JMLR: Workshop and Conference Proceedings 1–5, 2015 9.520 Project Report

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

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:

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

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 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:

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

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:

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

The contraction mapping can be written as:

$$\mathcal{T}_\sigma(f) = (\mathbf{I} - \pi_{\frac{\tau}{\sigma}K})\big(\big[(1-\frac{\mu}{\sigma})\boldsymbol{\alpha} - \frac{1}{\sigma n}(\mathbf{K}\boldsymbol{\alpha} - \mathbf{y})\big]^T\mathbf{k}\big) \qquad \text{where} \tag{2}$$

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

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**

$\boldsymbol{\alpha}_0^p = (1 - \frac{\mu}{\sigma})\boldsymbol{\alpha}^{p-1} - \frac{1}{\sigma n}(\mathbf{K}\boldsymbol{\alpha}^{p-1} - \mathbf{y})$
$\boldsymbol{\alpha}^p = \mathbf{S}_{\frac{\tau}{\sigma}}(K, \boldsymbol{\alpha}_0^p)$

end for

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

1.3. Implementation Detail

1.3.1. Normalization

1.3.2. Choice of Stepsize

The convergence of the aforementioned procedure is guaranteed by Banach Fixed Point theorem, given proper choice of σ. Based on Bach et al. (2004), it can be shown that a suitable choice of σ 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. $\nabla^2_{\mathbf{f}} Q(\mathbf{f}, \mathbf{y}) = 2$), 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:

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

Barzilai and Borwein (1988)
$\Delta\boldsymbol{\alpha} = \boldsymbol{\alpha}^p - \boldsymbol{\alpha}^{p-1}$
$\Delta\boldsymbol{g} = u * (\boldsymbol{\alpha}^p - \boldsymbol{\alpha}^{p-1}) + \frac{1}{n}\mathbf{K}(\boldsymbol{\alpha}^p - \boldsymbol{\alpha}^{p-1}) = (u\mathbf{I} + \frac{1}{n}\mathbf{K})\Delta\boldsymbol{\alpha}$

$$\frac{\langle \Delta\boldsymbol{\alpha}, \Delta\boldsymbol{g} \rangle}{\langle \Delta\boldsymbol{g}, \Delta\boldsymbol{g} \rangle} = \frac{\sum_{m=1}^{M} \Delta\boldsymbol{\alpha}_m^T \Delta\boldsymbol{g}_m}{\sum_{m=1}^{M} \Delta\boldsymbol{g}_m^T \Delta\boldsymbol{g}_m}$$

1.3.3. Choice of μ

(Rosasco et al., 2009)
Contraction mapping
$||(1 - \frac{\mu}{\sigma})\mathbf{I} - \frac{1}{\sigma n}\mathbf{K}||$

1.3.4. 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:

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

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.

J Barzilai and J Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141148, 1988.

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.