Mahalanobis distance–based kernel supervised machine learning in spectral dimensionality reduction for hyperspectral imaging remote sensing

Jing Liu and Yulong Qiao

Abstract
Spectral dimensionality reduction is a crucial step for hyperspectral image classification in practical applications. Dimensionality reduction has a strong influence on image classification performance with the problems of strong coupling features and high band correlation. To solve these issues, we propose the Mahalanobis distance–based kernel supervised machine learning framework for spectral dimensionality reduction. With Mahalanobis distance matrix–based dimensional reduction, the coupling relationship between features and the elimination of the scale effect are removed in low-dimensional feature space, which benefits the image classification. The experimental results show that compared with other methods, the proposed algorithm demonstrates the best accuracy and efficiency. The Mahalanobis distance–based multiples kernel learning achieves higher classification accuracy than the Euclidean distance kernel function. Accordingly, the proposed Mahalanobis distance–based kernel supervised machine learning method performs well with respect to the spectral dimensionality reduction in hyperspectral imaging remote sensing.

Keywords
Hyperspectral sensing, dimensionality reduction, kernel learning, metric learning.

Date received: 27 February 2020; accepted: 24 September 2020

Handling Editor: Feng-Jang Hwang

Introduction
Hyperspectral sensing remote systems are widely used in energy exploration, social safety, military monitoring, and other areas. Hyperspectral remote sensing provides accurate representations of the different materials with high spectral resolution on airborne and satellite platforms. Machine learning is a promising method for hyperspectral data analysis. Since the relationship between spectral curves is nonlinear and complex, spectral classification is a classic complex and nonlinear problem. Among these machine learning methods, a feasible and effective nonlinear method utilizes a kernel technique. As the spectral resolution increases, the coupling between the spectral bands becomes stronger, and the correlation becomes greater. The different spectral bands have different weights on the particular classification problem. Some bands affect the classification results. In feature space, each sample corresponds to a point in space, and the distance between the points reflects the degree of similarity between the sample points. Many classification algorithms do not use...
sample features directly but use the distance between features, that is, the degree of similarity of the samples as the object of analysis. Previous machine learning methods have often used the Euclidean distance to measure this degree of similarity. However, the character of Euclid space assumes that each feature of the sample is equally important and independent of each other, which often does not correspond to the actual spectral characteristics, resulting in a Euclidean distance that does not produce satisfactory results. However, a suitable similarity measure should be related to a specific problem and not be absolutely constant. Generally, for the different classification tasks, the methods used to measure similarity should be different. Metric learning extracts the similarity between different samples with a similarity function. Therefore, the procedural parameters are computed according to the training sample data. We can improve the classification performance by optimizing the matrix learning function in dimensionality reduction.

Depending on the sample used in the study, the metric learning algorithm can be divided into two categories: unsupervised and supervised. The unsupervised learning algorithm does not require class label information during the training stage. Alternatively, an implicit manifold structure is obtained that can maintain the geometric relationship between the sample points in the space. Typical unsupervised metric learning includes multidimensional scaling, nonnegative matrix factorization (NMF), independent component analysis (ICA), neighborhood preserving embedding, locality preserving projection (LPP), and other computing methods. In previous works, many researchers have developed dimensionality reduction methods for different application fields, such as generalized discriminant analysis, uncorrelated discriminant vector analysis (a criterion for optimizing kernel parameters), and kernel machine-based one-parameter regularized Fisher discriminant methods. In addition, other recognition algorithms have been applied in other application areas, such as vehicle estimation. As another feature extraction method, this article proposes an improved kernel function supervised kernel–based LPP method, a local structure supervised feature extraction, kernel subspace linear discriminant analysis (LDA) method, a kernel minimum squared error (MSE) algorithm, and quasiconformal mapping–based kernel machine method. Kernel optimization learning is based on feature extraction to improve kernel-based learning. Regarding the kernel model selection problem, in previous works, many kernel learning algorithms—for example, sparse multiple kernel learning (MKL), large-scale MKL, and Lp-norm MKL—have been proposed to improve the performance accuracy of practical learning systems. Moreover, hyperspectral data classification is the classical problem of high-dimensional data classification, and it is difficult to classify the original data space with the classifier, that is, the so-called “curse of dimensionality,” so dimensionality reduction is the crucial preprocessing step of the classification. The performance of dimensionality reduction directly affects the final classification performance, so dimensionality reduction is a necessary and important step for high-dimensional data classification. Therefore, this topic focuses on dimensionality reduction but for the purposes of classification. Learning-based dimensionality reduction is an effective algorithm for classification. On the dimensionality reduction and classification of hyperspectral, some advanced machine learning methods are presented for hyperspectral data to solve the classification problem, for example, the extreme learning machines. Moreover, the optimization learning methods are applied to hyperspectral image classification, for example, particle swarm optimization–based learning method. Compared with unsupervised metric learning, supervised metric learning makes full use of the label information of the sample so that a better performance metric function can be obtained. On the measure of the “closeness” of samples via the information-theoretic metric learning technique, it is effective to solve such problems directly. In previous studies, the authors presented information-theoretic metric learning algorithms and an improved version, and excellent performance was achieved. At the same time, different annotation information can be set to meet different evaluation criteria. Supervised metric learning is roughly divided into a metric learning algorithm based on pairwise constraints and a metric learning algorithm based on unpaired constraints in the literature. The pairwise constraint information refers to a priori information, and the class labels are easy to obtain and widely used in machine learning. According to the different criteria followed by the optimization process, the metric learning algorithm based on pairwise constraints can be divided into four categories: sample-pair distance sum, information theory, probability theory, and cosine similarity. Unpaired constraints generally refer to priori information other than pairwise constraints, often using ternary constraint information that represents the relative relationship between samples. In addition, machine learning methods are divided into online or local metric learning methods. According to different sample input methods, these methods can also be divided into offline learning and online learning.

In this article, we present the proposed Mahalanobis distance–based MKL-based algorithm, with the advantage of the nonlinear relationship of the spectral bands under hyperspectral imaging conditions, such as the lighting, atmospheric environment, geographical environment, temperature, and humidity. In low-dimensional nonlinear feature space, we use the
Mahalanobis distance–based metric to learn the feature similarity representation. The previous kernel learning represents feature similarity with the Euclidean distance, but the Euclidean distance–based representation performs well on the global differences of two vectors because of the mean square deviation of the two feature vectors. However, the method does not distinguish between some single spectral bands because the difference in the single spectral band can be ignored with mean square deviation computing. In contrast, in this article, the Mahalanobis distance–based similarity method is more effective for individual element differences because the difference in the single spectral band is weighted with the Mahalanobis weight matrix, that is, if some spectral elements in the spectral vectors are meaningful for hyperspectral image classification, then the Mahalanobis weight is large. It is important for hyperspectral image classification. Therefore, the Mahalanobis distance–based similarity measurement is more effective for hyperspectral data classification than the Euclidean distance. The Mahalanobis distance–based feature metric describes the nonlinear relations of spectral bands under hyperspectral imaging conditions such as lighting, atmospheric environment, geographical environment, temperature, and humidity. Therefore, compared with the other methods, the proposed dimensionality reduction method performs better with respect to extracting the nonlinear features of hyperspectral images for classification.

**Proposed Mahalanobis distance–based kernel supervised machine learning**

**Framework**

A single Mahalanobis distance–based kernel function cannot sufficiently process multidimensional and heterogeneous data. Therefore, we propose multikernel learning with the Mahalanobis distance kernel function so that a better performing Mahalanobis distance multikernel function can be obtained. In particular, the expression for the Mahalanobis Gaussian kernel function is

\[
k_{M}(x, y) = \sum_{h=1}^{H} d_h k_{Mh}(x, y)
\]

\[
= \sum_{h=1}^{H} d_h \exp \left( -\frac{1}{2\sigma_h^2} (x - y)^T M (x - y) \right)
\]

where \(x, y\) are the hyperspectral vectors; \(H\) is the number of basic kernels, which are selected according to the Mahalanobis distance–based kernel function classification. \(k_{M}(x, y)\) has the optimal kernel function with high performance of describing the class discriminative ability in the nonlinear feature space; \(k_{Mh}(x, y)\) is the basic kernel function, which is calculated with the input data; \(\{d_h\}_{h=1}^{H}\) is the kernel combination coefficient for the basic kernels, \(d_h \geq 0, \sum_{h=1}^{H} d_h = 1; M\) is the weight matrix for Mahalanobis kernels. By selecting different scale parameters \(\sigma_h\) and \(H\), we can achieve the Mahalanobis distance basic kernels, and the Mahalanobis distance multikernel model can be computed. It can be seen from the above formula that the Mahalanobis distance multikernel learning model is based on a suitable Mahalanobis distance matrix \(M\) and a set of combination coefficients \(\{d_h\}_{h=1}^{H}\), so that the combined kernel function \(k_{M}(x, y)\) has a better nonlinear feature mapping ability.

Considering metric learning, especially Mahalanobis distance metric learning for mining the potential structure inside the data, reducing the coupling relationship between features, and adjusting the weight of the features according to the correlation, researchers began to study how to integrate metric learning with existing classification algorithms to improve classification performance. In the previous work Abe,\(^{25}\) the Mahalanobis distance instead of the Euclidean distance was used in the radial basis function (RBF) kernel function to construct the Mahalanobis distance kernel function; this function was used to support vector machine (SVM) classification and obtained a better classification effect. The kernel function was used to construct the Mahalanobis distance kernel function to SVM classification and to obtain a better classification effect. In Wang et al.,\(^{26}\) the researchers theoretically analyzed how the Mahalanobis distance kernel function improves the classification performance of SVMs. When using a SVM for classification, the kernel function based on Euclidean distance uses the information contained in the support vector. For the whole sample space, this approach is equivalent to using the local information of the sample, and it is easy to produce a situation where the actual distribution of the sample does not match the hyperplane, resulting in a certain risk of misjudgment. When using the Mahalanobis distance kernel function, the Mahalanobis matrix can introduce the global information of the sample to obtain a more reasonable classification hyperplane.

The Mahalanobis distance–based kernel function is constructed by extending the existing Euclidean distance–based kernel function. For a specific metric learning algorithm, if the mapping of matrix \(A\) is easier to obtain, then the sample can be directly mapped to a new domain \(x \rightarrow A^T x\). The Euclidean distance kernel function is used in the new mapping domain, that is, \(k_{M}(x, y) = \langle A^T x, A^T y \rangle\). If the Mahalanobis matrix \(M\) is easier to obtain, then the existing Euclidean distance–based kernel function can be extended to construct a Mahalanobis distance–based kernel function.\(^{27}\)

According to the different ways of using sample features, the existing common kernel functions can be
roughly divided into two types: kernel functions based on feature similarity and kernel functions based on the feature inner product. Since the expressions of these two types of kernel functions are different, the method to construct the Mahalanobis distance kernel function is also different.

**Learning criteria**

We proposed a large margin nearest neighbor criterion based on similar/dissimilar–based learning. The metric learning algorithm utilizes ternary constraint information. Better consistency is achieved with the perspective of feature space after the metric mapping is beneficial for the subsequent classification operation.

The closest sample in the same samples. Then, the target loss function is

\[
\begin{align*}
\mathcal{L}(A) &= \sum_{y} \eta_{y} \|A(x_{i} - x_{j})\|^2 \\
&\quad + c \sum_{y_k} \eta_{y}(1 - y_{k}) \left[1 + \|A(x_{i} - x_{j})\|^2 - \|A(x_{i} - x_{j})\|^2\right]
\end{align*}
\]

(2)

where the training set \( \{(x_{i}, y_{i})|x_{i} \in \mathbb{R}^{N}, y_{i} \in \{0, 1\}, i = 1, \ldots, l\} \) indicates whether the samples \( x_{i} \) and \( x_{j} \) belong to the same category, and \( \eta_{y_{i}} \in \{0, 1\} \) indicates whether the samples \( x_{i} \) and \( x_{j} \) belong to a neighbor relationship, that is, where \( \eta_{y_{i}} = \max(z, 0) \), \( c \) is a normal number used to adjust the weight between the two loss terms. The first term of the penalty function in the above equation is used to adjust the distance between the sample and the neighbor, and a similar sample is compressed toward the center by a “push” method. The second term is used to adjust the distance between the boundaries of heterogeneous samples, and the boundary interval of the heterogeneous samples is expanded by a “pull” method.

Furthermore, by introducing the slack variable \( \xi_{ijk} \), the metric optimization problem can be transformed into a semidefinite programming problem of the form so that the optimal Mahalanobis distance matrix can be obtained by the standard algorithm for solving the semidefinite programming

\[
\min_{M, \xi} \sum_{y} \eta_{y} \|A(x_{i} - x_{j})\|^2 + c \sum_{y} \eta_{y}(1 - y_{k}) \xi_{ijk}
\text{s.t.} \forall (i, j, k), (x_{i} - x_{j})^{T}M(x_{i} - x_{j}) - (x_{i} - x_{j})^{T}M(x_{i} - x_{j}) = 1 - \xi_{ijk}
\forall (i, j, k) \xi_{ijk} = 0
\]

\[
M \succeq 0
\]

(3)

where \( x, y \) are the hyperspectral vectors; \( H \) is the number of basic kernels, which are selected according the task of hyperspectral curves vectors classification. \( \eta_{y_{i}} \in \{0, 1\} \) indicates whether the samples \( x_{i} \) and \( x_{j} \) belong to a neighbor relationship, \( M \) is the weight matrix for Mahalanobis kernels, and \( \xi_{ijk} \) is the slack variable. As a different projection function, metric learning algorithms have linear and nonlinear abilities for feature extraction. The initial distance metrics are computed with the Euclidean distances without the loss of generality. Nonlinear metric learning is computed by kernelizing the linear model.

**Discussion on metric learning**

Information theoretic–based metric learning is widely used in designing optimization goals. As a general method, the information entropy, Kullbacle–Leibler (KL) divergence, is used to measure the difference between two probability distributions. For two probability distributions \( P \) and \( Q \) in continuous space \( X, p(x) \) and \( q(x) \) represent their probability density functions, respectively, and then the KL divergence is defined as

\[
D_{KL} = \int_{X} p(x) \log \frac{p(x)}{q(x)} dx
\]

(4)

However, for a metric matrix \( M \), since it is a semipositive definite matrix, a multivariate Gaussian distribution \( p(x; M) = (1/2)\exp((1/2)d_{M}(x, \mu)) \) can be found, where \( z \) is the normalization factor and \( \mu \) is the mean. Using the matrix \( M \) as a parameter of the probability distribution, a relationship between the metric matrix and the multivariate probability distribution is established. A priori metric matrix \( M_{0} \) should be defined first, typically using the inverse of the sample covariance matrix or the Euclidean distance matrix. The learned metric matrix \( M \) should be as close as possible to the a priori metric matrix if the constraints are satisfied, that is, the relative entropy represented by the two matrices should be minimal.

The distance from the same class labels is not greater than the threshold \( u \) and is not less than a certain threshold \( l \) as a constraint. After determining the objective function and constraints, the metric learning problem based on the relative entropy can be expressed as

\[
\min_{M} D_{KL}(p(x; M_{0}) \parallel p(x; M))
\text{s.t.} \quad M \succeq 0
\]

\[
d_{M}(x_{i}, x_{j}) \leq u, \quad \text{if} (x_{i}, x_{j}) \in W
\]

\[
d_{M}(x_{i}, x_{j}) \geq l, \quad \text{if} (x_{i}, x_{j}) \in B
\]

(5)

where \( d_{M}(x_{i}, x_{j}) \) is the Mahalanobis distance between \( x_{i}, x_{j} \). In general, the mean values of the Gaussian
distributions corresponding to \( M_0 \) and \( M \) are the same, and the KL divergence value can be obtained by defining a convex function \( f(X) = \log \det (X) \) and calculating the Bregman divergence of the mapping, that is

\[
D_{KL}(p(x; M_0) \parallel p(x; M)) = \frac{1}{2} D_{LD}(M_0^{-1}, M^{-1})
\]

\[
= \frac{1}{2} \left( tr(MM_0^{-1}) - \log \det (MM_0^{-1}) - d \right)
\]

where \( M \) is the weight matrix for Mahalanobis kernels, \( D_{LD}(\cdot) \) represents the Bregman divergence, and \( \log \det (X) \) represents the logarithm of the determinant of the matrix \( X \). Furthermore, the Bregman iterative optimization algorithm can be used to solve the problem.

**Algorithm procedure**

The Mahalanobis distance matrix and the combination parameters are solved through the actual solution process. It is difficult to convert the equation into an optimization problem. Therefore, we propose a two-stage optimization method to solve the matrix \( M \) and the weights. Combined with the MKL algorithm, the typical Mahalanobis distance multikernel learning algorithm is based on pairwise constraints. On the MKL algorithm, we adopt wrapper MKL algorithms in Saeid et al.,\(^ {28} \) and the wrapper algorithms use a two-step optimization method to solve the matrix weights and the parameters of the classifier.\(^ {28} \) In the wrapper MKL algorithms, we use the simple MKL methods, as an L1-MKL algorithm, which estimates the weights on a simplex by optimizing the MKL dual problem,\(^ {29} \) and the optimal weights are solved with the projected gradient descendant optimization algorithm (Figure 1).

**Experiments and discussion**

**Experimental setting**

We implement some experiments to test the feasibility of the proposed algorithm and compare the performance with that of the other algorithms. The experiments are implemented on two hyperspectral sensing data sets: the Indian Pine data set and the Pavia University data set. The accuracy of the spectrum classification is an important index for evaluating the performance of the spectrum classification. The Indian Pine data set is based on an airborne platform at various spectral and spatial resolutions. The data include 224 0.4–2.5 \( \mu m \) bands. Nine kinds of 145 \( \times \) 145 pixel images are realized in the experiment. The data collection at the University of Pavia is based on a reflective optical system imaging spectrometer (ROSIS). The data include 115 bands. In the experiment, the performance of nine kinds of 610 \( \times \) 340 images is verified. Except for the feature dimensions of the participating categories, the two experiments were identical. Regarding the use of classification features, considering the computational efficiency and stability of the Mahalanobis matrix, the dimensions of the original spectral features are reduced by principal component analysis (PCA). After dimension reduction, the features are normalized to eliminate the deviation caused by the
sampling method. For the first experiment, the top 30 principal components are selected to participate in the classification, that is, the feature dimension is 30. For experiment 2, the first 40 principal components are selected to participate in the classification, that is, the feature dimension is 40. Regarding the classifier settings, the preset parameter values are selected by cross-validation by a standard multiclass SVM. In the kernel function setting, the Gaussian kernel function and the Mahalanobis Gaussian kernel function are used as the basis kernel functions. The scale parameter \( \sigma \) is set between [0.01, 2], and the number of basis kernels is 10. The overall classification accuracy (OA) and Kappa coefficient (KC) are used to evaluate the classification accuracy and to evaluate the computation efficiency with the classifier training time and test time. In the comparison method, the average multiple kernel matrix and different MKL methods are used. The Gaussian kernel and polynomial kernel are used with the Mahalanobis and Euclidean distances. In the following experiments, we implement some experiments on the Indian Pine data set and Pavia University data set to test the feasibility of the proposed algorithm and compare its performance results with that of the other algorithms.

Performance regarding the difference dimensionality

In these experiments, we implement four algorithms as follows. Euclidean-MKL1\textsuperscript{27}: the Euclidean distance kernel function. Each kernel function is combined according to the same weight, that is, the combination coefficient of each kernel function is the reciprocal of the number of kernel functions. See the description of Gonen and Alpaydin\textsuperscript{27} for details. Mahalanobis-MKL1: the Mahalanobis distance kernel function is used for kernel learning. Euclidean-MKL2\textsuperscript{30}: the Euclidean distance kernel function is used, which describes the combination coefficient and is described in Gu et al.\textsuperscript{30} Mahalanobis-MKL2: the Mahalanobis distance kernel function is used, and the kernel learning is the same as Euclidean-MKL2. During the experiment, a certain number of samples are randomly selected from the sample data as training samples, and the remaining samples were used as test samples. The number of training samples is 10, 20, 30, 40, and 50. For each parameter setting, 10 experiments are repeated and averaged, while the variance values are calculated.

**Performance of the Indian Pine data set.** Figure 2 shows the classification OA results for the four methods with respect to the Indian Pine data set. Combined with the KC results shown in Figure 3, it can be seen that the classification accuracy and KC of the corresponding algorithm are improved after using the Mahalanobis distance kernel function. Figures 2 and 3 describe the recognition accuracy under the same feature dimension, for the dimension of feature, 10, 20, 30, 40, and 50. The highest recognition accuracy is 77.12\%, under the 50 dimensional feature. And the Mahalanobis-MKL2 method has the highest recognition accuracy Indian Pine data set. And with the similar results, the performance with KC of different methods on the Indian Pine data set is described in Figure 3, and the Mahalanobis-MKL2 method has the highest recognition accuracy compared with other results. In terms of the extent of OA improvement, Mahalanobis-MKL1 has a 3\% increase in the Euclidean-MKL1 algorithm, and the amplitude of the increase does not change as the number of training samples increases. The experimental results show that the Mahalanobis distance kernel combined with metric learning can make full use of the sample information when the number of training samples is small, thus improving the separability of the sample. Similarly, Mahalanobis-MKL2 also improves the Euclidean-MKL2 algorithm, but the increase is relatively low, and its amplitude does not increase as the number of training samples increases.
As shown in Table 1, the classifier using the Mahalanobis distance kernel function has the lowest computation time for classifier training and testing. Under the mapping of the Mahalanobis distance matrix, the distance between samples of different categories is larger, but the distance between similar samples is smaller. As shown in Table 2, the test times are computed as a function of the number of support vectors. For the training time of the classifier, although the difference is not very obvious, the trend is the same as the test time of the classifier. As the number of samples increases, the difference in time is more obvious. Table 3 shows the actual mapping effect of the four methods when the number of training samples is 50.

### Table 1. Training time of different methods on the Indian Pine data set (s).

| Method           | Feature dimension |
|------------------|-------------------|
|                  | 10    | 20    | 30    | 40    | 50    |
| Euclidean-MKL1   | 0.445 | 0.468 | 0.518 | 0.528 | 2.438 |
| Mahalanobis-MKL1 | 0.443 | 0.464 | 0.494 | 0.504 | 1.794 |
| Euclidean-MKL2   | 0.442 | 0.463 | 0.493 | 0.513 | 1.703 |
| Mahalanobis-MKL2 | 0.442 | 0.461 | 0.481 | 0.491 | 1.584 |

### Table 2. Testing time of different methods on the Indian Pine data set (s).

| Method           | Feature dimension |
|------------------|-------------------|
|                  | 10    | 20    | 30    | 40    | 50    |
| Euclidean-MKL1   | 5.123 | 8.523 | 10.523| 12.523| 13.223|
| Mahalanobis-MKL1 | 4.125 | 6.245 | 7.545 | 8.015 | 8.815 |
| Euclidean-MKL2   | 4.235 | 6.425 | 7.825 | 8.825 | 9.425 |
| Mahalanobis-MKL2 | 3.324 | 5.224 | 5.824 | 6.223 | 7.034 |

### Table 3. Number of support vectors of different methods on the Indian Pine data set.

| Method           | Feature dimension |
|------------------|-------------------|
|                  | 10    | 20    | 30    | 40    | 50    |
| Euclidean-MKL1   | 1010  | 1650  | 2090  | 2450  | 2570  |
| Euclidean-MKL2   | 810   | 1250  | 1490  | 1610  | 1790  |
| Mahalanobis-MKL1 | 890   | 1310  | 1510  | 1780  | 1980  |
| Mahalanobis-MKL2 | 700   | 990   | 1210  | 1350  | 1450  |

As shown in Table 1, the classifier using the Mahalanobis distance kernel function has the lowest computation time for classifier training and testing. Under the mapping of the Mahalanobis distance matrix, the distance between samples of different categories is larger, but the distance between similar samples is smaller. As shown in Table 2, the test times are computed as a function of the number of support vectors. For the training time of the classifier, although the difference is not very obvious, the trend is the same as the test time of the classifier. As the number of samples increases, the difference in time is more obvious. Table 3 shows the actual mapping effect of the four methods when the number of training samples is 50.

Performance using University of Pavia data set. Regarding the classification accuracy in the different dimensions, Figure 4 shows the classification OA for the four methods with the University of Pavia data set, and Figure 5 shows the corresponding KCs. As shown in Figure 4, the recognition accuracy under the same feature dimension is denoted, for the dimension of feature, 10, 20, 30, 40, and 50. The highest recognition accuracy is 79.86%, under the 50 dimensional feature on the Pavia University data set. And the Mahalanobis-MKL2 method has the highest recognition accuracy, which has the same conclusion to the Indian Pine data set. And with the similar results, the performance with KC of different methods on the Pavia University data set is described in Figure 5. Compared with the Indian Pine data set, the Mahalanobis distance kernel function is lower with respect to the OA, only by approximately 1%–2%, but the overall trend is the same as that of the Indian Pine data set.

By analyzing the classifier training and test times given in Tables 4 and 5 and accordingly the number of support vectors given in Table 6, we can see the role of the Mahalanobis distance kernel in reducing the classifier runtime. In the case of the fixed training samples, because the number of samples in each category of the University of Pavia data set is larger, the corresponding
number of test samples is larger, and the Mahalanobis distance kernel is more effective in shortening the test time of the classifier.

Comparing the computational time, Tables 1, 2, 4, and 5 show the computational times for the two data sets with the different vector dimensions. The computational cost was recorded via a PC with a 2.6 GHz i5-3320 processor and 4 GB RAM. All the results are averages of 10 repeated experiments. As shown in the tables, different computational costs are achieved with the different features. The proposed algorithms omit the parameter optimization process under the same dimension of the feature vector, so that high computation efficiency is achieved. Both Euclidean-MKL1 and Mahalanobis-MKL1 adopt NMF to optimize the kernel weights, and higher dimensional features require more time because they need more memory to save the kernel matrix and have more dimensions to compute. Therefore, Mahalanobis-MKL1 has a higher computational efficiency than Euclidean-MKL1 with the same feature dimensions.

### Table 4. Training time of different methods for the Pavia University data set (s).

| Method               | Feature dimension |
|----------------------|-------------------|
|                      | 10    | 20    | 30    | 40    | 50    |
| Euclidean-MKL1       | 0.22  | 0.30  | 0.38  | 0.48  | 1.80  |
| Mahalanobis-MKL1     | 0.23  | 0.25  | 0.32  | 0.42  | 1.32  |
| Euclidean-MKL2       | 0.21  | 0.28  | 0.35  | 0.46  | 1.72  |
| Mahalanobis-MKL2     | 0.19  | 0.24  | 0.30  | 0.40  | 1.28  |

### Table 5. Testing time of different methods for the Pavia University data set (s).

| Method               | Feature dimension |
|----------------------|-------------------|
|                      | 10    | 20    | 30    | 40    | 50    |
| Euclidean-MKL1       | 15.23 | 25.23 | 35.24 | 43.23 | 53.33 |
| Mahalanobis-MKL1     | 12.37 | 19.33 | 25.25 | 30.63 | 34.63 |
| Euclidean-MKL2       | 15.28 | 25.38 | 34.68 | 40.34 | 48.54 |
| Mahalanobis-MKL2     | 12.35 | 18.55 | 24.95 | 28.43 | 33.53 |

### Table 6. Number of support vectors of different methods for the Pavia University data set.

| Method               | Feature dimension |
|----------------------|-------------------|
|                      | 10    | 20    | 30    | 40    | 50    |
| Euclidean-MKL1       | 520   | 1090  | 1510  | 1920  | 2230  |
| Mahalanobis-MKL1     | 500   | 800   | 1120  | 1280  | 1490  |
| Euclidean-MKL2       | 530   | 1100  | 1490  | 1890  | 2180  |
| Mahalanobis-MKL2     | 490   | 780   | 1100  | 1230  | 1450  |

### Table 7. Performance comparisons of two databases.

| Data sets            | Methods | Indian Pine data set | Pavia University data set |
|----------------------|---------|----------------------|---------------------------|
|                      | OA (%)  | KC (%)               | OA (%)                    | KC (%)                   |
| RBF26               | 73.23   | 63.73                | 75.43                     | 68.76                    |
| POLY26              | 75.77   | 66.28                | 78.08                     | 72.07                    |
| Mahal-RBF31         | 76.92   | 67.79                | 76.26                     | 69.87                    |
| Mahal-Poly31        | 77.64   | 68.62                | 79.07                     | 73.24                    |
| SK-CV-RBF32         | 67.03   | 64.23                | 75.71                     | 69.14                    |
| SK-POLY32           | 69.37   | 66.96                | 77.62                     | 71.37                    |
| NMF-MKL33           | 67.48   | 64.81                | 71.57                     | 64.42                    |
| KNMF-MKL33          | 68.22   | 65.63                | 72.80                     | 65.81                    |
| Euclidean-MKL127    | 74.23   | 69.63                | 78.69                     | 72.11                    |
| Euclidean-MKL230    | 76.07   | 72.84                | 79.24                     | 72.92                    |
| Mahalanobis-MKL1    | 75.90   | 73.25                | 79.78                     | 73.43                    |
| Mahalanobis-MKL2    | 78.22   | 74.25                | 79.86                     | 74.16                    |

OA: overall accuracy; KC: Kappa coefficient; NMF: nonnegative matrix factorization; KNMF: kernel-based nonnegative matrix factorization. Bold value signifies the proposed algorithm.

**Performance comparisons**

For the comparisons, we have some experiments to compare the performance of the proposed algorithm, and the following 12 methods are implemented as the comparison. The experimental results are shown in Table 7. They are as follows:

1. RBF: RBF Euclidean kernel as the kernel function in the SVM.
2. POLY: polynomial Euclidean kernel as the kernel function in the SVM.
3. Mahal-RBF: Mahalanobis distance–based RBF kernel as the kernel function in the SVM.
4. Mahal-Poly: Mahalanobis distance–based polynomial kernel as the kernel function in the SVM.
5. SK-CV (RBF): an SVM with a single kernel and adopting the RBF kernel as the kernel function in the SVM.
6. SK-POLY: standard SVM with a single kernel and adopting a polynomial kernel as the kernel function in the SVM.
7. NMF-MKL: the NMF-MKL proposed by Gu et al., which combines multiple kernels with NMF.
8. KNMF-MKL: the kernel-based nonnegative matrix factorization (KNMF)-MKL method, also proposed by Gu et al., which combines multiple kernels with the KNMF method.
9. Euclidean-MKL1, the Euclidean distance kernel function. Each kernel function is combined according to the same weight, that is, the combination coefficient of each kernel function is the reciprocal of the same
kernel functions. See the description in Gonen and Alpaydin\textsuperscript{27} for details.

10. Euclidean-MKL\textsuperscript{2}: the Euclidean distance kernel function is used, which describes the combination coefficient, as described in Gu et al.\textsuperscript{30}

11. Mahalanobis-MKL\textsuperscript{2}: proposed the Mahalanobis distance–based multiple kernel function, and the learning criterion is the same as that of Euclidean-MKL\textsuperscript{1}.

12. Mahalanobis-MKL\textsuperscript{2}: the Mahalanobis distance kernel function is used, and the learning criterion is the same as that of Euclidean-MKL\textsuperscript{2}.

In Table 7, the experiments show that the proposed algorithms perform better than the other algorithms. The data sets are constructed with the images under different conditions, including lighting, atmospheric environment, geographical environment, temperature, and humidity. Therefore, from the experimental results, the proposed Mahalanobis distance–based MKL-based algorithm has the advantage of the nonlinear relationships of the spectral bands under hyperspectral imaging conditions. In low-dimensional nonlinear feature space, we use the Mahalanobis distance–based metric to learn the feature similarity representation. The Mahalanobis distance–based similarity method is more effective for individual element differences because the difference in the single spectral band is weighted with the Mahalanobis weight matrix, that is, if some spectral elements in the spectral vectors are meaningful for hyperspectral image classification, then the Mahalanobis weight is large. The experimental results show that the proposed algorithm performs better than the other kernel matrix learning methods. Compared with the Euclidean distance kernel function, the Mahalanobis distance kernel learning achieves higher classification accuracy. The number of support vectors is reduced by reducing the boundary of similar samples, and accordingly, the algorithm performs more efficiently.

Discussion

The experimental results show that the addition of metric learning can effectively improve the sample features and the classification performance of the classifier. Under the Indian Pine data set, while the classification accuracy increased by 3\%, the classifier training time was reduced by 23\%, and the test time was reduced by 31\%. Under the University of Pavia data set, while the classification accuracy increased by 1.5\%, the classifier training time was reduced by 25\%, and the test time was reduced by 33\%. Compared with the multikernel learning method using the Euclidean distance kernel function, the Mahalanobis multikernel learning algorithm has higher classification accuracy, reduces the number of support vectors, and shortens the training time and test time of the classifier by the cohesion of similar samples.

Conclusion

Hyperspectral features have a different importance for each band, strong coupling between features, and challenge the classification method with the Euclidean distance as a measure. This article optimizes sample features from the perspective of improving the sample metrics. First, we introduce the basic concepts of metric learning, as well as some typical Mahalanobis distance metrics. By mining the distribution information and discriminative information contained in the sample data, the Mahalanobis distance matrix can remove the dimensional influence and coupling relationship in the spectral features. In addition, this matrix can also map the features to a feature space with a smaller distance within the class, and a larger space between classes is distanced, making subsequent classification operations easier. On this basis, from the perspective of kernel function design, the Mahalanobis distance kernel function is constructed, and a Mahalanobis distance multikernel learning method is proposed.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Science Foundation of China under grant no. 61871142.

ORCID iD

Yulong Qiao \textsuperscript{16} https://orcid.org/0000-0003-1329-6974

References

1. Li J, Pan J and Chu S. Kernel class-wise locality preserving projection. \textit{Inform Sci} 2008; 178(7): 1825–1835.

2. Afzal A and Ashraf S. Deep kernel learning in core vector machines. \textit{Pattern Anal Appl} 2018; 21(3): 721–729.

3. Du Z, Pan J, Chu S, et al. Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks. \textit{IEEE Access} 2019; 8: 8583–8594.

4. Baudat G and Anouar F. Generalized discriminant analysis using a kernel approach. \textit{Neural Comput} 2000; 12(10): 2385–2404.
5. Wang L, Chan L and Xue P. A criterion for optimizing kernel parameters in KBDA for image retrieval. *IEEE T Syst Man Cyb Part B: Cybernetics* 2005; 35(3): 556–562.

6. Pan JS, Hu P and Chu SC. Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power. *Processes* 2019; 7(11): 845–856.

7. Pan J and Lee C. Novel systolization of subquadratic space complexity multipliers based on Toeplitz matrix-vector product approach. *IEEE T Very Large Scale Integr Syst* 2019; 27(7): 1614–1622.

8. Chen C, Lee C and Lo C. Vehicle localization and velocity estimation based on mobile phone sensing. *IEEE Access* 2016; 4: 803–817.

9. Chen C. A cell probe-based method for vehicle speed estimation. *IEICE T Fund Electr Commun Comput Sci* 2010; 103(1): 265–267.

10. Zhao H, Sun S, Jing Z, et al. Local structure based supervised feature extraction. *Pattern Recogn* 2006; 39(8): 1546–1550.

11. Huang J, Yuen P, Chen W, et al. Kernel subspace LDA with optimized kernel parameters on face recognition. In: *Proceedings of the sixth IEEE international conference on automatic face and gesture recognition*, Seoul, 19 May 2004, pp.187–192. New York: IEEE.

12. Tian A, Chu S and Pan J. A compact Pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station. *Sustainability* 2020; 12(3): 345–351.

13. Xie X, Li B and Chai X. A framework of quasiconformal mapping-based kernel machine with its application to hyperspectral remote sensing. *Measurement* 2016; 80: 270–280.

14. Pan JS, Li JB and Lu ZM. Adaptive quasiconformal kernel discriminant analysis. *Neurocomputing* 2008; 71(13–15): 2754–2760.

15. Chen WS, Yuen PC, Huang J, et al. Kernel machine-based one-parameter regularized Fishers discriminant method for face recognition. *IEEE T Syst Man Cyb Part B: Cybernetics* 2005; 35(4): 659–669.

16. Xiong H, Swamy MNS and Ahmad MO. Optimizing the kernel in the empirical feature space. *IEEE T Neural Netw* 2005; 16(2): 460–474.

17. Subrahmanya N and Shin Y. Sparse multiple kernel learning for signal processing applications. *IEEE T Pattern Anal Mach Intell* 2020; 32(5): 788–798.

18. Sören S, Rätsch G and Schäfer C. Large scale multiple kernel learning. *J Machine Learn Res* 2006; 7: 1531–1565.

19. Marius K, Bréfeld U and Sonnenburg S. Lp-norm multiple kernel learning. *J Mach Learn Res* 2011; 12: 953–997.

20. Haut JM, Paoletti ME, Plaza J, et al. Fast dimensionality reduction and classification of hyperspectral images with extreme learning machines. *J Real Time Image Proc* 2018; 15(3): 439–462.

21. Yang H, Du Q and Chen G. Particle swarm optimization-based hyperspectral dimensionality reduction for urban land cover classification. *IEEE J Select Topic Appl Earth Observ Remote Sens* 2012; 5(2): 544–554.

22. Davis JV, Kulis B, Jain P, et al. Information-theoretic metric learning. In: *Proceedings of the 24th international conference on machine learning*, Corvalis, OR, June 2007, pp.209–216. New York: ACM.

23. Dong Y, Bo Du B, Zhang L, et al. Hyperspectral target detection via adaptive information-theoretic metric learning with local constraints. *Remote Sens* 2018; 10(9): 1415.

24. Dong Y, Du B, Zhang L, et al. Dimensionality reduction and classification of hyperspectral images using ensemble discriminative local metric learning. *IEEE T Geosci Remote Sens* 2017; 55(5): 2509–2524.

25. Abe S. Training of support vector machines with Mahalanobis kernels. In: *International conference on artificial neural networks: formal models & their applications*, Warsaw, 11–15 September 2005, pp.234–241. New York: Springer.

26. Wang D, Yeung DS and Tsang ECC. Weighted Mahalanobis distance kernels for support vector machines. *IEEE T Neural Netw* 2007; 18(5): 1453–1462.

27. Gonen M and Alpaydin E. Multiple kernel learning algorithms. *J Mach Learn Res* 2011; 12: 2211–2268.

28. Saeid N, Begüm D, Lorenzo B, et al. Multiple kernel learning for remote sensing image classification. *T Geosci Remote Sens* 2018; 56(3): 1425–1443.

29. Rakotomamonjy A, Bach F, Canu S, et al. Simple MKL. *J Mach Learn Res* 2008; 9: 2491–2521.

30. Gu Y, Wang C and You D. Representative multiple kernel learning for classification in hyperspectral imagery. *IEEE T Geosci Remote Sens* 2012; 50(7): 2852–2865.

31. Li L, Sun C, Lin L, et al. A Mahalanobis metric learning-based polynomial kernel for classification of hyperspectral images. *Neural Comput Applic* 2018; 29: 1103–1113.

32. Li L, Sun C, Lin L, et al. A dual-layer supervised Mahalanobis kernel for the classification of hyperspectral images. *Neurocomputing* 2016; 214: 430–444.

33. Gu Y, Wang Q, Wang H, et al. Multiple kernel learning via low-rank nonnegative matrix factorization for classification of hyperspectral imagery. *IEEE J Sel Top Appl Earth Obs Remote Sens* 2015; 8(6): 2739–2751.