A Binary Classification Framework for Two Stage Kernel Learning

Abhishek Kumar
Dept. of Computer Science
University of Maryland
abhishek@cs.umd.edu

Alexandru Niculescu-Mizil
NEC Laboratories America
alex@nec-labs.com

Koray Kavukcoglu
NEC Laboratories America
koray@nec-labs.com

Hal Daumé III
Dept. of Computer Science
University of Maryland
hal@umiacs.umd.edu

In this abstract we show that the Multiple Kernel Learning (MKL) problem of learning a "good" combination of pre-specified base kernels can be reduced to binary classification in a new instance space. Framing MKL in this way has the distinct advantage that it makes it easy to leverage the extensive binary classification research to develop better performing and more scalable MKL techniques. Arguably, this approach also has to benefit of being conceptually simpler, easier to implement, and more accessible to practitioners than most current MKL systems.

Consider a classification problem where instances (x, y) are drawn from a distribution P over $\mathcal{X} \times \mathcal{Y}$, with $\mathcal{Y}$ a finite discrete set of labels. We assume that we have access to a finite set of p positive semi-definite (PSD) base kernel functions $K_1, \cdots, K_p$ with $K_i : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Our goal is to learn a combination of these kernel functions that is itself a positive semi-definite function and is a "good" kernel for the classification task at hand. To achieve this, we define another binary classification problem over a new instance space $\{(z_{xx'}, t_{yy'}) | ((x, y), (x', y')) \sim P \times P\} \subset \mathbb{R}^p \times \{\pm 1\}$ where

$$z_{xx'} = (K_1(x, x'), \cdots, K_p(x, x')) \qquad t_{yy'} = 2 \cdot \mathbf{1}\{y = y'\} - 1 \tag{1}$$

We call this new instance space the *K-space*. Any function $h : \mathbb{R}^p \to \mathbb{R}$ in this space induces a similarity function $\tilde{K}_h$ between instances in the original space:

$$\tilde{K}_h(x, x') = h(z_{xx'}) = h(K_1(x, x'), \cdots, K_p(x, x'))$$

If $\tilde{K}_h$ is also positive semi-definite, hence a valid kernel, we say that h is a *K-classifier*. For example, all linear functions with positive coefficients (i.e. $h_\mu(z_{xx'}) = \mu \cdot z_{xx'}$ with $\mu \geq 0$) are K-classifiers with the induced kernels $\tilde{K}_\mu$ being linear combinations of the p base kernels.

The key insight is that a good K-classifier h will induce a good kernel $\tilde{K}_h$, which in turn will allow for good performance on the original classification task. Intuitively, if a K-classifier h is a good classifier in the K-space, then $\tilde{K}_h(x, x') = h(z_{x,x'})$ will likely be positive if x and x' belong to the same class and negative otherwise. This makes $\tilde{K}_h$ a good kernel for the original classification task. This intuition is backed up by the following result:

Theorem 1 *Let h be a K-classifier, $\tilde{K}_h$ be the kernel induced by h, and R be a constant such that $\tilde{K}_h(x, x) \leq R^2 \ \forall x \in \mathcal{X}$. Let $z_{xx'}$ and $t_{yy'}$ be as defined in equation 1. Then, with probability at least $1 - \delta$, a classifier $\hat{f}(\cdot) = \sum_{i=1}^n \hat{\alpha}_i \tilde{K}_h(x_i, \cdot)$ with generalization error*

$$P_{(x,y)} \left[\!\!\left[y\hat{f}(x) \leq 0 \right]\!\!\right] \leq E_{(x,y)(x',y')} \left[\!\!\left[\left[1 - \frac{t_{yy'} h(z_{xx'})}{\gamma} \right]_+ \right]\!\!\right] + \mathcal{O}\left(\sqrt{\frac{R^4 \ln(1/\delta)}{\gamma^2 n}} \right)$$

can be learned efficiently from a training sample of n IID instances.

To the best of our knowledge, this result represent the first finite sample bound for two-stage kernel learning, improving on previous bounds that were only asymptotic.

Thus the problem of learning a good kernel is reduced to the problem of learning a good K-classifier in the newly defined K-space: given a training sample $(x_i, y_i)_{i=1}^n$ for the original classification task, construct a K-training set $(z_{ij}, t_{ij})_{1 \leq i \leq j \leq n}$ and learn a K-classifier h from this sample. Any learning algorithm can be used for learning h provided that the induced kernel can be guaranteed to be a valid PSD kernel.

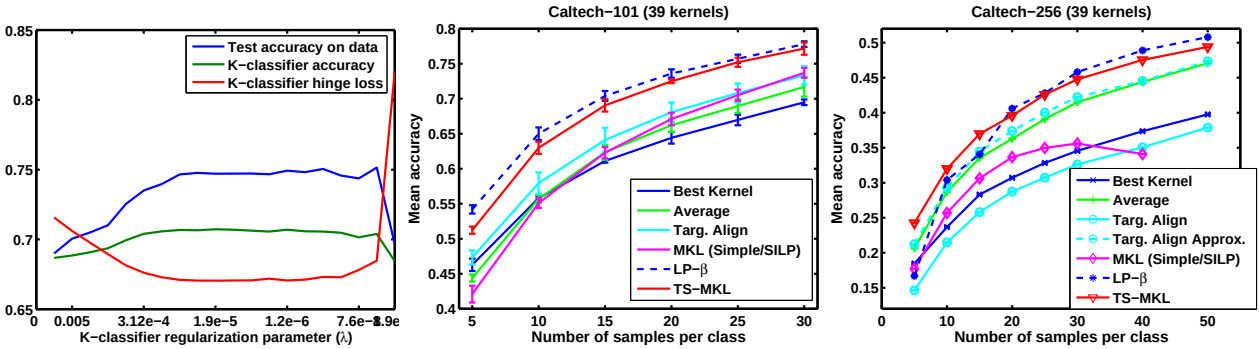

Figure 1: **Left:** Correlation between hinge loss (and accuracy) on K-examples and test data accuracy on Caltech 101 dataset. **Middle:** Caltech-101 results: mean accuracy over all classes for different sample sizes, averaged over 5 splits.**Right:** Caltech-256 results: mean accuracy over all classes for different sample sizes. Results for LP-β and MKL are taken from (Gehler & Nowozin, 2009).

To scale up and/or parallelize the algorithm one can readily take advantage of the recent advances in large scale learning. Our implementation, for instance, learns a K-classifier by training an L_2 regularized linear SVM with positive weights using the stochastic projected sub-gradient descent method from Pegasos (Shalev-Shwartz et al., 2007). This makes our approach very scalable with respect to the number of training points, the number of base kernels, as well as the number of classes in the original problem.

We run a comprehensive evaluation on two object recognition datasets (Caltech 101 and 256), and three bioinformatics datasets (Psort+, Psort-, Plant). We compare our method (TS-MKL) with a number of baselines: best base kernel, linear uniform combination of base kernels (Average), one-stage MKL algorithms like SILP (Sonnenburg et al., 2006) and SimpleMKL (Rakotomamonjy et al., 2007), and two-stage MKL algorithms based on Target Alignment (Cortes et al., 2010). On the Caltech problems (Figure 1) TS-MKL significantly outperforms the other MKL algorithms, and achieves performance comparable to the state of the art method, LP-β (Gehler & Nowozin, 2009)[1]. On the bioinformatics datasets (Table 1) all MKL methods perform equally well, and better than "best kernel" and "average kernel" baselines.

	Psort+		Psort–		Plant
	Full test	Filtered	Full test	Filtered	Full test
Best Kernel	81.30(4.69)	86.26(4.96)	85.95(1.54)	91.53(1.04)	72.19(3.94)
Average	84.75(3.97)	89.48(4.97)	88.03(1.10)	93.95(1.14)	86.72(3.38)
Target Alignment	88.14(3.99)	92.82(3.99)	89.91(1.42)	95.22(1.33)	89.13(2.75)
MKL (SILP/Simple)	89.05(3.02)	93.89(3.37)	91.01(1.10)	96.01(1.51)	89.32(2.76)
MC-MKL(Zien & Ong, 2007)	–	93.8	–	96.1	89.1
TS-MKL(Our approach)	89.08(3.32)	93.50(2.74)	90.15(1.33)	95.63(1.31)	88.86(3.26)

Table 1: Average accuracy measures (%) over 10 splits for Psort+, Psort- and Plant datasets. Numbers in parentheses are the std. deviations.

References

Cortes, C., Mohri, M., & Rostamizadeh, A. (2010). Two-Stage Learning Kernel Algorithms. *International Conference on Machine Learning*.

Gehler, P., & Nowozin, S. (2009). On Feature Combination for Multiclass Object Detection. *International Conference on Computer Vision*.

Ong, C. S., Smola, A. J., & Williamson, R. C. (2005). Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, *6*, 1043–1071.

Rakotomamonjy, A., Bach, F., Canu, S., & Grandvalet, Y. (2007). More efficiency in multiple kernel learning. *International Conference on Machine Learning*.

Shalev-Shwartz, S., Singer, Y., & Srebro, N. (2007). Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. *International Conference on Machine Learning*.

Sonnenburg, S., Ratsch, G., Schafer, C., & Scholkopf, B. (2006). Large scale multiple kernel learning. *Journal of Machine Learning Research*, *7*.

Zien, A., & Ong, C. S. (2007). Multiclass Multiple Kernel Learning. *International Conference on Machine Learning*.

[1]LP-β is an ensemble based method that learns an ensemble of SVM classifiers, each of which is trained on an individual kernel.