# Feature Extraction for Hyperspectral Imagery via Ensemble Localized Manifold Learning

Fan Li, *Student Member, IEEE*, Linlin Xu, *Member, IEEE*, Alexander Wong, *Member, IEEE*, and David A. Clausi, *Senior Member, IEEE*

***Abstract*—A feature extraction approach for hyperspectral image classification has been developed. Multiple linear manifolds are learned to characterize the original data based on their locations in the feature space, and an ensemble of classifier is then trained using all these manifolds. Such manifolds are localized in the feature space (which we will refer to as "localized manifolds") and can overcome the difficulty of learning a single global manifold due to the complexity and nonlinearity of hyperspectral data. Two state-of-the-art feature extraction methods are used to implement localized manifolds. Experimental results show that classification accuracy is improved using both localized manifold learning methods on standard hyperspectral data sets.**

***Index Terms*—Ensemble learning, feature extraction, hyperspectral image classification, manifold learning.**

## I. Introduction

CLASSIFICATION of hyperspectral imagery data has been investigated by researchers in both remote sensing and computer science fields in the last decade. The main difficulty of processing hyperspectral data is the "Hughes phenomenon" [1] caused by the high dimensionality of spectral bands, which tends to decrease the classification performance. To minimize the impact of this problem, various feature extraction techniques have been developed as a preprocessing step before classification, so that useful information such as feature structure and class separability can be maintained in the new feature space, while the dimensionality of data is significantly reduced.

In remote sensing, principal component analysis (PCA) is commonly used for feature extraction. PCA determines projections that can preserve maximum variance without using any label information, so the extracted features are not directly related to classification. Linear discriminant analysis (LDA), in contrast, seeks a transformation matrix to minimize the within-class scatter and maximize the between-class scatter, both of which are calculated using label information. In recent years, new algorithms have been developed to improve the standard LDA. Typical methods include nonparametric weighted feature extraction (NWFE) [2] and local Fisher's discriminant analysis (LFDA) [3], which both adopt a locally adaptive scatter matrix and overcome the rank deficiency problem of LDA. These methods have been demonstrated to perform very well for hyperspectral images [2], [4].

The above are linear methods that aim to learn an embedded subspace by a single linear transformation matrix. However, the structure of hyperspectral imaging data is typically so complex that the use of a single linear manifold may result in the loss of useful information, particularly when we want the dimensions of extracted features to be low. A traditional approach to capture nonlinear manifolds of high-dimensional data is using the "kernel trick," such as kernel PCA [5] and generalized discriminant analysis [6].

An alternative method to capture nonlinear manifolds is to use local manifolds or metrics based on the assumption that the data structure is locally linear [7]–[10]. An early attempt is the discriminant adaptive nearest neighbor classification algorithm (DANNC) [7]. This algorithm first uses local LDA to learn metrics for computing neighborhoods and then adapts local neighborhoods based on local decision boundaries by a neighborhood-based classifier. However, the local metrics are learned independently, and thus, the method is prone to overfitting. A more recent approach is the multimetric version of the large-margin nearest neighbor method (LMNN-MM) [8], which first uses $k$-means to split training data into disjoint clusters and then learns a Mahalanobis distance metric for each cluster. Although the number of local metrics is reduced from the number of training samples in DANNC to the number of clusters, overfitting is still unavoidable because the metrics are learned from separate parts of training samples independently. Moreover, such local methods will cause a discontinuity of the metrics near the $k$-means decision boundary. There are also approaches where the local manifolds of the training samples are represented by the weighted linear combination of multiple manifolds [9], [10], but a large number of parameters need to be estimated.

In the last decade, ensemble learning has been widely used to improve the classification performance [11]–[13]. Combining multiple weak classifiers into a classifier ensemble can reduce the variance by a single classifier and thus prevent overfitting. All of the aforementioned issues related to manifold learning, along with the benefits of ensemble learning, motivate the proposed feature extraction method in this letter. Unlike traditional approaches that use feature extraction as a preliminary step, followed by fitting a classifier on the extracted features, here, we first learn multiple manifolds localized in the feature space, and then fit multiple classifiers on features projected in different localized manifolds, and finally combine the classifiers to provide the unified classification decision.

Manuscript received March 18, 2015; revised July 23, 2015 and August 26, 2015; accepted September 23, 2015. Date of publication November 4, 2015; date of current version November 11, 2015. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, by the Canadian Space Agency, and by the Canada Research Chairs Program.

The authors are with the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: lifan1001@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2015.2487226

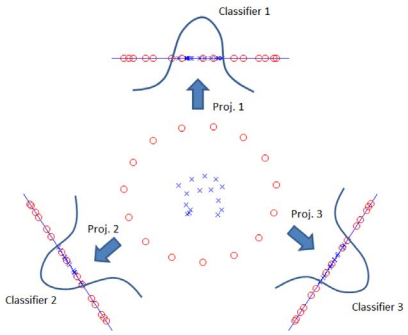


Fig. 1. Simulated example showing why multiple manifolds are better than a single manifold. The red "o" and blue "x" represent two classes in two-dimension feature space. It is obvious that no linear dimension reduction method could project all the samples to one-dimension space and still keep them separable. However, we can project them to three one-dimension manifolds and fit three nonlinear classifiers to each set of projected samples. Although, in each base classifier, some of the o are misclassified as x, these misclassified samples can be finally corrected by majority voting.

The rest of this letter is organized as follows. Section II summarizes the proposed algorithm. In Section III, the implementation details of localized manifolds are covered, and the localization for two feature extraction methods, i.e., L-NWFE and L-LFDA, based on a novel weighting scheme is shown. In Section IV, the performance of the algorithm is evaluated by comparing with those based on a single global manifold on two standard hyperspectral data sets. Section V concludes this letter by summarizing the previous sections.

## II. Overview of the ELML Algorithm

This section overviews the ensemble localized manifold learning (ELML), a joint feature extraction and classification framework that captures the nonlinear structure using multiple linear manifolds. First, the feature space is partitioned into $K$ clusters using a data clustering algorithm. Note that such a cluster is independent of the classes of training samples; thus, one cluster could have training samples of multiple classes. Then, localized manifolds are learned, each focused on one cluster. After $K$ manifolds are learned, $K$ sets of features are extracted using the manifolds. Afterward, a set of base classifiers are trained on each set of features. All the classifiers are finally combined into a final classifier ensemble to improve generalization performance. Fig. 1 shows a simulated example showing why multiple manifolds are better than a single manifold.

**Algorithm 1** The ELML algorithm

**Input**: training data $X$, number of clusters $K$;
**Output**: A classifier ensemble $H(X)$;
1: Partition $X$ into $K$ clusters $\mathcal{C}_k$ $(k = 1, \ldots, K)$;
2: **for** $k = 1$ to $K$: **do**
3: Learn a manifold $\mathcal{M}_k$ that focuses on $\mathcal{C}_k$ : $\mathcal{M}_k \leftarrow \{X, \mathcal{C}_k\}$ $(k = 1, \ldots, K)$;
4: Learn a classifier $h_k$ on data projected in the manifold: $h_k \leftarrow \mathcal{M}_k(X)$;
5: **end for**
6: Combine all the classifiers into the final classifier ensemble: $H(X) = \cup\{h_k(X)\}$ $(k = 1, \ldots, K)$.

The ELML algorithm shown in Algorithm 1 shows in the high level. The implementation details of Algorithm 1 are as follows. For Step 1, we use k-means to partition the data into $K$ clusters due to its simplicity and fast convergence. Step 3, which implements the localized manifolds, is the most important step in the ELML algorithm. We will focus on this step in Section III and present two approaches to learn the manifolds for each cluster. In Step 4, the 1-nearest neighbor (1-NN) classifier is used as the base classifier. 1-NN is a very simple nonlinear classifier with no tuning parameter. It is guaranteed to converge to an error rate less than twice the Bayes' error when the number of samples approaches infinity. NN-based classifiers are capable of capturing the variations of features; thus, they have been widely used to test the performance of feature extraction methods and achieve comparable performance to more complicated classifiers such as support vector machines [14], [15]. In the ELML algorithm, this property can help generate more diversity between the base classifiers. In Step 6, standard majority voting is used to aggregate the predictions by all the base classifiers.

## III. Learning Localized Manifolds

From the theory of ensemble learning, the individual learners should have small bias and be diverse from each other [16]. Different from random sampling methods such as bootstrap sampling [17] and random subspace [18] commonly used in ensemble methods, the diversity of base classifiers is achieved by learning localized features in ELML. The key is thereby to learn the localized manifolds. If the manifold is learned only on a local cluster of data, it might be incapable of reflecting global data structure, and the bias will be thus increased. Moreover, there might be insufficient training samples to learn a manifold in the local part. Therefore, in this letter, the localized manifold is learned from all the training samples using a localization weighting scheme.

Here, we will present two localized algorithms—NWFE [2] and LFDA [3], [4]—that are modified to be able to learn localized manifolds and thus are called localized NWFE (L-NWFE) and localized LFDA (L-LFDA). Like LDA, both L-NWFE and L-LFDA aim to find a transformation $T$ to minimize the within-class scatter matrix and maximize the between-class scatter matrix, i.e.,

$$T = \arg \max_{T \in \mathcal{R}^{d \times r}} \left[ \text{tr} \left\{ \left( T^T S^{(w)} T \right)^{-1} T^T S^{(b)} T \right\} \right] \quad (1)$$

where $S^{(w)}$ and $S^{(b)}$ are the within-class scatter and the between-class scatter in the original feature space, respectively; and $d$ and $r$ are the number of features in the original space and the new space, respectively.

This optimization can be solved by eigendecomposition, i.e., the extracted $r$ features are the $r$ eigenvectors associated with the largest $r$ eigenvalues of $\left(S^{(w)}\right)^{-1} S^{(b)}$. Therefore, the key of this category of algorithms is to define $S^{(w)}$ and $S^{(b)}$. In both proposed methods, new $S^{(w)}$ and $S^{(b)}$ are defined using a novel weighting scheme to incorporate localization, where sample points closer to the local clusters are assigned larger weights. The details of these two methods are described in the rest of this section.

### A. L-NWFE

NWFE is a nonparametric feature extraction method based on scatter matrices. It is modified from the standard LDA by replacing the class mean with the local mean for each sample point. Moreover, the scatter matrices are weighted based on the Euclidean distances from sample points to their local means. More details of the original NWFE algorithm are provided by Kuo and Landgrebe [2].

Given $L$ classes, the nonparametric between-class scatter matrix is defined as [2]

$$S^{(b)} = \sum_{i=1}^{L} P_i \sum_{\substack{j=1 \\ j\neq i}}^{L} \sum_{l=1}^{N_i} \frac{\lambda_l^{i,j}}{N_i} \left(\mathbf{x}_l^i - \mathbf{M}_j\left(\mathbf{x}_l^i\right)\right) \cdot \left(\mathbf{x}_l^i - \mathbf{M}_j\left(\mathbf{x}_l^i\right)\right)^T \tag{2}$$

$$S^{(w)} = \sum_{i=1}^{L} P_i \sum_{l=1}^{N_i} \frac{\lambda_l^{i,i}}{N_i} \left(\mathbf{x}_l^i - \mathbf{M}_i\left(\mathbf{x}_l^i\right)\right) \cdot \left(\mathbf{x}_l^i - \mathbf{M}_i\left(\mathbf{x}_l^i\right)\right)^T \tag{3}$$

where $\mathbf{x}_l^i$ is a feature vector, $N_i$ is the number of samples in class $i$, $P_i$ is the prior probability of class $i$, and the nonparametric local weight $\lambda_l^{i,j}$ is defined as

$$\lambda_l^{i,j} = \frac{\text{dist}\left(\mathbf{x}_l^i, \mathbf{M}_j\left(\mathbf{x}_l^i\right)\right)^{-1}}{\sum_{t=1}^{N_i} \text{dist}\left(\mathbf{x}_t^i, \mathbf{M}_j\left(\mathbf{x}_t^i\right)\right)^{-1}} \tag{4}$$

where $\text{dist}(\cdot,\cdot)$ is the Euclidean distance between two points. The local mean $\mathbf{M}_j(\mathbf{x}_l^i)$ is the weighted mean of $N_j$ sample points in class $j$, i.e.,

$$\mathbf{M}_j\left(\mathbf{x}_l^i\right) = \sum_{m=1}^{N_j} w_{lm}^{i,j} \mathbf{x}_m^j \tag{5}$$

$$w_{lm}^{i,j} = \frac{\text{dist}\left(\mathbf{x}_l^i, \mathbf{x}_m^j\right)^{-1}}{\sum_{t=1}^{N_j} \text{dist}\left(\mathbf{x}_l^i, \mathbf{x}_t^j\right)^{-1}}. \tag{6}$$

To implement localization, we introduce a localized weight $\eta_l^{i,k}$ for the $i$th sample of the $l$th class based on cluster $\mathcal{C}_k$, which is denoted by

$$\eta_l^{i,k} = \exp\left\{-\frac{\text{dist}^2\left(\mathbf{x}_l^i, \mathcal{C}_k\right)}{\sigma^2}\right\} \tag{7}$$

where $\sigma$ is a smoothness constant, and $\text{dist}(\mathbf{x}_l^i, \mathcal{C}_k)$ denotes the Euclidean distance from $\mathbf{x}_l^i$ to cluster $\mathcal{C}_k$. When $\mathbf{x}_l^i$ is in $\mathcal{C}_k$, $\text{dist}(\mathbf{x}_l^i, \mathcal{C}_k) = 0$; otherwise, it is the minimum distance from $\mathbf{x}_l^i$ to any point in $\mathcal{C}_k$, i.e.,

$$\text{dist}\left(\mathbf{x}_l^i, \mathcal{C}_k\right) = \min_j \text{dist}\left(\mathbf{x}_l^i, c_{kj}\right), \quad j = 1, \ldots, N_k \tag{8}$$

where $c_{kj}$ is the $j$th sample point in cluster $\mathcal{C}_k$. Therefore, the localized within-class and between-class scatter matrices are formulated as

$$S_k^{(b)} = \sum_{i=1}^{L} P_i \sum_{\substack{j=1 \\ j\neq i}}^{L} \sum_{l=1}^{N_i} \frac{\eta_l^{i,k} \lambda_l^{i,j}}{N_i} \left(\mathbf{x}_l^i - \mathbf{M}_j\left(\mathbf{x}_l^i\right)\right) \cdot \left(\mathbf{x}_l^i - \mathbf{M}_j\left(\mathbf{x}_l^i\right)\right)^T \tag{9}$$

$$S_k^{(w)} = \sum_{i=1}^{L} P_i \sum_{l=1}^{N_i} \frac{\eta_l^{i,k} \lambda_l^{i,i}}{N_i} \left(\mathbf{x}_l^i - \mathbf{M}_i\left(\mathbf{x}_l^i\right)\right) \cdot \left(\mathbf{x}_l^i - \mathbf{M}_i\left(\mathbf{x}_l^i\right)\right)^T. \tag{10}$$

The localized weight $\sigma$ is used to determine the degree of localization for learning manifolds. When $\sigma \to 0$, the manifold $\mathcal{M}_k$ is learned only from samples in $\mathcal{C}_k$, which is the strategy used in the LMNN-MM method [8]. When $\sigma \to +\infty$, the method is reduced to a single manifold learning algorithm.

### B. L-LFDA

LFDA is another feature extraction and dimension reduction method based on LDA. It combines LDA with locality-preserving projection [19], which aims to preserve the local structure of the data. Similar to LDA and NWFE, LFDA requires to determine within-class and between-class scatter matrices, and the feature extraction problem can be solved by the same generalized eigendecomposition technique. The difference between LFDA and LDA is that the local weights are used for calculating the within-class and between-class scatter matrices [3], i.e.,

$$\tilde{S}^{(w)} = \frac{1}{2} \sum_{i,j=1}^{N} \tilde{W}_{i,j}^{(w)} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \tag{11}$$

$$\tilde{S}^{(b)} = \frac{1}{2} \sum_{i,j=1}^{N} \tilde{W}_{i,j}^{(b)} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \tag{12}$$

where

$$\tilde{W}_{i,j}^{(w)} = \begin{cases} \frac{A_{i,j}}{N_l}, & \text{if } y_i = y_j = l \\ 0, & \text{if } y_i \neq y_j \end{cases} \tag{13}$$

$$\tilde{W}_{i,j}^{(b)} = \begin{cases} A_{i,j}\left(1/N - 1/N_l\right), & \text{if } y_i = y_j = l \\ \frac{1}{N}, & \text{if } y_i \neq y_j \end{cases} \tag{14}$$

where $N_l$ is the number of samples in class $l$, $N$ is the total number of samples, and $A$ is the a sparse affinity matrix, with $A_{i,j}$ representing the similarity between sample points $\mathbf{x}_i$ and $\mathbf{x}_j$. In this letter, $A_{i,j}$ is defined using the local scaling method [20], i.e., $A_{i,j} = \exp\{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(\gamma_i\gamma_j)\}$, where $\gamma_i = \|\mathbf{x}_i - \mathbf{x}_i^{(K')}\|$, and $\mathbf{x}_i^{(K')}$ is the $K'$th nearest neighbor of $\mathbf{x}_i$ and is set to 7 as suggested by Zelnik-Manor and Perona [20]. The details of LFDA can be referred to previous publications by Sugiyama [3] and Li *et al.* [4].

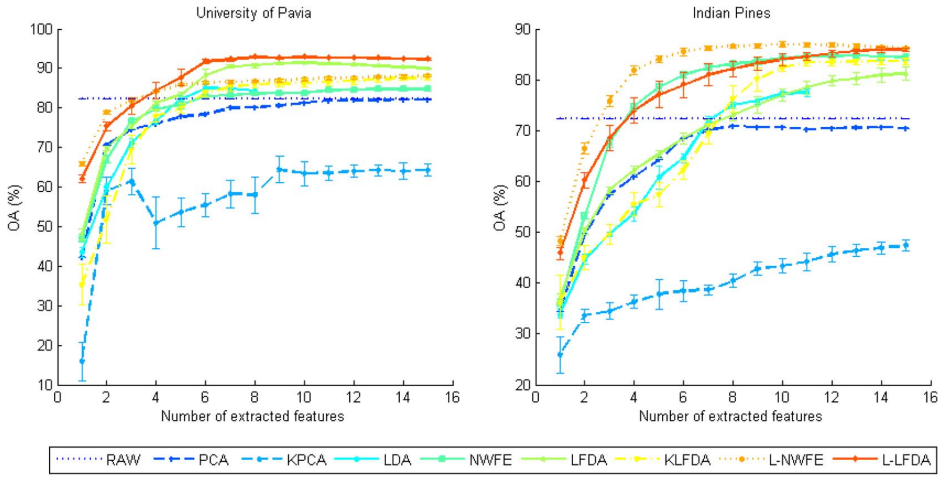


Fig. 2. Classification accuracy (vertical axis) using 1-NN on different numbers of extracted features (horizontal axis) on the two data sets. The dotted horizontal line tagged "RAW" is the classification accuracy using all original features.

TABLE I
HIGHEST CLASSIFICATION ACCURACY ACHIEVED FOR THE TWO DATA SETS (WITH THE CORRESPONDING NUMBER OF EXTRACTED FEATURE DIMENSIONS)

| Dataset | RAW | PCA | KPCA | LDA | NWFE | LFDA | KLFDA | L-NWFE | L-LFDA |
|---|---|---|---|---|---|---|---|---|---|
| University of Pavia | 82.3 | 82.2±0.5 (15) | 64.4±1.0 (9) | 86.6±0.0 (10) | 84.7±0.9 (14) | 91.4±0.0 (10) | 87.6±0.5 (15) | 88.1±0.5 (15) | **92.8±0.5 (8)** |
| Indian Pines | 72.2 | 70.9±0.5 (8) | 47.4±0.5 (15) | 77.5±0.9 (13) | 84.7±0.9 (13) | 81.2±0.5 (15) | 83.7±0.5 (15) | **86.9±0.0 (10)** | 85.9±0.5 (15) |

Similar to L-NWFE, we use a localized weight $\eta_k^{i,j}$ for both $\mathbf{x}_i$ and $\mathbf{x}_j$ to implement localization

$$\tilde{S}_k^{(w)} = \frac{1}{2}\sum_{i,j=1}^{N} \eta_k^{(i,j)} \tilde{W}_{i,j}^{(w)} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \quad (15)$$

$$\tilde{S}_k^{(b)} = \frac{1}{2}\sum_{i,j=1}^{N} \eta_k^{(i,j)} \tilde{W}_{i,j}^{(b)} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \quad (16)$$

where

$$\eta_k^{(i,j)} = \exp\left\{-\frac{\text{dist}^2(\mathbf{x}_i, \mathcal{C}_k) + \text{dist}^2(\mathbf{x}_j, \mathcal{C}_k)}{2\sigma^2}\right\} \quad (17)$$

where $\text{dist}(\mathbf{x}_i, \mathcal{C}_k)$ is again calculated using (8).

## IV. EXPERIMENTAL RESULTS AND ANALYSIS

The classification performances using 1-NN classifiers on the original features and on the extracted features by eight methods (PCA, KPCA, LDA, NWFE, LFDA, KLFDA, L-NWFE, and L-LFDA) are compared. The first six methods are applied to a single classifier, and the last two methods are applied to multiple classifiers. For KPCA and KLFDA, the Gaussian kernel is used, with the optimized smoothness parameters by grid search. For L-NWFE and L-LFDA, the number of clusters in the $k$-means algorithm is fixed to 10. The smoothness parameter $\sigma$ is selected by grid search ($\sigma = 0, 2^0/10, 2^1/10, \ldots, 2^5/10$) and twofold cross validation using training samples.

Two standard hyperspectral data sets are used to evaluate the performance of the approaches: the University of Pavia data set and the Indian Pines data set. The University of Pavia data set acquired from the Reflective Optics System Imaging Spectrometer (ROSIS) has 103 spectral reflectance bands. The Indian Pines data set acquired from the Airborne Visible/Infrared Imaging Spectrometer sensor has 200 bands after removing the water absorption bands. For the University of Pavia data set, there are a total of 42 776 pixels with ground truth. We randomly select 5% of pixels in each class for training and the rest for testing. For the Indian Pines data set, there are 10 155 samples with labels. We randomly select 20% of pixels due to the data set's higher dimensionality and fewer pixels. To guarantee the training efficiency for all methods, we make sure there are at least 100 training samples for each class in both data sets, and we remove the classes that have fewer than 100 pixels in the Indian Pines data set. The overall classification accuracy (OA) is used for performance evaluation. All the statistics are averaged after testing ten times with different randomly selected training samples.

The result is shown in Fig. 2. The number of extracted features ranges from 1 to 15. All feature extraction methods, except PCA and KPCA, improve OA over using the original features. Among the supervised feature extraction methods, NWFE and LFDA, which use local information and overcome rank deficiency, both perform better than LDA. For KLFDA, it performs worse than LFDA on the University of Pavia data set, but better when the number of extracted features is greater than 8 on the Indian Pines data set. From the results of both data sets in Fig. 2, we can see that both L-LFDA and L-NWFE improve OA compared with the original LFDA and NWFE, particularly when the number of extracted features is low. Moreover, when the single manifold does not perform very well, the improvement by ELML is more significant. For example, L-NWFE on the University of Pavia data set and L-LFDA on the Indian Pines data set improve OA by 5.1% and 9.4% compared with NWFE and LFDA, respectively, on average. The highest OA achieved by each feature extraction methods with number of extracted features (maximum 15) is shown in Table I. L-NWFE and L-LFDA outperform NWFE

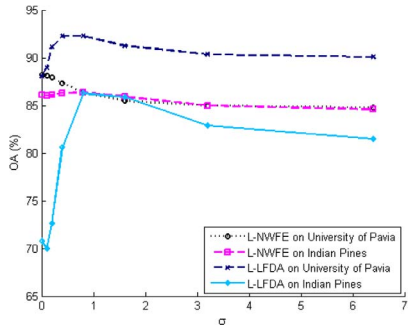


Fig. 3. Classification accuracy (OA%) as a function of $\sigma$ for both L-NWFE and L-LFDA (the numbers of extracted features are both 15) on the two data sets.

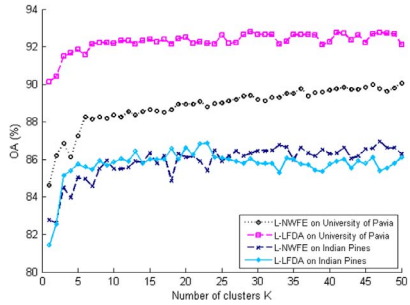


Fig. 4. Classification accuracy (OA%) as a function of the number of clusters for both L-NWFE and L-LFDA (the numbers of extracted features are both 15) on the two data sets.

and LFDA by 3.4% and 1.4%, respectively, on the University of Pavia data set, and by 2.2% and 4.7%, respectively, on the Indian Pines data set.

There are two parameters in the proposed algorithm, i.e., the smoothness constant $\sigma$ and the number of clusters $K$. The relationship between $\sigma$ and OA is shown in Fig. 3. All the OA reach the highest when $\sigma$ is chosen between 0 and 1 and decrease slowly when $\sigma$ increases beyond 1. Specifically, the classification performance of L-NWFE is relatively robust to the variation of $\sigma$. For L-LFDA, OA reaches a peak around $\sigma = 0.8$ for both data sets. When $\sigma$ becomes smaller, the manifolds are too focused on local structure, and OA tends to decrease.

The sensitivity of the number of clusters $K$ (base classifiers) is shown in Fig. 4. We can see that the classification accuracy improves as the number of clusters increases. Theoretically, the classification accuracy is better with larger $K$, but the computation time will also increase, and the improvement is marginal when $K$ is larger than 10. Therefore, we set the parameter $K$ to 10 in our experiment to make a balance between classification performance and computation time.

## V. Conclusion

In this letter, we have presented the novel ELML algorithm, a feature extraction approach for hyperspectral image classification. Considering the complexity and nonlinearity of high-dimensional data structure, multiple linear localized manifolds are learned from the data. Then, multiple sets of features are extracted using these manifolds, and a classifier ensemble is trained on the features to obtain the final result. To implement ELML, L-NWFE and L-LFDA are modified from NWFE and LFDA using a localization weighting scheme in order to learn the localized manifolds, and the 1-NN classifier is used for classification. Experiments show that both L-NWFE and L-LFDA compare favorably with respect to the referenced classification approaches in terms of OA on two hyperspectral data sets. Our future work is to extend the algorithm to the situation when the number of labeled training samples is limited.

## References

[1] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. 14, no. 1, pp. 55–63, Jan. 1968.
[2] B.-C. Kuo and D. A. Landgrebe, "Nonparametric weighted feature extraction for classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 5, pp. 1096–1105, May 2004.
[3] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis," *J. Mach. Learn. Res.*, vol. 8, pp. 1027–1061, 2007.
[4] W. Li, S. Prasad, J. E. Fowler, and L. Mann Bruce, "Locality-preserving dimensionality reduction and classification for hyperspectral image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 4, pp. 1185–1198, Apr. 2012.
[5] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Proc. ICANN*, 1997, pp. 583–588.
[6] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Comput.*, vol. 12, no. 10, pp. 2385–2404, Oct. 2000.
[7] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 6, pp. 607–616, Jun. 1996.
[8] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.
[9] T.-K. Kim and J. Kittler, "Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 318–327, Mar. 2005.
[10] J. Wang, A. Kalousis, and A. Woznica, "Parametric local metric learning for nearest neighbor classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1601–1609.
[11] G. J. Briem, J. A. Benediktsson, and J. R. Sveinsson, "Multiple classifiers applied to multisource remote sensing data," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 10, pp. 2291–2299, Oct. 2002.
[12] P. Zhong and R. Wang, "A multiple conditional random fields ensemble model for urban area detection in remote sensing optical images," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 3978–3988, Dec. 2007.
[13] X. Huang and L. Zhang, "An SVM ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 257–272, Jan. 2013.
[14] W. Liao, A. Pizurica, P. Scheunders, W. Philips, and Y. Pi, "Semisupervised local discriminant analysis for feature extraction in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 184–198, Jan. 2013.
[15] Y. Zhou, J. Peng, and C. P. Chen, "Dimension reduction using spatial and spectral regularized local discriminant embedding for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 2, pp. 1082–1095, Feb. 2015.
[16] T. Hastie *et al.*, *The Elements of Statistical Learning*, vol. 2. Berlin, Germany: Springer-Verlag, 2009.
[17] B. Efron and B. Efron, *The Jackknife, the Bootstrap and other Resampling Plans*, vol. 38. Philadelphia, PA, USA: SIAM, 1982.
[18] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.
[19] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 153–160.
[20] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1601–1608.