LABEL-BASED MULTIPLE KERNEL LEARNING FOR CLASSIFICATION

Bing Yang[1], Qian Li[1], Lujia Song[1], Changhe Fu[1], Ling Jing[1]

[1]College of Science, China Agricultural University, Beijing 100083, P.R. China
jingling@cau.edu.cn

Keywords: Kernel methods, Multiple kernel learning, Similarity-based classification, Support Vector Machine.

Abstract

This paper provides a novel technique for multiple kernel learning within Support Vector Machine framework. The problem of combining different sources of information arises in several situations, for instance, the classification of data with asymmetric similarity matrices or the construction of an optimal classifier from a collection of kernels. Often, each source of information can be expressed as a similarity matrix. In this paper we propose a new method in order to produce a single optimal kernel matrix from a collection of kernel (similarity) matrices with the label information for classification purposes. Then, the constructed kernel matrix is used to train a Support Vector Machine. The key ideas within the kernel construction are twofold: the quantification, relative to the classification labels, of the difference of information among the similarities; and the linear combination of similarity matrices to the concept of functional combination of similarity matrices. The proposed method has been successfully evaluated and compared with other powerful classifiers on a variety of real classification problems.

1 Introduction

Kernel based methods such as Support Vector Machine (SVM) have proven to be powerful for a wide range of different data analysis problems. They employ a so-called kernel function $k(x_i, x_j)$, which intuitively computes the similarity between two examples x_i and x_j .The specific literature on the combination of matrix-like sources of information is rather in its beginnings [16,17]. For the particular case of information arising from kernel matrices, a usual approach is to consider linear combinations of the matrices. This is the proposal in [9], which is based on the solution of a semi-definite programming problem [13] to calculate the coefficients of the linear combination. Special purpose implementations, in order to improve the computational cost required for the solution of this type of optimization problems, are supplied [4, 5]. The main difference between both approaches is the way in which the weights within the semi-definite programming problem are found. The ideas introduced by [9] are extended by [11]. This work is based on the definition of a kernel (called hyperkernel) in the space of kernels itself, leading to the semi-definite optimization problem. Finally, it is worthwhile to mention the proposal in [3]. The method, called MARK-L, builds a classifier (not the specific kernel matrix) by a boosting type algorithm. So far, in multiple kernel learning problem, especially for the co-correction of original kernel function, label information has not been adopted.

Label information has been well used in machine learning problem. Yu-Feng Li [15] has been successfully applied label mean to control the number of the positive points in the unlabeled data for semi-supervised learning problems. Xin Geng [14] has constructed probabilistic regression model from label contributions for facial age estimation problem. Javier M. Moguerza [10] combined the multi-source similarity matrices with label information by heuristic methods for classification problem, but in the paper, no quantitative function has been proposed for the construction of the optimal kernel matrix. In this paper, we propose a quantitative function to construct the de-noise kernel matrices with label information, and use the MKL (multiple kernel learning) model to solve the classification problems.

This paper is organized as follows. The general framework is presented in Sect. 2. In Sect. 3, the problem at hand is motivated. The experimental setup and results on artificial and real data sets are resumed in Sect. 4, Sect. 5 concludes.

2 Methods

2.1 Multiple Kernel Learning (MKL)

As the background of this paper, this section will introduce the basic idea of multiple kernel learning (MKL) and the standard multiple kernel learning within SVM framework.

Multiple kernel learning (MKL) aims at simultaneously learning a kernel and the associated predictor in supervised learning settings. Let $\{x_i, y_i\}_{i=1}^{l}$ is the learning set, where x_i belongs to some input space X and y_i is the target value for pattern x_i . For kernel algorithms in SVM, the solution of the learning problem is of the form

$$f(x) = \sum_{i=1}^{l} \alpha_i^* K(x, x_i) + b^* \qquad (2\text{-}1\text{-}1)$$

where α_i^* and b^* are some coefficients to be learned from examples, while $K(\cdot,\cdot)$ is a given positive definite kernel associated with a reproducing kernel Hilbert space (RKHS) H.

In some situations, a machine learning practitioner may be interested in more flexible models. Recent applications have shown that using multiple kernels instead of a single one can enhance the interpretability of the decision function and improve performances. In such cases, a convenient approach is to consider that the kernel $K(x,x')$ is actually a convex combination of basis kernels:

$$K(x,x') = \sum_{m=1}^{M} d_m K_m(x,x'), \quad with \quad d_m \geq 0, \quad \sum_{m=1}^{M} d_m = 1 \qquad (2\text{-}1\text{-}2)$$

where M is the total number of kernels. Each basis kernel K_m may either use the full set of variables describing x or subsets of variables stemming from different data sources. Alternatively, the kernels K_m can simply be classical kernels (such as Gaussian kernels) with different parameters. Within this framework, the problem of data representation through the kernel is then transferred to the choice of weights d_m.

In the MKL-SVM methodology, the decision function is of the form

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i \sum_{m=1}^{M} d_m K_m + b \qquad (2\text{-}1\text{-}3)$$

Where the optimal parameters d_m, α_i and b are obtained by solving the dual of the following optimization problem [1, 2, 5, 7, 12]:

$$\begin{aligned} &\min_{\{f_m\},b,\xi,d} \frac{1}{2}\sum_m \frac{1}{d_m}\|f_m\|_{H_m}^2 + C\sum_i \xi_i \\ &s.t. \quad y_i \sum_m f_m(x_i) + y_i b \geq 1-\xi_i \quad \forall i \\ &\xi_i \geq 0 \quad \forall i \\ &\sum_m d_m = 1, \quad d_m \geq 0 \quad \forall m \end{aligned} \qquad (2\text{-}1\text{-}4)$$

a) Label-based MKL (LB-MKL)

In this section, we will introduce the basic principle for the construction of label-based kernel function, and give the general formula for the kernel function. Then, two-step method will be proposed to solve the label-based multiple kernel learning problem.

i. Label Information Added in the Kernel Function

For the classification problem, how to measure the similarity between the individuals? The natural idea is that individuals belonging to the same class should be similar, and individuals belonging to different classes should be un-similar. Since kernel function is one of the metrics of similarity between individuals, then the value of kernel function between the individuals belonging to the same class should be large, and individuals belonging to different classes should be small.

So, it is reasonable to add the label information to the construction of kernel function. The formula of Label-based kernel function can be defined as: $K^*(x_i,x_j) = h(K(x_i,x_j),Y)$, Which the kernel funcition is multi-variable about the original kernel matrix and the label information.

In this paper, we use an exponential weighted approach to combine the label information with the original kernel matrices; we define the kernel function as:

$$K^*(x_i,x_j) = (1 + y_i y_j e^{-beta\|x_i - x_j\|^2}) K(x_i,x_j) \qquad (2\text{-}2\text{-}1)$$

In the formula, beta is a given parameter with $beta \geq 0$, which can be qualified by cross-validation. The factor $1 + y_i y_j e^{-beta\|x_i - x_j\|^2}$ is called adjusting factor, the range of which is [0, 2].

The adjusting factor is restricted by the label information and the distance of the two points x_i and x_j. If the two points x_i and x_j are very close in distance metric belonging to the same class, the factor is calculated large, the original kernel $k(x_i,x_j)$ is expanded by the factor, otherwise, if the two points x_i and x_j are very close in distance metric belonging to the different classes, the factor is calculated small, the original kernel $k(x_i,x_j)$ is co-corrected by compression.

The adjusting factor has the following properties:

(1). When the two points x_i and x_j are close to each other in the distance metrics and belong to the same class , the factor is larger than 1, and the original kernel function is expanded. This means that the original kernel function is trustable and should be strengthened.

(2). When the two points are far away from each other in the distance metrics and do not belong to the same class, the factor is smaller than 1. The original kernel function is compressed. This means that the original kernel function is not trustable and should be un-strengthened

(3). When the two points x_i and x_j are close to each other in the distance metrics and do not belong to the same class, the factor is smaller than 1, and the original kernel function is compressed, This means that the original kernel function is not trustable and should un-strengthened.

(4). When the two points x_i and x_j are far away from each other and belong to the same class, the factor is larger than 1, and the original kernel function is expanded. This means that the original kernel function is trustable and should be strengthened.

The formula of kernel function with label information can be other ways so long as it satisfies the four above properties.

After the weighting approach, it is necessary to transform the matrix $\mathbf{K}^*=(K^*(x_i,x_j))$ to a semi-definite positive matrix $\mathbf{K}^{**}=(K^{**}(x_i,x_j))$, so that the matrix $\mathbf{K}^{**}$ is a Gram matrix.

Proposition: For the two points x_i and x_j ($i,j=1,\cdots,l$), their kernel function $K^*(x_i,x_j)$ is defined by Equation (5), so we get the kernel matrix $\mathbf{K}^*$. Let $K^{**}=K^*\times K^*$, then the matrix $\mathbf{K}^{**}$ is a SDP matrix.

Proof: Since similarity matrix $\mathbf{K}^*$ is symmetric, then we consider the spectral decomposition of matrix $\mathbf{K}^*$:

$$K^*=U\ \Lambda U^T,$$

Where the matix $U=[u_1,u_2,\cdots,u_n]$ is an orthonormal matrix, whose columns are the corresponding eigenvectors, and Λ is the a diagonal matrix containing the eigenvalues of K^*.

Since U is an orthonormal matrix, then

$$u_i^T\cdot u_j=\begin{cases}0, & i\neq j, \quad i,j=1,\cdots,n\\ 1, & i=j, \quad i,j=1,\cdots,n.\end{cases}$$

So, we have $U^TU=E$, where E is an identity matrix. Then

$$K^{**}=K^*\times K^*=U\ \Lambda(U^TU)\Lambda U^T$$
$$=U\Lambda(E)\Lambda U^T=U(\Lambda^2)U^T,$$

where Λ^2 is a diagonal matrix containing the eigenvalues of K^{**}.

Since All the elements of the diagonal eigenvalue matrix Λ^2 is non-negative, so the matrix $\mathbf{K}^{**}=(K^{**}(x_i,x_j))$ is a SDP matrix.

ii. Label-based multiple kernel learning (LB-MKL)

After the construction of the kernel function, we propose two-step method to solve the problem. And the framework of LB-MKL approach is shown in Fig.1.

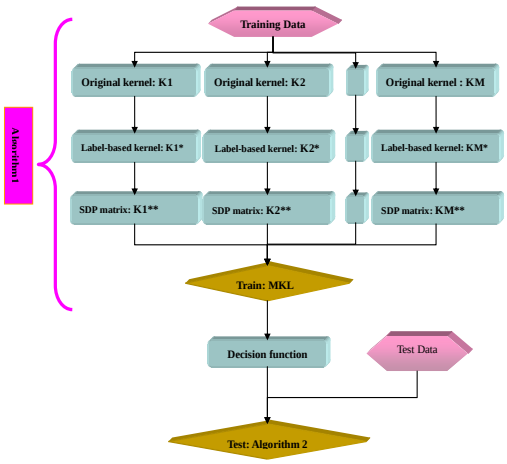

Figure 1. The framework of LB-MKL

In the framework of LB-MKL approach, there are two algorithms, in corresponding to the training process and the testing process. They are described in detail below.

Algorithm 1:Training Algorithm

1. Let $\mathbf{K}_1,\mathbf{K}_2,\cdots,\mathbf{K}_M$ be a set of normalized input similarity matrices calculated from the training data points $\{x_1,x_2,\cdots,x_n\}$,drawn from a possibly unknown statistical distribution X.

2. Build a single symmetric similarity matrix using the proposed weighting approach for each original similarity matrix. Get a set of correcting similarity matrices $\mathbf{K}_1^*,\mathbf{K}_2^*,\cdots,\mathbf{K}_M^*$.

3. Transform each $\mathbf{K}_m^*$ into a PSD matrix $\mathbf{K}_m^{**}$, respectively.(m=1, ...,M)

4. Use $\mathbf{K}_1^{**},\mathbf{K}_2^{**},\cdots,\mathbf{K}_M^{**}$ to train a multiple kernel learning within the framework of MKL-SVM, for the computation of the vector of weights alpha that will be used to build the discrimination rule at testing time.

Given an unlabeled data point x, $\mathbf{K}_1^*,\mathbf{K}_2^*,\cdots,\mathbf{K}_M^*$ has to be evaluated. Since labels are needed to evaluate $\mathbf{K}_1^*,\mathbf{K}_2^*,\cdots,\mathbf{K}_M^*$, we can calculate two different values for $\mathbf{K}_1^*,\mathbf{K}_2^*,\cdots,\mathbf{K}_M^*$: the first one is

$\mathbf{K}_{m+}^*(x,x_i),(m=1,\cdots,M,i=1,\cdots,l)$, assuming x belongs to class +1.

And the second one is

$\mathbf{K}_{m-}^*(x,x_i),(m=1,\cdots,M,i=1,\cdots,l)$, assuming x belongs to class -1.

For each assumption, all we have to do is to predict the class x belongs to. This can be made by calculating the conditional decision hyperplanes under each assumption, that is $f_+(x)$ and $f_-(x)$. Then, using a voting scheme, the a posteriori class for x can be predicted. These stages are summarized in Algorithm 2.

Algorithm 2 *Testing algorithm*

1.Consider an unlabeled point x.

2.Calculate

$$f_+(x)=\sum\alpha_i y_i(\sum_d d_m K_{m+}^*)\ \text{and}\ f_-(x)=\sum\alpha_i y_i(\sum_d d_m K_{m-}^*),$$

where K_{m+}^* and K_{m-}^* correspond, respectively to K_m^* assuming x belong to class +1 an -1.

3.Calculate

3 Results

In this section, to test the performance of the proposed method, we first perform the artificial experiments to show the label-based kernel function is more powerful for the classification problems. And then the experiments on real world data is performed, compared with other algorithms.

3.1 Simulation

In this section, we illustrate the proposed method is more useful to the classification purpose. We demonstrate the distribution of data set by proposed method.

We build a data set made up of 100 two-dimensional points (50 per class). To build the set, we first generate two uniform distribution with the interval of (-1, 0) and (0, 1), respective. Then we add the noise to the data set. The noise is Guassian distribution with variance of sigma and mean of 0. In this section, we set sigma for 0, 0.5, 1 and 2. For the original data set, we use the proposed method to build the label-based kernel matrix, and the original kernel function is Guassian kernel function. For the label-based kernel matrix, we adopt eigenvalue decomposition method to take the two principle cmponoents with the first two large eigenvalues. The Distributions of the original data and decomposition data are shown in Fig. 2 (*a*) ~ (*l*). They have the following characters:

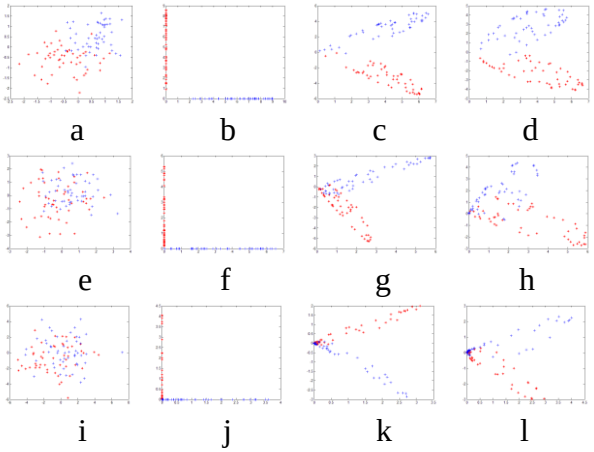

Figure 2. The Distributions of the original data and decomposition data

(*a*): Original dataset added by the noise with *mean*=0 and standard variance $\sqrt{sigma}$ =0.5.

(*b*): Distribution of data in hyperspace after the decomposition of kernel matrix with *beta*=0.

(*c*): Distribution of data in hyperspace after the decomposition of kernel matrix in the kernel space with *beta*=1.

(*d*): Distribution of data in hyperspace after the decomposition of kernel matrix with *beta*=2.

(*e*) : Original data dataset added by the noise with *mean*=0 and standard variance $\sqrt{sigma}$ =1.

(*f*): Distribution of data in hyperspace after the decomposition of kernel matrix with *beta*=0.

(*g*): Distribution of data in hyperspace after the decomposition of kernel matrix with *beta*=0.5.

(h): Distribution of data in hyperspace after the decomposition of kernel matrix with *beta*=1

(*i*): Original data dataset added by the noise with *mean*=0 and standard variance $\sqrt{sigma}$ =2

(*j*): Distribution of data in hyperspace after the decomposition of kernel matrix with *beta*=0.

(*k*): Distribution of data in hyperspace after the decomposition of kernel matrix with *beta*=0.5.

(*l*): Distribution of data in hyperspace after the decomposition of kernel matrix with *beta*=1.

Fig. 2, it is clear that after adding the label information to the kernel function, the data points in the hyperspace are more separable than in the original space. So the label-based kernel function is more useful for classification.

3.2 Experiments on Real World Datasets

We apply the proposed method to five UCI data sets [10], the cancer data set, the ionosphere data set, the heart disease data set, and the vote recording data set. The description of the data sets can be consulted in Table 1.

Since the proposed label-based kernel function has not been used in single kernel method ever, we perform the label-based kernel function in the framework of single kernel method SVM, called LB-S-SVM(label-based single kernel SVM). Furthermore, in the experiments, we perform LB-MKL(label based multiple kernel learning), LB-S-SVM(label-based single kernel SVM) on the datasets. And the proposed methods are compared with SSVM(standard SVM), LDA(linear discriminate analysis), and K-NN(k-nearest neighborhood). And the experimental results of SSVM, LDA and K-NN are cited from the literature [9].

Table 1. Summary of the data sets used in the experiments

Database	Number of data	Dimension	Classes
Cancer	683	9	2
Ionosphere	351	34	2
Heart	462	9	2
Hepatitis	155	19	2
Vote	435	16	2

For each dataset, we have used 80% of the data for training and 20% for testing. The parameters for each algorithm are selected from the set $\{2^i | i = -7, \ldots, 7\}$ by ten-fold cross validation on each training data. The results of average error ratio on training data and testing data over 10 runs of the experiments are shown in Table 2.

Table 2. The results of average error ratio on train and test data

Database	LB-MKL (Train Test)	MKL (Train Test)	LB-S-SVM (Train test)	SSVM (Train Test)	LDA (Train Test)	K-NN (Train Test)
Cancer	**1.45**	1.58	1.57	1.6	3.8	2.2
	3.71	3.70	3.82	3.9	3.9	**2.6**
Ionosphere	**0**	0	2.74	2.7	7.7	9.5
	5.87	7.60	7.51	6.5	14.1	16.1
Heart	**23.04**	23.05	21.5	21.6	25.2	27.8
	25.52	28.44	29.6	29.1	28.7	26.5
Hepatitis	5.52	6.45	**5.00**	5.1	9.2	12.8
	14.96	15.17	17.61	18	17.6	17.4
Vote	**1.06**	1.77	1.52	1.5	4.4	5.1
	3.76	3.90	3.80	**3.**7	4.3	6.7

From the experimental results, our proposed method is powerful for classification. When the dimension of the dataset is high, the proposed method is much powerful than other methods.

3.3 The Sensitivity of The Parameter Beta

In LB-S-SVM and LB-MKL, *Beta* is a given parameter, when *beta* is too small, the clarifier will be over-fitting, and when *beta* is too large, the effect of label information will not be imposed. In this experiment we show how the value of *beta* impacts on the accuracy of the test sets. For each real world data set , we set *beta* for 1,2,4,8,16,32, And the results of the average accuracy on the test data of 10 runs are shown in Fig. 3.

From the five figures, it is clear that when *beta* is getting larger, the accuracy of test data is flat with high accuracy, we can get the highest accuracy with finite *beta*. To find the best *beta* for the training data set, we can adopt grid searching in the feasible interval of *beta* until the highest accuracy shows up.

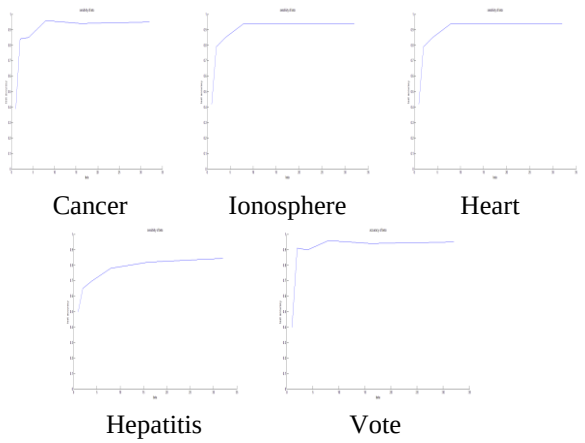

Figure 3. The results of the average accuracy with different parameter *Beta*

4 Conclusion

In this paper we have proposed a novel technique for multiple kernel learning problems within the context of SVM classifiers. The proposed framework is based on the natural idea that individuals belonging to the same class should be similar. This is supported by the fact that the suggested method compares favorably theoretically and computationally to other well established classification techniques in a variety of data sets.

Regarding further research, a natural extension is to study the other formula of kernel function based on label information, and the application of this methodology to other kernel-based classification methods.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (10971223, 11071252), Chinese Universities Scientific Fund (2012YJ130, 2013YJ010).

References

[1] Alain Rakotomamonjy and Francis Bach, 2008, SimpleMKL, Journal of Machine Learning Research 9 ,pp:2491-2521.

[2] Alain Rakotomamonjy and Francis Bach, 2007, More Efficiency in Multiple Kernel Learning, Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR.

[3] Bennett K., Momma M., Embrechts, J., 2002. MARK: a boosting algorithm for heterogeneous kernel models. In Proceedings of SIGKDD international conference on knowledge discovery and data mining.

[4] Bousquet O., Herrmann D., 2003. On the complexity of learning the kernel matrix. In S. Becker,S. Thurn, & K. Obermayer (Eds.), Advances in neural information processing systems: Vol. 15 (pp. 415–422). Cambridge: MIT Press.

[5] Francis Bach., 2007, Consistency of the Group Lasso and Multiple Kernel Learning. the International Conference on Machine Learning (ICML).

[6] Francis Bach and Gert R. G. Lanckriet., 2004, Multiple kernel learning, conic duality, and the SMO algorithm. Proceedings of the 21th international conference on Machine learning.

[7] Francis Bach, Lanckriet G. and Jordan M. 2004. Multiple kernel learning, conic duality and the SMO algorithm. In Proceedings of the 21st international conference machine learning. New York: ACM.

[8] Isaac Martín de Diego, Alberto Muñoz, 2010, Javier M. Moguerza. Methods for the combination of kernel matrices within a support vector framework. Machine learning.

[9] Lanckriet N. et al., 2004, Learning the kernel matrix with semi-definite programming. Journal of Machine Learning Research, 5:27–72.

[10] Murphy P.M., Aha D.W., UCI Machine Learning Repository, 1992, www.ics.uci.edu/~mlearn/MLRepository.html.

[11] Ong C. S., Smola A. and Williamson R., 2005. Learning the kernel with Hyperkernels. Journal of Machine Learning Research, 6, 1043–1071.

[12] Sören Sonnenburg and Gunnar Rätsch, 2006, Large Scale Multiple Kernel Learning, Journal of Machine Learning Research 7 ,1531–1565 .

[13] Vandenberghe L. and Boyd S., 1996. Semidefinite programming. SIAM Review, 38(1), 49–95.

[14] Xin Geng, Kate Smith-miles, Zhi-Hua Zhou, 2010, Facial age estimation by learning form label distribution. Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence.

[15] Yu-Feng Li, James T.Kwok, Zhi-Hua Zhou, 2009, Semi-Supervised Learning Using Label Mean. Prpceeding of the 26th International Conference on Machine Learning, Montreal Canada.

[16] Zhe Wang, Wenbo Jie, Songcan Chen, Daqi Gao, 2012, Random projection ensemble learning with multiple empirical kernels, Knowledge-Based Systems, In Press.

[17] Zheng-Peng WU and Xue-Gong ZHANG, 2011, Elastic Multiple Kernel Learning, Acta Automatica Sinica, Volume 37, Issue 6, June 2011, Pages 693-699.