Kernel-based Informative Feature Extraction via Gradient Learning

Songhua Liu[1,2], Jiansheng Liu[1], Caiying Ding[1,3,4], Chaoquan Zhang[1]
[1]College of Science, Jiangxi University of Science and Technology, Ganzhou, P. R. China
[2]School of Computer Science and Technology, Xidian University, Xi'an, P. R. China
[3]Center of Interdisciplinary, Lanzhou University, Lanzhou, P. R. China
[4]Institute of Physics, Chinese Academy of Science, Beijing, P. R. China
Email: sooh.liu@gmail.com

***Abstract*—We consider the problem of feature extraction for kernel machines. One of the key challenges in this problem is how to detect discriminative features while mapping features into kernel spaces. In this paper, we propose a novel strategy to quantify the importance of features. Firstly, we derive an informative energy model to quantification of feature difference. Secondly, we move the features in the same class closer and push away those belong to different classes according to the model and derivate its objective function. Finally, gradient learning is employed to maximize this function. Experimental results on real data sets have shown the efficient and effective in dealing with projection and classification.**

***Index Terms*—Kernel methods, nonlinear transformation, feature extraction, gradient learning.**

I. Introduction

Obtaining important features with kernel machines is a challenging problem in classification tasks. It is also essential in exploratory data analysis, where the purpose is to map data onto a feature space for improved visualization. We are interested in methods that reveal discriminative features of the data sets. This can be achieved either by selection or by a transform from a large number of original features.

Feature selection methods [1-4] keep only useful features and discard others. Feature transform methods [5-9] construct new features out of the original features. However, algorithms that perform feature selection often lead to a combinatorial problem since all features need to be evaluated, but feature transform only need some criterion related to the performance of classifiers that would reflect the importance of a feature or a number of features. For this very reason, finding a transform might be easier than selection features. A further motivation for transforms is the ability to extract distributed relevant information across several original features, which produces a more compact representation than selection.

In this paper, we propose a novel method that aims to extract significant features by a new criterion within kernel framework in conjunction with gradient learning, called kernel informative feature extraction (KIFE). The algorithm has some distinct characteristics: (1) It can extract high order statistics and nonlinear discriminative features; (2) this method can avoid the high time usage associated with eigenvalue decomposition in existing methods; (3) traditional criterion mutual information (MI) can be derived within the proposed KIFE framework, which helps users obtaining important features.

The rest of the paper is organized as follows. In Section II, we briefly summarize background and prior work. Then, the main algorithm is derived in Section III. Section IV presents the experimental results, and finally conclusions are drawn in Section V.

II. Background And Prior Work

A. Notations

Let us denote by X the original feature set and by variable C the class labels. x_{ci} is a sample in the input space X, where $c \in [1, C]$, $X \in R^{N \times d}$ and $t \in [1, N_c]$, N is the number of features in X, d is dimensionality of the features, N_c is the number of classes. We make use of a dual notation for the feature x_{ci} in the input space, it is written with a single subscript x_i when its class is irrelevant, index $1 \le i \le N$. If the class is relevant, assume that we have J_c features for c th class, we write x_{ci}, where the class index $1 \le c \le N_c$, and the index within class $1 \le i \le J_c$.

For a given kernel function $k : X \times X \to R$, the training features x_i, $1 \le i \le J_c$ are implicitly mapped to a feature space F with usually high dimensionality. Let $\phi(\cdot)$ denotes the mapping from X to F and $y_i = \phi(x_i) \in F$, then

$$k_{ij} = k(x_i, x_j) = \left\langle \phi(x_i), \phi(x_j) \right\rangle \tag{1}$$

B. Related Proir work

In general, feature extraction algorithms require certain criterions. Recently, research has been done on using different objective functions to address this problem. For example, Ref. [10] described a generalized discriminant

Manuscript received September 29, 2010; revised December 7, 2010; accepted January, 2011.
Corresponding author: Songhua Liu

doi:10.4304/jcp.7.11.2813-2820

analysis (GDA) method, which depends on the eigendecomposition of the kernel matrix, which bears high computational complexity. Invoked by this problem, Ref. [11] used a low-rank approximation to a complete eigendecomposition of the kernel matrix. Recently, Ref. [12] proposed a kernel based nonlinear feature extraction, which transforms this problem to a kernel parameter learning problem. Ref. [13] presented a method for learning discriminative feature transforms using as criterion the MI between class labels and transformed features. Another recent paper by [14] employed conditional information and information losses to extract main features in input features.

However, the similarity measure in many of these papers depends only to the Euclidean measure. When samples have equal Euclidean distances to training samples, the kernel mapped the samples into the same vectors. This may not perfectly fulfill the purpose of classification-oriented feature extraction. On the other hand, MI according to Shannon's definition is computationally expensive.

III. The Algorithm For The Kife

In this section, we describe KIFE as Algorithm 1, which trained with training data X and its class label set C. After the kernel matrix is obtained, the algorithm has three stages. In the first stage, we define a function to quantification of the feature difference. In the second stage, we derive the objective function for feature extraction. Then, gradient learning is used to optimize the objective function and find the optimal coefficient matrix.

A. Detailed Description of Our Main Algorithm

In the first stage of KIFE, we propose an informative energy model. The main idea is that we quantify the difference between features according to their graph energy [15]. Our goal is to transform the kernel space so that the distance in the transformed space correlated with the difference of the labels of features. So, we need to define informative energy for our method.

The graph energy in [15] is defined as

$$E(\sigma_j) = \frac{1}{Z}\sum_{i=1}^{N} G(x_j - x_i, 2\sigma^2 I) H(x_j, x_i)$$

where $G(y,\sigma^2) = \exp\left(-\frac{1}{2\sigma^2} y^T y\right)$ the Gaussian kernel function, $\sigma_j \in \sigma$ is the kernel width parameter, H is an indicator function, $Z = \sum_{j=1}^{N}\sum_{i=1}^{N} G(x_j - x_i, 2\sigma^2 I)$ is a normalization variable. The Energy value is 1 when the feature x_i and x_j are in the same class, otherwise 0 when they are in the different class.

For the feature $y_{ci} = \phi(x_{ci})$ in the kernel space, we define its informative energy according to the graph energy model. The main difference is that we consider each feature in the kernel space as a particle, and pull or push other particles in this space. This means that the resultant effect of a particle is the sum of the separate effects between the same class and the different classes. For each feature we defined two informative energy function: similar and dissimilar energy. For the features in the same class, the similar energy is computed as follows

$$E_c(y_{ci}) = \frac{1}{N}\sum_{j=1}^{J_c} G(y_{cj} - y_{ci}, 2\sigma^2 I) \tag{2}$$

where I is the identity matrix.

Then the dissimilar energy considering features between different classes is computed as

$$E_{p\neq c}(y_{ci}) = \frac{1}{N}\sum_{p=1}^{N_c}\sum_{l=1}^{J_p} G(y_{pl} - y_{ci}, 2\sigma^2 I) \tag{3}$$

These two energy functions vary between 0 and 1. A high E_c indicates that two features in the same class are quite similar. But a low $E_{p\neq c}$ indicates that two features in the different class are quite different. We can use these two values to quantify the difference between any feature pairs.

In the second stage, we derive the objective function for feature extraction. In order to improve the performance of the projection and classification, we need to move the features in the same class as close as we can. Meanwhile, the features belong to different classes are push away as far as possible.

As mentioned above, we have the simple idea that $E_c(y_{ci})$ should as large as possible, and $E_{p\neq c}(y_{ci})$ should be as small as possible. This can ensure the separation between the different classes and the aggregation within the same class. Then, the total resultant effect can be computed as

$$E(y) = \frac{1}{N}\sum_{c=1}^{N_c}\sum_{i=1}^{J_c} \alpha E_c(y_{ci}) - (1-\alpha) E_{p\neq c}(y_{ci}) \tag{4}$$

where $\alpha = \left[\left(1 - \frac{J_c}{k+1}\right)^2 + \sum_{\substack{p=1 \\ p\neq c}}^{N_c}\left(\frac{J_p}{k+1}\right)^2\right]$ presents the effects from the same class and the different class, k is the number of neighborhood of y_{ci}. The first term of α means effects of all features to y_{ci}, the second term means effects of other features to y_{ci} except the features in class c.

However, $y_{ci} = \phi(x_{ci})$ cannot be computed explicitly. So we transform it to a coefficient matrix learning problem. In the kernel space, we can project y_{ci} into a new feature space, and define this space as F for simplicity. Owing to the kernel trick, we can modify y_{ci} as following

$$y_{ci} = \langle v, \phi(x_{ci})\rangle = \sum_{s=1}^{C}\sum_{t=1}^{J_c} \beta_{st} K(x_{st}, x_{ci})$$

where K is the kernel matrix, $K(x_i, x_j) = k_{ij}$, β is the coefficient when project the original y_{ci} onto the direction v . In this form, we can compute the variables $y_{pl} - y_{ci}$ in (2) and (3) as

$$y_{pl} - y_{ci} = B\left(\sum_{s,t} \left(K(x_s, x_{ci}) - K(x_t, x_{ci}) \right) \right) \tag{5}$$

where elements of B is constructed by β as well as the kernel matrix.

In the third stage, we need to employ optimization methods to maximize the objective function (4). There are many algorithms we can used, such as traditional quotient method of GDA, but it bears eigendecomposition problem, which may result in high computational complexity. Recently, semi-definite programming (SDP) is widely used for this optimization [16]. But it is also quite time consuming when the number of features is large, so we adopt gradient ascent algorithm.

Substituting (5) into (4), we can transform the feature y learning problem to the coefficient matrix B learning problem as follow

$$E(B) = \frac{1}{N} \sum_{c=1}^{N_c} \sum_{i=1}^{J_c} \alpha E_c(y_{ci}) - (1-\alpha) E_{p \neq c}(y_{ci}) \tag{6}$$

Maximizing (6) creates a transformed feature space with wide separation of the different class and better clustering of the same class. The gradient for the objective function is

$$\frac{\partial E}{\partial B} = \frac{1}{N} \sum_{c=1}^{N_c} \sum_{i=1}^{J_c} \alpha \frac{\partial E_c(y_{ci})}{\partial y_{ci}} - (1-\alpha) \frac{\partial E_{p \neq c}(y_{ci})}{\partial y_{ci}} \tag{7}$$

In Eq. (7), the computation of $\frac{\partial E_c(y_{ci})}{\partial y_{ci}}$ by chain rule is $\frac{\partial E_c}{\partial B} = \frac{\partial E_c}{\partial y_{ci}} \frac{\partial y_{ci}}{\partial B}$ and $\frac{\partial E_{p \neq c}(y_{ci})}{\partial y_{ci}}$ is $\frac{\partial E_{p \neq c}}{\partial B} = \frac{\partial E_{p \neq c}}{\partial y_{ci}} \frac{\partial y_{ci}}{\partial B}$, where $\frac{\partial E_c}{\partial y_{ci}}$ is

$$\frac{\partial E_c}{\partial y_{ci}} = \frac{1}{N} \sum_{j=1}^{J_c} G(y_{cj} - y_{ci}, 2\sigma^2 I) \frac{(y_{cj} - y_{ci})}{\sigma^2} \tag{8}$$

and $\frac{\partial E_{p \neq c}}{\partial y_{ci}}$ is

$$\frac{\partial E_{p \neq c}}{\partial y_{ci}} = \frac{1}{N} \sum_{p=1}^{N_c} \sum_{l=1}^{J_p} G(y_{pl} - y_{ci}, 2\sigma^2 I) \frac{(y_{pl} - y_{ci})}{\sigma^2} \tag{9}$$

and $\frac{\partial y_{ci}}{\partial B}$ is

$$\frac{\partial y_{ci}}{\partial B} = \sum_{s} K(x_s, x_{ci}) \tag{10}$$

substituting (8) (9) and (10) into (7), we can obtain the gradient of the objective function.

Maximizing the objective function (6) using gradient ascent algorithm, we can get the final coefficient matrix B , which can be used for classification and projection.

Finally, the KIFE is briefly described by the following

Algorithm 1. KIFE

```
Input: Training data set $X$ , class label set $C$ , and integer $k$ .
Output: A coefficient matrix $B$ .
// Initialization
Compute the kernel matrix $K = (k_{ij})_{1 \le i \le N, 1 \le j \le N}$ .
// The first stage
Get the graph energy of each feature according to (2) and (3).
// The second stage
Construct the objective function according to (6).
// The third stage
Run the gradient learning optimization method according to (7), (8), (9) and (10).
Return the final coefficient matrix $B$ .
```

In this section we will explain how the gradient of the objective function (6) provides information on the geometry and statistical variables to predicting the class label given a new feature. Our KIFE is motivated by the following idea: the gradient is a local concept as it measures local changes of the objective function (6). In the optimization processing for the objective function, we found that B varying with the gradients of (6) as follows

$$B_{r+1} = B_r + \eta \frac{\partial E}{\partial B}$$

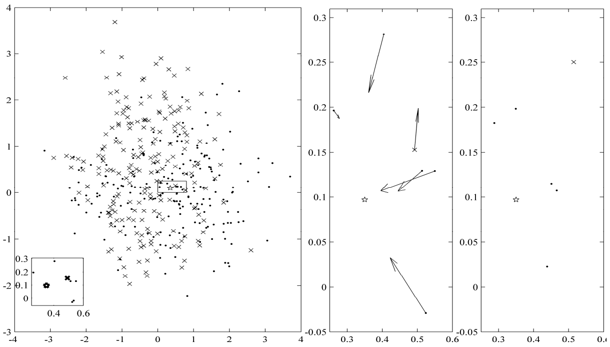
where r is the learning step, η is learning rate, these two parameters can be obtained using the cross-validation method.

We present visualization experiment with synthesized data. In this example we learn a nonlinear projection from a high-dimensional feature space onto a discriminative direction for visualization purposes, specially to visualize classification ability. The data is non-Gaussian densities. It is three-dimensional, and two classes. Class one has 200 samples from a bimodal Gaussian distribution, with centers at (1,0,0) and (-1,0,0). Class two has 200 samples, also from a bimodal Gaussian distribution, with centers at (0,1,0) and (0,-1,0).

Since we have two classes, KIFE is able to produce a two-dimensional projection. In this example, GDA was used as the initial state to learn the KIFE-projection. The result is presented in Figure 1. Using the gradient ascent algorithm, KIFE can converge to the global optimum in 16 iterations, which now exhibits much better separation.

Figure 1 shows our KIFE results on synthesized data, the synthesized data is presented in Figure 1(a), the sub graph in Figure 1(a) is reference features we consider in this example. We use “☆” denoting the feature in class one, the symbol “.” represents the other features in class one, the symbol “×” is those features in class two. Figure 1(b) shows the status of the original features in the input space. After KIFE learning, the final status is presented in Figure 1(c).

From Figure 1 we can found out several characteristics of this example: (1) features in the same class are clustered; (2) features in the different class are pushed away.



(a) The origianl data (b) Before learning (c) After learning

Figure 1. Projection results on synthesized data.

Recall that informative energy function (2) and (3) can reflect the distance of features in the kernel space. The central quantity is an estimate of the gradient of the difference of features, which is controlled by those neighborhoods. So we find k neighborhoods of y_{ci}, those in the same class are moving closer and in the different class are push away. From the view of nearest neighborhood (NN), this can improve the performance of classification and projections. Then our KIFE can be viewed as a graph, each node is the feature in transformed space and is connected with its nearest neighborhood.

B. Relation to MI criterion

We show that different setting of the trade-off parameter α will lead to special versions of KIFE algorithm, which are highly related to the popular criterion MI.

We have the following theorem:

Theorem 1. Let α in (6) a constant, when $k = N-1$, then our objective function (6) is the same as MI criterion.

Proof. In [13], MI is computed by Renyi entropy, according to definitions in our method, we can rewrite the MI computed in [13] as

$$MI = V_{IN} + V_{ALL} - 2V_{BTW} \tag{11}$$

where the quantities appearing in (11) are as follows

$$V_{IN} = \frac{1}{N^2}\sum_{p=1}^{N_c}\sum_{k=1}^{J_p}\sum_{l=1}^{J_p} G(y_{pk}-y_{pl}, 2\sigma^2 I) \tag{12}$$

$$V_{ALL} = \frac{1}{N^2}\left(\sum_{p=1}^{N_c}\left(\frac{J_p}{N}\right)^2\right)\sum_{k=1}^{N}\sum_{l=1}^{N} G(y_k - y_l, 2\sigma^2 I) \tag{13}$$

$$V_{BTW} = \frac{1}{N^2}\sum_{p=1}^{N_c}\frac{J_p}{N}\sum_{j=1}^{J_p}\sum_{k=1}^{N} G(y_{pj}-y_k, 2\sigma^2 I) \tag{14}$$

substituting (12), (13) and (14) into (11), we can obtain the MI values.

In our KIFE, the objective function is

$$E(B) = \frac{1}{N}\sum_{c=1}^{N_c}\sum_{i=1}^{J_c}\alpha E_c(y_{ci}) - (1-\alpha)E_{p\neq c}(y_{ci})$$

where $\alpha = \left[\left(1-\frac{J_c}{k+1}\right)^2 + \sum_{\substack{p=1\\p\neq c}}^{N_c}\left(\frac{J_p}{k+1}\right)^2\right]$.

When we consider all $N-1$ features in the kernel space as neighborhoods of certain feature y_{ci}, this means $k = N-1$. Then, we can computed the parameter in (6) as

$$\alpha = 1 + \left(\frac{J_c}{N}\right)^2 + \sum_{\substack{p=1\\p\neq c}}^{N_c}\left(\frac{J_p}{N}\right)^2 - \frac{2J_c}{N} \tag{15}$$

substituting (15) into (6), we rewrite (6) as

$$E(B) = \frac{1}{N^2}\left(E^1 + E^2 + E^3\right)$$

where E^1, E^2, E^3 are computed as follows, for convenient, we write $G(\cdot-\cdot, 2\sigma^2 I)$ as $G(\cdot-\cdot)$

$$E^1 = \sum_{c=1}^{N_c}\sum_{i=1}^{J_c}\sum_{j=1}^{J_c} G(y_{cj}-y_{ci})$$

$$\begin{aligned} E^2 = &\sum_{c=1}^{N_c}\left(\frac{J_c}{N}\right)^2\sum_{i=1}^{J_c}\left(\sum_{j=1}^{J_c} G(y_{cj}-y_{ci}) + \sum_{\substack{p=1\\p\neq c}}^{N_c}\sum_{l=1}^{J_p} G(y_{pl}-y_{ci})\right) \\ &+\sum_{\substack{p=1\\p\neq c}}^{N_c}\left(\frac{J_P}{N}\right)^2\sum_{c=1}^{N_c}\left(\sum_{i=1}^{J_c} G(y_{cj}-y_{ci}) + \sum_{l=1}^{J_p} G(y_{pl}-y_{ci})\right)\end{aligned}$$

$$E^3 = -2\sum_{c=1}^{N_c}\frac{J_c}{N}\sum_{i=1}^{J_c}\left(\sum_{j=1}^{J_c} G(y_{cj}-y_{ci}) + \sum_{\substack{p=1\\p\neq c}}^{N_c}\sum_{l=1}^{J_p} G(y_{pl}-y_{ci})\right)$$

For E^1, it computes informative energy in the same class, this is the same as V_{IN} in (12) except the normalization factor $\frac{1}{N^2}$. So, $E^1 = \frac{1}{N^2}V_{IN}$.

For E^2, we can find that $\sum_{j=1}^{J_c} G(y_{cj}-y_{ci})$ computes informative energy for class $c(c\neq p)$, the other term $\sum_{l=1}^{J_p} G(y_{pl}-y_{ci})$ computes informative energy for class $p(p\neq c)$. The sum of these two terms can be merged into one term without considering the class label. Then E^2 can be modified as

$$E^2 = \left(\sum_{p=1}^{N_c} \left(\frac{J_p}{N} \right)^2 \right) \sum_{k=1}^{N} \sum_{l=1}^{N} G(y_k - y_l, 2\sigma^2 I)$$

For E^3 , $\sum_{j=1}^{J_c} G(y_{cj} - y_{ci}) + \sum_{\substack{p=1 \\ p \neq c}}^{N_c} \sum_{l=1}^{J_p} G(y_{pl} - y_{ci})$ computes informative energies for class $c(c \neq p)$ and class $p(p \neq c)$, they can be merged into one term as $\sum_{i=1}^{J_c} \sum_{k=1}^{N} G(y_k - y_{ci}, 2\sigma^2 I)$. Then E^3 can be modified as

$$E_3 = -2 \sum_{p=1}^{N_c} \frac{J_p}{N} \sum_{j=1}^{J_p} \sum_{k=1}^{N} G(y_{pj} - y_k, 2\sigma^2 I)$$

substituting E^1 , E^2 and E^3 into $E(B) = \frac{1}{N^2}\left(E^1 + E^2 + E^3 \right)$, we obtain

$$E(B) = V_{IN} + V_{ALL} - 2V_{BTW}$$

Compare E^1 , E^2 and E^3 with (12), (13) and (14), we know that when $k = N - 1$, $E(B)$ is equal to MI.

Then, MI criterion is a special case of our objective function (6). □

As mentioned above, we can get the same criterion as MI, the main difference is that we can change the parameter k to obtain the high performance of classification and projections, and this is the same idea as NN algorithms. However, we cannot necessarily hope to preserve the quality of quantification of difference of features, we may sacrifice its power to obtain corresponding gains in classification accuracy and computational efficiency. Experimental results show that KIFE appears promising in the contexts of classification and projections. This feature is especially desirable for kernel-based methods such as those yield very large kernel matrices for important feature extraction.

C. KIFE for Dimension Reduction via Gradient Learning

In this section, we show how to using KIFE for dimension reduction. The main problem in the kernel space is that its dimensionality depends on the number of features. So we need reduce the dimensionality for visualization and complexity reduction. Here, we employ the gradient learning algorithm [17]. It has some merit suit for our model: (1) it is simple to run; (2) it holds for Euclidean spaces as well as the manifold setting.

Firstly, we define graph Laplacian

$$L = I - D^{-1/2} W D^{-1/2} \tag{16}$$

where I is the identity matrix, $D_{ii} = \sum_j W_{ij}$, W_{ij} is a similarity metric between two points y_i and y_j .

In literature [17], they defined a regression function to compute W_{ij} . Here, we propose a novel similarity metric according to the difference quantification function (4)

$$\begin{aligned} W_{ij} &= W_E(y_i, y_j) \\ &= \exp\left(-\frac{\left\| y_i - y_j \right\|^2}{\sigma^1} - \frac{\left| \frac{1}{2}\left(\nabla E(y_i) + \nabla E(y_j) \right) \cdot \left(y_i - y_j \right) \right|^2}{\sigma^2} \right) \end{aligned}$$

where $\nabla E(y_i) = \frac{\partial E}{\partial y_i}$, σ^1 and σ^2 are parameters.

In the similarity metric W_{ij} , $\left\| y_i - y_j \right\|^2$ represents the local geometry of the marginal distribution and the second term pastes together gradient estimates between neighboring features.

The main differences between the metric in [17] and ours are: (1) the first term in [17] is used in unsupervised dimension reduction such as Laplacian eigenmaps or diffusion maps, but we can use it considering its class label; (2) the second term in [17] can be interpreted as a first order Taylor expansion leading to the following approximation

$$\frac{1}{2}\left(\nabla E(y_i) + \nabla E(y_j) \right) \cdot \left(y_i - y_j \right) \approx E(y_i) - E(y_j) .$$

But ours can be computed explicitly according to the function (4); (3) the method in [17] is defined in the input space X , but ours is defined in the feature space F .

Once the similarity metric W_{ij} is obtained, we can compute the graph Laplacian according to (16). Then, dimension reduction is achieved by projection onto a spectral decomposition of the matrix L .

The pseudo-code of KIFE for dimension reduction is in the following form.

Algorithm 2. KIFE for dimension reduction using gradient learning

```
Input: Training data set $X$ , class label set $C$ , and integer $k$ .
Output: A dimension reduction matrix.
// Initialization
Give an integer $D$ , it represent the final dimension.
// The first step
Compute the similarity metric $W_{ij}$ .
// The second step
Compute the graph Laplacian $L$ according to (16).
// The third step
Run decomposition on $L$ .
// The fourth step
Pick the top $D$ eigenvectors.
Return the final dimension reduction matrix.
```

IV. Experimental Results

In this paper, we develop KIFE for feature extraction based on kernel gradient learning. Our main results are the algorithms 1 and 2 of Section III. The algorithm 1 is mainly used as a preprocessing for classification tasks. The algorithm 2 is used for dimensional reduction.

In this section, we evaluated KIFE on real benchmark data sets of varying size and difficulty. The Phoneme set is available with maximize mutual information (MMI) algorithm in [13]. The rest of data sets are cited from the UCI data sets (http://archive.ics.uci.edu/ml/). The data sets and some of their characteristics are presented in Table 1.

In Table 1, Training/Sampling means that the number of the original features/the number of features sampling for training, D is the dimension reduction number for projection, d is the original dimension of the data set.

TABLE I.
CHARACTERISTICS OF THE DATA SETS AND PARAMETER SETTINGS

Data	Training/Sampling	Testing/Sampling	N_c / d / D
Iris	150/105	150/45	3/4/2
Statlog	4435/1800	2000/2000	6/36/2
Letter	16000/2000	4000/1500	26/16/8
Phoneme	1962/1962	1961/1961	20/20/9

We will conduct two experiments: dimension reduction for projection and classification. We compare KIFE with two existing methods. In order to facilitate the comparison, we duplicate the GDA in [10] and the MMI in [13]. The kernel width σ is selected by the method in [13].

A. *Visualization of Class Separation*

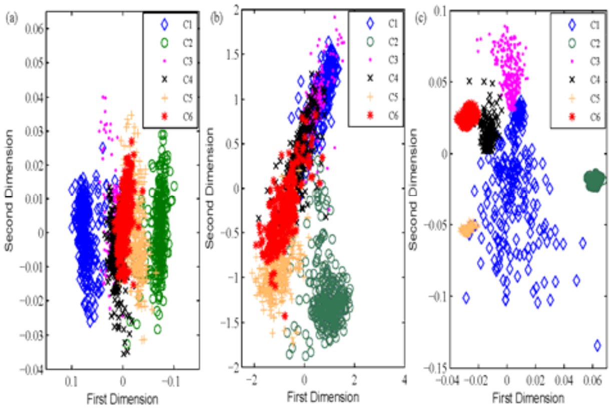
In the first experiment, we illustrate a projection from 36-dimensional Statlog feature space onto two. We employ algorithm 2, the extension of KIFS, to perform dimension reduction. Two existing dimension reduction methods are duplicated for comparison: GDA and MMI. GDA uses the label information to find informative projection such that the separation of data of different classes can be maximized and data of same classes should be highly aggregated. To that end, GDA tries to maximize the inter-class scatter matrix and minimize the intra-class scatter matrix simultaneously. However, GDA suffers from the singular problem when dealing with high-dimensional data, and its dimension reduction number is controlled by the number of classes. MMI is a non-parametric learning method. Promising performance of MMI have been shown in [13] for classification, but it is mainly used as a linear projection algorithm. Since our quantification method is based on a energy function which is similar to MMI, we empirically compared our method with GDA and MMI.

The Statlog is the Statlog satellite image database from UCI Machine Learning Repository. It has 4435 features for training, and 2000 for testing. Its dimensional is 36 and the number of classes is six. For dimension reduction task, we randomly sample 1800 from the training data and use total testing data. Those six classes are according to the label in Figure 2, where C1 represents red soil, C2 is cotton crop, C3 is grey soil, C4 is damp grey soil, C5 is soil with vegetation stubble, and C6 is very damp grey soil. This data is one of the many sources of information available for a scene. The interpretation of a scene by integrating spatial data of diverse types and resolutions including multispectral and radar data, is a difficult task. The GDA projection is presented in Figure 2(a), MMI-projection in Figure 2(b), and KIFE-projection in Figure 2(c).

From the Figure 2, we observed that GDA separates C1, C2 and C5 very well. MMI separates C2 and C5 but places other four classes almost on the top to each other. The criterion of KIFE is a combination of representing each class as compactly as possible and as separated from each other as possible. Figure 2(c) has achieved this: all classes are represented as quite compact clusters. But we should note that C1 is scattered and has small cross parts with other classes. We can obtain the following conclusions: (1) GDA can separate the red soil, cotton crop and vegetation stubble. The other three soil features are overlapping with each other. (2) MMI can separate the cotton crop and vegetation stubble, but the other four soil features cannot be separated well. (3) KIFE firstly separate the cotton crop from other soil features, the other four soil features are separate well except the red soil. This means that the gradient information in the transform processing may be useful for classification problem.

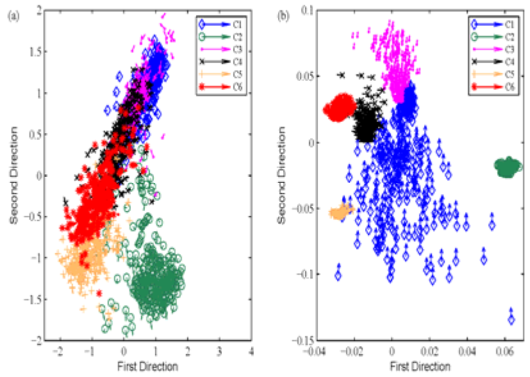
In order to evaluate how the features transform according to KIFE, we plot the gradient of each feature until the coefficient matrix is obtained. In this setting, linear discriminant analysis (LDA) is used as an initial state of MMI, and GDA for KIFE. We also used LDA as an initial state of KIFE, experimental result has shown that our method is not sensitive to the initialization, but need 41 steps to converge to the optimal solution.

Figure 3 shows the iteration status of 24 steps using gradient learning. Figure 3(a) shows the gradient learning about MMI, we can find out why MMI cannot separate other four classes. The direction of C3 has no regular arrangement, and it overlaps with C1. However, KIFE shows better results. All classes have better clustering performance except C1. From the direction of C1 in figure 3(b), we find out that features in C1 have the tendency toward the within class center, and C1 has little overlapping area with C3.



(a) GDA (b) MMI (c) KIFE

Figure 2. Projection results on Statlog.



(a) Iteration status of MMI (b) Iteration status of KIFE

Figure 3. Iteration status of 24 steps using gradient learning.

This result is useful when the feature is difficult to classify. We can predict its potential class label according to its move direction in feature space.

B. Classification Results

In the second experiment, we applied our feature extraction method to improve the classification of nearest neighbor algorithm knnclassify in Matlab.

We must first determine k, the number of the nearest neighbor to construct the objective function. We selected the k from $\{3,5,7,9,11\}$ according to their classification performance. We averaged over 100 runs with random sampling of Landsat data set according to Table 1. Figure 4 shows the interaction between the value of k and the accuracy of nearest neighbor algorithm.

It can be observed that we can obtain high classification performance when $k = 5$. Therefore, we assigned $k = 5$.

In order to evaluate whether our KIFE feature extraction method actually results in a new space that characterizes the difference between features better, we perform our algorithm using other data sets showed in Table 1. Iris is perhaps the best known data used in the pattern recognition literature. The data set contains three classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two, the latter one are not linearly separable from each other. For this small sample, we use cross-validation to random select features for training and testing. The Phoneme is widely used resource for research in speech recognition. It contains 20 classes of 20 dimensional. The Letters is a database of character image features. It contains 26 capital letters, we randomly sample 2000 from 16000 for training and 1500 from 4000 for testing. Detailed description is listed in Table 1, we run experiments 10 runs, the performance of these methods are show in Table 2.

On the other hand, we report the computation time of the KIFE algorithm, as KIFE avoids the eigendecomposition of GDA. For MMI and KIFE, we terminate the algorithms when the iteration step is up to 24. We can observe that KIFE has the similar computational complexity, and they are all superior to GDA. However, the classification performance of GDA is superior to MMI, because GDA can extract nonlinear features. KIFE combines the merits of these two algorithms, so it shows high performance on these data sets.

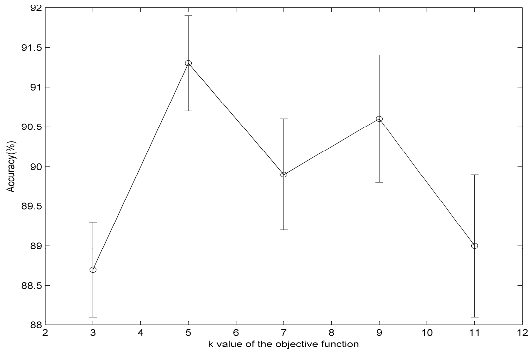


Figure 4. Influence of the k values.

TABLE II.
TEST PERFORMANCE OF PROJECTION ALGORITHMS(%/S)

	Iris	Statlog	Phoneme	Letters
GDA	96.67/0.098	90.5/35.126	86.7/30.354	89.9/13.446
MMI	96.67/0.084	89.5/3.804	85.3/2.745	88.6/1.806
KIFE	97.33/0.093	91.3/4.059	86.7/2.743	92.1/1.849

The above experiments show that it may be beneficial to combine KIFE with classifiers, because KIFE can extract important features and can be used for dimension reduction.

Other advantages of our algorithm are

- KIFE can be easily extended to dimension reduction based on gradient learning. It can be used in many applications, such as in [18-20].
- It can extract high order statistics and nonlinear statistics from data sets. Contrary to GDA and MMI, our method holds both for Euclidean and manifold setting.
- Computational efficiency. Our approach naturally draws advantages from gradient ascent implementation. But the extension of KIFE for dimension reduction needs eigendecomposition, which can be considered as a preprocessing step for classification or regression tasks.
- Extraction ability. The important feature extracted by our method can improve the visual ability of projection, and can be used for classification task when the number of features is large.

The informative energy function can be used as a quantification of feature difference. We expect that this function can be used in bioinformatics in future work.

V. CONCLUSION

A novel algorithm, KIFE, for feature extraction is presented. The algorithm works in an iterative fashion and the final coefficient matrix is obtained during successive iterations. Experimental shows that KIFE has lower time complexity than GDA, and it is superior to MMI in classification performance.

Another algorithm for dimension reduction is also evaluated, which is an extension of KIFE. This can be considered as a preprocessing step for classification. Nevertheless, the fascinating idea of using our approach is that we build a connection between feature extraction and gradient learning.

In future work we intend to apply the proposed method to larger data sets, especially for bioinformatics. We also want to modify our algorithm to parallel implementations. In addition, how gradient of the objective function influence the performance of the classifiers is an interesting topic.

ACKNOWLEDGMENT

The authors wish to thank Editor and referees. This work was supported by the Doctoral Foundation of Jiangxi University of Science and Technology, Jiangxi Educational Committee Foundation for Youths, National Natural Science Foundation of China (61070137, 60702063).

REFERENCES

[1] M. Yang, F. Wang, P. Yang. “A novel feature selection algorithm based on hypothesis-margin”, Journal of Computers, 2008, 3(12): 27-34

[2] L. Li, M. Li, Y. Lu, Y. Zhang. “A new multi-objective genetic algorithm for feature subset selection in fatigue fracture image identification”, Journal of Computers, 2010, 5(7): 1105-1111

[3] I. Rodriguez-Lujan, R. Huerta, C. Elkan, and C. S. Cruz. “Quadratic programming feature selection”, Journal of Machine Learning Research, 2010, 11: 1491-1516

[4] S. Zhu, D. Wang, K. Yu, T. Li. “Feature selection for gene expression using model-based entropy”, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2010, 7(1): 25-36

[5] F. Oveisi, A. Erfanian. “A minimax mutual information scheme for supervised feature extraction and its application to EEG-based brain-computer interfacing”, EURASIP Journal on Advances in Singnal Processing, 2008

[6] K. Lee. “Exploration on feature extraction schemes and classifiers for shaft testing system”, Journal of Computers, 2010, 5(5): 679-686

[7] K. Torkkola, W. M. Campbell. “Mutual information in learning feature transformations”, Proceedings of the 17th International Conference on Machine Learning, Stanford, USA, 2000, 1015-1022

[8] U. Ozertem, D. Erdogmus, R. Jessen. “Spectral feature projections that maximize Shannon mutual information with class labels”. Pattern Recognition, 2006, 39: 1241-1252

[9] X. Yuan, B. Hu. “Robust feature extraction via information theoretic learning”, Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada, 2009, 1193-1200

[10] G. Baudat, F. Anouar. “Generalized discriminant analysis using a kernel approach”, Neural Computation, 2001, 12: 2385-2404

[11] A. R. Teixeira, A. M. Tome, E. W. Lang. “Feature extraction using low-rank approximations of the kernel matrix”, Proceedings of the 5th International Conference on Image Analysis and Recognition, Lecture Notes In Computer Science, Povoa de Varzim, Portugal, 2008, 5112: 404-412

[12] M. Wu, J. Farquhar. “A subspace kernel for nonlinear feature extraction”, Proceedings of the 20th International Joint Conference on Artifical Intelligence, Hyderabad, India, 2007, 1125-1130

[13] K. Torkkola. “Feature extraction by non-parametric mutual information maximization”, Journal of Machine Learning Research, 2003, 3: 1415-1438

[14] R. Kamimura. “An information-theroretic approach to featuren extraction in competitive learning”, Neurocomputing, 2009, 72: 2693-2704

[15] X. Zhu, Z. Ghahramani. “Learning from labeled and unlabeled data with label propagation”, Technical Report CMU-CMLD-02-107, Carnegie Mellon University, Pittsburg, PA, 2000

[16] C. Shen, H. Li, M. J. Brooks. “Feature extraction using sequential semidefinite programming”, Proceedings of the 9th biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications, Glenelg, Australia, 2007, 430-437

[17] Q. Wu, J. Guinney, M. Maggioni, S. Mukherjee. “Learning gradients: predictive models that infer geometry and statistical dependence”, Journal of Machine Learning Research, 2010, 11: 2175-2198

[18] J. Cai, H. Wang, D. Zhou. “Gradient learning in a classification setting by gradient descent”, Journal of Approximation Theory, 2009, 161: 674-692

[19] N. D. Ratliff, J. A. Bagnell. “Kernel Conjugate Gradient for Fast Kernel Machines”, Proceedings of the 20th International Joint Conference On Artifical Intelligence, Hyderabad, India, 2007, 1017-1022

[20] G. Tzimiropoulos, V. Argyriou, S. Zafeiriou, T. Stathaki. “Robust FFT-based scale-invariant image registration with image gradients”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(10): 1899-1906

Songhua Liu received his PhD degree in Technology of Computer Application from Xidian University, Xi’an, China, in 2011. He is currently a lecturer in the College of Science in Jiangxi University of Science and Technology, Ganzhou, China. His research interest covers machine learning and intelligent information processing, image processing, machine learning, bioinformatics.

Jiansheng Liu He is currently a Professor in the College of Science in Jiangxi University of Science and Technology, Ganzhou, China. His research interests cover from intelligent information processing, image processing, machine learning and pattern recognition.