rls_mkl: A Projection-based MATLAB Implementation for Multiple Kernel Learning

(Jeremiah) Zhe Liu zhl112@mail.harvard.edu
Department of Biostatistics
Harvard School of Public Health
Boston, MA, 02115

Contents

1 Introduction **2**
1.1 Multiple Kernel Learning Problem 2
1.2 Iterative Projection Algorithm 2
1.3 Implementation Detail 3
1.3.1 Choice of Stepsize 3
1.3.2 Parallel 3
1.3.3 Alternative Regularization through Early Stopping 3

2 Software Structure and Usage **3**

3 Example **3**

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 reproducting kernel $\mathbf{k} \in \mathcal{K} = \{\sum_{i=1}^M c_i \mathbf{k}_i | (c_i \geq 0 \forall i) \wedge \sum_{i=1} c_i = 1\}$ 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:

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

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

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

1.2. Iterative Projection 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 σ:

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

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

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

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:

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

The contraction mapping can be written as:

$$\mathcal{T}_\sigma(f) = (\mathbf{I} - \pi_{\frac{\tau}{\sigma}K})\big((1-\frac{\mu}{\sigma})\boldsymbol{\alpha} - \frac{1}{\sigma n}(\mathbf{K}\boldsymbol{\alpha} - \mathbf{y})\mathbf{k}\big) \qquad \text{where}$$

$$\pi_{\frac{\tau}{\sigma}K}(g)_j = min\{1, \frac{||g_j||_{\mathcal{H}_j}}{\tau/\sigma}\} * \frac{g_j}{||g_j||_{\mathcal{H}_j}} = min\{1, \frac{\sqrt{\boldsymbol{\alpha}_j^T\mathbf{K}_j\boldsymbol{\alpha}_j}}{\tau/\sigma}\} * \frac{\boldsymbol{\alpha}_j^T\mathbf{k}_j}{\sqrt{\boldsymbol{\alpha}_j^T\mathbf{K}_j\boldsymbol{\alpha}_j}}$$

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

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

Above discussions lead to below algorithm:

Algorithm 1: MKL Algorithm

set $\boldsymbol{\alpha}^0 = \mathbf{0}$

for $p = 1$ to `MAX_ITER` **do**

$\boldsymbol{\beta}^{p-1} = (1 - \frac{\mu}{\sigma})\boldsymbol{\alpha}^{p-1} - \frac{1}{\sigma n}(\mathbf{K}\boldsymbol{\alpha}^{p-1} - \mathbf{y})$
$\boldsymbol{\alpha}^p = \mathbf{S}_{\frac{\tau}{\sigma}}(K, \boldsymbol{\beta}^{p-1})$

end for

return $f^{\texttt{MAX\_ITER}} = (\boldsymbol{\alpha}^{\texttt{MAX\_ITER}})^T\mathbf{k}$

1.3. Implementation Detail

1.3.1. Choice of Stepsize

Micchelli and Pontil (2005)

1.3.2. Parallel

1.3.3. Alternative Regularization through Early Stopping

2. Software Structure and Usage

3. Example

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.

S Mosci, L Rosasco, M Santoro, A Verri, and S Villa. Solving structured sparsity regularization with proximal methods. *Machine Learning and Knowledge Discovery in Databases*, II:418, September 2010.

L Rosasco, S Mosci, M Santoro, A Verri, and S Villa. Iterative projection methods for structured sparsity regularization. Technical report, MIT, 2009.