ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

The Minimum Effort Maximum Output Principle applied to Multiple Kernel Learning

Ivano Lauriola*, Mirko Polato and Fabio Aiolli

University of Padova - Department of Mathematics
Via Trieste 63, Padova - Italy

Abstract. The Multiple Kernel Learning (MKL) paradigm aims at learning the representation from data reducing the effort devoted to the choice of the kernel's hyperparameters. Typically, the resulting kernel is obtained as the maximal margin combination of a set of base kernels. When too expressive base kernels are provided to the MKL algorithm, the solution found by these algorithms can overfit data. In this paper, we propose a novel MKL algorithm which takes into consideration the expressiveness of the obtained representation in its objective function in such a way that a trade-off between large margins and simple hypothesis spaces can be found. Moreover, an empirical comparison against hard baselines and state-of-the-art MKL methods on several real-world datasets is presented showing the merits of the proposed algorithm especially with respect to the robustness to overfitting.

1 Introduction

Kernel methods are recognized state-of-the-art methods for classification. They rely on the concept of kernel which implicitly defines the data representation. However, choosing the right kernel for a given problem remains an hard task. In a typical learning pipeline the user tries several kernels with different values of the hyperparameters, guided by some prior knowledge or via a validation procedure. This process is computationally expensive when the number of possible values for the kernels hyperparameters is large. To overcome this issue, methods to directly learn kernels from data have been recently proposed. Multiple Kernel Learning (MKL) [1] is one of the most popular frameworks for kernel learning [2, 3, 4, 5, 6]. The idea behind these methods is to combine a set of base kernels. This is usually done by finding the kernel combination that maximizes the margin in feature space. However, the complexity (or expressiveness) of the induced representation is generally neglected, with the risk of obtaining a too complex representation for the task at hand. To overcome this problem, more recent approaches tried to regularize the combination, for instance, by considering the radius of the Minimum Enclosing Ball (MEB) in their objective function [7, 8]. The main contribution of this paper is the proposal of a novel MKL algorithm, here dubbed MEMO-MKL (Minimum Effort Maximum Output MKL), which finds a combination of base kernels trading-off between margin maximization and low expressiveness of the resulting representation. An extensive experimental assessment has been performed comparing the proposed method against strong baselines/state-of-the-art MKL methods on several benchmark datasets.

*The PhD grant of Ivano Lauriola is funded by the Bruno Kessler Foundation.

2 Background and notation

We focus on binary classification problems. Let $\mathcal{D} \equiv \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ be the training set where $\mathbf{x}_i \in \mathbb{R}^n$ are n-dimensional input vectors and $y_i \in \{+1, -1\}$ are the associated labels. A kernel function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defines the dot-product in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$, by means of the function $\phi : \mathcal{X} \to \mathcal{H}$, which maps vectors from the input space $\mathcal{X}$ onto the embedding (or feature) space $\mathcal{H}$, such that $\kappa(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$ for two generic input vectors $\mathbf{x}$ and $\mathbf{z}$. The Kernel (or *Gram*) matrix $\mathbf{K} \in \mathbb{R}^{l \times l}$ is the matrix containing the evaluation of the kernel function for all pairs of training vectors. Let $\mathbf{K}$ be a kernel matrix, the (squared) margin achieved by a hard-SVM in feature space can be computed as the value of the objective of a quadratic optimization problem, namely $\min_{\boldsymbol{\gamma} \in \Gamma} \boldsymbol{\gamma}^\intercal \mathbf{YKY}\boldsymbol{\gamma}$, where $\mathbf{Y}$ is the diagonal matrix containing the labels and $\Gamma = \{\boldsymbol{\gamma} \in \mathbb{R}_+^l \,|\, \sum_{i:y_i=+1} \gamma_i = \sum_{i:y_i=-1} \gamma_i = 1\}$. In the following, the Homogeneous Polynomial Kernel (HPK) $\kappa_{\text{HP}}^d(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^d$, $d \in \mathbb{N}$ and its normalized version $\widetilde{\kappa}_{HP}^d(\mathbf{x}, \mathbf{z}) = \frac{\kappa_{\text{HP}}^d(\mathbf{x},\mathbf{z})}{\sqrt{\kappa_{\text{HP}}^d(\mathbf{x},\mathbf{x})\kappa_{\text{HP}}^d(\mathbf{z},\mathbf{z})}}$ will be used as base kernels.

2.1 Multiple Kernel Learning

MKL [1] is a method to combine a set of base kernels that represent different similarity measures or different data sources (e.g. audio and video). Many functional forms exist to combine kernels. In this paper, the convex combination of base kernels is considered, that is:

$$\kappa_{\boldsymbol{\mu}}(\mathbf{x}, \mathbf{z}) = \sum_{r=1}^{P} \mu_r \kappa_r(\mathbf{x}, \mathbf{z}) = \sum_{r=1}^{P} \mu_r \langle \phi_r(\mathbf{x}), \phi_r(\mathbf{z}) \rangle, \quad \mu_r \geq 0, \quad \sum_r \mu_r = 1.$$

where the r-th kernel $\kappa_r(\mathbf{x}, \mathbf{z}) = \langle \phi_r(\mathbf{x}), \phi_r(\mathbf{z}) \rangle$ consists of the dot-product in feature space defined by the mapping function ϕ_r. In particular, we consider HPKs of increasing degrees as base kernels. It has been shown that, under mild conditions, any dot-product kernel of the form $f(\langle \mathbf{x}, \mathbf{z} \rangle)$ can actually be seen as a convex combination of HPKs [9]. In other words, applying MKL on HPKs can be seen as selecting a representation from the space of dot-product kernels.

2.2 Expressiveness of base kernels

The expressiveness (or complexity) of a kernel function can be defined as the number of dichotomies that can be realized by hyperplanes inside such feature space. It has been demonstrated that the expressiveness of a kernel is related to the rank of its induced kernel matrix. Specifically, given a set of instances S and the associated kernel matrix of rank r, then there is at least r instances in S that can be fragmented in the feature space [9]. However, using the rank of the kernel matrix for estimating the expressiveness can be very expensive since it requires the decomposition of the kernel matrix. To this end, in [9], a simpler measure able to approximate the rank of a kernel has been proposed, namely the

Spectral Ratio (SR). Specifically, given a kernel matrix $\mathbf{K}$, the SR is the ratio between the trace norm $\|\cdot\|_T$ and the Frobenius norm $\|\cdot\|_F$ of $\mathbf{K}$:

$$\mathcal{C}(\mathbf{K}) = \frac{\|\mathbf{K}\|_T}{\|\mathbf{K}\|_F} = \frac{\sum_i \mathbf{K}_{ii}}{\sqrt{\sum_{ij} \mathbf{K}_{ij}^2}}. \tag{1}$$

In the same work, it has been shown that $1 \le \mathcal{C}(\mathbf{K}) \le \sqrt{rank(\mathbf{K})}$ always holds. Furthermore, the SR has been related to other complexity measures, such as the Empirical Rademacher Complexity and the radius of the Minimum Enclosing Ball (MEB) in the feature space, that is the radius of the smallest hypersphere enclosing the data in the kernel space. Note that, the maximal SR is obtained by the identity kernel matrix, where examples are orthogonal to each other $\mathcal{C}(\mathbf{I}_l) = \sqrt{l}$, whereas the constant kernel $\mathcal{C}(\mathbf{1}_l\mathbf{1}_l^\intercal) = 1$ has the minimal SR. Notably, when a kernel is normalized ($\|\phi(\mathbf{x})\|_2 = 1$) the SR formulation in (1) can be further simplified, that is $\mathcal{C}(\mathbf{K}) = l\|\mathbf{K}\|_F^{-1}$.

3 MEMO-MKL

MKL algorithms generally aim at finding mixing coefficients maximizing the margin between the positive and negative classes. One of the main issues of MKL is that the combined kernel may be too expressive for a given problem, with the risk of overfitting. To overcome this problem a regularization term is usually included in the optimization criterion of these algorithms. However, as far as we know, only the mixing coefficients are regularized while no terms related to the resulting kernel complexity is taken into consideration in the objective function. In the MEMO-MKL algorithm presented in this paper we propose to find the mixing coefficients $\boldsymbol{\mu}$ which jointly minimize the SR and maximize the margin. Note that, normalized kernels and convex combinations of them have constant trace norm. Hence, minimizing the SR corresponds to maximizing the Frobenius norm, that can be characterized for a combined kernel as in the following:

$$\begin{aligned}\|\mathbf{K}_{\boldsymbol{\mu}}\|_F^2 &= \sum_{i,j}\left(\sum_r \mu_r \mathbf{K}_{ij}^{(r)}\right)^2 = \sum_{i,j}\sum_{r,s}\mu_r\mu_s\mathbf{K}_{ij}^{(r)}\mathbf{K}_{ij}^{(s)} \\ &= \sum_{r,s}\mu_r\mu_s \underbrace{\left(\sum_{i,j}\mathbf{K}_{ij}^{(r)}\mathbf{K}_{ij}^{(s)}\right)}_{Q_{rs}} = \boldsymbol{\mu}^\intercal\boldsymbol{Q}\boldsymbol{\mu}, \quad \boldsymbol{Q} \in \mathbb{R}^{P\times P}.\end{aligned}$$

In order to make the MKL problem unconstrained, a new vector of variables $\boldsymbol{\beta}$ is introduced, such that $\mu_r(\boldsymbol{\beta}) = e^{\beta_r}/\|e^{\boldsymbol{\beta}}\|_1$ and $\mathbf{K}_{\boldsymbol{\mu}(\boldsymbol{\beta})} = \sum_{r=1}^P \mu_r(\boldsymbol{\beta})\mathbf{K}^{(r)}$.

Given the change of variables above, the proposed algorithm maximizes the following unconstrained objective function, that is a trade-off between simpler representation (low SR) and large margin solutions:

$$\Psi(\boldsymbol{\beta}) = \hat{\gamma}(\boldsymbol{\beta})^\intercal\mathbf{Y}\mathbf{K}_{\boldsymbol{\mu}(\boldsymbol{\beta})}\mathbf{Y}\hat{\gamma}(\boldsymbol{\beta}) + \frac{\theta}{2}\boldsymbol{\mu}(\boldsymbol{\beta})^\intercal\boldsymbol{Q}\boldsymbol{\mu}(\boldsymbol{\beta}), \tag{2}$$

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

$$\text{where} \quad \hat{\gamma}(\boldsymbol{\beta}) = \arg\min_{\boldsymbol{\gamma}\in\Gamma} \boldsymbol{\gamma}^{\intercal}\mathbf{Y}\mathbf{K}_{\boldsymbol{\mu}(\boldsymbol{\beta})}\mathbf{Y}\boldsymbol{\gamma}, \tag{3}$$

Note that, when $\theta = 0$, the complexity of the representation becomes irrelevant, and the algorithm will find the coefficients maximizing the margin.
In order to maximize $\Psi(\boldsymbol{\beta})$ which is not convex, an iterative optimization algorithm is applied. We start from the uniform distribution of $\boldsymbol{\mu}(\boldsymbol{\beta})$, corresponding to $\boldsymbol{\beta} = \mathbf{0}$. At each step, the algorithm first finds the coefficients $\hat{\gamma}(\boldsymbol{\beta})$ by optimizing (3) with the current $\boldsymbol{\beta}$. Then, a step of gradient ascent is performed on $\Psi(\boldsymbol{\beta})$ where the vector $\hat{\gamma} = \hat{\gamma}(\boldsymbol{\beta})$ is considered locally constant. With these assumptions, the derivatives of $\Psi(\boldsymbol{\beta})$ can be easily computed as

$$\frac{\partial \Psi(\boldsymbol{\beta})}{\partial \beta_r} = \hat{\gamma}^{\intercal}\mathbf{Y}\mathbf{K}^{(r)}\mathbf{Y}\hat{\gamma} + \theta\boldsymbol{\mu}(\boldsymbol{\beta})^{\intercal}\boldsymbol{Q}_r.$$

and the mixing weights updated by $\beta_r \leftarrow \beta_r + \eta\frac{\partial\Psi(\boldsymbol{\beta})}{\partial\beta_r}$, $\mu_r(\boldsymbol{\beta}) = e^{\beta_r}/||e^{\boldsymbol{\beta}}||_1$, $\forall r \in \{1,\ldots,P\}$. The learning rate η is selected dinamically during the optimization by using the backtracking line-search method. The optimization loop ends when a maximum number of iterations is reached or the improvement of the objective function is below a fixed threshold.

4 Experimental assessment

To evaluate the effectiveness of the proposed MKL algorithm, a comparison against hard baselines has been performed on several benchmark problems, described in Table 1. These datasets are freely available on the UCI[1] and libsvm repositories[2].

The datasets were randomly split in training (70%) and test (30%) sets, and features were rescaled between 0 and 1 to prevent negative values. Multiclass problems have been mapped into binary ones keeping the distribution of positive and negative examples balanced. A 5-fold cross validation procedure on the training set has been used to find the best combination of the hyperparameters. After fitting the models, the accuracy score is computed on the test set. This procedure has been applied 30 times, and the average accuracies were recorded. Finally, normalized HPKs with degrees $d \in [1, 20]$, and the identity matrix kernel are considered for combination by MKL. The compared MKL methods are:

- average of base kernels, which is known to be a hard baseline, $\mu_r = \frac{1}{P}$;
- EasyMKL, a fast and effective margin maximization MKL algorithm [10];
- (fixed) linear kernel, $\mu_1 = 1$, $\mu_i = 0\ \forall i \neq 1$;
- MEMO-MKL, the algorithm proposed in this paper.

[1] M. Lichman. UCI machine learning repository, 2013.

[2] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

In all the experiments, an hard-margin SVM is used as base learner. The hyperparameters are $\lambda \in \{0, .01, .1, .2, .3, .9, .95, 1\}$ for EasyMKL, and $\theta \in \{10^i : -4 \leq i \leq 4\}$ for MEMO-MKL. Results are summarized in Table 1, where the average of accuracies, standard deviations, datasets informations, and the ranks of all used algorithms are shown. The results clearly show that MEMO-MKL is better than EasyMKL when the base kernels are too complex for the task considered and this especially happens when the number of features is large with respect to the number of examples.

dataset	**l × n**	**average**	**EasyMKL**	**MEMO-MKL**	**linear**
duke	44 × 7129	59.70$_{\pm 19.87}$	64.24$_{\pm 19.77}$	**84.55**$_{\pm 9.72}$	**84.55**$_{\pm 9.72}$
colon	62 × 2000	65.83$_{\pm 9.78}$	71.67$_{\pm 13.38}$	**83.96**$_{\pm 7.85}$	83.75$_{\pm 7.50}$
leukemia	72 × 7129	64.81$_{\pm 8.53}$	76.11$_{\pm 11.84}$	**96.85**$_{\pm 3.71}$	**96.85**$_{\pm 3.71}$
liver-disorder	145 × 5	65.59$_{\pm 5.75}$	68.29$_{\pm 5.75}$	70.00$_{\pm 7.46}$	**73.87**$_{\pm 5.51}$
iris	150 × 4	89.30$_{\pm 4.75}$	**91.67**$_{\pm 5.22}$	89.47$_{\pm 4.80}$	89.21$_{\pm 4.47}$
wine	178 × 13	97.63$_{\pm 3.39}$	**97.78**$_{\pm 3.14}$	**97.78**$_{\pm 1.99}$	97.33$_{\pm 2.10}$
sonar	208 × 60	85.58$_{\pm 5.61}$	85.64$_{\pm 5.63}$	**86.15**$_{\pm 5.40}$	73.59$_{\pm 3.37}$
glass	214 × 9	77.84$_{\pm 5.69}$	77.59$_{\pm 6.17}$	**78.58**$_{\pm 5.47}$	69.20$_{\pm 6.51}$
heart	270 × 13	77.25$_{\pm 3.94}$	78.28$_{\pm 3.68}$	80.39$_{\pm 4.17}$	**82.25**$_{\pm 4.26}$
ionosphere	351 × 34	**96.36**$_{\pm 1.59}$	96.33$_{\pm 1.49}$	96.33$_{\pm 1.57}$	89.05$_{\pm 3.51}$
breast-cancer	683 × 9	**96.47**$_{\pm 1.33}$	96.41$_{\pm 1.23}$	96.20$_{\pm 1.24}$	96.32$_{\pm 1.47}$
australian	690 × 14	81.23$_{\pm 2.92}$	84.89$_{\pm 2.81}$	**85.18**$_{\pm 2.73}$	84.74$_{\pm 2.50}$
diabetes	768 × 8	74.25$_{\pm 2.31}$	76.77$_{\pm 2.56}$	77.19$_{\pm 2.24}$	**77.27**$_{\pm 2.44}$
vehicle	846 × 18	96.98$_{\pm 1.17}$	**97.11**$_{\pm 1.20}$	**97.11**$_{\pm 1.23}$	91.15$_{\pm 1.61}$
fourclass	862 × 2	80.11$_{\pm 2.41}$	79.95$_{\pm 2.58}$	**80.29**$_{\pm 2.64}$	75.96$_{\pm 3.00}$
vowel	990 × 10	**99.37**$_{\pm 0.58}$	99.35$_{\pm 0.54}$	99.34$_{\pm 0.56}$	85.55$_{\pm 1.82}$
german-number	1000 × 24	73.33$_{\pm 2.47}$	73.09$_{\pm 2.61}$	76.47$_{\pm 2.57}$	**76.73**$_{\pm 2.36}$
splice	1000 × 60	84.85$_{\pm 1.54}$	**84.93**$_{\pm 2.12}$	84.80$_{\pm 1.55}$	80.05$_{\pm 2.27}$
dna	2000 × 180	93.97$_{\pm 0.85}$	**94.61**$_{\pm 0.84}$	94.12$_{\pm 0.83}$	89.91$_{\pm 1.41}$
madelon	2000 × 500	57.80$_{\pm 1.82}$	**59.55**$_{\pm 1.81}$	57.78$_{\pm 1.81}$	54.58$_{\pm 2.15}$
rank		2.85	2.20	**1.80**	2.90

Table 1: Datasets informations, accuracy scores (%) and rank of the proposed method and baselines. Best results are marked as bold.

The weights computed by MEMO-MKL are analyzed in Fig. 1 (left), showing their behavior when the hyperparameter θ is changed. As expected, increasing the value θ, the distribution of weights is concentrated on low degree HPKs (lower expressiveness), whereas more expressive HPKs will have higher weights when the algorithm optimizes the margin (lower θs). In Fig. 1 (right) a comparison between the weights computed by MEMO-MKL with $\theta = 0$ and EasyMKL is depicted, showing different distributions when both the methods focuses on large margin solutions.

5 Conclusions and future work

In this paper a novel MKL algorithm have been proposed which finds the kernels combination that maximizes the margin keeping low the complexity of the resulting representation. The algorithm has been evaluated empirically on several benchmark problems, showing better accuracies compared with three baselines. Furthermore, the solution computed is generally very sparse, and the weights

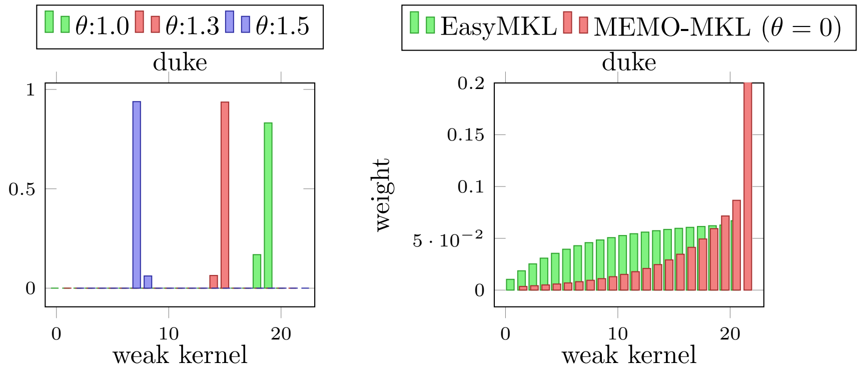

Fig. 1: Weights computed by MEMO-MKL varying the value θ (left) and weights comparison between EasyMKL and MEMO-MKL with $\theta = 0$ (right).

depend strictly on the selection of the regularization hyperparameter θ. In the future, the behavior of the proposed algorithm when using different kernel functions will be better analyzed, a comparison against other MKL approaches will be considered, and the sparsity property will be more theoretically studied.

References

[1] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(Jul):2211–2268, 2011.

[2] Soujanya Poria, Haiyun Peng, Amir Hussain, Newton Howard, and Erik Cambria. Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis. *Neurocomputing*, 2017.

[3] Nora K Speicher and Nico Pfeifer. Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery. *Bioinformatics*, 31(12):i268–i275, 2015.

[4] Blake Anderson, Curtis Storlie, and Terran Lane. Multiple kernel learning clustering with an application to malware. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 804–809. IEEE, 2012.

[5] Ivano Lauriola, Michele Donini, and Fabio Aiolli. Learning dot-product polynomials for multiclass problems. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2017.

[6] Alexander Zien and Cheng Soon Ong. Multiclass multiple kernel learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1191–1198. ACM, 2007.

[7] Ivano Lauriola, Mirko Polato, and Fabio Aiolli. Radius-margin ratio optimization for dot-product boolean kernel learning. In *International Conference on Artificial Neural Networks (ICANN)*, 2017.

[8] Alexandros Kalousis and Huyen T. Do. Convex formulations of radius-margin based support vector machines. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 169–177, 2013.

[9] Michele Donini and Fabio Aiolli. Learning deep kernels in the space of dot product polynomials. *Machine Learning*, pages 1–25, 2016.

[10] Fabio Aiolli and Michele Donini. Easymkl: a scalable multiple kernel learning algorithm. *Neurocomputing*, pages 215–224, 2015.