Non-sparse Multiple Kernel Learning

Marius Kloft[†], **Ulf Brefeld**[†], **Pavel Laskov**[‡], **Sören Sonnenburg**[‡]
[†] TU Berlin, Franklinstr. 28/29, 10587 Berlin
[‡] Fraunhofer Institute FIRST, Kekuléstr. 7, 12489 Berlin

Abstract

Approaches to multiple kernel learning (MKL) employ ℓ_1-norm constraints on the mixing coefficients to promote sparse kernel combinations. When features encode orthogonal characterizations of a problem, sparseness may lead to discarding useful information and may thus result in poor generalization performance. We study non-sparse multiple kernel learning by imposing an ℓ_2-norm constraint on the mixing coefficients. Empirically, ℓ_2-MKL proves robust against noisy and redundant feature sets and significantly improves the promoter detection rate compared to ℓ_1-norm and canonical MKL on large scales.

1 Introduction

A natural way to an automatic selection of optimal kernels is to learn a linear combination $K = \sum_{j=1}^{m} \beta_j \, K_j$ with mixing coefficients $\boldsymbol{\beta}$ together with the model parameters. This framework, known as multiple kernel learning (MKL), was first introduced by [2] where two kinds of constraints on $\boldsymbol{\beta}$ and K have been considered leading to either semi-definite programming or QCQP approaches, respectively. The SDP approach was also shown to be equivalent to sparse regularization over $\boldsymbol{\beta}$ by means of a standard simplex constraint $||\boldsymbol{\beta}||_1 = 1$.

Intuitively, sparseness of $\boldsymbol{\beta}$ makes sense when the expected number of meaningfull kernels is small. Requiring that only a small number of features contributes to the final kernel implicitly assumes that most of the features to be selected are equally informative. In other words, sparseness is good when the kernels already contain a couple of good features that alone capture almost all of the characteristic traits of the problem. This also implies that features are highly redundant. However, when features inherently encode “orthogonal” characterizations of a problem, enforcing sparseness may lead to discarding useful information and as a result, degradation of generalization performance.

We develop a *non-sparse* MKL, in which the ℓ_1-norm in the regularization constraint on $\boldsymbol{\beta}$ is replaced with the ℓ_2-norm. Although the constraint $||\boldsymbol{\beta}||_2 = 1$ is non-convex, a tight convex approximation can be obtained whose solution is always attained at the boundary $||\boldsymbol{\beta}||_2 = 1$, provided that kernel matrices are strictly positive definite. We develop a semi-infinite programming (SIP) formulation of non-sparse MKL. Our method proves robust against noisy and non-redundant feature sets. Large-scale experiments on promoter detection show a moderate but significant improvement of predictive accuracy compared to ℓ_1 and canonical MKL.

2 Non-sparse Learning with Multiple Kernels

We focus on binary classification problems where we are given labeled data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1\ldots,n}$, where $\boldsymbol{x} \in \mathcal{X}$ for some input space $\mathcal{X}$, and where $y \in \{+1, -1\}$. When learning with multiple kernels, we are additionally given p different feature mappings $\psi_1, \ldots, \psi_p$. Every mapping $\psi_j : \mathcal{X} \rightarrow \mathcal{H}_j$ gives rise to a reproducing kernel k_j of $\mathcal{H}_j$ given by $k_j(\boldsymbol{x}, \bar{\boldsymbol{x}}) = \langle \psi_j(\boldsymbol{x}), \psi_j(\bar{\boldsymbol{x}}) \rangle_{\mathcal{H}_j}$. In the remainder we will use ψ_j, k_j, and matrix $K_j = (k_j(\boldsymbol{x}_i, \boldsymbol{x}_m))_{i,m=1,\ldots,n}$ interchangeably for convenience. We now aim at finding a linear combination $\sum_{j=1}^{p} \beta_j K_j$ and parameters $\boldsymbol{w}, b$

simultaneously, such that the resulting hypothesis f has a small expected risk, where f is given by

$$f(\boldsymbol{x}) \;=\; \sum_{k=1}^{p} \sqrt{\beta_j}\, \boldsymbol{w}_j' \psi_j(\boldsymbol{x}) + b \;=\; \boldsymbol{w}' \psi_{\boldsymbol{\beta}}(\boldsymbol{x}) + b, \tag{1}$$

where $\boldsymbol{w} = (\boldsymbol{w}_j)_{k=1,\ldots,p}$, $\psi_{\boldsymbol{\beta}}(\boldsymbol{x}_i) = (\sqrt{\beta_j}\psi_j(\boldsymbol{x}_i))_{j=1,\ldots,p}$, and mixing coefficients $\beta_j \geq 0$.

Common approaches to multiple kernel learning impose ℓ_1-norm constraints on the mixing coefficients [1, 3] thus promoting sparse solutions lying on a standard simplex. By contrast, we aim at studying non-sparse multiple kernel learning, that is we employ an ℓ_2 regularization to allow for non-sparse kernel mixtures. The primal optimization problem can be stated as Given data $\mathcal{D}$, feature mappings $\psi_1, \ldots, \psi_p$, and $\eta > 0$.

$$\min_{\boldsymbol{\beta},\boldsymbol{w},b,\boldsymbol{\xi}} \; \frac{1}{2}\boldsymbol{w}'\boldsymbol{w} + \eta\|\boldsymbol{\xi}\|_1 \;\text{ s.t. }\; \forall_{i=1}^{n}:\; y_i\left(\boldsymbol{w}'\psi_{\boldsymbol{\beta}}(\boldsymbol{x}_i) + b\right) \geq 1 - \xi_i;\; \boldsymbol{\xi} \geq \mathbf{0}\,;\; \boldsymbol{\beta} \geq \mathbf{0}\,;\; \|\boldsymbol{\beta}\|_2 = 1.$$

The optimization problem is inherently non-convex since the boundary of the unit ball given by $\{\boldsymbol{\beta} : \|\boldsymbol{\beta}\|_2 = 1\}$ is not a convex set. As a remedy, we relax the constraint on $\boldsymbol{\beta}$ to become an inequality constraint, i.e., $\|\boldsymbol{\beta}\|_2 \leq 1$. We will later show that the resulting approximation error is zero under reasonable assumptions. Another non-convexity is caused by the products $\beta_j \boldsymbol{w}_j$ which, however, can be easily removed by a variable substitution $\boldsymbol{v}_j := \beta_j \boldsymbol{w}_j$. We arrive at the following optimization problem which is convex.

$$\min_{\boldsymbol{\beta},\boldsymbol{v},b,\boldsymbol{\xi}} \; \frac{1}{2}\sum_{j=1}^{p} \frac{\boldsymbol{v}_j'\boldsymbol{v}_j}{\boldsymbol{\beta}_j} + \eta\|\boldsymbol{\xi}\|_1 \;\text{ s.t. }\; \forall_{i=1}^{n} : y_i\Big(\sum_{j=1}^{p} \boldsymbol{v}_j'\psi_j(\boldsymbol{x}_i) + b\Big) \geq 1 - \xi_i;\; \boldsymbol{\xi},\boldsymbol{\beta} \geq \mathbf{0};\; \|\boldsymbol{\beta}\|_2 = 1.$$

Fixing $\boldsymbol{\beta} \in \Lambda$, where $\Lambda = \{\boldsymbol{\beta} \in \mathbb{R}^n \mid \boldsymbol{\beta} \geq \mathbf{0}\;,\;\; \|\boldsymbol{\beta}\|_2 \leq 1\}$, we build the partial Lagrangian with respect to $\boldsymbol{v}$, b, and $\boldsymbol{\xi}$. Setting the partial derivatives of the Lagrangian with respect to the primal variables to zero yields the relations $0 \leq \alpha_i \leq \eta$, $\sum_i \alpha_i y_i = 0$, and $\boldsymbol{v}_j = \sum_i \alpha_i y_i \beta_j \psi_j(\boldsymbol{x}_i)$ for $1 \leq i \leq n$ and $1 \leq j \leq p$. The KKT conditions trivially hold and resubstitution gives rise to the min-max formulation

$$\min_{\boldsymbol{\beta}\geq\mathbf{0}} \; \max_{\mathbf{0}\leq\boldsymbol{\alpha}\leq\mathbf{1}\eta} \; \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,m=1}^{n}\alpha_i\alpha_m y_i y_m \sum_{j=1}^{p}\beta_j k_j(\boldsymbol{x}_i,\boldsymbol{x}_m) \;\text{ s.t. }\; \sum_{i=1}^{n} y_i\alpha_i = 0;\; \|\boldsymbol{\beta}\|_2 \leq 1.$$

The above problem can either be solved directly by gradient-based techniques exploiting the smoothness of the objective [1] or translated into an equivalent semi-infinite program (SIP) as follows. Suppose $\boldsymbol{\alpha}^*$ is optimal, then denoting the value of the target function by $t(\boldsymbol{\alpha},\boldsymbol{\beta})$, we have $t(\boldsymbol{\alpha}^*,\boldsymbol{\beta}) \geq t(\boldsymbol{\alpha},\boldsymbol{\beta})$ for all $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Hence we can equivalently minimize an upper bound Θ on the optimal value. We thus arrive at Optimization Problem 1.

Optimization Problem 1 (SIP) *Let* $Q_j = YK_jY$ *for all* $1 \leq j \leq p$ *where* $Y = \operatorname{diag}(y)$,

$$\min_{\Theta,\boldsymbol{\beta}} \;\Theta \quad \textit{s.t.} \quad \Theta \geq \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}'\sum_{j=1}^{p}\beta_j Q_j \boldsymbol{\alpha}; \quad \|\boldsymbol{\beta}\| \leq 1; \quad \boldsymbol{\beta} \geq \mathbf{0}$$
$$\forall\boldsymbol{\alpha} \in \mathbb{R}^n \quad \textit{with} \quad \mathbf{1}'\boldsymbol{\alpha} \leq \eta\mathbf{1}' \quad \textit{and} \quad \boldsymbol{y}'\boldsymbol{\alpha} = 0 \quad \textit{as well as} \quad \boldsymbol{\alpha} \geq \mathbf{0}.$$

Note, that the above SIP is only a relaxation of the primal problem. However, Theorem 1 shows that the approximation error is zero if the employed kernel functions are positive definite.

Theorem 1 *Let* $(\Theta^*, \boldsymbol{\beta}^*)$ *be optimal points of Optimization Problem 1 and* $K_1, \ldots, K_p$ *be positive definite. Then we always have* $\|\boldsymbol{\beta}^*\|_2 = 1$. *(Proof omitted for lack of space)*

3 Discussion

The SIP in Optimization Problem 1 can be efficiently solved by interleaving cutting plane algorithms. The solution of a quadratic program (here the regular SVM) generates the most strongly violated constraint for the actual mixture $\boldsymbol{\beta}$. The optimal $(\boldsymbol{\beta}^*, \Theta)$ is then computed by solving a quadratically constrained program (QCP) with respect to set of active constraints. The described

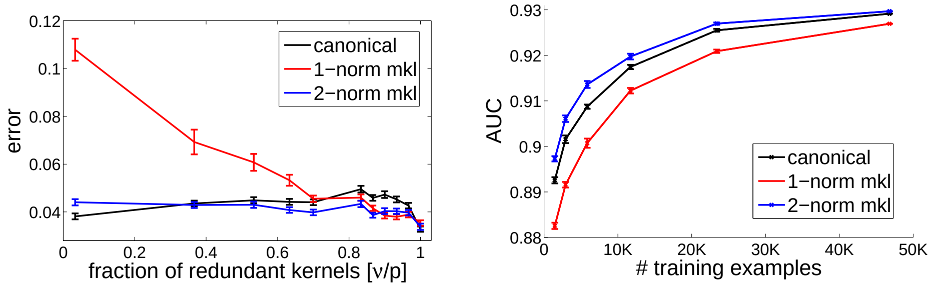

Figure 1: Left: Test errors for the artificial data set. Right: Results for the real-world experiment.

algorithm is a special case of SIP algorithms known as *exchange methods*. Exchange methods are known to converge if the feasible region can be covered by a ball with finite radius $r > 0$. However, no convergence rate for such algorithms are known. Recently, promising alternative strategies for optimizing the ℓ_1-MKL, based on gradient-based [3] and level-set [7] optimization, have been proposed.

Obviously, the regular support vector machine is contained as a special case for learning with only one kernel (i.e., $p = 1$). Moreover, our approach can be easily extended to a one-class setting when the kernel matrices are appropriately normalized (Section 4.2). Our approach is moreover contained in [6] as a special case for $p = 0, q = 1$, however, their approach is not discussed or evaluated for these parameters settings.

4 Empirical Results

4.1 Toy 1: Measuring the Impact of Redundant Kernels

The first experiment investigates the strengths and weaknesses of the canonical kernel combination, ℓ_1- and ℓ_2-MKL for different "levels of independence" of the kernel matrices.

The aim of the following procedure is to generate a fixed number of p Kernel matrices, where the degree of independence is parameterized by ν. To this end we generate a $d-$dimensional sample of size n from two Gaussian distributions with $\Sigma = I$. We decompose the n examples into ν disjoint feature sets $X_1, \ldots, X_\nu$, where $X_i \in \mathbb{R}^{\frac{d}{\nu}\times n}$, $\forall i = 1 \ldots \nu$. Then we sample $p - \nu$ copies from these feature sets, by randomly picking one by one from $X_1, \ldots, X_\nu$ with replacement[1]. For each of these p sets we randomly generate a linear transformation matrix $A_1, \ldots, A_n$ with $A_i \in \mathbb{R}^{\tau\frac{d}{\nu}\times\frac{d}{\nu}}$. Finally the kernel matrices are computed as $K_i = X_i' A_i' A_i X_i$. The randomization not only alters the attribute sets that would otherwise be identical but also enriches the dimensionality of the X_i by a factor τ. Using varying values for ν allows us to generate kernel matrices for different "levels of independence".

Throughout the experiment we fix $d = 60$, $p = 30$, and $\tau = 4$. For each value of $\nu \in \{1, 2, 3, 4, 6, 8, 12, 15, 20, 30\}$, we generate a sample of size 900 encoded in the p kernel matrices using the procedure above. The matrices are then equally split into training, validation, and test kernel matrices. We compare the performance of ℓ_1-MKL and ℓ_2-MKL with a baseline SVM using the canonical mixture kernel $K = \frac{1}{p}\sum_{j=1}^{p} K_j$. Optimal soft-margin parameters $\eta \in [0.001, 10]$ are determined using the validation set. We report on averaged test errors of 100 repetitions of this procedure; error bars indicate standard errors. Note that for each repetition the kernel matrices are generated from scratch. All matrices are normalized according to $k(\boldsymbol{x}, \bar{\boldsymbol{x}}) \mapsto k(\boldsymbol{x}, \bar{\boldsymbol{x}})/(\frac{1}{n}\sum_{i=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_i) - \frac{1}{n^2}\sum_{i,j=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_j))$.

The results are shown in Figure 1 (left). The x-axis depicts the ratio of information carrying kernels given by ν/p. Obviously, ℓ_1-MKL performs best when the relevant information is contained in only a few kernels. However, its performance deteriorates quickly with a decrease in redundancy. In the

[1] That is, feature sets can be picked multiple times.

extreme, where all relevant information is spread uniformly among the kernels such that there is no redundant information shared, the canonical mixture intuitively represents the optimal kernel.

With increasing redundancy, ℓ_2-MKL outperforms the canonical mixture that now incorporates more and more information that is either already contained in other kernels or irrelevant noise. By contrast, ℓ_2-MKL effectively determines appropriate kernel mixtures for all redundancy ratios. In the other extreme, where all kernel matrices encode the full knowledge about the data, all methods perform equally well and effectively counterbalance the random linear transformations by ensemble-effects.

4.2 Real World: Identifying Transcription Start Sites

This task on real-world data aims at detecting transcription start sites (TSS) of RNA Polymerase II binding genes in genomic DNA sequences. In general, transcription start site finders exploit that the features of promoter regions and the transcription start sites are different from features of other genomic DNA. Many such detectors thereby rely on a combination of feature sets which makes the learning task appealing for MKL.

For our experiments we use the dataset from [4] which contains a curated set of 8508 TSS annotated genes utilizing dbTSS version 4 [5] and refseq genes. These are translated into positive training instances by extracting windows of size $[-1000, +1000]$ around the TSS. From the interior of the gene 85042 negative instances are generated using the same window size. We employ five different kernels representing the TSS signal (weighted degree with shift), the promoter (spectrum), the 1st exon (spectrum), angles (linear), and energies (linear). Kernel parameters are specified according to prior knowledge or intuition and are reported in [4]. Every kernel is normalized according to $k(\boldsymbol{x}, \bar{\boldsymbol{x}}) \longmapsto k(\boldsymbol{x}, \bar{\boldsymbol{x}}) / \sqrt{k(\boldsymbol{x}, \boldsymbol{x}) k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}})}$.

As in [4], our training sets consist of 46794 instances, and the remaining 46756 examples are split into fixed tuning $(1/3)$ and test $(2/3)$ sets. Model selection is performed for $\eta \in \{2^{-2.5}, 2^{-2}, \ldots, 2^{2.5}\}$. We report on average AUC values over 10 repetitions with randomly drawn training instances; error bars indicate standard error. The results for varying training set sizes are shown in Figure 1 (right). The sparse mixture found by ℓ_1-norm MKL performs worst and is clearly outperformed by a canonical mixture for all sample sizes. By contrast, ℓ_2-MKL effectively learns a non-sparse kernel mixture and leads to significantly higher detection rates compared to the canonical mixture for all but the rightmost point. Non-sparse MKL outperforms its classical ℓ_1-norm counterpart significantly for all sample sizes.

5 Conclusions

We studied a non-sparse approach to multiple kernel learning (MKL). Our approach is motivated by the observation that sparseness may not always be desirable for a combination of multiple kernels. Large scale experiments on finding transcription start sites revealed the effectiveness of ℓ_2-MKL in the case where ℓ_1-MKL was even outperformed by a canonical mixture. The ℓ_2-MKL achieved the highest predictive performance in our experiments.

References

[1] F. Bach, G. Lanckriet, and M. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

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

[3] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. In *ICML*, pages 775–782, 2007.

[4] S. Sonnenburg, A. Zien, and G. Rätsch. ARTS: Accurate Recognition of Transcription Starts in Human. *Bioinformatics*, 22(14):e472–e480, 2006.

[5] Y. Suzuki, R. Yamashita, K. Nakai, and S. Sugano. dbTSS: Database of human transcriptional start sites and full-length cDNAs. *Nucleic Acids Research*, 30(1):328–331, 2002.

[6] M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy. Composite kernel learning. In *ICML*, 2008.

[7] Z. Xu, R. Jin, I. King, and M. R. Lyu. An extended level method for efficient multiple kernel learning. In *Advances in Neural Information Processing Systems*, 2009, to appear.