l_p-norm Multiple Kernel Learning with Diversity of Classes

Dayin Zhang[1,2] and Hui Xue[1,2,3,⋆]

[1] School of Computer Science and Engineering, Southeast University,
Nanjing, 210096, P.R. China
[2] Key Laboratory of Computer Network and Information Integration
(Southeast University), Ministry of Education, Nanjing, 210096, P.R. China
[3] State Key Laboratory for Novel Software Technology, Nanjing University,
Nanjing, 210093, P.R. China
{zhangdayin,hxue}@seu.edu.cn

Abstract. Multiple Kernel Learning (MKL) can learn an appropriate kernel combination from multiple base kernels for classification problems. It is often used to handle binary problems. However, multi-class problems appear in many real applications. In this paper, we propose a novel model, l_p-norm multiple kernel learning with diversity of classes (LMKLDC), for the multi-class multiple kernel learning problem. LMKLDC focuses on diversity of classes and aims to learn different kernel combinations for different classes to enhance the flexibility of our model. LMKLDC also utilizes l_p-norm ($0 < p \leq 1$) to promote the sparsity. However, LMKLDC boils down to a non-convex optimization problem when $0 < p < 1$. In virtue of the constrained concave convex procedure (CCCP), we convert the non-convex optimization problem into a convex one and present a two-stage optimization algorithm. Experimental results on several datasets show our model selects fewer kernels and improves the classification accuracy.

Keywords: Multiple kernel learning, Multi-class classification, Diversity of classes, l_p-norm.

1 Introduction

So far, Multiple Kernel Learning (MKL) has attracted much attention from the community of machine learning. MKL helps users to select the most suitable kernel for learning problems at hand. Besides, MKL can deal with one case in which there are many heterogeneous data sources, especially in bioinformatics. Related researches include Lanckriet et al. (2004), Bach et al. (2004), Rakotomamonjy et al. (2008), Kloft et al. (2009), Xu et al. (2010) and Xu et al. (2013).

MKL methods above can be applied to the binary learning problem. However, we often face multi-class learning problems in the real world. So it is necessary to design effective MKL algorithms to solve multi-class problems. Such a MKL

[⋆] Corresponding author.

Y.S. Kim et al (Eds.): PKAW 2014, LNCS 8863, pp. 38–47, 2014.

algorithm is called MCMKL. Zien et al. (2007) proposed a model of MCMKL based on the joint feature maps (Tsochantaridis et al. 2004) and the multi-class loss function (Crammer et al. 2002). Ye et al. (2008) decomposed a multi-class problem into multiple binary problems via some kind of coding. Kumar et al. (2012) converted the classification problem in the original input space into the binary classification problem in the proposed K-space. Recently, Cortes et al. (2013) introduced the notion of multi-class kernel margin and constructed the corresponding MCMKL model. However, most of these MCMKL methods learn the same weights of combining kernels for all classes. When there exists the diversity of classes, doing so seems unreasonable. Especially, as mentioned in (Zien et al. 2007), if the user is only interested in which kernel can distinguish one class from the rest, disadvantages of sharing a common feature space for all classes arise. Hence, it is obvious that learning the same weights for all classes restricts the flexibility of the model and will decrease its classification performance.

In addition, there are some researchers who are interested in enhancing the sparsity of MKL models by constraining different norms of the weights of combining kernels. The sparsity can lower the complexity of MKL models to improve the generalization performance and save the computation cost. In (Lanckriet et al. 2004), (Bach et al. 2004), (Rakotomamonjy et al. 2008) and (Xu et al. 2013), they directly or indirectly used the l_1-norm (it has a better sparsity than l_2-norm) of the weights. In Bach et al. (2008), the equivalence between group lasso and MKL is demonstrated. Szafranski et al. (2010) and Nath et al. (2009) combined MKL with the mixed-norm, still using l_1-norm to promote the sparsity. Basically, many MKL methods constrain l_1-norm of the weights to improve the sparsity of models. However, using l_p-norm $(0 < p < 1)$ can get a better sparsity than using l_1-norm in plenty of computational studies (Chartrand 2007; Chartrand et al. 2008; Xu et al. 2012). So it is worth trying combining MKL and l_p-norm $(0 < p \leq 1)$.

In this paper, we expand MKL to multi-class problems and propose a novel algorithm LMKLDC which considers diversity of classes. LMKLDC learns different weights of combining kernels for different classes to improve the flexibility of the model. Meanwhile, LMKLDC also utilizes l_p-norm $(0 < p \leq 1)$ to promote the sparsity, which can reduce the computation cost. However, LMKLDC boils down to a non-convex optimization problem when $0 < p < 1$. Unlike (Rakotomamonjy et al. 2011), separately solving optimization problems for $p = 1$ and $0 < p < 1$, we present an unified method to solve relevant optimization problems, no matter what the value of p takes. We apply the two-stage approach to find the classifier's parameters and the weights of combining kernels. We utilize the constrained concave convex procedure (CCCP) (Smola et al. 2005) to transform the non-convex optimization problem into a convex one, which is a quadratic programming and can be solved by the existing toolbox CVX.

The rest of the paper is organized as follows. In section 2, we discuss the multi-class kernel-based learning problem. Section 3 presents the proposed LMKLDC. Section 4 shows the experimental results and conclusions are drawn in Section 5.

2 Multi-Class Kernel-Based Learning (MCKL)

Given the multi-class data $x \in \mathbf{X}$ and the labels $y \in \mathbf{Y} = \{1, ..., m\}$, $m > 2$ and (x, y) is from some unknown distribution P over $\mathbf{X} \times \mathbf{Y}$. According to the Represent Theorem (Girosi 1998; Schölkopf et al. 2001) as well as the previous research of (Grammer et al. 2002), the output function for multi-class problems can be formulated as below,

$$f(x) = \left(\sum_{i=1}^{n} c_{r,i} k(x, x_i) \right), r = 1, ..., m, \tag{1}$$

where k is a kernel function that is positive definite. Then, the corresponding decision function is

$$\hat{y}(x^*) = \arg\max_{r=1}^{m} \left\{ \sum_{i=1}^{n} c_{r,i} k(x^*, x_i) \right\}, \tag{2}$$

where x^* is a new example. That implies that the predicted label for a new example x is the one that gets the largest score $\sum_{i=1}^{n} c_{r,i} k(x, x_i)$.

Based on the results above, we can cast the multi-class kernel-based learning into the following optimization problem,

$$\min_{C} \sum_{i=1}^{n} \sum_{r=1}^{m} l(y_r(x_i) f_r(x_i)) + \gamma \|C\|_2^2, \tag{3}$$

where l is a loss function and $f_r(x_i)$ denotes the rth element of $f(x_i)$. The matrix $C \in \mathrm{R}^{m \times n}$ is the parameters of the classifier, which is consisted of $(c_{r,\cdot})$, $r = 1, ..., m$ and $c_{r,\cdot} = (c_{r,1}, ..., c_{r,n}) \in \mathrm{R}^n$. γ is the regularization parameter. $\|\cdot\|_2$ denotes the l_2-norm of a matrix. And $y_r(x_i)$ is 1 if $r = y_i$ and -1 otherwise.

In the paper, we use the smooth quadratic hinge loss $l(z) = \max(0, 1 - z)^2$ so that Eq. (3) is convex. So, we can solve the optimization problem via letting the derivation about C of Eq. (3) to be zero to get the stable point, that is the optimal solution.

3 l_p-norm Multiple Kernel Learning with Diversity of Classes (LMKLDC)

3.1 Multi-class Multiple Kernel Learning with l_p-norm

Now, we can generalize the multi-class kernel-based learning to the multi-class multiple kernel learning. Unlike the general MKL algorithms which often learn the same weights of combining kernels for all classes, we take the diversity of classes into consideration. LMKLDC will learn different weights of combining kernels for each class. So, we can generalize the output function Eq. (1) to be

$$f(x) = \left(\sum_{i=1}^{n} c_{r,i} \sum_{k=1}^{q} \beta_{r,k} k_k(x, x_i) \right), r = 1, ..., m \ . \tag{4}$$

Here $\beta_{r,\cdot} = (\beta_{r,1}, ..., \beta_{r,q}) \in \mathrm{R}^q$ is the weights of q base kernels for the rth class. As usual, we add the simplex constrain for $\beta_{r,\cdot}$, $r = 1, ..., m$. That is,

$$\forall r, \beta_{r,\cdot} \in \varDelta_1 := \left\{ \beta_{r,\cdot} \middle| \forall k : \beta_{r,k} \geq 0, \sum_{k=1}^{q} \beta_{r,k} = 1 \right\} \ .$$

Correspondingly, the decision function becomes

$$\hat{y}(x^*) = \arg\max_{r=1}^{m} \left\{ \sum_{i=1}^{n} c_{r,i} \sum_{k=1}^{q} \beta_{r,k} k_k(x^*, x_i) \right\} \ . \tag{5}$$

Moreover, the regularization term in LMKLDC is in the form of the l_p-norm $(0 < p \leq 1)$ to promote the sparsity. As a result, the optimization problem of multi-class multiple kernel learning with l_p-norm is

$$\begin{aligned} &\min_{B,C} \sum_{i=1}^{n} \sum_{r=1}^{m} l(y_r(x_i) f_r(x_i)) + \alpha \left\| B \right\|_{p,2}^2 + \gamma \left\| C \right\|_2^2 \\ &s.t. \ \ \forall r : \beta_{r,\cdot} \cdot e = 1, \\ &\qquad \forall r : \forall k : \beta_{r,k} \geq 0, \end{aligned} \tag{6}$$

where $\mathrm{B} = [\beta_{r,k}]_{r=1,...,m;k=1,...,q} \in \mathrm{R}^{m \times q}$, $p \in (0, 1]$ and α and γ are regularization parameters. $e \in \mathrm{R}^q$ is a column vector whose elements are 1. $\left\| \cdot \right\|_{p,2}$ is a matrix norm defined as (Wang et al. 2013).

Considering diversity of classes, LMKLDC can learn different weights of combining kernels for each class to cater to the feature space of each class. Unfortunately, the objective function in Eq. (6) happens to be non-convex when $p \in (0, 1)$, which makes it difficult to solve Eq. (6). Hence, it is necessary to design an efficient algorithm to solve Eq. (6) with $p \in (0, 1]$.

3.2 Optimization Method

At first, the objective function of Eq.(6) can be rewritten as

$$\sum_{r=1}^{m} \min_{\beta_{r,\cdot}, c_{r,\cdot}} \sum_{i=1}^{n} l\left(y_r(x_i) f_r(x_i)\right) + \alpha \left\| \beta_{r,\cdot} \right\|_p^2 + \gamma \left\| c_{r,\cdot} \right\|_2^2, \tag{7}$$

so we can decompose the initial optimization problem Eq.(6) into m small optimization problems as follows,

$$\begin{aligned} &\min_{\beta_{r,\cdot}, c_{r,\cdot}} \sum_{i=1}^{n} l\left(y_r(x_i) f_r(x_i)\right) + \alpha \left\| \beta_{r,\cdot} \right\|_p^2 + \gamma \left\| c_{r,\cdot} \right\|_2^2 \\ &s.t. \ \ \beta_{r,\cdot} \cdot e = 1, \\ &\qquad \forall k : \beta_{r,k} \geq 0, \end{aligned} \tag{8}$$

with $r = 1, ..., m$. Substituting the smooth quadratic hinge loss into Eq. (8), we can get

$$\begin{aligned} \min_{\beta_{r,\cdot}, c_{r,\cdot}} & \sum_{i=1}^{n} [\max(0, 1 - y_r(x_i) f_r(x_i))]^2 + \alpha \|\beta_{r,\cdot}\|_p^2 + \gamma \|c_{r,\cdot}\|_2^2 \\ s.t. \ & \beta_{r,\cdot} \cdot e = 1, \\ & \forall k : \beta_{r,k} \geq 0 \ . \end{aligned} \tag{9}$$

So, the key to solving Eq. (6) is to solve the optimization problem Eq. (9). We apply a two-stage approach to solve the optimization problem Eq. (9). Firstly, initialize $\beta_{r,\cdot}$.

Since knowing $\beta_{r,\cdot}$, we can yield the value for $c_{r,\cdot}$ by solving Eq. (9). Once getting $c_{r,\cdot}$, $\beta_{r,\cdot}$ can be updated by solving Eq. (9). The process is repeated until $\beta_{r,\cdot}$ and $c_{r,\cdot}$ converge.

Optimize $c_{r,\cdot}$. After fixing the vector $\beta_{r,\cdot}$, Eq. (9) degenerates to the following unconstrained optimization problem,

$$\min_{c_{r,\cdot}} g(c_{r,\cdot}) := \frac{1}{\gamma_1^2} \sum_{i=1}^{n} [\max(0, 1 - y_r(x_i) f_r(x_i))]^2 + \|c_{r,\cdot}\|_2^2 \tag{10}$$

where $\gamma_1^2 = \gamma$.

It can be found that the objective function of Eq. (10) is a convex function. So, we can work out the stable point about $c_{r,\cdot}$, that is just right the optimal solution.

Firstly, the partial derivative about $c_{r,\cdot}$ of $g(c_{r,\cdot})$ can be calculated as

$$\begin{aligned} \frac{\partial g(c_{r,\cdot})}{\partial c_{r,\cdot}} = & \frac{1}{\gamma_1^2} \sum_{i=1}^{n} 2 y_r(x_i)^2 \cdot c_{r,\cdot} \cdot K(x_i) \cdot \beta_{r,\cdot}^T \cdot \beta_{r,\cdot} \cdot K(x_i)^T \\ & - \frac{1}{\gamma_1^2} \sum_{i=1}^{n} 2 y_r(x_i) \cdot \beta_{r,\cdot} \cdot K(x_i)^T + 2 c_{r,\cdot}. \end{aligned} \tag{11}$$

Here, $K(x) := \begin{bmatrix} k_1(x, x_1) & k_2(x, x_1) & ... & k_q(x, x_1) \\ k_1(x, x_2) & k_2(x, x_2) & ... & k_q(x, x_2) \\ .. & ... & ... & ... \\ k_1(x, x_n) & k_2(x, x_n) & ... & k_q(x, x_n) \end{bmatrix} \in \mathrm{R}^{n \times q}$.

Then, by letting Eq. (11) to be zero, we can get

$$c_{r,\cdot} = \beta_{r,\cdot} \sum_{i=1}^{n} y_r(x_i) K(x_i)^T \cdot \left(\sum_{i=1}^{n} K(x_i) \beta_{r,\cdot}^T \beta_{r,\cdot} K(x_i)^T + \gamma_1^2 I_n \right)^{-1}, \tag{12}$$

where I_n denotes unit matrix.

Optimize $\boldsymbol{\beta_{r,\cdot}}$. As above, fixing the vector $c_{r,\cdot}$ makes Eq. (9) become

$$\begin{aligned} &\min_{\beta_{r,\cdot}} h(\beta_{r,\cdot}) := \frac{1}{\gamma_2^2}\sum_{i=1}^{n}\left[\max\left(0, 1 - y_r(x_i)f_r(x_i)\right)\right]^2 + \left\|\beta_{r,\cdot}\right\|_p^2 \\ &s.t.\ \beta_{r,\cdot}\cdot e = 1, \\ &\quad\ \forall k : \beta_{r,k} \geq 0, \end{aligned} \tag{13}$$

where $\gamma_2^2 = \alpha$ and $p \in (0, 1]$.

No matter what the value of p takes, we always convert Eq. (13) into a convex optimization problem via CCCP (Smola et al. 2005). The concrete steps are: (1) get an initial value x_0 for $\beta_{r,\cdot}$, which satisfies the simplex constrain; (2) calculate the 1th order Taylor expansion $T_1\left\{\left\|\beta_{r,\cdot}\right\|_p^2, x_0\right\}(\beta_{r,\cdot})$ of $\left\|\beta_{r,\cdot}\right\|_p^2$ at location x_0 , that is

$$\begin{aligned} T_1\left\{\left\|\beta_{r,\cdot}\right\|_p^2, x_0\right\}(\beta_{r,\cdot}) &= \left\|x_0\right\|_p^2 + \left\langle \beta_{r,\cdot} - x_0, \frac{\partial\left\|\beta_{r,\cdot}\right\|_p^2}{\partial\beta_{r,\cdot}}\Big|_{\beta_{r,\cdot}=x_0}\right\rangle \\ &= \left\|x_0\right\|_p^2 + \left\langle \beta_{r,\cdot} - x_0, 2\left\|x_0\right\|_p^{2-p}\cdot\Psi_p\left(x_0\right)\right\rangle \quad . \end{aligned} \tag{14}$$

Here, $\Psi_p(\cdot)$ is defined as $\Psi_p(x) := \left(x_1^{p-1}, ..., x_q^{p-1}\right)$ with $x \in \mathrm{R}^q$. (3) approximate $\left\|\beta_{r,\cdot}\right\|_p^2$ using its 1th order Taylor expansion.

Now, we substitute Eq. (14) into Eq. (13) and obtain the following optimization function

$$\begin{aligned} &\min_{\beta_{r,\cdot}} h(\beta_{r,\cdot}) := \frac{1}{\gamma_2^2}\sum_{i=1}^{n}\left[\max\left(0, 1 - y_r(x_i)f_r(x_i)\right)\right]^2 + \left\|x_0\right\|_p^2 \\ &\qquad + \left\langle \beta_{r,\cdot} - x_0, 2\left\|x_0\right\|_p^{2-p}\cdot\Psi_p\left(x_0\right)\right\rangle \\ &s.t.\ \beta_{r,\cdot}\cdot e = 1, \\ &\quad\ \forall k : \beta_{r,k} \geq 0 \quad . \end{aligned} \tag{15}$$

We can find that Eq. (15) is a QP, which can be solved by the efficient solver. In the implementation, we use CVX with the solver SeDuMi for QP.

4 Experimental Results

In the section, to evaluate the performance of our proposed model, LMKLDC, we compare it with the SimpleMKL (Rakotomamonjy et al. 2008) and the unweighted MKL for multi-class classification, which corresponds to LMKLDC with the same weights of combining kernels for every class and for each base kernel. We perform some experiments on a toy dataset and UCI[1] datasets. The detailed experimental setting and analysis of results are presented as below.

[1] The dataset is available from '`http://www.ics.uci.edu/ mlearn/MLRepository.html`'

4.1 Toy Dataset

In the toy problem, we construct a toy dataset to test the effectiveness of LMKLDC. The toy dataset has three classes and the data of every class is respectively from a two-dimensional Gaussian distribution. Every class has 100 examples. The toy dataset is shown in Figure 1. *mu* and *sigma* respectively denote the average value and the covariance matrix of a Gaussian distribution. We randomly split the toy dataset into 70% training and 30% test set. And the process is repeated ten times. Like SimpleMKL, base kernels include the Gaussian kernels with the bandwidths $\{0.5, 1, 2, 5, 7, 10, 12, 15, 17, 20\}$ and the polynomial kernels with the degrees $\{1, 2, 3\}$. The average results of ten splits are reported.

We set the same parameters $\gamma_1 = 2^{-7}$, $\gamma_2 = 2^2$ and $p = 0.5$ for our algorithm and the unweighted MKL. If LMKLDC is effective, it should learn a different kernel combination for each class and has a better classification performance. The weights of combining kernels for each class of LMKLDC are presented in Figure 2. From the figure, we can find our algorithm does select a few different base kernels for each class. Class 2 and Class 3 select the similar base kernels, but the weights of these base kernels are different. Compared with Class 2 and Class 3, Class 1 selects entirely different base kernels. Besides, we compare our algorithm with the unweighted MKL about the classification accuracy. The result is reported in Table 1. We can find that LMKLDC gets a better classification accuracy. It implies that when the weights of combining kernels for all classes are restricted to be the same, the classification performance of MCMKL algorithms is decreased. Hence, considering diversity of classes in MCMKL algorithms is necessary.

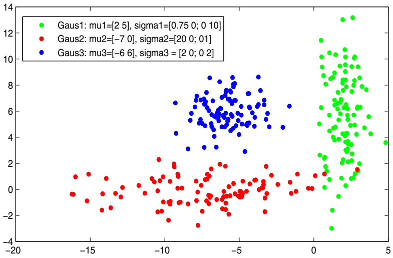

Fig. 1. Toy dataset

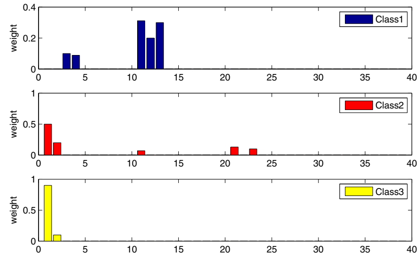

Fig. 2. The weights of combining kernels for each class of LMKLDC. The x-axis denotes the base kernels.

4.2 UCI Dataset

In the subsection, we perform a serial of experiments to evaluate the performance of the LMKLDC algorithm. Eight UCI datasets are used in our experiments. They include *Balance (3,4,625)*, *Iris (3,4,150)*, *Soybean (4,35,47)*, *Lenses*

Table 1. Accuracy of two algorithms on the toy dataset

Dataset	LMKLDC	Unweighted MKL
toy dataset	0.9833	0.9778

(3,4,24), Wine (3,13,178), Tae (3,5,151), Ecoli (6,6,332) and Cmc(3,9,1473), where the number of classes, the dimension and number of samples are listed in the bracket. Likewise, for every dataset, we randomly split it into 70% training and 30% test set. And the process is repeated ten times. The regularization parameters γ_1 and γ_2 are tuned via grid searching in the set $\left\{2^{-10}, 2^{-9}, .., 2^{9}, 2^{10}\right\}$ and p in the set $\{0.25, 0.5, 0.75, 1\}$. Similarly, we select the Gaussian kernels with the bandwidths $\{0.5, 1, 2, 5, 7, 10, 12, 15, 17, 20\}$ and the polynomial kernels with the degrees $\{1, 2, 3\}$ as base kernels. The average results of ten splits are reported in the experiments.

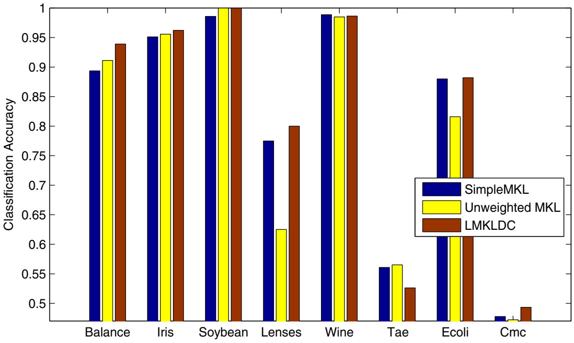

Fig. 3. Classification accuracy of algorithms on UCI datasets

For every dataset, we report the average classification accuracies of these compared algorithms in Figure 3. From the figure, it can be seen that our algorithm gets a better classification accuracy on most of datasets. Though the accuracy of SimpleMKL is higher than the accuracy of LMKLDC in Wine and Tae, LMKLDC selects fewer base kernels than SimpleMKL. And in Tae, the unweighted MKL, which is a special case of LMKLDC, gets the highest accuracy. In a sense, that also shows the superior classification performance of LMKLDC.

Moreover, we also perform an experiment to examine the sparsity of our model, which is measured by the number of kernels selected from base kernels. Considering our model learns different kernel combinations for each class, we use the average number of selected kernels of all classes as the final number of selected kernels for our model. The corresponding result is presented in Table 2. Kernels whose weight is greater than 0.0001 are selected. It is obvious that our model selects the less number of kernels from the base kernels than SimpleMKL.

It indicates the superiority of l_p-norm ($0 < p \leq 1$). Especially in Balance, Iris, Soybean, Wine and Cmc, the sparsity of LMKLDC makes it select far fewer base kernels than SimpleMKL. Hence, our algorithm LMKLDC can get a better classification accuracy and the sparsity simultaneously.

Table 2. The percentage of selected kernels from base kernels

Dataset	SimpleMKL	LMKLDC
Balance	7.69% (5/65)	2.56% (1.667/65)
Iris	7.69% (5/65)	1.54% (1/65)
Soybean	39.87% (181/454)	0.39% (1.75/454)
Lenses	4.62% (3/65)	4.10% (2.667/65)
Wine	10.43% (19/182)	1.10% (2/182)
Tae	6.41% (5/78)	5.13% (4/78)
Ecoli	10.99% (10/91)	4.76% (4.333/91)
Cmc	60.00% (78/130)	1.54% (2/130)

5 Conclusion

In this paper, we combine MKL with the multi-class classification and propose an effective algorithm LMKLDC. It considers the diversity of classes and utilizes the sparsity of l_p-norm ($0 < p \leq 1$) to promote the computational efficiency. Some experiments demonstrate its superiority.

In the paper, LMKLDC employs positive definite kernels as base kernels. How to expand LMKLDC with indefinite kernels as base kernels becomes an interesting issue for future work.

Acknowledgements. This work was supported in part by National Natural Science Foundations of China (Grant Nos. 61375057 and 61300165), Natural Scinence Foundation of Jiangsu Province of China (Grant No. BK20131298), and the Scientific Research Startup Project of New Doctorial Faculties of Southeast University.

References

Lanckriet, G.R.G., Cristianini, N., Bartlett, P., Ghaoui, L.E., Jordan, M.I.: Learning the kernel matrix with semidefinite programming. The Journal of Machine Learning Research 5, 27–72 (2004)

Bach, F.R., Lanckriet, G.R.G., Jordan, M.I.: Multiple kernel learning, conic duality, and the SMO algorithm. In: Proceedings of the Twenty-first International Conference on Machine Learning, vol. 6. ACM (2004)

Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: SimpleMKL. Journal of Machine Learning Research 9(11) (2008)

Kloft, M., Brefeld, U., Sonnenburg, S., Laskov, P., Müller, K.R., Zien, A.: Efficient and accurate l_p-norm multiple kernel learning. NIPS 22(22), 997–1005 (2009)

Xu, Z., Jin, R., Yang, H., King, I., Lyu, M.R.: Simple and efficient multiple kernel learning by group lasso. In: Proceedings of the 27th International Conference on Machine Learning, ICML, pp. 1175–1182 (2010)

Xu, X., Tsang, I.W., Xu, D.: Soft margin multiple kernel learning. IEEE Transactions on Neural Networks and Learning Systems 24, 749–761 (2013)

Zien, A., Ong, C.S.: Multiclass multiple kernel learning. In: Proceedings of the 24th International Conference on Machine Learning, pp. 1191–1198. ACM (2007)

Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and sturcutured output spaces. In: Proceedings of the 16th International Conference on Machine Learning (2004)

Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. The Journal of Machine Learning Research 2, 265–292 (2002)

Ye, J., Ji, S., Chen, J.: Multi-class discriminant kernel learning via convex programming. The Journal of Machine Learning Research 9, 719–758 (2008)

Kumar, A., Niculescu-Mizil, A., Kavukcuoglu, K., Daumé, H.: A binary classification framework for two-stage multiple kernel learning. arXiv preprint arXiv:1206.6428 (2012)

Cortes, C., Mohri, M., Rostamizadeh, A.: Multi-Class Classification with Maximum Margin Multiple Kernel. In: Proceedings of the 30th International Conference on Machine Learning, ICML, pp. 46–54 (2013)

Bach, F.R.: Consistency of the group lasso and multiple kernel learning. The Journal of Machine Learning Research 9, 1179–1225 (2008)

Szafranski, M., Grandvalet, Y., Rakotomamonjy, A.: Composite kernel learning. Machine Learning 79(1-2), 73–103 (2010)

Nath, J.S., Dinesh, G., Raman, S., Bhattacharyya, C., Ben-Tal, A., Ramakrishnan, K.R.: On the Algorithmics and Applications of a Mixed-norm based Kernel Learning Formulation. In: NIPS, pp. 844–852 (2009)

Chartrand, R.: Exact reconstruction of sparse signals via nonconvex minimization. IEEE Signal Processing Letters 14(10), 707–710 (2007)

Chartrand, R., Yin, W.: Iteratively reweighted algorithms for compressive sensing. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2008, pp. 3869–3872. IEEE (2008)

Xu, Z., Chang, X., Xu, F., Zhang, H.: $L_{1/2}$ regularization: A Thresholding Representation Theory and a Fast Solver. IEEE Transaction on Neural Networks and Learning Systems 23(7) (2012)

Rakotomamonjy, A., Flamary, R., Gasso, G., Ganu, S.: l_p-l_q penalty for sparse linear and sparse multiple kernel multitask learning. IEEE Transactions on Neural Networks 22(8), 1307–1320 (2011)

Smola, A., Vishwanathan, S.V.N., Hofmann, T.: Kernel methods for missing variables (2005)

Girosi, F.: An equivalence between sparse approximation and support vector machines. Neural Computation 10(6), 1455–1480 (1998)

Schölkopf, B., Herbrich, R., Smola, A.J.: A generalized representer theorem. In: Proceedings of the 14th Annual Conference on Computational Learning Theory, pp. 416–426 (2001)

Wang, L., Chen, S.: $l_{2,p}$-Matrix norm and its application in feature selection. arXiv preprint arXiv:1303.3987 (2013)