

# rls\_dual\_mkl: A PFBS-based Implementation for Multiple Kernel Learning

(Jeremiah) Zhe Liu

ZHL112@MAIL.HARVARD.EDU

*Department of Biostatistics*

*Harvard School of Public Health*

*Boston, MA, 02115*

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Multiple Kernel Learning Problem . . . . .	2
1.2	Iterative PFBS Algorithm . . . . .	2
1.3	Implementation Detail . . . . .	3
1.3.1	Choice of Stepsize . . . . .	3
1.3.2	Choice of $\mu$ and $\tau$ . . . . .	4
1.3.3	Parallel . . . . .	4
<b>2</b>	<b>Software Structure and Usage</b>	<b>4</b>
<b>3</b>	<b>Example</b>	<b>5</b>

## 1. Introduction

### 1.1. Multiple Kernel Learning Problem

Multiple kernel learning (MKL) (Bach et al. (2004)) is the process of finding an optimal kernel from a prescribed (convex) set  $\mathcal{K}$  of basis kernels, for learning a real-valued function by regularization. In this work, we consider a RKHS  $\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2 \cdots \oplus \mathcal{H}_M$  with reproducing kernel  $\mathbf{k} \in \mathcal{K} = \{\sum_{i=1}^M c_i \mathbf{k}_i | (c_i \geq 0 \forall i) \wedge \sum_{i=1}^M c_i = 1\}$  such that  $f = \sum_{i=1}^M f_i, f_i \in \mathcal{H}_i$ . By Micchelli and Pontil (2005), the problem of multiple kernel learning corresponds to find  $f^*$  such that:

$$\arg \min_{f \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^n Q\left(\sum_{j=1}^M f_j(\mathbf{x}), \mathbf{y}\right) + \tau g\left(\left(\sum_{j=1}^M \|f_j\|_{\mathcal{H}}\right)^2\right) \right\}$$

Rosasco et al. (2009) generalized above problem by taking  $Q$  to be square loss,  $g(\cdot) = \sqrt{\cdot}$  and also impose L1 regularization, leading to the elastic-net-regulated problem:

$$\arg \min_{f \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^M f_j(x_i) - y_i\right)^2 + \mu \sum_{j=1}^M \|f_j\|_{\mathcal{H}}^2 + 2\tau \sum_{j=1}^M \|f_j\|_{\mathcal{H}} \right\} \quad (1)$$

### 1.2. Iterative PFBS Algorithm

By Theorem 1 of Rosasco et al. (2009), since the penalty function is lower semicontinuous, coercive, convex and one-homogenous, solution to problem 1  $f^*$  is the unique fixed point of the the contractive mapping with step size  $\sigma$ :

$$\mathcal{T}_{\sigma}(f) = (\mathbf{I} - \pi_{\frac{\tau}{\sigma}K})(f - \frac{1}{2\sigma} \nabla_f [\frac{1}{n} \|f - y\|^2])$$

where  $\pi_{\frac{\tau}{\sigma}K}(g)$  is a project operator which project  $g$  to  $\mathcal{H}' = \{f \in \mathcal{H} | \|f_j\|_{\mathcal{H}_j} \leq \frac{1}{\tau/\sigma} \forall j\}$ , or more rigorously:

$$\pi_{\frac{\tau}{\sigma}K}(g) = \frac{\tau}{\sigma} v, \quad \text{where } v = \arg \min_{v \in \mathcal{H}, \|v_j\| \leq 1} \left\| \frac{\tau}{\sigma} v - g \right\|_{\mathcal{H}}^2$$

Above mapping can also be written in terms of Kernel matrices by generalizing representer theorem and write  $f_j^*(x) = \sum_{i=1}^n \alpha_{ji}^T k_j(x_i, x) = \boldsymbol{\alpha}_j^T \mathbf{k}_j(x)$ , where  $\boldsymbol{\alpha}_j$  and  $\mathbf{k}_j(x)$  are  $n \times 1$  vectors. Further, if denote:

$$\begin{aligned} \boldsymbol{\alpha}_{Mn \times 1} &= (\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_M)^T \\ \mathbf{k}(x)_{Mn \times 1} &= (\mathbf{k}_1(x)^T, \dots, \mathbf{k}_M(x)^T)^T \\ \mathbf{K}_{Mn \times Mn} &= \begin{bmatrix} \mathbf{K}_1 & \dots & \mathbf{K}_M \\ \vdots & \ddots & \vdots \\ \mathbf{K}_1 & \dots & \mathbf{K}_M \end{bmatrix}, \text{ where } \mathbf{K}_i = \mathbf{k}_i(\cdot) \mathbf{k}_i(\cdot)^T \\ \mathbf{y}_{Mn \times 1} &= (y_{n \times 1}^T, \dots, y_{n \times 1}^T)^T \end{aligned}$$

The contraction mapping can be written as:

$$\begin{aligned} \mathcal{T}_\sigma(f) &= (\mathbf{I} - \pi_{\frac{\tau}{\sigma}K})\left(\left[\left(1 - \frac{\mu}{\sigma}\right)\boldsymbol{\alpha} - \frac{1}{\sigma n}(\mathbf{K}\boldsymbol{\alpha} - \mathbf{y})\right]^T \mathbf{k}\right) \quad \text{where} \\ \pi_{\frac{\tau}{\sigma}K}(g)_j &= \min\left\{1, \frac{\|g_j\|_{\mathcal{H}_j}}{\tau/\sigma}\right\} * \frac{g_j}{\|g_j\|_{\mathcal{H}_j}} = \min\left\{1, \frac{\sqrt{\boldsymbol{\alpha}_j^T \mathbf{K}_j \boldsymbol{\alpha}_j}}{\tau/\sigma}\right\} * \frac{\boldsymbol{\alpha}_j^T \mathbf{k}_j}{\sqrt{\boldsymbol{\alpha}_j^T \mathbf{K}_j \boldsymbol{\alpha}_j}} \end{aligned} \quad (2)$$

Thus the projection  $\mathbf{I} - \pi_{\frac{\tau}{\sigma}K}$  corresponds to the soft-thresholding operator for  $\boldsymbol{\alpha}_j$ :

$$\mathbf{S}_{\frac{\tau}{\sigma}}(K, \boldsymbol{\alpha})_j = \frac{\boldsymbol{\alpha}_j^T}{\sqrt{\boldsymbol{\alpha}_j^T \mathbf{K}_j \boldsymbol{\alpha}_j}} \left( \sqrt{\boldsymbol{\alpha}_j^T \mathbf{K}_j \boldsymbol{\alpha}_j} - \frac{\tau}{\sigma} \right)_+$$

Above discussions lead to below algorithm:

---

**Algorithm 1:** MKL Algorithm

---

```

set  $\boldsymbol{\alpha}^0 = \mathbf{0}$ 
for  $p = 1$  to MAX_ITER do
     $\boldsymbol{\alpha}_0^p = \left(1 - \frac{\mu}{\sigma}\right)\boldsymbol{\alpha}^{p-1} - \frac{1}{\sigma n}(\mathbf{K}\boldsymbol{\alpha}^{p-1} - \mathbf{y})$ 
     $\boldsymbol{\alpha}^p = \mathbf{S}_{\frac{\tau}{\sigma}}(K, \boldsymbol{\alpha}_0^p)$ 
end for
return  $f^{\text{MAX\_ITER}} = (\boldsymbol{\alpha}^{\text{MAX\_ITER}})^T \mathbf{k}$ 
    
```

---

### 1.3. Implementation Detail

#### 1.3.1. CHOICE OF STEPSIZE

The convergence of the aforementioned procedure is guaranteed by Banach Fixed Point theorem, given proper choice of  $\sigma$ . Based on [Bach et al. \(2004\)](#), it can be shown that a suitable choice of  $\sigma$  is  $\sigma = \frac{1}{4}(a * L_{\min} + b * L_{\max}) + \mu$ , where  $(b, a)$  denotes the lower/upper bound on the eigenvalue of  $\mathbf{K}$ , and  $(L_{\min}, L_{\max})$  denotes the lower/upper bound of  $\nabla^2 Q(\mathbf{f}, \mathbf{y})$ .

In the context where  $Q$  is square loss (i.e.  $\nabla_{\mathbf{f}}^2 Q(\mathbf{f}, \mathbf{y}) = 2$ ), we have:

$$\sigma = \frac{1}{2}(a + b) + \mu$$

Approximate eigenvalue of  $\mathbf{K}$  using the PerronFrobenius upper bound, i.e.  $\sigma_{\max} \leq \sum_j \mathbf{K}_{ij}$ .

Approximate eigenvalue of  $\mathbf{K}$  using its upper bound  $\sigma = \|\mathbf{K}\| \leq M * \sum_{m=1}^M \|\mathbf{K}_m\|$ , since:

$$\begin{aligned}
\|\mathbf{K}\| &= \underset{\boldsymbol{\alpha} \in \mathbb{R}^{Mn}}{\operatorname{argmax}} \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} = \left( \sum_{i=1}^M \boldsymbol{\alpha}_i \right)^T \left( \sum_{j=1}^M \mathbf{K}_j \boldsymbol{\alpha}_j \right) \\
&\leq \left\| \sum_{i=1}^M \boldsymbol{\alpha}_i \right\| * \left\| \sum_{j=1}^M \mathbf{K}_j \boldsymbol{\alpha}_j \right\| \leq \left\| \sum_{j=1}^M \mathbf{K}_j \right\| * \left\| \sum_{i=1}^M \boldsymbol{\alpha}_i \right\|^2 = \|\mathbf{I}_{n \times Mn} \boldsymbol{\alpha}\|^2 * \sum_{j=1}^M \|\mathbf{K}_j\| \\
&\leq \|\mathbf{I}_{n \times Mn}\|^2 \|\boldsymbol{\alpha}\|^2 * \sum_{j=1}^M \|\mathbf{K}_j\| \\
&= M * \sum_{m=1}^j \|\mathbf{K}_j\|
\end{aligned}$$

### 1.3.2. CHOICE OF $\mu$ AND $\tau$

Zou and Hastie (2005)

Rosasco et al. (2009)

Contraction mapping

$$\|(1 - \frac{\mu}{\sigma})\mathbf{I} - \frac{1}{\sigma n} \mathbf{K}\|$$

Continuation strategy.

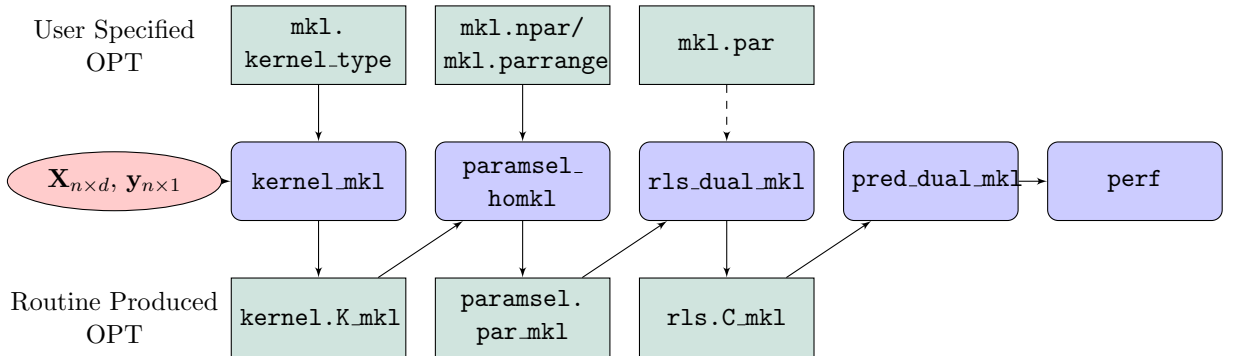
$$\frac{\tau}{\sigma_{max}} = \frac{\|y\|^2}{\|\mathbf{K}\|}$$

### 1.3.3. PARALLEL

Notice that in (2),  $\mathcal{T}_\sigma$  updates  $\boldsymbol{\alpha}$  by group, it is thus possible to write  $\mathcal{T}$  at  $p^{th}$  step as:

$$\begin{aligned}
\mathcal{T}_\sigma^p &= [\mathcal{T}_{\sigma,1}^p, \mathcal{T}_{\sigma,2}^p, \dots, \mathcal{T}_{\sigma,M}^p] \quad \text{with} \\
\mathcal{T}_{\sigma,j}^p &= \mathbf{S}_{\frac{\tau}{\sigma}} \left( \mathbf{K}, (1 - \frac{\mu}{\sigma}) \boldsymbol{\alpha}_j^{p-1} - \frac{1}{n\sigma} * \boldsymbol{\epsilon}^{p-1} \right) \quad \text{where } \boldsymbol{\epsilon}^{p-1} = \left( \sum_{m=1}^M \mathbf{K}_m \boldsymbol{\alpha}_m^{p-1} - y \right)
\end{aligned}$$

## 2. Software Structure and Usage



1. Kernel Generation

- kernel/kernel\_mkl

2. Parameter Selection

- kernel/paramsel\_homkl
- util/paramsel\_L1ratioguesses

3. Optimization

- optimizers/rls\_dual\_mkl
- optimizers/rls\_dual\_mkl\_pfbs
- util/ConsoleProgressBar

4. Prediction

- optimizers/pred\_dual\_mkl

5. Performance assessment

- perf/perf\_rmsestd

### 3. Example

1

---

1. LIBSVM Data Repository: '<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html>'

## References

- F Bach, G Lanckriet, and M Jordan. Multiple kernel learning, conic duality, and the smo algorithm. *ACM International Conference Proceeding Series*, 69, 2004.
- C Micchelli and M Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:10991125, 2005.
- L Rosasco, S Mosci, M Santoro, A Verri, and S Villa. Iterative projection methods for structured sparsity regularization. Technical report, MIT, 2009.
- H Zou and T Hastie. Regularization and variable selection via the elastic net. *J. R. Statist. Soc. B*, 67, Part 2:301320, 2005.