Regularization Strategies and Empirical Bayesian Learning for MKL

Ryota Tomioka **Taiji Suzuki**
Department of Mathematical Informatics, The University of Tokyo,
7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan.

Abstract

Multiple kernel learning (MKL) has received considerable attention recently. In this paper, we show how different MKL algorithms can be understood as applications of different types of regularization on the kernel weights. Within the regularization view we consider in this paper, the Tikhonov-regularization-based formulation of MKL allows us to consider a generative probabilistic model behind MKL. Based on this model, we propose learning algorithms for the kernel weights through the maximization of marginalized likelihood.

1 Introduction

In this paper, we consider the problem of combining multiple data sources in a kernel-based learning framework. More specifically, we assume that a data point $x \in \mathcal{X}$ lies in a space $\mathcal{X}$ and we are given M candidate kernel functions $k_m : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ $(m = 1, \ldots, M)$. Each kernel function corresponds to one data source. A conical combination of k_m $(m = 1, \ldots, M)$ gives the combined kernel function $\bar{k} = \sum_{m=1}^{M} d_m k_m$, where d_m is a nonnegative weight. Our goal is to find a good set of kernel weights based on some training examples.

Various approaches have been proposed for the above problem under the name multiple kernel learning (MKL) [12, 3]. Kloft et al. [11, 10] have recently shown that many MKL approaches can be understood as application of the Tikhonov or Ivanov regularization. However they only showed that there is a regularization constant μ that makes the Tikhonov regularization and Ivanov regularization equivalent, and argued that the Ivanov regularization is preferable to the Tikhonov regularization because it does not require selecting the constant μ. The first contribution of this paper is to show that actually the constant μ that makes the two formulations equivalent can be obtained *analytically*; thus we show that the two formulations are completely equivalent. In addition, we show a connection between the Tikhonov-regularization-based formulation and the generalized block-norm formulation considered in Tomioka & Suzuki [23].

The second contribution of this paper is to derive an empirical Bayesian learning algorithm for MKL motivated by the Tikhonov regularization formulation. Although Bayesian approaches have been applied to MKL earlier in a transductive nonparametric setting [27], and a setting similar to the relevance vector machine [22] in [9, 7], we believe that our formulation is more coherent with the correspondence between Gaussian process classification/regression and kernel methods [17]. In addition, we propose two iterative algorithms for the learning of kernel weights through the maximization of marginalized likelihood. One algorithm iteratively solves a reweighted MKL problem and the other iterates between a classifier training for a fixed kernel combination and a kernel weight update.

2 Learning with fixed kernel combination

We assume that we are given N training examples $(x_i, y_i)_{i=1}^N$ where x_i belongs to an input space $\mathcal{X}$ and y_i belongs to an output space $\mathcal{Y}$ (usual settings are $\mathcal{Y} = \{\pm 1\}$ for classification and $\mathcal{Y} = \mathbb{R}$ for regression).

We first consider a learning problem with fixed kernel weights. More specifically, we fix nonnegative kernel weights $d_1, d_2, \ldots, d_M$ and consider the RKHS $\bar{\mathcal{H}}$ corresponding to the combined kernel function $\bar{k} = \sum_{m=1}^M d_m k_m$. The squared RKHS norm of a function $\bar{f}$ in the combined RKHS $\bar{\mathcal{H}}$ can be represented as follows:

$$\|\bar{f}\|_{\bar{\mathcal{H}}}^2 := \min_{\substack{f_1 \in \mathcal{H}_1, \\ \ldots, f_M \in \mathcal{H}_M}} \sum_{m=1}^M \frac{\|f_m\|_{\mathcal{H}_m}^2}{d_m} \quad \text{s.t. } \bar{f} = \sum_{m=1}^M f_m, \tag{1}$$

where $\mathcal{H}_m$ is the RKHS that corresponds to the kernel function k_m. See Sec 6 in [2], and also Lemma 25 in [14] for the proof. We also provide some intuition for a finite dimensional case in Appendix B

Using the above representation, a supervised learning problem with a fixed kernel combination can be written as follows:

$$\underset{\substack{f_1 \in \mathcal{H}_1, \\ \ldots, f_M \in \mathcal{H}_M, \\ b \in \mathbb{R}}}{\text{minimize}} \sum_{i=1}^N \ell\left(y_i, \textstyle\sum_{m=1}^M f_m(x_i) + b\right) + \frac{C}{2} \sum_{m=1}^M \frac{\|f_m\|_{\mathcal{H}_m}^2}{d_m}, \tag{2}$$

where $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a loss function and we assume that ℓ is convex in the second argument; for example, the loss function can be the hinge loss $\ell_H(y_i, z_i) = \max(0, 1 - y_i z_i)$, or the quadratic loss $\ell_Q(y_i, z_i) = (y_i - z_i)^2/(2\sigma_y^2)$.

It might seem that we are making the problem unnecessarily complex by introducing M functions f_m to optimize instead of simply optimizing over $\bar{f}$. However, explicitly handling the kernel weights enables us to consider various regularization strategies on the weights as we see in the next section.

3 Learning kernel weights

Now we are ready to also optimize the kernel weights d_m in the above formulation. Clearly there is a need for regularization, because the objective (2) is a monotone decreasing function of the kernel weights d_m. Intuitively speaking, d_m corresponds to the complexity allowed for the mth regression function f_m; the more complexity we allow, the better the fit to the training examples becomes. Thus without any constraint on d_m, we can get a severe overfitting problem.

There are essentially two ways to prevent such overfitting [11]. One is to enforce some constraints on d_m, which is called Ivanov regularization and the other is to add a penalty term to the objective, which is called Tikhonov regularization.

In this section, we discuss the Tikhonov regularization. See Kloft et al [11, 10] and Appendix C for the Ivanov regularization. Table 1 summarizes the regularization strategies we discuss in this paper.

3.1 Tikhonov regularization

One way to penalize the complexity is to minimize the objective (2) together with the regularizer $h(d_m)$ as follows:

$$\underset{\substack{f_1 \in \mathcal{H}_1, \ldots, f_M \in \mathcal{H}_M, \\ b \in \mathbb{R}, \\ d_1 \geq 0, \ldots, d_M \geq 0}}{\text{minimize}} \sum_{i=1}^N \ell\left(y_i, \textstyle\sum_{m=1}^M f_m(x_i) + b\right) + \frac{C}{2} \sum_{m=1}^M \left(\frac{\|f_m\|_{\mathcal{H}_m}^2}{d_m} + \mu h(d_m)\right), \tag{3}$$

where h is a convex nondecreasing function defined the nonnegative reals, and the regularization constant $\mu > 0$ is introduced to make a correspondence between the above formulation to the Ivanov-regularization-based formulation (see Appendix C).

MKL model	$g(x)$	$h(d_m)$	μ	Equality in (5)
block 1-norm MKL	$\sqrt{x}$	d_m	1	$d_m = \|f_m\|_{\mathcal{H}_m}$
ℓ_p-norm MKL	$\frac{1+p}{2p} x^{p/(1+p)}$	d_m^p	$1/p$	$d_m = \|f_m\|_{\mathcal{H}_m}^{2/(1+p)}$
Uniform-weight MKL (block 2-norm MKL)	$x/2$	$I_{[0,1]}(d_m)$	$+0$	$d_m = 1$
block q-norm MKL ($q > 2$)	$\frac{1}{q} x^{q/2}$	$d_m^{-q/(q-2)}$	$-(q-2)/q$	$d_m = \|f_m\|_{\mathcal{H}_m}^{2-q}$
Elastic-net MKL	$(1-\lambda)\sqrt{x} + \frac{\lambda}{2} x$	$\frac{(1-\lambda)d_m}{1-\lambda d_m}$	$1-\lambda$	$d_m = \frac{\|f_m\|_{\mathcal{H}_m}}{(1-\lambda)+\lambda\|f_m\|_{\mathcal{H}_m}}$

Table 1: Correspondence of the concave function g in the block-norm formulation (6), and the regularizer h and constant μ in the Ivanov and Tikhonov formulations (15) and (3). $I_{[0,1]}$ denotes the indicator function of the interval $[0, 1]$; i.e., $I_{[0,1]}(x) = 0$ (if $x \in [0, 1]$), and $I_{[0,1]}(x) = \infty$ (otherwise).

For some choices of h and μ, it is easy to eliminate the kernel weights d_m in Eq. (3) and obtain a block-norm formulation. For example, if $h(d_m) = d_m$ (a linear function) and $\mu = 1$, we obtain the block 1-norm formulation (see also [4, 24]) as follows:

$$\underset{f_1 \in \mathcal{H}_1, \ldots, f_M \in \mathcal{H}_M, b \in \mathbb{R}}{\text{minimize}} \sum_{i=1}^{N} \ell\left(y_i, \textstyle\sum_{m=1}^{M} f_m(x_i) + b\right) + C \sum_{m=1}^{M} \|f_m\|_{\mathcal{H}_m}. \tag{4}$$

For general regularizer h, we can use a convex upper-bounding technique [15] to derive the corresponding block-norm formulation. In order to do this, we first let $\tilde{h}(y) = -\mu h(1/y)$. Note that $\tilde{h}$ is a concave function, because $1/y$ is a convex function for $y > 0$, h is a nondecreasing convex function, and $\mu > 0$ (see Sec. 3.2.4 in [5]). Then by the definition of concave conjugate, we have

$$\begin{aligned} \frac{\|f_m\|_{\mathcal{H}_m}^2}{d_m} + \mu h(d_m) &= \frac{\|f_m\|_{\mathcal{H}_m}^2}{d_m} - \tilde{h}(1/d_m) \\ &\geq \tilde{h}^*(\|f_m\|_{\mathcal{H}_m}^2) =: 2g(\|f_m\|_{\mathcal{H}_m}^2), \end{aligned} \tag{5}$$

where $\tilde{h}^*$ denotes the concave conjugate function of the concave function $\tilde{h}$; for convenience, we defined the concave function g as above. The equality is obtained when $d_m = 1/(2g'(\|f_m\|_{\mathcal{H}_m}^2))$, where g' is the derivative of g. For example, $h(d_m) = d_m$ and $\mu = 1$ give $\tilde{h}(y) = -1/y$, $g(x) = \sqrt{x}$, and the equality is obtained when $d_m = \|f_m\|_{\mathcal{H}_m}$. See Table 1 for more examples.

3.2 Generalized block-norm formulation

The resulting generalized block-norm formulation can be written as follows:

$$\underset{f_1 \in \mathcal{H}_1, \ldots, f_M \in \mathcal{H}_M, b \in \mathbb{R}}{\text{minimize}} \sum_{i=1}^{N} \ell\left(y_i, \textstyle\sum_{m=1}^{M} f_m(x_i) + b\right) + C \sum_{m=1}^{M} g(\|f_m\|_{\mathcal{H}_m}^2), \tag{6}$$

where g is the concave function defined in Eq. (5).

In Tomioka & Suzuki [23], the following elastic-net regularizer g was considered:

$$g(x) = (1-\lambda)\sqrt{x} + \frac{\lambda}{2} x. \tag{7}$$

With the above concave regularizer g, Eq. (6) becomes

$$\underset{\substack{f_1 \in \mathcal{H}_1, \ldots, f_M \in \mathcal{H}_M, \\ b \in \mathbb{R}}}{\text{minimize}} \sum_{i=1}^{N} \ell\left(y_i, \textstyle\sum_{m=1}^{M} f_m(x_i) + b\right) + C \sum_{m=1}^{M} \left((1-\lambda)\|f_m\|_{\mathcal{H}_m} + \frac{\lambda}{2}\|f_m\|_{\mathcal{H}_m}^2\right), \tag{8}$$

which reduces to the block 1-norm regularization (Eq. (4)) for $\lambda = 0$ and the uniform-weight combination ($d_m = 1$ in Eq. (2)) for $\lambda = 1$.

In order to derive the Tikhonov regularization problem (3) corresponding to the elastic-net regularization (8), we only need to compute the relation (5) backwards as follows:

$$\mu h(d_m) = -\tilde{h}(1/d_m) = -(2g)^*(1/d_m) = -2g^*(1/(2d_m)). \tag{9}$$

For the concave regularizer (7), we can easily obtain

$$\mu h(d_m) = \frac{(1-\lambda)^2 d_m}{1-\lambda d_m}.$$

The Ivanov-regularization-based formulation for the Elastic-net MKL (8) can also be derived analytically. See Appendix C.

4 Empirical Bayesian multiple kernel learning

The Tikhonov regularization formulation (3) allows a probabilistic interpretation as a hierarchical maximum a posteriori (MAP) estimation problem. The loss term can be considered as a negative log-likelihood. The first regularization term $\|f_m\|_{\mathcal{H}_m}^2/d_m$ can be considered as the negative log of a Gaussian process prior with variance scaled by the hyper-parameter d_m. The last regularization term $\mu h(d_m)$ corresponds to the negative log of a hyper-prior distribution $p(d_m) \propto \exp(-\mu h(d_m))$. In this section, instead of a MAP estimation, we maximize the marginalized likelihood (evidence) to obtain the kernel weights.

We rewrite the Tikhonov regularization problem (3) as a probabilistic generative model as follows:

$$\begin{aligned}
d_m &\sim \frac{1}{Z_1(\mu)} \exp(-\mu h(d_m)) \qquad (m = 1, \dots, M), \\
f_m &\sim GP(f_m; 0, d_m k_m) \qquad (m = 1, \dots, M) \\
y_i &\sim \frac{1}{Z_2} \exp(-\ell(y_i, f_1(x_i) + f_2(x_i) + \cdots + f_M(x_i))),
\end{aligned}$$

where $Z_1(\mu)$ and Z_2 are normalization constants; $GP(f; 0, k)$ denotes the Gaussian process [17] with mean zero and covariance function k. We omit the bias term for simplicity.

When the loss function is quadratic $\ell(y_i, z_i) = (y_i - z_i)^2/(2\sigma_y^2)$, we can analytically integrate out the Gaussian process random variable $(f_m)_{m=1}^M$ and compute the negative log of the marginalized likelihood as follows:

$$-\log p(\boldsymbol{y}|\boldsymbol{d}) = \frac{1}{2}\boldsymbol{y}^\top \bar{\boldsymbol{K}}(\boldsymbol{d})^{-1}\boldsymbol{y} + \frac{1}{2}\log\left|\bar{\boldsymbol{K}}(\boldsymbol{d})\right| \tag{10}$$

where $\boldsymbol{d} = (d_1, \dots, d_M)^\top$, $\boldsymbol{K}_m = (k_m(x_i, x_j))_{i,j=1}^N$ is the Gram matrix, and

$$\bar{\boldsymbol{K}}(\boldsymbol{d}) := \sigma_y^2 \boldsymbol{I}_N + \sum_{m=1}^M d_m \boldsymbol{K}_m.$$

We could directly minimize (e.g., by gradient descent) the marginalized likelihood (10) to obtain a hyperparameter maximum likelihood estimation. However this could be challenging because of the nonconvexity of the marginalized likelihood.

We present two alternative approaches for the maximization of the marginalized likelihood (10). The first approach is based on upper-bounding both terms in Eq. (10); since the upper-bound takes a form of the Tikhonov regularization problem (3), we can minimize this efficiently using various algorithms for MKL proposed recently [20, 6, 21]. The second approach uses the same upper-bound on the quadratic term in Eq. (10) but leaves the log determinant term as it is. Then we perform a fixed-point iteration known as the MacKay update [13, 25] for the optimization of the kernel weights.

For the first approach, we first express the quadratic term in the negative log-likelihood (10) as a minimization over $\boldsymbol{f}_m \in \mathbb{R}^N$ $(m = 1, \dots, M)$ as follows (see e.g., [25]):

$$\frac{1}{2}\boldsymbol{y}^\top \bar{\boldsymbol{K}}(\boldsymbol{d})^{-1}\boldsymbol{y} = \min_{\substack{\boldsymbol{f}_1 \in \mathbb{R}^N, \\ \dots, \boldsymbol{f}_M \in \mathbb{R}^N}} \left(\frac{1}{2\sigma_y^2} \left\| \boldsymbol{y} - \sum_{m=1}^M \boldsymbol{f}_m \right\|^2 + \frac{1}{2}\sum_{m=1}^M \frac{\|\boldsymbol{f}_m\|_{\boldsymbol{K}_m}^2}{d_m} \right), \tag{11}$$

where $\boldsymbol{f}_m := (f_m(x_1), \ldots, f_m(x_N))^\top$, and $\|\boldsymbol{f}_m\|^2_{\boldsymbol{K}_m} = \boldsymbol{f}_m^\top \boldsymbol{K}_m^{-1} \boldsymbol{f}_m$. Note that the above expression corresponds to the first two terms in the Tikhonov regularization problem (3).

Next, we express the log determinant term in Eq. (10) as a minimization. Noticing that the function $\psi(\boldsymbol{d}) := \log \left|\bar{\boldsymbol{K}}(\boldsymbol{d})\right|$ is concave in d_m (see p73 in [5]), we have

$$\log \left|\bar{\boldsymbol{K}}(\boldsymbol{d})\right| = \min_{\boldsymbol{z} \in \mathbb{R}^M_+} \left(\sum_{m=1}^M z_m d_m - \psi^*(\boldsymbol{z}) \right), \tag{12}$$

where $z_m > 0$ ($m = 1, \ldots, M$) and ψ^* is the concave conjugate function of ψ. See [26, 19] for the details and other approaches (upper-bound and lower-bound) to approximate the log determinant term.

Combining the two upper-bounds (11) and (12), we have

$$-\log p(\boldsymbol{y}|\boldsymbol{d}) = \min_{\substack{\boldsymbol{f}_1 \in \mathbb{R}^N, \\ \ldots \boldsymbol{f}_M \in \mathbb{R}^N, \\ \boldsymbol{z} \in \mathbb{R}^M_+}} \left(\frac{1}{2\sigma_y^2} \left\| \boldsymbol{y} - \sum_{m=1}^M \boldsymbol{f}_m \right\|^2 + \frac{1}{2} \sum_{m=1}^M \left(\frac{\|\boldsymbol{f}_m\|^2_{\boldsymbol{K}_m}}{d_m} + z_m d_m \right) - \frac{1}{2} \psi^*(\boldsymbol{z}) \right).$$

Comparing the above expression to the Tikhonov problem (3), we can see that minimization of the right-hand side with respect to $(\boldsymbol{f}_m)_{m=1}^M$ and $\boldsymbol{d}$ is a Tikhonov regularization problem with $\mu h(d_m) = z_m d_m$. Accordingly, we obtain a *weighted* block 1-norm MKL problem using the relation (5) (or simply the inequality of arithmetic and geometric means) as follows:

$$\min_{\boldsymbol{d}} -\log p(\boldsymbol{y}|\boldsymbol{d}) = \min_{\substack{\boldsymbol{f}_1 \in \mathbb{R}^N, \\ \ldots \boldsymbol{f}_M \in \mathbb{R}^N, \\ \boldsymbol{z} \in \mathbb{R}^M_+}} \frac{1}{2\sigma_y^2} \left\| \boldsymbol{y} - \sum_{m=1}^M \boldsymbol{f}_m \right\|^2 + \sum_{m=1}^M \sqrt{z_m} \|\boldsymbol{f}_m\|_{\boldsymbol{K}_m} - \frac{1}{2} \psi^*(\boldsymbol{z}).$$

Once we solve the weighted block 1-norm MKL for a fixed variational parameter $\boldsymbol{z}$, we can minimize Eq. (12) over $\boldsymbol{z}$ to tighten the upper-bound. Accordingly the iteration can be written as follows:

$$\begin{aligned}
(\boldsymbol{f}_m)_{m=1}^M &\leftarrow \operatorname*{argmin}_{(\boldsymbol{f}_m)_{m=1}^M} \left(\frac{1}{2\sigma_y^2} \left\| \boldsymbol{y} - \textstyle\sum_{m=1}^M \boldsymbol{f}_m \right\|^2 + \textstyle\sum_{m=1}^M \sqrt{z_m} \|\boldsymbol{f}_m\|_{\boldsymbol{K}_m} \right), \\
z_m &\leftarrow \operatorname{Tr}\left((\sigma_y^2 \boldsymbol{I}_N + \textstyle\sum_{m=1}^M d_m \boldsymbol{K}_m)^{-1} \boldsymbol{K}_m \right),
\end{aligned}$$

where $d_m = \|f_m\|_{\mathcal{H}_m} / \sqrt{z_m}$ in the second line. It can be shown that this procedure converges to a local minimum of the negative log-likelihood [25].

The second approach computes the derivative of the negative log likelihood to derive a fixed-point iteration. By minimizing the right-hand side of Eq. (11), we have

$$-\log p(\boldsymbol{y}|\boldsymbol{d}) = \frac{1}{2\sigma_y^2} \left\| \boldsymbol{y} - \sum_{m=1}^M \boldsymbol{f}_m^{\mathrm{MAP}} \right\|^2 + \frac{1}{2} \sum_{m=1}^M \frac{\|\boldsymbol{f}_m^{\mathrm{MAP}}\|^2_{\boldsymbol{K}_m}}{d_m} + \frac{1}{2} \log \left| \sigma_y^2 \boldsymbol{I}_N + \sum_{m=1}^M d_m \boldsymbol{K}_m \right|,$$

where $\boldsymbol{f}_m^{\mathrm{MAP}}$ is the minimizer of the right-hand side of Eq. (11); note that this minimization is a fixed kernel weight learning problem (2).

Taking the derivative of the above expression with respect to d_m we have

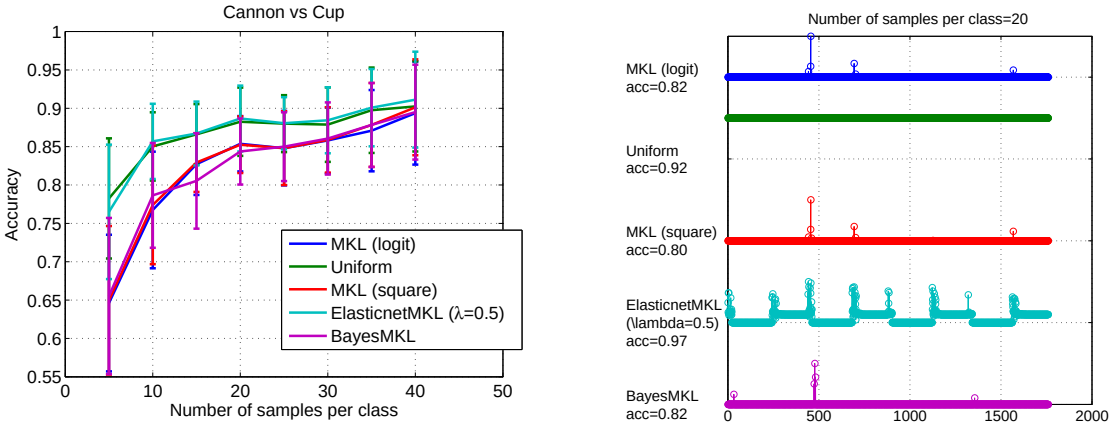
$$-\frac{\|\boldsymbol{f}_m^{\mathrm{MAP}}\|^2_{\boldsymbol{K}_m}}{d_m^2} + \operatorname{Tr}\left((\sigma^2 \boldsymbol{I}_N + \textstyle\sum_{m=1}^M d_m \boldsymbol{K}_m)^{-1} \boldsymbol{K}_m \right) = 0.$$

Therefore, we use the following iteration:

$$(\boldsymbol{f}_m)_{m=1}^M \leftarrow \operatorname*{argmin}_{(\boldsymbol{f}_m)_{m=1}^M} \left(\frac{1}{2\sigma_y^2} \left\| \boldsymbol{y} - \textstyle\sum_{m=1}^M \boldsymbol{f}_m \right\|^2 + \frac{1}{2} \textstyle\sum_{m=1}^M \dfrac{\|\boldsymbol{f}_m\|^2_{\boldsymbol{K}_m}}{d_m} \right) \tag{13}$$

$$d_m \leftarrow \frac{\|\boldsymbol{f}_m\|^2_{\boldsymbol{K}_m}}{\operatorname{Tr}\left((\sigma^2 \boldsymbol{I}_N + \sum_{m=1}^M d_m \boldsymbol{K}_m)^{-1} d_m \boldsymbol{K}_m \right)}. \tag{14}$$

The convergence of this procedure is not established mathematically, but it is known to converge rapidly in many practical situations [22].



(a) Accuracy averaged over 40 train/test splits (b) Obtained kernel weights

Figure 1: Caltech 101 dataset.

5 Numerical experiments

Figure 1 shows the result of applying different MKL algorithms on a binary classification task (Cannon vs Cup) from the Caltech 101 dataset [8]. We have generate 1760 kernel functions by combining four SIFT features, 22 spacial decompositions (including the spatial pyramid kernel), two kernel functions, and 10 kernel parameters. See [23] for more details[1].

In order to make the comparison between the Bayesian and non-Bayesian MKL methods easy, we use the squared loss for all MKL algorithms. We also included the block 1-norm MKL with the logistic loss ("MKL (logit)"). Since the difference between MKL (logit) and MKL (square) is small, we expect that the discussion here is not specific to the squared loss. For the Elastic-net MKL (8), we fix the constant λ as $\lambda = 0.5$. For the empirical Bayesian MKL, we use the MacKay update (13)-(14). The regularization constant C was chosen by 2×4-fold cross validation on the training-set for each method.

From Fig. 1(a), we can see that Elastic-net MKL and uniformly-weighted MKL perform clearly better than other MKL methods. empirical Bayesian MKL seems to be slightly worse than block 1-norm MKL when the number of samples per class is smaller than 20. Although Elastic-net MKL performs almost the same as uniform MKL in terms of accuracy, Fig. 1(b) shows that Elastic-net can find important kernel components automatically. More specifically, Elastic-net MKL chose 88 Gaussian RBF kernel functions and 792 χ^2 kernel functions. Thus it prefers χ^2 kernels to Gaussian RBF kernels. This agrees with the common choice in CV literature. In addition, Elastic-net MKL consistently chose the band width parameter $\gamma = 0.1$ for the Gaussian RBF kernels but it never chose $\gamma = 0.1$ for the χ^2 kernels; instead it averaged all χ^2 kernels from $\gamma = 1.2$ to $\gamma = 10$.

6 Summary

We have shown that various MKL algorithms including ℓ_p-norm MKL and Elastic-net MKL can be seen as applications of different regularization strategies. Extending the arguments in Kloft et al. [11], we have shown the exact correspondence between the Ivanov regularization and Tikhonov regularization, thus rejected the false rumour that the Tikhonov regularization has more tuning parameters than the Ivanov regularization. Moreover, we have presented a generalized block-norm formulation that uses a concave function and shown how it corresponds to Ivanov and Tikhonov regularizations with a general convex increasing regularizer; see Table 1. The Tikhonov regularization-based formulation allows us to view MKL as a hierarchical Gaussian process model. Motivated by this view, we proposed two iterative algorithms for the maximization of marginalized likelihood; one of them iteratively solves a reweighted block 1-norm MKL and the other solves a fixed kernel weight

[1] Preprocessed data is available from `http://www.ibis.t.u-tokyo.ac.jp/ryotat/prmu09/data/`.

learning problem. A preliminary experiment on a visual categorization task from Caltech 101 with 1760 kernels has shown that Elastic-net MKL can achieve comparable classification accuracy to uniform kernel combination with roughly half of the candidate kernels and provide information about the usefulness of the candidate kernels. Further analysis and empirical validation are necessary to gain more insights about the empirical Bayesian learning procedure.

Acknowledgement

We would like to thank Hisashi Kashima and Shinichi Nakajima for helpful discussions. This work was partially supported by MEXT Kakenhi 22700138, 22700289.

References

[1] J. Aflalo, A. Ben-Tal, C. Bhattacharyya, J. S. Nath, and S. Raman. Variable sparsity kernel learning — algorithms and applications. *J. Mach. Learn. Res. (submitted)*, 2009.

[2] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[3] F. Bach, G. Lanckriet, and M. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *the 21st International Conference on Machine Learning*, pages 41–48, 2004.

[4] F. R. Bach, R. Thibaux, and M. I. Jordan. Computing regularization paths for learning multiple kernels. In *Advances in Neural Information Processing Systems 17*, pages 73–80. MIT Press, 2005.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

[6] Olivier Chapelle and Alain Rakotomamonjy. Second order optimization of kernel parameters. In *NIPS 2008 Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, Whistler, 2008.

[7] T. Damoulas and M. A. Girolami. Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection. *Bioinformatics*, 24(10):1264–1270, 2008.

[8] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *IEEE. CVPR 2004 Workshop on Generative-Model Based Vision*, 2004.

[9] M. Girolami and S. Rogers. Hierarchic bayesian models for kernel learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 241–248. ACM, 2005.

[10] M. Kloft, U. Rückert, and P. L. Bartlett. A unifying view of multiple kernel learning. In *Proc. ECML 2010*, 2010.

[11] Marius Kloft, Ulf Brefeld, Soeren Sonnenburg, Pavel Laskov, Klaus-Robert Müller, and Alexander Zien. Efficient and accurate lp-norm multiple kernel learning. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 997–1005. 2009.

[12] G. Lanckriet, N. Cristianini, L. El Ghaoui, P. Bartlett, and M. Jordan. Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[13] D. J. C. MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

[14] C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005.

[15] Jason Palmer, David Wipf, Kenneth Kreutz-Delgado, and Bhaskar Rao. Variational em algorithms for non-gaussian latent variable models. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1059–1066. MIT Press, Cambridge, MA, 2006.

[16] A. Rakotomamonjy, F. Bach, S. Canu, and Grandvalet Y. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[17] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[18] Bernhard Schölkopf and Alex Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA, 2002.

[19] M. Seeger and H. Nickisch. Large scale variational inference and experimental design for sparse generalized linear models. Technical report, arXiv:0810.0901, 2008.

[20] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

[21] Taiji Suzuki and Ryota Tomioka. SpicyMKL. Technical report, arXiv:0909.5026, 2009.

[22] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, 2001.

[23] Ryota Tomioka and Taiji Suzuki. Sparsity-accuracy trade-off in MKL. Technical report, arXiv:1001.2615, 2010.

[24] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *IEEE 11th International Conference on Computer Vision (ICCV)*, pages 1–8. 2007.

[25] D. Wipf and S. Nagarajan. A new view of automatic relevance determination. In *Advances in NIPS 20*, pages 1625–1632. MIT Press, 2008.

[26] D. Wipf and S. Nagarajan. A unified bayesian framework for meg/eeg source imaging. *NeuroImage*, 44(3):947–966, 2009.

[27] Z. Zhang, D.Y. Yeung, and J.T. Kwok. Bayesian inference for transductive learning of kernel matrix using the tanner-wong data augmentation algorithm. In *Proceedings of the Twenty-First International Conference on Machine Learning*, page 118. ACM, 2004.

[28] A. Zien and C.S. Ong. Multiclass multiple kernel learning. In *Proceedings of the 24th international conference on machine learning*, pages 11910–1198. ACM, 2007.

A A representer theorem for the fixed kernel weight learning problem (2)

The representer theorem [18] holds for the learning problem (2), and importantly, the expansion coefficients are the same for all functions f_m (except the kernel weight d_m). In order to see this, we take the Fréchet derivative of the objective (2) and set it to zero as follows:

$$\left\langle h_m,\, -\sum_{i=1}^N \alpha_i k_m(\cdot, x_i) + C\frac{f_m}{d_m} \right\rangle_{\mathcal{H}_m} = 0 \qquad (\forall h_m \in \mathcal{H}_m, \forall m),$$

$$\sum_{i=1}^N \alpha_i = 0,$$

$$\partial\ell(y_i, \textstyle\sum_{m=1}^M f_m(x_i) + b) \ni -\alpha_i \qquad (i = 1, \ldots, N),$$

where $\partial\ell$ denotes the subdifferential of the loss function ℓ with respect to the second argument. From the first equation, we have the kernel expansion

$$f_m(x) = \frac{d_m}{C}\sum_{i=1}^N \alpha_i k_m(x, x_i) \qquad (m = 1, \ldots, M),$$

from which the overall predictor can be written as follows:

$$\bar{f}(x) + b = \frac{1}{C}\sum_{i=1}^N \alpha_i \sum_{m=1}^M d_m k_m(x, x_i) + b.$$

B Proof of Eq. (1) in a finite dimensional case

In this section, we provide a proof of Eq. (1) when $\mathcal{H}_1, \ldots, \mathcal{H}_m$ are all finite dimensional. We assume that the input space $\mathcal{X}$ consists of N points $x_1, \ldots, x_N$, for example the training points. The function $f_m \in \mathcal{H}_m$ is completely specified by the function values at the N-points $\boldsymbol{f}_m = (f_m(x_1), \ldots, f_m(x_N))^\top$. The kernel function k_m is also specified by the Gram matrix $\boldsymbol{K}_m = (k_m(x_i, x_j))_{i,j=1}^N$. The inner product $\langle f_m, g_m\rangle_{\mathcal{H}_m}$ is written as $\langle f_m, g_m\rangle_{\mathcal{H}_m} = \boldsymbol{f}_m^\top \boldsymbol{K}_m^{-1}\boldsymbol{g}_m$, where $\boldsymbol{g}_m$ is the N-dimensional vector representation of $g_m \in \mathcal{H}_m$, assuming that the Gram matrix $\boldsymbol{K}_m$ is positive definite. It is easy to check the reproducibility; in fact, $\langle f_m, k_m(\cdot, x_i)\rangle = \boldsymbol{f}_m^\top \boldsymbol{K}_m^{-1}\boldsymbol{K}_m(:, i) = f(x_i)$, where $\boldsymbol{K}_m(:, i)$ is a column vector of the Gram matrix $\boldsymbol{K}_m$ that corresponds to the ith sample point x_i.

The right-hand side of Eq. (1) is written as follows:

$$\min_{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M \in \mathbb{R}^N} \sum_{m=1}^M \frac{\boldsymbol{f}_m^\top \boldsymbol{K}_m^{-1}\boldsymbol{f}_m}{d_m} \quad \text{s.t.} \quad \sum_{m=1}^M \boldsymbol{f}_m = \bar{\boldsymbol{f}}.$$

Forming the Lagrangian, we have

$$\begin{aligned}
&\sum_{m=1}^{M} \frac{\boldsymbol{f}_m^\top \boldsymbol{K}_m^{-1} \boldsymbol{f}_m}{d_m} \\
&= \sum_{m=1}^{M} \frac{\boldsymbol{f}_m^\top \boldsymbol{K}_m^{-1} \boldsymbol{f}_m}{d_m} + 2\boldsymbol{\alpha}^\top \left(\bar{\boldsymbol{f}} - \textstyle\sum_{m=1}^{M} \boldsymbol{f}_m \right) \\
&\geq -\boldsymbol{\alpha}^\top \left(\textstyle\sum_{m=1}^{M} d_m \boldsymbol{K}_m \right) \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \bar{\boldsymbol{f}} \\
&\xrightarrow{\max_{\boldsymbol{\alpha}}} \bar{\boldsymbol{f}}^\top \left(\textstyle\sum_{m=1}^{M} d_m \boldsymbol{K}_m \right)^{-1} \bar{\boldsymbol{f}},
\end{aligned}$$

where the equality is obtained for

$$\boldsymbol{f}_m = d_m \boldsymbol{K}_m \left(\textstyle\sum_{m=1}^{M} d_m \boldsymbol{K}_m \right)^{-1} \bar{\boldsymbol{f}}.$$

C Ivanov regularization

Another way to penalize the complexity is to enforce some constraint on the kernel weights for the minimization of the objective (2) as follows (see [3, 20, 28, 16]):

$$\underset{\substack{f_1 \in \mathcal{H}_1, \ldots, f_M \in \mathcal{H}_M, \\ b \in \mathbb{R}, \\ d_1 \geq 0, \ldots, d_M \geq 0}}{\text{minimize}} \sum_{i=1}^{N} \ell \left(y_i, \textstyle\sum_{m=1}^{M} f_m(x_i) + b \right) + \frac{\tilde{C}}{2} \sum_{m=1}^{M} \frac{\|f_m\|_{\mathcal{H}_m}^2}{d_m} \quad \text{s.t.} \qquad \sum_{m=1}^{M} h(d_m) \leq 1, \tag{15}$$

where $h(d_m)$ is a convex increasing function over the nonnegative reals. For example, the ℓ_p-norm MKL can be obtained by choosing the regularizer $h(d_m)$ as $h(d_m) = d_m^p$; see [11, 14].

In order to obtain the Ivanov regularization problem (15) corresponding to the elastic-net regularization (8), we need to identify the function h (without the constant μ). Choosing $h(d_m) = (1 - \tilde{\lambda}) d_m / (1 - \tilde{\lambda} d_m)$ (note that $\tilde{\lambda}$ is different from λ), the regularization term in the Ivanov regularization problem (15) can be written as

$$\begin{aligned}
\sum_{m=1}^{M} \frac{\|f_m\|_{\mathcal{H}}^2}{d_m} &= \sum_{m=1}^{M} \frac{1 - \tilde{\lambda} d_m + \tilde{\lambda} d_m}{d_m} \|f_m\|_{\mathcal{H}}^2 \\
&= \sum_{m=1}^{M} \left(\frac{1 - \tilde{\lambda}}{h(d_m)} + \tilde{\lambda} \right) \|f_m\|_{\mathcal{H}}^2 \\
&\geq (1 - \tilde{\lambda}) \left(\textstyle\sum_{m=1}^{M} \|f_m\|_{\mathcal{H}} \right)^2 + \tilde{\lambda} \textstyle\sum_{m=1}^{M} \|f_m\|_{\mathcal{H}}^2,
\end{aligned}$$

where we used Jensen's inequality in the last line. The Ivanov regularization problem (15) with the above regularizer $h(d_m)$ is equivalent to the elastic-net problem (8) by suitably converting the pair (C, λ) and $(\tilde{C}, \tilde{\lambda})$.

D Derivation of the block q-norm regularization from the Tikhonov regularization (3)

We choose the regularizer $h(d_m)$ as $h(d_m) = d_m^p$ and $\mu = 1/p$. Then,

$$\begin{aligned}
\frac{\|f_m\|_{\mathcal{H}_m}^2}{d_m} + \frac{1}{p} d_m^p &= \frac{1+p}{p} \left(\frac{p}{1+p} \frac{\|f_m\|_{\mathcal{H}_m}^2}{d_m} + \frac{1}{1+p} d_m^p \right) \\
&\geq \frac{1+p}{p} \|f_m\|_{\mathcal{H}_m}^{2p/(1+p)} = \frac{1+p}{p} \|f_m\|_{\mathcal{H}_m}^q,
\end{aligned}$$

where we used Young's inequality, which is the inequality of arithmetic and geometric means when $p = 1$; the equality is obtained by taking $d_m = \|f_m\|_{\mathcal{H}_m}^{2/(1+p)}$. The resulting block-norm formulation can be written as follows:

$$\underset{f_1 \in \mathcal{H}_1, \ldots, f_M \in \mathcal{H}_M, b \in \mathbb{R}}{\text{minimize}} \sum_{i=1}^{N} \ell\left(y_i, \sum_{m=1}^{M} f_m(x_i) + b\right) + \frac{C}{q} \sum\nolimits_{m=1}^{M} \|f_m\|_{\mathcal{H}_m}^q, \tag{16}$$

where we define $q = 2p/(1+p)$. Clearly, when $q = 1$ ($p = 1$), Eq. (16) reduces to the block 1-norm MKL (4).

Let us consider the block q-norm MKL for $q > 2$ of Aflalo et al. [1] in the Tikhonov regularization framework. Aflalo et al.'s approach can be interpreted as a nonconvex regularization on the kernel weights. The easiest way to see this is to extrapolate the mapping between p and q also for $q > 2$, which gives the regularization term $\mu h(d_m)$ as follows:

$$\mu h(d_m) = -\frac{q-2}{q} d_m^{-q/(q-2)}. \tag{17}$$

This is a concave increasing function. Young's inequality cannot be used to see how the above regularizer (17) is related to the block q-norm regularization, because $p = -q/(q-2)$ is negative. However, by explicitly computing the minimum, we have for $2 < q < \infty$,

$$\frac{\|f_m\|_{\mathcal{H}_m}}{d_m} - \frac{q-2}{q} d_m^{-q/(q-2)} \geq \frac{2}{q} \|f_m\|_{\mathcal{H}_m}^q,$$

where the minimum is obtained for $d_m = \|f_m\|_{\mathcal{H}_m}^{2-q}$.