, 30% and 40% of the data are randomly selected for testing and training, respectively. The testing dataset is disjoint from the data used for cross-validation. In our experiments, dictionary was learned or randomly selected from unlabeled pixels.

In the followings, we first present the effects of those four parameters on the classification performances. We then compare the two dictionary learning methods and finally, the final classification results will be compared with a recent successful algorithm, supervised locally linear embedding weighted *k* nearest neighbor (SLLE-W*k*NN) classifier [9].

4.3 Dictionaries learned

We obtained basis functions either by sparse coding or random selection. In Figure 4, two set of normalized basis functions resulting from sparse coding and randomly sampled patches are illustrated, respectively.



(a) (b)

Figure 4. Dictionaries obtained by (a) sparse coding (b) randomly selected patches.

4.4 Effects of the number of basis functions and the splitting number

We investigated the effect of the number of basis functions on classification performances by varying *K* from 50 to 350 with a step size of 50. Effect of the splitting numbers (*k* = 4 to 10) was also tested and results are listed in Table 2. It is observed that increasing the number of basis function will improve the classification performance until a maximum value is reached. The splitting number, *k*, which is used in pooling step, also increases the accuracy though it is less sensitive. The best performance can be obtained with different combinations of *K* and *k.* It should be noted that this table was generated based on a random basis function dictionary. Similar results were observed from sparse coding. For the other parameters in this comparison, we set the receptive field *b* as120, and the value, *t*, as 0.1 for soft thresholding.

Table 2. Effect of number of futures and splitting number on performance.

Number of bases (*K*) \ Splitting Number (*k*)	4	5	6	7	8	9	10
	Three-fold Cross-validation results (%) (Numbers in parenthesis indicates the length of final vector)						
50	93.20 (400)	93.34 (500)	93.70 (600)	93.84 (700)	93.95 (800)	93.80 (900)	93.69 (1000)
100	94.01 (800)	94.27 (1000)	94.30 (1200)	94.36 (1400)	94.29 (1600)	94.21 (1800)	94.13 (2000)
150	94.17 (1200)	94.32 (1500)	94.32 (1800)	94.32 (2100)	94.26 (2400)	94.21 (2700)	94.21 (3000)
200	94.36 (1600)	94.41 (2000)	94.48 (2400)	**94.57** (2800)	94.32 (3200)	94.26 (3600)	94.26 (4000)
250	94.41 (2000)	94.50 (2500)	**94.57** (3000)	94.47 (3500)	94.47 (4000)	94.38 (4500)	94.41 (5000)
300	94.41 (2400)	94.42 (3000)	94.52 (3600)	94.56 (4200)	94.56 (4800)	94.38 (5400)	94.41 (6000)
350	94.43 (2800)	94.40 (3500)	94.52 (4200)	94.54 (4900)	94.52 (5600)	94.41 (6300)	94.41 (7000)

4.5 Effect of the receptive field length

One of the important parameters in the algorithm is the receptive field length, *b*. It is observed that a larger receptive field could allow recognizing more complex patterns in the image [19]. On the other hand, a larger receptive field will extract less number of patches for each input HSI pixel decreasing the discriminating power.

Let the number of overlapped patches that can be extracted for each input pixel is (*B*-*b*+1), where *B* is the number of bands and *b* is the receptive field length, it yields (*B*-*b*+1)-by-*b* matrix to represent each input pixel before encoding. Table 3 illustrates the evaluation results with different receptive field lengths (varying from 20 to 160 with a step size of 20). It is observed that the best accuracy can be obtained if the receptive field length is 120. Note that this table was generated based on a randomly selected dictionary. Similar results were observed using a dictionary learned by sparse coding. For the other parameters, we used K=250 for the number of basis functions, k=6 for the splitting number and t=0.1 for the soft thresholding value in encoding.

Table 3. Effect of receptive field length

Receptive field length (*b*)	20	40	60	80	100	120	140	160
Number of patches for each pixel (*B*-*b*+1)	157	137	117	97	77	57	37	17
Number of elements to represent a pixel. (*B*-*b*+1)-by-*b*	3140	5480	7020	7760	7700	6840	5180	2720
Performance (%)	92.56	93.25	94.26	94.51	95.18	95.21	94.92	93.93

4.6 Effect of the threshold value in soft thresholding

The last parameter of the algorithm is the threshold, *t*, used in the soft thresholding step for encoding. As pointed out earlier, the soft threshold function was successfully used in some algorithms to mimic the sparse coding. This function is also called shrinkage function since it eliminates the insignificant representations of pixels. Table 4 illustrates the effect of threshold value *t*, used in encoding with the other parameters set as K=250, k=6 and b=120. It is observed that the performance is not sensitive to the threshold and t=0.1 gives the best accuracy for this dataset.

Table 4. Effect of fixed threshold point, *t*.

Threshold for Encoding (*t*)	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40
Performance (%)	94.37	94.70	94.67	94.58	94.46	94.43	94.34	94.30

4.7 Final classification results

As stated in the previous sections, one of our goals is to compare the random selection method and the sparse coding technique for dictionary learning in remote sensing classification. With the best parameters (k=6, b=120 and t=0.1) we obtained in cross-validation, we performed 10 replications of experiments with different number of basis functions in the dictionary to compare these two approaches. It should be noted that 40% of labeled pixels were used for training and 30% of labeled pixels were used for testing in each of the experiments. As seen in the Table 5, the randomly selected dictionaries can achieve similar or slightly better results than those obtained by sparse coding. The success of the randomly selected dictionaries implies that we can implement a very efficient system for remote sensing.

Table 5. Comparison of dictionary learning algorithms

Number of basis functions (*K*)	50	100	150	200	250	300
Performance of sparse coding (%)	94.18	94.64	94.73	94.87	94.85	94.69
Performance of random functions (%)	94.27	94.80	95.09	95.15	95.21	95.12

We also compared our results with those obtained by a recently proposed method for HSI classification, supervised locally linear embedding weighted *k* nearest neighbor (SLLE-W*k*NN) classifier [9], in terms of overall accuracy. In SLLE-W*k*NN classifier, a kernel function of locally linear embedding (LLE) algorithm is employed to determine the weights of a weighted *k*NN classifier. In other words, the local structure of the distribution of the manifold is learned by employing LLE algorithm in conjunction with the *k*NN classifier [31].

In the comparison, we used the same training and testing datasets for different classifiers. The parameters of the proposed algorithm were K= 250, k=6, b=120 and t=0.1. The parameters of SLLE-W*k*NN were set the same as those reported in [9]. Results from the 10 replications of experiments are shown in Table 6. It is observed that the proposed algorithm with random dictionary is significantly better than the SLLE-W*k*NN method ($p = 8.78 \times 10^{-8}$) and the system with a learned dictionary is also significantly better than the SLLE-WkNN method ($p = 9.43 \times 10^{-7}$). The system with a learned dictionary performs similarly to that with a randomly selected dictionary ($p = 0.1522$).

Table 6. Comparison of proposed algorithm and SLLE-W*k*NN classifier.

Classifier	Classification Accuracy (%)	Standard Deviation	*p* value with respect to SLLE-W*k*NN
Linear SVM using sparse coding for dictionary learning	94.83	0.59	9.43×10^{-7}
Linear SVM with random dictionary	95.21	0.45	8.78×10^{-8}
SLLE-W*k*NN (*k*=35)	93.13	0.52	

Since the quantity of the training data has critical importance in classification tasks, experiments were performed using different number of training data samples, where the ratios of data used for training varied from 10% to 70% and that rate for testing was fixed to 30%. In Table 7, it is clearly that the proposed framework has better overall accuracies than those from the SLLE-W*k*NN algorithm.

Table 7. Comparison of proposed algorithm with the SLLE-W*k*NN classifier with different training rates.

Classifier \ Training Rate	10%	20%	30%	40%	50%	60%	70%
SVM with learned dictionary	91.01	93.29	94.15	94.85	95.13	95.55	95.62
SVM with random dictionary	91.53	93.88	94.59	95.21	95.33	95.56	95.65
SLLE-WkNN (*k*=35)	89.40	91.48	92.62	93.13	93.67	93.91	94.34

5. CONCLUSION

We applied the sparse coding framework to HSI classification and experimentally showed that a randomly selected dictionary can achieve slightly better results than the dictionary learned by sparse coding. Both sparse learning techniques outperformed a recently proposed advanced algorithm, SLLE-WkNN, on a well-known HSI data set collected at KSC. To best of our knowledge, this is the first time to apply the sparse coding framework to this hyperspectral dataset for land cover classification and experimentally proved that a randomly selected dictionary can achieve very good results making a simple and efficient HSI data classification system possible.

6. REFERENCES

[1] http://www.nrcan.gc.ca/sites/www.nrcan.gc.ca.earth-sciences/files/pdf/resource/tutor/fundam/pdf/fundamentals_e.pdf

[2] Chang, C., [Hyperspectral Data Exploitation: Theory and Applications], Wiley, New York, (2007).

[3] Jolliffe, I., [Principal component analysis], Springer-Verlag, New York, (1986).

[4] He X., and Niyogi, P., "Locality Preserving Projections," NIPS, (2003).

[5] Tenenbaum, J. B., Silva, V., and Langford, J. C., "A global geometric framework for nonlinear dimensionality reduction," Science 290(5500), 2319–2323 (2000).

[6] Roweis, S. T. and Saul, L. K., "Nonlinear dimensionality reduction by locally linear embedding," Science 290(5500), 2323-2326 (2000).
[7] Zhang, Z. Y. and Zha, H. Y. L., "Principal manifolds and nonlinear dimension reduction via local tangent space alignment," SIAM J. Sci. Comput. 26(1), 313-338 (2004).
[8] Belkin, M. and Niyogi, P., "Laplacian eigenmaps for dimensionality reduction and data representation," Neural Comput. 15(6), 1373-1396 (2003).
[9] Ma, L., Crawford, M. M., and Tian, J., "Local manifold learning-based k-nearest-neighbor for hyperspectral image classification," IEEE Trans. Geosci. Remote Sens. 48(11), 4099-4109 (2010).
[10] Sprechmann, P., Sapiro, G., "Dictionary learning and sparse coding for unsupervised clustering," ICASSP, (2010).
[11] Mairal, J., Bach F., Ponce J., Sapiro G., Zisserman A., "Discriminative learned dictionaries for local image analysis," CVPR, 1-8 (2008).
[12] Mairal, J., Leordeanu M., Bach F., Hebert M., Ponce J., "Discriminative sparse image models for class-specific edge detection and image interpretation," ECCV, (2008).
[13] Raina R., Battle A., Lee H., Packer B., Ng A.Y., "Self-taught learning: Transfer learning from unlabeled data," ICML, (2007).
[14] Wright J., Yang A. Y., Ganesh A., Sastry S. S., Ma Y., "Robust face recognition via sparse representation," Pattern Analysis and Machine Intelligence 31(2), 210-227 (2008).
[15] Coates, A., Ng, A. Y., Mall, S., "The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization," Proceedings of the 28th International Conference on Machine Learning (ICML), (2011).
[16] Wu, T.T., and Lange, K., "Coordinate descent algorithms for lasso penalized regression," Annals of Applied Statistics 2(1), (2008).
[17] Castrodad A., Xing Z., Greer J.B., Bosch E., Carin L., and Sapiro G., "Learning discriminative sparse representations for modeling, source separation, and mapping of hyperspectral imagery," IEEE Trans. Geosci. Remote Sens. 49(11), 4263-4281 (2011).
[18] Jarrett, K., Kavukcuoglu, K., Ranzato, M., and Le-Cun, Y. "What is the best multi-stage architecture for object recognition?" In International Conference on Computer Vision, (2009).
[19] Coates, A., Honlak, L., Ng, A. Y., "An analysis of single-layer networks in unsupervised feature learning," In International Conference on AI and Statistics, (2011).
[20] H. Lee, A., Battle, R., Raina, and A. Ng., "Efficient sparse coding algorithms", In Advances in Neural Information Processing Systems, (2006).
[21] Boureau, Y., Bach, F., LeCun, Y., and Ponce, J., "Learning mid-level features for recognition," "Computer Vision and Pattern Recognition", (2010).
[22] Charles A. S., Olshausen B. A., and Rozell C. J., "Sparse coding for spectral signatures in hyperspectral images," Proc. Asilomar Conf. Signals Syst. Comput., (2010)
[23] Charles A. S., Olshausen B. A. and Rozell C. J., "Learning sparse codes for hyperspectral imagery," IEEE J. Sel. Topics Appl. Earth Observations Remote Sens., 5(5), 963–978 (2011).
[24] Iordache M. D., Bioucas-Dias J., Plaza A., "Sparse unmixing of hyperspectral data," IEEE Trans. Geosci. Remote Sens. 49(6), 2014-2039 (2011).
[25] Pati, Y. C., Rezaifar, R., and Krishnaprasad, P. S. "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition", In Asilomar Conference on Signals Systems and Computers, (1993).
[26] Kavukcuoglu, K., Ranzato, M., and LeCun, Y., "Fast inference in sparse coding algorithms with applications to object recognition," Technical Report CBLL-TR-2008-12-01 Computational and Biological Learning Lab Courant Institute NYU, (2008).
[27] Nair, V. and Hinton, G. E., "Rectified Linear Units Improve Restricted Boltzmann Machines," International Conference on Machine Learning, (2010).
[28] Krizhevsky, A., "Convolutional Deep Belief Networks on CIFAR-10". Unpublished manuscript, (2010).
[29] Hyvarinen, A., and Oja, E., "Independent component analysis: algorithms and applications," Neural networks 13(4-5), 411-430, (2000).
[30] Ham, J., Yangchi, C., Crawford, M.M., Ghosh, J., "Investigation of the random forest framework for classification of hyperspectral data," Geoscience and Remote Sensing IEEE Transactions 43(3), 492-501 (2005).
[31] Chanussot, J., Crawford, M. M., Kuo, B.-C., "Foreword to the special issue on hyperspectral image and signal processing," Geoscience and Remote Sensing IEEE Transactions 48(11), 3871-3876 (2010).