An Informative Feature Extraction Algorithm for Kernel Machines

J. S. Liu[1]
[1]College of Science, Jiangxi University of Science and Technology,
Ganzhou, P. R. China
jxgzjscn@126.com

***Abstract*—In this paper we propose a novel method for feature extraction tasks. The algorithm contains three stages: quantifying of feature difference to determining the importance of all features; constructing a feature extraction model according to the traditional nearest neighbour principle and optimizing this model using gradient based methods. Experimental results on benchmark data set have validated the effectiveness of the proposed method.**

***Index Terms*—Feature extraction, kernel machines, image processing.**

I. INTRODUCTION

Kernel machines play an important role in modern engineering applications, but one challenging problem is how to obtaining important features. This is essential in exploratory data analysis, where we need to map data onto a kernel feature space. Obviously, it can be achieved either by selection methods [1], [2] or transformation [3]–[5] from the input space. The former keeps only useful features and discard others, and the latter constructs new features from the input spaces. However, algorithms that perform feature selection often lead to a combinatorial problem since all features need to be evaluated, but feature extraction only need certain criterion related to the performance of classifiers that can reflect the importance of a feature or a number of features. A further motivation for transforms is the ability to extract distributed relevant information across several original features, which produces a sparse representation.

High-dimensional data are nowadays found in many applications areas: image and signal processing, biological and medical data analysis, etc. The availability of traditional methods to these areas is in general more difficult to analyse, because of the curse of dimensionality, but not for kernel methods.

In this paper, we consider the problem of extracting significant features by a new framework based on kernel methods. Our contributions are as follows: (1) It can extract high order statistics and nonlinear discriminative features; (2) It can avoid the high time usage associated with the traditional eigendecomposition problem; (3) The classic criterion mutual information (MI) can be derived within the proposed method, and it can be extended to other kernel machines easily.

Manuscript received April 6, 2012; accepted June 26, 2013.

II. BACKGROUND AND PRIOR 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. [6] described a generalized discriminant analysis (GDA) method, which depends on the eigendecomposition of the kernel matrix, which bears high computational complexity. Invoked by this problem, Ref. [7] used a low-rank approximation to a complete eigendecomposition of the kernel matrix. Relatedly, Ref. [8] proposed a kernel based nonlinear feature extraction, which transforms this problem to a kernel parameter learning problem. Ref. [9] presented a method for learning discriminative feature transforms using as criterion the MI between class labels and transformed features. Another recent paper by [10] 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[11], [12]. On the other hand, MI according to Shannon's definition is computationally expensive.

III. THE MAIN ALGORITHM

In this section, we describe our method as KFE (Kernel Feature Extraction), 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.

In the first stage of KFE, we propose an informative energy model. The main idea is that we quantify the difference between features according to their graph energy. 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.

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 functions: similar and dissimilar energy. For the features in the same class, the similar energy is computed as follows

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

where I is the identity matrix, G is the Gaussian kernel function, σ is the kernel width parameter.

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

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

These two energy functions vary between 0 and 1. A high E_c indicates that two features in the same class are quite similar. 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

$$E(y) = \frac{1}{N}\sum_{c=1}^{N_c}\sum_{i=1}^{J_c} \alpha E_c(y_{ci}) - (1-\alpha)E_{p\neq c}(y_{ci}), \quad (3)$$

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

$$y_{ci} = \langle v, \phi(x_{ci})\rangle = \sum_{s=1}^{C}\sum_{t=1}^{J_c} \beta_{st} K(x_{st}, x_{ci}), \quad (4)$$

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 (1) and (2) as

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

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 (3). There are many algorithms we can use, such as traditional quotient method of GDA, but it bears eigendecomposition problem, which may result in high computational complexity.

Substituting (5) into (3), we can transform the feature y learning problem to the coefficient matrix B learning problem as follow

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

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

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

In (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

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

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

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

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

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

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 [13]–[16] as following

$$B_{d+1} = B_d + \eta \frac{\partial E}{\partial B} = B_d + \eta \frac{\partial E}{\partial y_{ci}} \frac{\partial y_{ci}}{\partial B}. \quad (11)$$

From (11), the final coefficient matrix B can be obtained, and can be used for classification and projection.

As mentioned above, we can found that our KFE meets with the gradient of linear invariability:

Theorem 1. (Gradient of linear invariability) when computing the energy of KFE according to (7), the optimization of energy function is independent of mapping ϕ of the kernel methods, and this nonlinear transformation can be linearized during the gradient ascent process.

Proof. When kernel transformation is introduced into (6), we can found that the gradient optimization of KFE's energy needs computing items $\partial y_{ci}/\partial B$ and $\partial E/\partial y_{ci}$:

1) The first term can be computed according to the (10);
2) The second term can be derived from the (8) and (9).

The (1) is computed by the Gaussian function as following

$$G(y,\sigma^2) = \exp(-\frac{1}{2\sigma^2} y^T y). \quad (12)$$

Then, the partial derivative of the Gaussian function is

$$\frac{\partial G(y_{cq} - y_{ci}, 2\sigma^2 I)}{\partial y_{ci}} = G(y_{cq} - y_{ci}, 2\sigma^2 I) \frac{y_{cq} - y_{ci}}{2\sigma^2}. \quad (13)$$

Combining this equation and (4), we can found that kernel transformation can be introduced into this optimization.

As mentioned above, the gradient of energy function can be rewritten as the function of the terms $y_{cq} - y_{ci}$ and $y_{pq} - y_{ci}$. The partial derivative of $y_{cq} - y_{ci}$ can be computed by (8), which means similar energy of those samples belong to the same class. In other hand, $y_{pq} - y_{ci}$ can be computed by (9), which means dissimilar energy of those samples belongs to different class.

Then, we can obtain the coefficient matrix B according to (11). In this way, the nonlinear transform by ϕ is converted into the linear transform by terms $y_{cq} - y_{ci}$ and $y_{pq} - y_{ci}$, which means this nonlinear transformation can be linearized during the gradient ascent process.

IV. EXPERIMENTAL RESULTS

We will conduct two experiments: dimension reduction for projection and classification. We compare KFE with two existing methods. In order to facilitate the comparison, we duplicate the GDA in [6] and the MMI in [9]. The kernel width σ is selected by the method in [9].

A. Test our method on synthesized data

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 class separability. 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).

For visualization of the classification ability, we perform our algorithm on this synthesized data, and add the third class which is a single Gaussian distribution, we sample 200 samples. Then, we compare it with some existing methods, such as GDA、MMI and KFE. Fig. 1 shows the projection results.

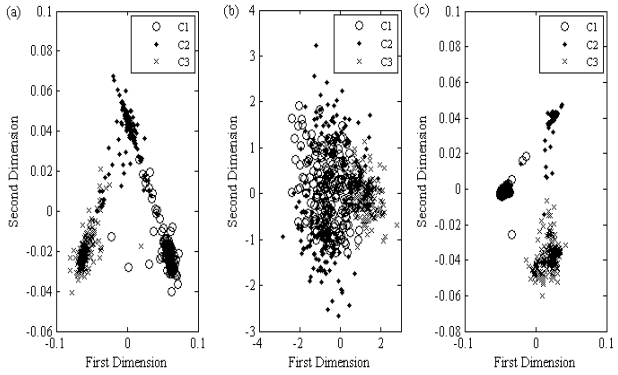

Fig. 1. Visualization of results: (a) GDA; (b) MMI; (c) KFE

From Fig. 1 we can observe that GDA and KFE can produce the better classification ability. Because of the linear properties, MMI cannot give a discriminative projection

In order to explain why KFE can give a better classification result, we show the iteration status of MMI and KFE. Fig. 2 shows this result.

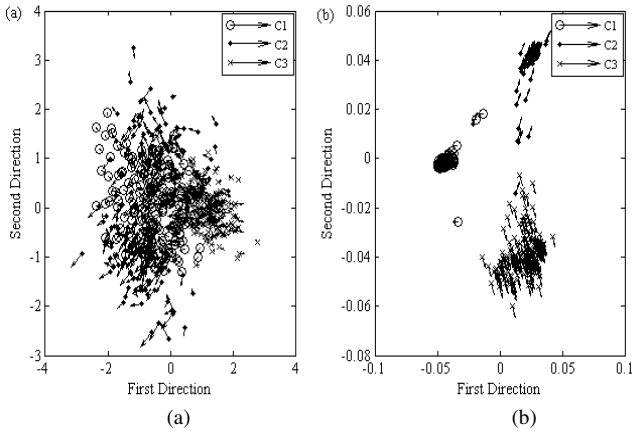

Fig. 2. Status of projection using gradient learning: (a) MMI; (b) KFE

From Fig. 2, we observed that at the certain iteration steps, the samples using GDA are intend to gathering into the centers of these three classes, but MMI can give a more obviously tendency. This means that the gradient information in the transform processing may be useful for classification problem.

B. Test our method on benchmark data

In this section, we evaluated KFE on real benchmark data sets of varying size and difficulty. The Phoneme set is available with maximize mutual information (MMI) algorithm in [9]. The rest of data sets are cited from the UCI data sets. The data sets and some of their characteristics are presented in Table I.

TABLE I. CHARACTERISTICS OF THE DATA SETS AND PARAMETERS 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

In Table II, 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 dataset.

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

	Iris	Statlog	Phoneme	Letters
GDA	96.67/0.09	90.5/35.12	86.7/30.35	89.9/13.44
MMI	96.67/0.08	89.5/3.80	85.3/2.74	88.6/1.80
KFE	97.33/0.09	91.3/4.05	86.7/2.74	92.1/1.84

We applied our feature extraction method to improve the classification of nearest neighbor algorithm [17] knnclassify in Matlab. Table II shows the classification performance of three methods: GDA、MMI and KFE.

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

V. CONCLUSIONS

A novel algorithm, KFE, 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 KFE has lower time complexity than GDA, and it is superior to MMI in classification performance.

Another algorithm, an extension of KFE, is also evaluated for dimension reduction. 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.

REFERENCES

[1] W. 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 Computer*, vol. 5, no. 8, pp. 1105–1111, 2010.

[2] I. Rodriguez–Lujan, R. Huerta, C. Elkan, C. S. Cruz, "Quadratic programming feature selection", *Journal of Machine Learning Research*, no. 8. 1491–1516, 2010.

[3] 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

[4] K. Lee, "Exploration on feature extraction schemes and classifiers for shaft testing system", *Journal of Computers*, vol. 5, no. 5, pp. 679–686, 2010. [Online]. Available: http://dx.doi.org/10.4304/jcp.5.5.679-686

[5] X. Yuan, B. Hu, "Robust feature extraction via information theoretic learning", in *Proc. of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009, pp. 1193–1200.

[6] G. Baudat, F. Anouar, "Generalized discriminant analysis using a kernel approach", *Neural Computation*, no. 3, pp. 2385–2404, 2001.

[7] A. R. Teixeira, A. M. Tome, E. W. Lang, "Feature extraction using low–rank approximations of the kernel matrix", in *Proc. of the 5th International Conference on Image Analysis and Recognition, Lecture Notes In Computer Science,* Povoa de Varzim, Portugal, 2008, no. 5112, pp. 404–412.

[8] M. Wu, J. A. Farquhar, "Subspace kernel for nonlinear feature extraction", in *Proc. of the 20th International Joint Conference on Artifical Intelligence*, Hyderabad, India, 2007, pp. 1125–1130.

[9] K. Torkkola, "Feature extraction by non–parametric mutual information maximization", *Journal of Machine Learning Research*, no. 3, pp. 1415–1438, 2003.

[10] R. Kamimura, "An information–theroretic approach to featuren extraction in competitive learning", *Neurocomputing*, no. 72, pp. 2693–2704, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2008.09.013

[11] 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.

[12] C. Shen, H. Li, M. J. Brooks, "Feature extraction using sequential semidefinite programming", in *Proc. of the 9th biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, Glenelg, Australia, 2007, pp. 430–437.

[13] Q. Wu, J. Guinney, M. Maggioni, S. Mukherjee, "Learning gradients: predictive models that infer geometry and statistical dependence", *Journal of Machine Learning Research*, no. 11, pp. 2175–2198, 2010.

[14] J. Cai, H. Wang, D. Zhou, "Gradient learning in a classification setting by gradient descent", *Journal of Approximation Theory*, no.161, pp. 674–692, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.jat.2008.12.002

[15] N. D. Ratliff, J. A. Bagnell, "Kernel Conjugate Gradient for Fast Kernel Machines", in *Proc. of the 20th International Joint Conference On Artifical Intelligence*, Hyderabad, India, 2007, pp. 1017–1022.

[16] 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, no. 32, pp. 1899–1906. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2010.107

[17] B. Schölkopf, A. J. Smola, *Learning with kernels*. MIT Press, Cambridge, MA, 2002.