AdaMKL: A Novel Biconvex Multiple Kernel Learning Approach

Ziming Zhang, Ze-Nian Li, Mark S. Drew
School of Computing Science, Simon Fraser University, Vancouver, B.C., Canada
zza27, li, mark}@cs.sfu.ca

Abstract

In this paper, we propose a novel large-margin based approach for multiple kernel learning (MKL) using biconvex optimization, called Adaptive Multiple Kernel Learning (AdaMKL). To learn the weights for support vectors and the kernel coefficients, AdaMKL minimizes the objective function alternately by learning one component while fixing the other at a time, and in this way only one convex formulation needs to be solved. We also propose a family of biconvex objective functions with an arbitrary $\ell_p$-norm ($p \geq 1$) of kernel coefficients. As our experiments show, AdaMKL performs comparably with state-of-the-art convex optimization based MKL approaches, but its learning is much simpler and faster.

1 Introduction

Multiple Kernel Learning (MKL) [1] aims to generate an optimal kernel automatically by combining a set of basic kernels linearly as well as learn the weights for support vectors simultaneously. Given $N$ labeled data $\mathcal{D} = \{(x_i, y_i)\}_{i=1, \ldots, N}$ where $\forall i, x_i \in \mathcal{X}$ is an input feature vector and $y_i \in \{+1, -1\}$ is its class label for binary classification, and $M$ feature mapping functions $\Phi = \{\phi_j\}_{j=1, \ldots, M}$, each of which maps an input feature into a Hilbert space $\mathcal{H}_j$, a reproducing kernel $K_j$ is defined as an inner product of $\phi_j(x_m)$ and $\phi_j(x_n)$, that is, $K_j(x_m, x_n) = \langle \phi_j(x_m), \phi_j(x_n) \rangle_{\mathcal{H}_j}$, and the optimal kernel $K_{opt}$ learned by MKL can be represented as Eqn. 1, where $\gamma$ are kernel coefficients.

$$K_{opt} = \sum_{j=1}^{M} \gamma_j K_j, \quad \forall j, \gamma_j \geq 0 \quad (1)$$

By putting MKL into the context of large-margin classifiers, several MKL approaches have been proposed [1, 4, 5, 7, 2, 3, 6]. Typically the objective functions in the existing approaches are convex, with a norm constraint on $\gamma$, e.g. $\|\gamma\|_1=1$ (sparse learning) [1, 4, 5, 7, 6], or $\|\gamma\|_2=1$ (non-sparse learning) [2], or even $\|\gamma\|_p=1$ ($p \geq 1$) [3]. Also, different optimization approaches have been used in MKL, e.g. second-order cone programming [1], semi-definite programming [4], semi-infinite programming [7, 2], and gradient-based approaches [5, 3, 6].

Commonly, learning kernel coefficients efficiently is one of the major difficulties in MKL due to the existence of the norm constraint and non-negative constraint on kernel coefficients. In this paper, we propose a biconvex optimization based MKL approach, denoted Adaptive Multiple Kernel Learning (AdaMKL). A function $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is called biconvex if $f$ is convex both in $x \in \mathcal{X}$ for fixed $y \in \mathcal{Y}$ and in $y \in \mathcal{Y}$ for fixed $x \in \mathcal{X}$. The main contribution of this paper is that we propose a family of biconvex, rather than convex, objective functions to learn the parameters in MKL by introducing an arbitrary $\ell_p$-norm ($p \geq 1$) constraint into the objective function so that the norm constraint and non-negative constraint are hidden in the dual of AdaMKL when learning the weights of support vectors without the need for explicit consideration. It turns out that the traditional sparse (or non-sparse) learning due to the norm constraint is separable, which makes the learning of AdaMKL much simpler and faster compared to the convex optimization based MKL approaches.

The rest of this paper is organized as follows. Section 2 describes our approach in detail. Section 3 shows our experimental results. We conclude the paper in Section 4.

2 Adaptive Multiple Kernel Learning

In this paper, we focus on binary classification using AdaMKL. We follow the notations in Section 1.

2.1 Motivation

Eqn. 2 shows one type of common decision function in MKL, where $\theta$ are the coefficients w.r.t. kernels, w

$$f(x) = \langle \theta, \Phi(x) \rangle + b.$$
are the normal vectors for SVM, and $b$ is the bias term:

$$f(x) = \sum_{m=1}^{M} \theta_m (w_m, \phi_m(x)) + b$$ (2)

Letting $\theta_m \geq 0$, $w_m = \sqrt{\theta_m} w_m$ and fitting Eqn. 2 into the original SVM formulation, Kloft et al. [3] showed that the corresponding MKL formulation is actually a convex optimization problem (OP). However, if directly fitting Eqn. 2 into an SVM without any constraint on $\theta$, we can obtain Eqn. 3, where $C$ is a constant and $\xi$ are errors:

$$\min_{\theta, w, b, \xi} \frac{1}{2} \sum_m \theta_m^2 \parallel w_m \parallel_2^2 + C \sum_i \xi_i$$ (3)

s.t. \[
\forall i, \ y_i \left[ \sum_m \theta_m (w_m, \phi_m(x_i)) + b \right] \geq 1 - \xi_i \\
\xi_i \geq 0, \ C \geq 0
\]

Clearly, Eqn. 3 defines a biconvex OP, and its objective function can be easily relaxed to one of its upper bounds as shown in Eqn. 4, which also defines a biconvex OP and allows the introduction of an arbitrary $\ell_p$-norm of kernel coefficients into AdaMKL:

$$\sum_m \theta_m^2 \parallel w_m \parallel_2^2 \leq \parallel \theta^2 \parallel_p \parallel w \parallel_2^2$$ (4)

$$\leq \parallel \theta^2 \parallel_1 \parallel w \parallel_2 \parallel w \parallel_2$$

2.2 Formulation and Optimization

We define the primal of our AdaMKL as Eqn. 5, which contains a family of biconvex objective functions, and learn $w$ and $\theta$ alternately.

$$\min_{\theta, w, b, \xi} \frac{1}{2} \mathcal{N}_p(\theta) \parallel w \parallel_2^2 + C \sum_i \xi_i$$ (5)

s.t. \[
\forall i, \ y_i \left[ \sum_m \theta_m (w_m, \phi_m(x_i)) + b \right] \geq 1 - \xi_i \\
\xi_i \geq 0, \ C \geq 0
\]

where $\mathcal{N}_0(\theta) = \parallel \theta \parallel_1^2$, $\mathcal{N}_p(\theta) = \parallel \theta^2 \parallel_{p\geq1}$

2.2.1 Learning Parameter $w$

When learning parameter $w$, parameter $\theta$ is fixed. Using Lagrange multipliers, we rewrite our primal in Eqn. 5 and obtain its dual in Eqn. 6, where $\alpha$ is the set of Lagrange multipliers. Accordingly, we can calculate each $w_m$ using Eqn. 7 and rewrite Eqn. 1 as Eqn. 8.

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j,m} \alpha_i \alpha_j y_i y_j \theta_m^2 \mathcal{N}_p(\theta) K_m(x_i, x_j)$$

s.t. \[
\forall i, \ 0 \leq \alpha_i \leq C, \ \sum_i \alpha_i y_i = 0
\]

$$w_m = \sum_i \alpha_i y_i \theta_m \mathcal{N}_p(\theta) \phi_m(x_i)$$ (7)

$$K_{opt} = \sum_m \gamma_m K_m = \sum_m \theta_m^2 \mathcal{N}_p(\theta) K_m$$ (8)

Notice that if $\mathcal{N}_p(\theta) = \parallel \theta^2 \parallel_{p\geq1}$ and we denote $\gamma_m = \frac{\theta_m^2}{\mathcal{N}_p(\theta)}$, then we actually obtain $\parallel \gamma \parallel_{p=1}$. Therefore, an arbitrary $\ell_p$-norm of kernel coefficients is hidden in the dual formulation without the need for explicit consideration, and since $\forall m, \ \frac{\theta_m^2}{\mathcal{N}_p(\theta)} \geq 0$ always holds, it is guaranteed that $K_{opt}$ is a valid kernel as long as $\gamma_m, K_m$ is valid.

2.2.2 Learning Parameter $\theta$

We employ linear programming (LP) or quadratic programming (QP) to learn parameter $\theta$ while fixing parameter $w$, and the corresponding primal formulations are shown in Eqn. 9.

$$\min_{\theta, w, b, \xi} \mathcal{L}_1(\theta) + C \sum_i \xi_i$$ (9)

s.t. \[
\forall i, \ y_i \left[ \sum_m \theta_m \phi_m^w(x_i) + b \right] \geq 1 - \xi_i \\
\xi_i \geq 0, \ C \geq 0, \ n = 1, 2
\]

where $\mathcal{L}_1(\theta) = \sum_m |\theta_m|$, $\mathcal{L}_2(\theta) = \frac{1}{2} \sum_m \theta_m^2$

$$\phi_m^w(x_i) = \langle w_m, \phi_m(x_i) \rangle$$

To summarize our optimization algorithm, Algorithm 1 lists all the details as follows.

Algorithm 1: AdaMKL

\begin{algorithmic}
\Input $K, y$
\Output $\alpha, \theta, b$
\Initialize $\theta$
\Calculate $\alpha$, $b$ using Eqn. 6;
\Repeat
\Update $\theta$ using Eqn. 9;
\Update $\alpha$, $b$ using Eqn. 6;
\Until Satisfy some conditions;
\Return $\alpha, \theta, b$
\end{algorithmic}

2.3 Discussion

Our AdaMKL shares the same computational complexity as LP or QP, and in our implementation we set the initial value of $\theta$ to 1 as in some other convex optimization based MKL approaches (e.g. [6], [5]).
Another important issue for AdaMKL is its convergence, because this could be an important termination criterion. Generally speaking, we cannot guarantee that any formulation of AdaMKL will converge to its local minimum. However, letting $g_1(\theta, w, \xi_1) = \min \{ \frac{1}{2} N_d(\theta) \| w \|^2 + C \sum_i \xi_i \} \text{ and } g_2(\xi_2) = \min \{ L_n(\theta) + C \sum_i \xi_i \}$, in the following cases it is guaranteed that AdaMKL will converge.

**Proposition 1.** For hard-margin cases ($C = +\infty$), if Eqn. 6 can be solved at the initialization stage then AdaMKL will converge to a local minimum.

**Proof.** In this case, after each update we have $\xi_1 = 0$. Then considering the primal, at the $t^{th}$ and $s^{th}$ updates for $\theta$ and $w$, respectively, we have $g_1(\theta^{(t)}), w^{(s)}, \xi_1^{(s)}) \geq g_1(\theta^{(t)}), w^{(s+1)}, \xi_1^{(s+1)}) \geq g_1(\theta^{(t+1)}), w^{(s+1)}, \xi_1^{(s+1)})$ plus $\forall (\theta, w, \xi_1), g_1 \geq 0$, so AdaMKL will converge to a local minimum. \qed

**Proposition 2.** If $g_2$ converges, then AdaMKL has converged to a local minimum.

**Proof.** Suppose at the $t^{th}$ and $s^{th}$ updates for $\theta$ and $w$, respectively, $g_2$ converged; then $\theta^{(t)} = \theta^{(t+1)}$ and $\xi_1^{(s)} = \xi_2^{(t)} = \xi_2^{(t+1)} = \xi_1^{(s+1)}$. Due to the local convergence, we have $g_1(\theta^{(t+1)}, w^{(s)}, \xi_1^{(s+1)}) = g_1(\theta^{(t+1)}, w^{(s+1)}, \xi_1^{(s+1)})$, which means AdaMKL has converged. \qed

3 **Experiments**

We construct four different specific formulations of AdaMKL, viz., AdaMKL-$N_0 \mathcal{L}_1$, AdaMKL-$N_1 \mathcal{L}_1$, AdaMKL-$N_2 \mathcal{L}_2$, AdaMKL-$N_2 \mathcal{L}_2$, where $N$ and $\mathcal{L}$ follow the notations in Eqn. 5 and 9, respectively.

3.1 **Toy Example**

We first tested our approach on a toy dataset, where data is sampled randomly based on 2D Gaussian distributions. For positive data, the mean vector of the Gaussian distribution is $[0 \ 0]$ and the covariance matrix is $[0.3 \ 0; 0 \ 0.3]$. For negative data, the mean vectors of the Gaussian distributions are $[1 \ -1]$ and $[1 \ 1]$, and the covariance matrices are $[0.1 \ 0; 0.01]$ and $[0.2 \ 0; 0 \ 0.2]$, respectively. Each distribution generates 100 samples. We create 10 Gaussian kernels with parameters from 1 to 10, with step equal to 1. Here, we fix the parameter C in AdaMKL as $10^5$ without tuning.

Fig. 1 shows the behavior of the dual values of Eqn. 6 at each update using different AdaMKL. Since there are always some misclassified instances in this toy example, Prop. 1 cannot be demonstrated here and therefore their convergence behaviors are difficult to predict, especially for AdaMKL-$N_0 \mathcal{L}_1$. However, from this figure we can also see that: (1) The norm constraint makes AdaMKL converge to a local minimum more smoothly. (2) In general, most of the biggest changes occur in the first few updates, which indicates that our AdaMKL can be terminated in a few updates. In Fig. 2 we can see that for each of the four AdaMKL, the support vector distribution changes slightly after the $1^{st}$ update compared to the others. This also suggests that an early stop strategy can be applied in our AdaMKL to speed up its learning.

3.2 **Benchmark Datasets**

We also tested our AdaMKL on four benchmark datasets: breast-cancer, heart, thyroid, and titanic. Each dataset contains 100 pairs of training and test files, where the numbers of training patterns are 200, 170, 140, 150, the numbers of test patterns are 77, 100, 75, 2051, and the feature dimensions are 9, 13, 5, 3, respectively.

1These datasets can be downloaded from [http://ida.first.fraunhofer.de/projects/bench/](http://ida.first.fraunhofer.de/projects/bench/).
the parameters alternately. As a result, an arbitrary $\ell_p$-norm ($p \geq 1$) of kernel coefficients can be involved in our formulation, and the sparse (or non-sparse) learning is separated from this norm constraint, which makes the learning process much easier and faster. We mainly discuss the convergence issue of AdaMKL, and compare four specific AdaMKL with two state-of-the-art convex optimization based MKL approaches in our experiments. In general, the performance of each AdaMKL is comparable to the MKL approaches with much faster learning speed.

4 Conclusion

In this paper, Adaptive Multiple Kernel Learning (AdaMKL) is proposed based on the large-margin criterion. We propose a family of biconvex objective functions for AdaMKL which are minimized by updating the parameters alternately. As a result, an arbitrary $\ell_p$-norm ($p \geq 1$) of kernel coefficients can be involved in our formulation, and the sparse (or non-sparse) learning is separated from this norm constraint, which makes the learning process much easier and faster. We mainly discuss the convergence issue of AdaMKL, and compare four specific AdaMKL with two state-of-the-art convex optimization based MKL approaches in our experiments. In general, the performance of each AdaMKL is comparable to the MKL approaches with much faster learning speed.

References