Simple and Efficient Multiple Kernel Learning by Group Lasso

Zenglin Xu ZLXU@MPI-INF.MPG.DE
Cluster of Excellence MMCI, Saarland University and MPI Informatics, Saarbruecken, Germany

Rong Jin RONGJIN@CSE.MSU.EDU
Department of Computer Science & Engineering, Michigan State University, East Lansing, MI 48824 USA

Haiqin Yang, Irwin King, Michael R. Lyu {HQYANG, KING, LYU}@CSE.CUHK.EDU.HK
Department of Computer Science & Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

Abstract

We consider the problem of how to improve the efficiency of Multiple Kernel Learning (MKL). In literature, MKL is often solved by an alternating approach: (1) the minimization of the kernel weights is solved by complicated techniques, such as Semi-infinite Linear Programming, Gradient Descent, or Level method; (2) the maximization of SVM dual variables can be solved by standard SVM solvers. However, the minimization step in these methods is usually dependent on its solving techniques or commercial softwares, which therefore limits the efficiency and applicability. In this paper, we formulate a closed-form solution for optimizing the kernel weights based on the equivalence between group-lasso and MKL. Although this equivalence is not our invention, our derived variant equivalence not only leads to an efficient algorithm for MKL, but also generalizes to the case for L_p-MKL ($p \geq 1$ and denoting the L_p-norm of kernel weights). Therefore, our proposed algorithm provides a unified solution for the entire family of L_p-MKL models. Experiments on multiple data sets show the promising performance of the proposed technique compared with other competitive methods.

1. Introduction

Multiple kernel learning (MKL) has been an attractive topic in machine learning (Lanckriet et al., 2004b; Ong et al., 2005). It has been regarded as a promising technique for identifying the combination of multiple data sources or feature subsets and been applied in a number of domains, such as genome fusion (Lanckriet et al., 2004a), splice site detection (Sonnenburg et al., 2006), image annotation (Harchaoui & Bach, 2007) and so on.

Multiple kernel learning searches for a combination of base kernel functions/matrices that maximizes a generalized performance measure. Typical measures studied for multiple kernel learning, include maximum margin classification errors (Lanckriet et al., 2004b; Bach et al., 2004; Argyriou et al., 2006; Zien & Ong, 2007), kernel-target alignment (Cristianini et al., 2001), Fisher discriminative analysis (Ye et al., 2007), etc.

There are two active research directions in multiple kernel learning. One is to improve the efficiency of MKL algorithms. Following the Semi-Definite Programming (SDP) algorithm proposed in the seminal work of (Lanckriet et al., 2004b), in (Bach et al., 2004), a block-norm regularization method based on Second Order Cone Programming (SOCP) was proposed in order to solve medium-scale problems. Due to the high computation cost of SDP and SOCP, these methods cannot process more kernels and more training data. Recent studies suggest that alternating approaches (Sonnenburg et al., 2006; Rakotomamonjy et al., 2008; Xu et al., 2009a) are more efficient. These approaches alternate between the optimization of kernel weights and the optimization of SVM classifiers. In each step, given the current solution of kernel weights, it solves a classical SVM with the combined kernel; then a specific procedure is used to update the kernel weights. The advantage of the alternating scheme is that SVM solvers can be very efficient due to recent advances in large scale optimization (Bottou & Lin, 2007). However, the current approaches for updating the kernel weights are

Appearing in *Proceedings of the 27^{th} International Conference on Machine Learning*, Haifa, Israel, 2010.

still time consuming and most of them depend on commercial softwares. For example, the well-known Shogun (`www.shogun-toolbox.org/`) toolbox of MKL employs CPLEX for solving the linear programming problem.

The second direction is to improve the accuracy of MKL by exploring the possible combination ways of base kernels. L_1-norm of the kernel weights, also known as the simplex constraint, is mostly used in MKL methods. The advantage of the simplex constraint is that it leads to a sparse solution, i.e., only a few base kernels among manys carry significant weights. However, as argued in (Kloft et al., 2008), the simplex constraint may discard complementary information when base kernels encodes orthogonal information, leading to suboptimal performance. To improve the accuracy in this scenario, an L_2-norm of the kernel weights, known as a ball constraint, is introduced in their work. A natural extension to the L_2-norm is the L_p-norm, which is approximated by the second order Taylor expansion and therefore leads to a convex optimization problem (Kloft et al., 2009). Another possible extension is to explore the grouping property or the mixed-norm combination, which is helpful when there are principle components among the base kernels (Saketha Nath et al., 2009; Szafranski et al., 2008). Other researchers also study the possibility of non-linear combination of kernels (Varma & Babu, 2009; Cortes et al., 2009b). Although the solution space has been enlarged, the non-linear combination usually results in non-convex optimization problem, leading to even higher computational cost. Moreover, the solution of nonlinear combination is difficult to interpret. It is important to note that the choice of combination usually depends on the composition of base kernels and it is not the case that a non-linear combination is superior than the traditional L_1-norm combination.

To this end, we first derive a variation of the equivalence between MKL and group lasso (Yuan & Lin, 2006). Based on the equivalence, we transform the related convex-concave optimization of MKL into a joint-minimization problem. We further obtain a closed-form solution for the kernel weights, which therefore greatly improves the computational efficiency. It should be noted that although the consistency between MKL and group lasso is discussed in (Rakotomamonjy et al., 2008; Bach, 2008), our obtained variational equivalence leads to a stable optimization algorithm. On the other hand, the proposed optimization algorithm could also be instructive to the optimization of group-lasso. We further show that L_p-norm formulation for MKL is also equivalent to an optimization function with a different group regularizer, which does not appear in literature. Compared to the approach in (Kloft et al., 2009), our approach directly solves the related optimization problem without a Taylor approximation. Experimental results on multiple data sets show the promising performance of the proposed optimization method.

The rest of this paper is organized as follows. Section 2 presents the related work on multiple kernel learning. Section 3 first presents the variational equivalence between MKL and group lasso, followed by the optimization method and its generalization to L_p-norm MKL. Section 4 shows the experimental results and Section 5 concludes this paper.

2. Related Work

Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{n \times d}$ denote the collection of n training samples that are in a d-dimensional space. Let $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in \{-1, +1\}^n$ denote the binary class labels for the data points in $\mathbf{X}$. Multiple kernel learning is often cast into the following optimization problem:

$$\min_{f \in \mathcal{H}_\gamma} \frac{1}{2}\|f\|^2_{\mathcal{H}_\gamma} + C \sum_{i=1}^{n} \ell(y_i f(\mathbf{x}_i)), \tag{1}$$

where $\mathcal{H}_\gamma$ is a reproducing kernel Hilbert space parameterized by γ and $\ell(\cdot)$ is a loss function. $\mathcal{H}_\gamma$ is endowed with kernel function $\kappa(\cdot, \cdot; \gamma) = \sum_{j=1}^{m} \gamma_j \kappa_j(\cdot, \cdot)$.

When the Hinge loss is employed, the dual problem of MKL (Lanckriet et al., 2004b) is equivalent to:

$$\min_{\gamma \in \Delta} \max_{\alpha \in \mathcal{Q}} \mathbf{1}^\top \alpha - \frac{1}{2}(\alpha \circ \mathbf{y})^\top \left(\sum_{j=1}^{m} \gamma_j \mathbf{K}_j \right) (\alpha \circ \mathbf{y}), \tag{2}$$

where Δ is the domain of γ and $\mathcal{Q}$ is the domain of α. $\mathbf{1}$ is a vector of all ones, $\{\mathbf{K}_j\}_{j=1}^m$ is a group of base kernel matrices associated with $\mathcal{H}'_j$, and $\circ$ defines the element-wise product between two vectors. The domain $\mathcal{Q}$ is usually defined as:

$$\mathcal{Q} \quad = \quad \{\alpha \in \mathbb{R}^n : \alpha^\top \mathbf{y} = 0,\ 0 \le \alpha \le C\}. \tag{3}$$

It is interesting to discuss the domain of γ. When $\gamma \in \Delta$ lies in a simplex, i.e., $\Delta = \{\gamma \in \mathbb{R}^m_+ : \sum_{j=1}^m \gamma_j = 1,\ \gamma_j \ge 0\}$, we call it L_1-norm of kernel weights. Most MKL methods fall in this category. Correspondingly, when $\Delta = \{\gamma \in \mathbb{R}^m_+ : \|\gamma\|_p \le 1,\ \gamma_j \ge 0\}$, we call it L_p-norm of kernel weights and the resulting model L_p-MKL. A special case is L_2-MKL (Kloft et al., 2008; Cortes et al., 2009a).

It is easy to verify that the overall optimization problem (2) is convex on γ and concave on α. Thus the above optimization problem can be regarded as a convex-concave problem.

Based on the convex-concave nature of the above problem, a number of algorithms have been proposed, which alternate between the optimization of kernel weights and the optimization of the SVM classifier. In each step, given the current solution of kernel weights, they solve a classical SVM with the combined kernel; then a specific procedure is used to update the kernel weights. For example, the Semi-Infinite Linear Programming (SILP) approach, developed in (Sonnenburg et al., 2006; Kloft et al., 2009), constructs a cutting plane model for the objective function and updates the kernel weights by solving a corresponding linear programming problem. Although the SILP approach can be employed for large scale MKL problems, it often suffers from slow convergence. In (Rakotomamonjy et al., 2008), the authors address the MKL problem by a simple Subgradient Descent (SD) method, which is also known as simpleMKL. However, since the SD method is memoryless, it does not utilize the gradients computed in previous iterations, which could be very useful in boosting the efficiency of the search. More recently, a level set method has been proposed in (Xu et al., 2009a;b) to combine cutting plane models and a regularization method based on a projection to level sets. Most alternating methods for MKL can be summarized in Algorithm 1.

Algorithm 1 Alternating approach for solving MKL

1: Initialize a valid solution $\gamma^0 \in \Delta$ and $t = 0$
2: **repeat**
3: Solve the dual of SVM with kernel $\mathbf{K} = \sum_{j=1}^m \gamma_j^t \mathbf{K}_j$ and obtain optimal solution α^t
4: Update kernel weights via cutting plane methods, subgradient descent or level set projection.
5: Update $t = t + 1$
6: **until** Convergence

Despite the success of the alternating algorithms in a number of applications, they are usually inefficient in updating kernel weights. More specifically, these methods require solving a sub-problem of linear programming that usually resorts to commercial softwares, such as CPLEX and MOSEK, which limits their applications. To alleviate the dependency on the commercial softwares and to further accelerate the optimization, in the following section, we will present an alternative procedure for updating kernel weights.

3. Efficient MKL Algorithms

Instead of casting the optimization ofs kernel weight into a complex linear programming problem, we derive a closed-form solution for updating kernel weights based on an alternative formulation of MKL using group lasso. It is important to note that although the equivalence between MKL and group lasso is not our invention (Bach, 2008), the variational formulation derived in this work is more effective in motivating an efficient optimization algorithm for MKL. Furthermore, the alternative formulation will lead to the equivalence between L_p-MKL ($p \geq 1$) and group regularizer.

3.1. Connection between MKL and Group Lasso

For convenience of presentation, we first start from the setting of L_1-norm for kernel weights. To show the connection between MKL and group lasso, we first transform (1) by Theorem 1.

Theorem 1. *The optimization problem of (1) is equivalent to the following optimization problem (4):*

$$\min_{\gamma\in\Delta} \min_{\{f_j\in\mathcal{H}_j\}_{j=1}^m} \quad \frac{1}{2}\sum_{j=1}^m \gamma_j \|f_j\|_{\mathcal{H}_j}^2 + \quad C\sum_{i=1}^n \ell\left(\sum_{j=1}^m y_i\gamma_j f_j(\mathbf{x}_i)\right). \quad (4)$$

Proof. It is important to note that problem in (4) is non-convex, and therefore we cannot deploy the standard approach to convert the problem in (4) into its dual form. To transform (4) into (1), first, we rewrite $C\ell(z) = \max_{\alpha\in[0,C]} \alpha(1-z)$. Then the optimization problem (4) can be transformed as follows

$$\min_{\gamma\in\Delta} \min_{\{f_j\in\mathcal{H}_j\}_{j=1}^m} \quad \max_{\alpha\in[0,C]^n} \frac{1}{2}\sum_{j=1}^m \gamma_j \|f_j\|_{\mathcal{H}_j}^2 + \quad \sum_{i=1}^n \alpha_i\left(1 - \sum_{j=1}^m y_i\gamma_j f_j(\mathbf{x}_i)\right).$$

Since the problem is convex in f_j and concave in α, we can then switch the minimization of f_j with the maximization of α, i.e.,

$$\min_{\gamma\in\Delta} \max_{\alpha\in[0,C]^n} \quad \min_{\{f_j\in\mathcal{H}_j\}_{j=1}^m} \sum_{i=1}^n \alpha_i + \quad \sum_{j=1}^m \gamma_j\left(\frac{1}{2}\|f_j\|_{\mathcal{H}_j}^2 - \sum_{i=1}^n \alpha_i y_i f_j(\mathbf{x}_i)\right).$$

By taking the minimization of f_j, we have

$$f_j(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i \kappa_j(\mathbf{x}_i, \mathbf{x}). \quad (5)$$

Then the resulting optimization problem becomes

$$\min_{\gamma\in\Delta} \max_{\alpha\in[0,C]^n} \mathbf{1}^\top\alpha - \frac{1}{2}\sum_{j=1}^m \gamma_j(\alpha\circ\mathbf{y})^\top\mathbf{K}_j(\alpha\circ\mathbf{y}),$$

which is exactly the same dual problem of (1) as shown in (2). □

To further show the connection between group lasso and MKL, we need to decouple the interaction between weights γ and classification functions $f_j, j = 1, \ldots, m$.

Based on the result of Theorem 1, we define

$$\tilde{f}_j = \gamma_j f_j. \tag{6}$$

We then have the problem in (4) rewritten as

$$\begin{aligned} \min_{\gamma \in \Delta} \min_{\{\tilde{f}_j \in \mathcal{H}_j\}_{j=1}^m} \quad & \frac{1}{2} \sum_{j=1}^m \frac{1}{\gamma_j} \|\tilde{f}_j\|_{\mathcal{H}_j}^2 \\ + \quad & C \sum_{i=1}^n \ell \left(y_i \sum_{j=1}^m \tilde{f}_j(\mathbf{x}_i) \right). \end{aligned} \tag{7}$$

By taking the minimization over γ, we obtain:

$$\gamma_j = \frac{\|\tilde{f}_j\|_{\mathcal{H}_j}}{\sum_{j=1}^m \|\tilde{f}_j\|_{\mathcal{H}_j}}, j = 1, \ldots, m. \tag{8}$$

Based on the equations (5) and (6), $\|\tilde{f}_j\|_{\mathcal{H}_j}$ can be calculated as:

$$\|\tilde{f}_j\|_{\mathcal{H}_j}^2 = \gamma_j^2 (\alpha \circ \mathbf{y})^\top \mathbf{K}_j (\alpha \circ \mathbf{y}). \tag{9}$$

The following theorem below allows us to rewrite (4) in another form that is clearly related to group lasso.

Theorem 2. *The optimization problem in (4) is equivalent to the following optimization problem*

$$\min_{\{f_j\}_{j=1}^m} \frac{1}{2} \left(\sum_{j=1}^m \|f_j\|_{\mathcal{H}_j} \right)^2 + C \sum_{i=1}^n \ell \left(y_i \sum_{j=1}^m f_j(\mathbf{x}_i) \right),$$

where $f_j \in \mathcal{H}_j$ for $j = 1, \ldots, m$.

The above theorem can be easily obtained by substituting the equation (8) into the optimization problem (7).

Remark 1. The regularizer in the above can be regarded as a group-lasso regularizer for *f* (Bach, 2008). Theorem 2 states that the classical formulation of MKL essentially uses a group-lasso type regularizer, which is not obvious from the optimization problem in (1).

Remark 2. The formulation in (7) provides an alternative approach for solving MKL. Unlike the typical formulation that cast MKL into a convex-concave optimization problem, (7) cast MKL into a simple minimization problem. As a result, we could solve (7) by alternating optimization: optimize γ with fixed $\tilde{f}_j, j = 1, \ldots, m$ and optimize $\tilde{f}_j, j = 1, \ldots, m$ with fixed γ. It is important to note that the calculation of $\|f_j\|_{\mathcal{H}_j}$ has the same computational complexity as the subgradient calculation as used in the simpleMKL (Rakotomamonjy et al., 2008) and Shogun (Sonnenburg et al., 2006) toolboxes. Therefore, due to the closed-form solution to γ_j, the overall optimization could be very efficient compared with the complicated semi-infinite programming and subgradient descent in these toolboxes.

We summarize the alternating optimization discussed in the above into Algorithm 2. It is important to note that the optimization problem (7) is joint-convex in γ and $\tilde{f}_j$, which continuously decreases the objective. Therefore Algorithm 2 ensures that the final solution converges to the global one. Since the dual problem is convex, the optimality of the solution can be guaranteed.

Remark 3. Joint minimization techniques are also used in feature scaling (Grandvalet & Canu, 2002). As discussed in (Rakotomamonjy et al., 2008), the minimization over the scaling vector (similar to the kernel weights in our context) may suffer from instability especially when some elements approach 0. We however argue that this is not the case for Algorithm 2. This is because although $1/\gamma_j$ appears in (7), which could lead to singularity when γ_j approaches 0, γ_j is updated by (8), which does not have any source of singularity. This is further confirmed by our empirical study.

Algorithm 2 The Group-Lasso Minimization for Multiple Kernel Learning

1: Initialize $\gamma^0 = \mathbf{1}/m$
2: **repeat**
3: Solve the dual problem of SVM with $\mathbf{K} = \sum_{j=1}^m \gamma_j \mathbf{K}_j$ to obtain the optimal solution α
4: Calculate $\|f_j\|_{\mathcal{H}_j}$ and γ_j according to (9) and (8), respectively
5: **until** Convergence

3.2. Generalization to L_p-MKL

Given the equivalence between group-lasso and L_1-MKL, we then generalize the formulation of MKL by replacing $\sum_{j=1}^m \gamma_j \leq 1$ with $\sum_{j=1}^m \gamma_j^p \leq 1$ with $p > 0$.

Theorem 3. *The MKL problem with the general constraint $\|\gamma\|_p \leq 1$, where $p \geq 1$, i.e.,*

$$\min_{\|\gamma\|_p \leq 1} \min_{f \in \mathcal{H}_\gamma} \frac{1}{2} \|f\|_{\mathcal{H}_\gamma}^2 + C \sum_{i=1}^n \ell(y_i f(\mathbf{x}_i)),$$

is equivalent to the following optimization problem

$$\min_{\{f_j \in \mathcal{H}_j\}_{j=1}^m} \frac{1}{2} \left(\sum_{j=1}^m \|f_j\|_{\mathcal{H}_j}^q \right)^{\frac{2}{q}} + C \sum_{i=1}^n \ell \left(\sum_{j=1}^m y_i f_j(\mathbf{x}_i) \right),$$

with $q = \frac{2p}{1+p}$.

Proof. Similar to the proof of Theorem 1, we can rewrite the generalized MKL, which is originally in the form of

$$\min_{\|\gamma\|_p \le 1} \min_{f \in \mathcal{H}_\gamma} \frac{1}{2}\|f\|^2_{\mathcal{H}_\gamma} + C\sum_{i=1}^{n} \ell(y_i f(\mathbf{x}_i)),$$

into the following form

$$\min_{\|\gamma\|_p \le 1} \min_{\{f_j \in \mathcal{H}_j\}_{j=1}^m} \frac{1}{2}\sum_{j=1}^{m} \frac{1}{\gamma_j}\|f_j\|^2_{\mathcal{H}_j} + C\sum_{i=1}^{n} \ell\left(\sum_{j=1}^{m} y_i f_j(\mathbf{x}_i)\right) \quad (10)$$

By taking the minimization over γ, we have

$$\gamma_j = \frac{\|f_j\|_{\mathcal{H}_j}^{\frac{2}{1+p}}}{\left(\sum_{k=1}^{m} \|f_k\|_{\mathcal{H}_{\kappa_k}}^{\frac{2p}{1+p}}\right)^{\frac{1}{p}}}, \quad (11)$$

and

$$\begin{aligned}\sum_{j=1}^{m} \frac{\|f_j\|^2_{\mathcal{H}_j}}{\gamma_j} &= \left(\sum_{k=1}^{m} \|f_k\|_{\mathcal{H}_{\kappa_k}}^{\frac{2p}{1+p}}\right)^{\frac{1}{p}} \sum_{j=1}^{m} \|f_j\|_{\mathcal{H}_j}^{\frac{2p}{1+p}} \\ &= \left(\sum_{k=1}^{m} \|f_k\|_{\mathcal{H}_{\kappa_k}}^{\frac{2p}{1+p}}\right)^{\frac{p+1}{p}}.\end{aligned}$$

Define $q = \frac{2p}{1+p}$ with the constraint $q \le 2$, we have the regularizer $\left(\sum_{k=1}^{m} \|f_k\|^q_{\mathcal{H}_{\kappa_k}}\right)^{\frac{q}{2}}$, which leads to the result in the theorem. □

It is important to note when mapping the original MKL formulation to an L_q regularizer, q is in the range between 1 and 2. However, using the regularizer framework, q is allowed to have any value larger or equal to 1. The L_p-MKL can also be similarly solved by Algorithm 2.

4. Experiments

In this section, we conduct experiments with two objectives: (1) to evaluate the efficiency of the proposed MKL algorithm, and (2) to evaluate the effectiveness of the L_p-MKL when varying the value of p. It is important to note that although the algorithm scalability is also our concern, we will not focus on this perspective in the current version of this paper.

We denote by MKLGL our proposed MKL algorithm based on group lasso. We compare MKLGL with several state-of-the-art efficient MKL algorithms including: Semi-infinite Programming (SIP) (Kloft et al., 2009; Sonnenburg et al., 2006), the subgradient descent approach used in the SimpleMKL toolbox (Rakotomamonjy et al., 2008), and the level method (Xu et al., 2009a). It is important to note that the SIP involves sub-problems of linear programming (for L_1-MKL) or sub-problems of second order cone programming (for L_p-MKL and $p \ge 2$) and the level method involves sub-problems of quadratic programming. For all the above convex programming problems, we adopt two implementations: the CVX package (`www.stanford.edu/~boyd/cvx`) and the MOSEK package (`http://www.mosek.com`).

4.1. Comparison with SimpleMKL

We randomly select four UCI datasets, i.e., "iono", "breast", "sonar", and "pima", to evaluate the efficiency of the proposed MKL algorithm. We adopt the following settings in (Rakotomamonjy et al., 2008; Xu et al., 2009a) to construct the base kernel matrices, i.e.,

- Gaussian kernels with 10 different widths ($\{2^{-3}, 2^{-2}, \ldots, 2^6\}$) on all features and on each single feature
- Polynomial kernels of degree 1 to 3 on all features and on each single feature.

Each base kernel matrix is normalized to unit trace. We repeat all the algorithms 20 times for each data set. In each run, 80% of the examples are randomly selected as the training data and the remaining data are used for testing. The training data are normalized to have zero mean and unit variance, and the test data are then normalized using the mean and variance of the training data. The regularization parameter C in SVM is chosen by cross-validation. The duality gap defined in (Rakotomamonjy et al., 2008) is used as the stopping criterion for all algorithms. Other settings are the same as the simpleMKL toolbox.

We report the results of simpleMKL and MKLGL in Table 1. It can be observed that MKLGL greatly improves the computational time of simpleMKL. It is important to note that we omit the results of SIP and Level methods since both of them significantly slower than MKLGL no matter what optimization package, either MOSEK or CVX, is employed. Generally, the MOSEK implementation is more efficient than the CVX implementation for both algorithms. Take the "Iono" dataset as an example, under the MOSEK implementation, the SIP method spends 12 times of training time of MKLGL; and under the CVX implementation, the number is 20. For the level method, its computational cost is 2 times of MKLGL's under the MOSEK implementation; and more

Table 1. The performance comparison of simpleMKL and MKLGL algorithms. Here n and m denote the size of training samples and the number of kernels, respectively.

	Acc. (%)	Time(s)	#SVM
	Iono	$n = 280$	$m = 442$
SimpleMKL	91.5±3.0	79.9±13.8	826.8±220.9
MKLGL	92.0±2.9	12.0±2.2	72.1±19.4
	Breast	$n = 546$	$m = 117$
SimpleMKL	96.5±1.2	110.5±18.6	542.0±131.1
MKLGL	96.6±1.2	14.1±1.9	40.0±8.2
	Sonar	$n = 166$	$m = 793$
SimpleMKL	82.0±6.5	57.0±4.6	687.6±135.4
MKLGL	82.0±6.8	5.7±1.1	53.6±16.1
	Pima	$n = 614$	$m = 117$
SimpleMKL	73.4±2.3	94.5±18.1	294.3±67.4
MKLGL	73.5±2.5	15.1±1.9	15.1±1.9

Table 2. The performance comparison of L_p-MKL algorithm on the simulation data.

Norm	Algo.	Accuracy (%)	Time(s)
	sumKer	80.4±2.8	1.9±0.1
p=1	SimpleMKL	83.4±2.0	9.2±3.6
p=2	MKLGL	89.7±1.2	3.1±0.5
	MKLSIP	89.7±1.3	42.2±4.8
p=3	MKLGL	91.1±1.6	2.1±0.2
	MKLSIP	91.1±1.5	27.3±3.7
p=4	MKLGL	91.4±1.8	1.8±0.2
	MKLSIP	91.4±1.8	28.4±2.1
p=10	MKLGL	92.2±1.6	1.4 ±0.1
	MKLSIP	92.2±1.4	17.2±1.6
p=100	MKLGL	92.6±1.8	1.1±0.1
	MKLSIP	92.6±1.9	13.1±0.8
p=1000	MKLGL	92.8±2.0	0.8±0.0
	MKLSIP	92.8±2.0	6.8±0.1

than 5 times under the CVX implementation.

To better understand the properties of the algorithms, we plot the evolution of the objective value and the kernel weights into Figure 1 and Figure 4.1, respectively. First, from Figure 1, we observe that the MKLGL algorithm converges overwhelmingly faster than simpleMKL. This is because that MKLGL uses a closed-form to update the kernel weights, which has the same cost as the gradient computation in simpleMKL. However, simpleMKL additionally requires a large number of function evaluations in order to compute the optimal stepsize via a line search. Note that every function evaluation in the line search of simpleMKL requires solving an SVM problem. This phenomenon can also be observed from the 4-th column of Table 1. Second, to understand the evolution of kernel weights (i.e., γ), we plot the evolution curves of the five largest kernel weights in Figure 4.1. We observe that the values of γ computed by simpleMKL fluctuates significantly through iterations. In contrast, for the proposed MKLGL algorithm, the values of γ change smoothly through iterations. The instability phenomenon discussed in (Rakotomamonjy et al., 2008) does not appear in our case.

4.2. Experiment on L_p-MKL

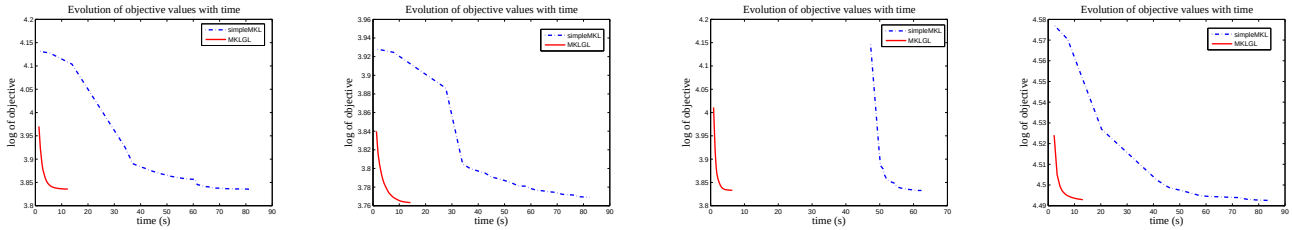
In this section, we examine the scenarios where L_p-MKL is more effective than L_1-MKL. Though L_p-MKL usually discards the sparse property when $p \geq 2$, it helps keep the complementary information among base kernels.

To see this, we design a synthesized dataset with 400 data points and 34 features satisfying that $\mathbf{y} = sign(\bar{\mathbf{X}})$ where $\bar{\mathbf{X}}$ is the mean of randomly generated data $\mathbf{X}$. We employ 50% of data for training and the rest for testing. For kernel composition, each kernel matrix is defined as a linear kernel on a single feature. Therefore, the kernels are complementary with each other. The other settings are similar to the previous experiments.

To see how the value of p affects the classification accuracy, we vary the value of p from 2 to 1000. We show the accuracy and computational time in Table 2. We also show the results of L_p-MKL with the SIP implementation, noted as MKLSIP, and the result of L_1-MKL with the simpleMKL implementation. As a baseline, the result of SVM using the summation of the base kernels, denoted by sumKer, is also reported.

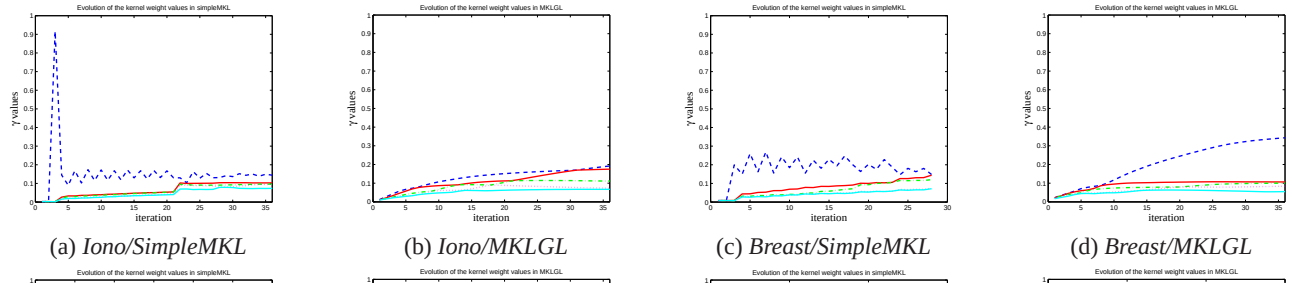
It can be observed that a larger p leads to a better accuracy since the kernels are all complementary to each other. On the other hand, although MKLSIP achieves almost the same accuracy, it takes significantly more computational time than MKLGL. This experiment suggests that our proposed MKLGL algorithm with L_p-norm can be both effective and efficient for the scenario that the base kernels are complementary to each other.

We further evaluate the proposed algorithm for L_p-MKL on three real-world datasets, i.e., "Bci", "Usps", and "Coil", from the benchmark repository (`www.kyb.tuebingen.mpg.de/ssl-book/`). Here the datasets of "Usps" and "Coil" are both with 1500 instances and 241 features, and "Bci" has 400 instances and 117 features. For kernel formation, we only adopt the base kernels which are calculated on all features. For each dataset, 50% of the data are used for training and the remaining are used for testing. Each kernel matrix is normalized with its trace being 1. Other settings are the same as in previous experiments. We report the accuracy of L_p-MKL for different settings of p in Table 3. For comparison, we also show the result of the summation of all base kernels. Note that due to



(a) *Iono* (b) *Breast* (c) *sonar* (d) *Pima*

Figure 1. Evolution of objective values over time (seconds) for datasets "Iono", "Breast", "Sonar" , and "Pima". The objective values are plotted on a logarithm scale (base 10) for better comparison.



(a) *Iono/SimpleMKL* (b) *Iono/MKLGL* (c) *Breast/SimpleMKL* (d) *Breast/MKLGL*

(e) *sonar/SimpleMKL* (f) *sonar/MKLGL* (g) *Breast/SimpleMKL* (h) *Breast/MKLGL*

Figure 2. The evolution curves of the five largest kernel weights for datasets "Iono", "Breast", "Sonar" and "Pima" computed by the two MKL algorithms

Table 3. The accuracy of the MKLGL algorithm when varying the value of p for real-world datasets.

Algo.	Bci	Usps	Coil
SumKer	52.9±3.8	80.5±1.1	78.1±2.1
SimpleMKL	55.1±5.8	87.2±1.3	84.8±1.5
MKLGL			
p=2	66.4±3.6	90.2±0.9	88.9±1.2
p=3	68.0±3.0	91.8±0.7	90.8±1.3
p=4	68.6±3.0	92.3±0.7	91.6±1.3
p=10	69.4±3.2	93.3±0.6	93.2±1.1
p=100	70.0±3.4	93.7±0.6	94.2±1.2
p=1000	70.1±3.4	93.7±0.6	94.3±1.2

space limit, we do not include the result of MKLSIP, as it obtains almost the same result as MKLGL but takes more time. From Table 3, we observe that L_p-MKL ($p \geq 2$) improves the classification accuracy when compared to L_1-MKL. Similar to the toy experiment, we find that the larger p corresponds the better classification accuracy.

In summary, our proposed MKLGL is an efficient algorithm for solving L_p-MKL that usually leads to better performance than L_1-MKL.

5. Conclusion

In this paper, we have presented an efficient algorithm for multiple kernel learning by discussing the connection between MKL and group-lasso regularizer. We calculate the kernel weights by a closed-form formulation, which therefore leverages the dependency of previous algorithms on employing complicated or commercial optimization softwares. Moreover, in order to improve the accuracy of traditional MKL methods, we naturally generalize MKL to L_p-MKL that constrains the p-norm ($p \geq 1$) the kernel weights. We show that our algorithm can be applied to the whole family of L_p-MKL models for all $p \geq 1$ without extra Taylor approximation. Experimental results on a number of benchmark datasets indicate the promising results of the proposed algorithm.

For the future work, we plan to employ our proposed algorithm to solve the real-world applications

(`http://mkl.ucsd.edu`). In addition, it is meaningful to find approaches to automatically determine the value of p in L_p-MKL. It is also desirable to derive MKL models which have sparse solutions and also keep the complementary or other structural information about data.

Acknowledgement

We thank the anonymous reviewers for the valuable comments. The work was supported by the National Science Foundation (IIS-0643494), National Institute of Health (1R01GM079688-01) and Research Grants Council of Hong Kong (CUHK4158/08E and CUHK4128/08E).

References

Argyriou, Andreas, Hauser, Raphael, Micchelli, Charles A., and Pontil, Massimiliano. A dc-programming algorithm for kernel selection. In *ICML '06: Proc. of the 23rd international conference on Machine learning*, pp. 41–48, 2006.

Bach, Francis R. Consistency of the group Lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9: 1179–1225, 2008.

Bach, Francis R., Lanckriet, Gert R. G., and Jordan, Michael I. Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML '04: Proceedings of the 21th international conference on Machine learning*, pp. 41–48, 2004.

Bottou, Léon and Lin, Chih-Jen. Support vector machine solvers. In Léon Bottou, Olivier Chapelle, Dennis DeCoste and Weston, Jason (eds.), *Large Scale Kernel Machines*, pp. 1–28. 2007.

Cortes, Corinna, Mohri, Mehryar, and Rostamizadeh, Afshin. l_2 regularization for learning kernels. In *UAI '09: Proc. of the 25th Conference on Uncertainty in Artificial Intelligence*, pp. 187–196, 2009a.

Cortes, Corinna, Mohri, Mehryar, and Rostamizadeh, Afshin. Learning non-linear combinations of kernels. In *Advances in Neural Information Processing Systems 22*, pp. 396–404. 2009b.

Cristianini, Nello, Shawe-Taylor, John, Elisseeff, André, and Kandola, Jaz S. On kernel-target alignment. In *Neural Information Processing Systems (NIPS 13)*, pp. 367–373, 2001.

Grandvalet, Yves and Canu, Stéphane. Adaptive scaling for feature selection in SVMs. In *NIPS*, pp. 553–560, 2002.

Harchaoui, Zaid and Bach, Francis. Image classification with segmentation graph kernels. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2007.

Kloft, Marius, Brefeld, Ulf, Laskov, Pavel, and Sonnenburg, Soren. Non-sparse multiple kernel learningg. In *NIPS workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, 2008.

Kloft, Marius, Brefeld, Ulf, Sonnenburg, Söoren, Laskov, Pavel, Müller, Klaus-Robert, and Zien, Alexander. Efficient and accurate lp-norm multiple kernel learning. In *Advances in Neural Information Processing Systems 22 (NIPS)*, pp. 997–1005. 2009.

Lanckriet, Gert R. G., Bie, Tijl De, Cristianini, Nello, Jordan, Michael I., and Noble, William Stafford. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004a.

Lanckriet, Gert R. G., Cristianini, Nello, Bartlett, Peter, Ghaoui, Laurent El, and Jordan, Michael I. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004b.

Ong, Cheng Soon, Smola, Alexander J., and Williamson, Robert C. Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, 6:1043–1071, 2005.

Rakotomamonjy, Alain, Bach, Francis R., Canu, Stéphane, and Grandvalet, Yves. SimpleMKL. *Journal of Machine Learning Research*, 9:1179–1225, 2008.

Saketha Nath, J., Dinesh, G., Raman, S., Bhattacharyya, C., Ben-Tal, A., and Ramakrishnan, K.R. On the algorithmics and applications of a mixed-norm based kernel learning formulation. In *Advances in Neural Information Processing Systems 22*, pp. 844–852. 2009.

Sonnenburg, Sören, Rätsch, Gunnar, Schäfer, Christin, and Schölkopf, Bernhard. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

Szafranski, Marie, Grandvalet, Yves, and Rakotomamonjy, Alain. Composite kernel learning. In *ICML '08: Proc. of the 25th international conference on Machine learning*, pp. 1040–1047, 2008.

Varma, Manik and Babu, Bodla Rakesh. More generality in efficient multiple kernel learning. In *ICML '09: Proc. of the 26th Annual International Conference on Machine Learning*, pp. 1065–1072, 2009.

Xu, Zenglin, Jin, Rong, King, Irwin, and Lyu, Michael. An extended level method for efficient multiple kernel learning. In *Advances in Neural Information Processing Systems 21 (NIPS)*, pp. 1825–1832. 2009a.

Xu, Zenglin, Jin, Rong, Lyu, Michael R., and King, Irwin. Discriminative semi-supervised feature selection via manifold regularization. In *IJCAI '09: Proceedings of the 21th International Joint Conference on Artificial Intelligence*, pp. 1303–1308, 2009b.

Ye, Jieping, Chen, Jianhui, and Ji, Shuiwang. Discriminant kernel and regularization parameter learning via semidefinite programming. In *ICML '07: Proc. of the 24th international conference on Machine learning*, pp. 1095–1102, 2007.

Yuan, M. and Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–67, 2006.

Zien, Alexander and Ong, Cheng Soon. Multiclass multiple kernel learning. In *ICML '07: Proc. of the 24th international conference on Machine learning*, pp. 1191–1198, 2007.