Localized Multiple Kernel Machines for Image Recognition

Mehmet Gönen GONEN@BOUN.EDU.TR
Ethem Alpaydın ALPAYDIN@BOUN.EDU.TR
Department of Computer Engineering
Boğaziçi University
TR-34342, Bebek, İstanbul, Turkey

Abstract

We review our work on localized multiple kernels (Gönen and Alpaydın, 2008, 2009) that allows kernels to be combined with different weights in different regions of the input space by using a gating model. We give example uses in image recognition for combining kernels of different representations and costs.

1. The Model

The multiple kernel learning (MKL) framework (Lanckriet et al., 2004; Bach et al., 2004) uses an unweighted summation of discriminant values in different feature spaces that corresponds to a weighted summation of kernel values:

$$f(\boldsymbol{x}) = \sum_{m=1}^{P} \langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}) \rangle + b$$

where m indexes kernels, $\boldsymbol{w}_m$ is the vector of weight coefficients and $\Phi_m(\boldsymbol{x})$ is the mapping function for feature space m. After eliminating $\boldsymbol{w}_m$ from the model by using the duality conditions, the discriminant function becomes:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i \sum_{m=1}^{P} \eta_m \underbrace{\langle \Phi_m(\boldsymbol{x}_i), \Phi_m(\boldsymbol{x}) \rangle}_{k_m(\boldsymbol{x}_i, \boldsymbol{x})} + b$$

where the kernel weights satisfy $\eta_m \geq 0$ and $\sum_{m=1}^{P} \eta_m = 1$.

The localized multiple kernel learning (LMKL) framework divides the input space into regions and assigns combination weights to kernels in a data-dependent way (Gönen and Alpaydın, 2008). The discriminant function for binary classification is rewritten as:

$$f(\boldsymbol{x}) = \sum_{m=1}^{P} \eta_m(\boldsymbol{x}|\mathbf{V}) \langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}) \rangle + b \tag{1}$$

where $\eta_m(\boldsymbol{x}|\mathbf{V})$ is a parametric *gating model* which assigns a weight to feature space m as a function of the input $\boldsymbol{x}$. This is similar to but also different from the mixture of experts framework (Jacobs et al., 1991) in the sense that the gating model combines kernel-based experts and is learned together with experts; the difference is that in the mixture of experts,

experts individually are classifiers whereas in our formulation, there is no discriminant per kernel.

Note that unlike in Mkl, in Lmkl it is not obligatory to combine different feature spaces; we can also use multiple copies of the same feature space in different regions of the input space and thereby obtain a more complex discriminant function. By using (1) and regularizing the discriminant coefficients of all the feature spaces together, Lmkl obtains the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\sum_{m=1}^{P}\|\boldsymbol{w}_m\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{with respect to} \quad & \boldsymbol{w}_m \in \mathbb{R}^{D_m}, b \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}_+^N, \mathbf{V} \in \mathbb{R}^{D_{\mathcal{G}}} \\ \text{subject to} \quad & y_i\left(\sum_{m=1}^{P}\eta_m(\boldsymbol{x}_i|\mathbf{V})\langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}_i)\rangle + b\right) \geq 1-\xi_i \qquad \forall i \end{aligned} \tag{2}$$

where C is the regularization parameter, D_m is the dimensionality of the feature space m, $\boldsymbol{\xi}$ is the vector of slack variables, $\mathbf{V}$ is the vector of gating model parameters, and $D_{\mathcal{G}}$ is the dimensionality of the space in which the gating model parameters are defined. The optimization problem in (2) is not convex due to the nonlinearity formed by using the gating model outputs in the separation constraints.

If we know the gating model parameters, the model becomes convex and we can write the dual formulation as:

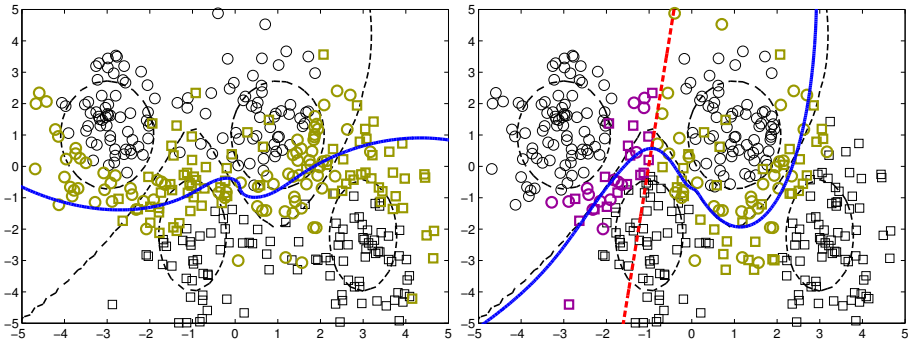
$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{with respect to} \quad & \boldsymbol{\alpha} \in \mathbb{R}_+^N \\ \text{subject to} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0 \\ & C \geq \alpha_i \geq 0 \qquad \forall i \end{aligned} \tag{3}$$

where the *locally combined kernel* is defined as:

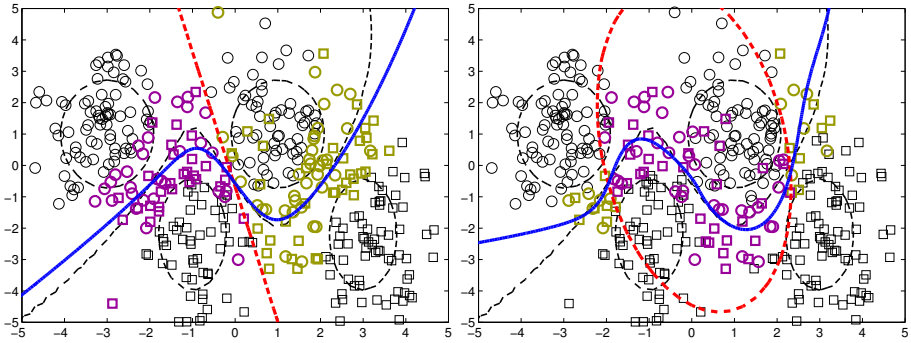
$$k_\eta(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{m=1}^{P}\eta_m(\boldsymbol{x}_i|\mathbf{V})k_m(\boldsymbol{x}_i, \boldsymbol{x}_j)\eta_m(\boldsymbol{x}_j|\mathbf{V}) \; .$$

By using the support vector coefficients obtained from (3) and the gating model parameters, we obtain the following discriminant function:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N}\alpha_i y_i k_\eta(\boldsymbol{x}_i, \boldsymbol{x}) + b \; .$$



(a) MKL (k_L-k_P) (Acc.:90.95 SV:38.23) (b) LMKL (k_L-k_P) (Acc.:91.78 SV:25.20)



(c) LMKL (k_P-k_P) (Acc.:91.60 SV:31.85) (d) LMKL (k_L-k_L) (Acc.:91.63 SV:21.98)

Figure 1: Fitted functions (solid lines) and support vectors (bold points) on GAUSS4 data set. Dashed lines show the Gaussians from which data are sampled and the optimal Bayes' discriminant. The thick dashed lines show the gating boundaries (where $\eta_i(\boldsymbol{x}) = \eta_j(\boldsymbol{x})$ and i, j are neighboring kernels) and the thick lines show the learned class boundaries. Test accuracies and stored support vector percentages are given. Note that the softmax function allows a smooth transition between regions. (a) MKL solution, (b) linear and quadratic kernels are combined with linear gating, (c) two quadratic kernels are combined with linear gating, and (d) two linear kernels are combined with quadratic gating.

We can use different gating models for assigning kernel weights in a data-dependent way:

$$\eta_m(\boldsymbol{x}|\mathbf{V}) = \frac{\exp(\langle \boldsymbol{v}_m, \Phi_{\mathcal{G}}(\boldsymbol{x})\rangle + v_{m0})}{\sum_{h=1}^{P} \exp(\langle \boldsymbol{v}_h, \Phi_{\mathcal{G}}(\boldsymbol{x})\rangle + v_{h0})} \qquad \forall m \tag{4}$$

where $\mathbf{V} = \{\boldsymbol{v}_1, v_{10}, \boldsymbol{v}_2, v_{20}, \ldots, \boldsymbol{v}_p, v_{p0}\}$, $\Phi_{\mathcal{G}}(\boldsymbol{x})$ is the mapping function for the gating feature space and there are $D_{\mathcal{G}} = P(D_g + 1)$ parameters where D_g is the dimensionality of the gating feature space. See Figure 1 for an example.

In some application areas such as bioinformatics, $\boldsymbol{x}$ vectors may appear in a non-vectorial format such as sequences, trees, and graphs. In such a case where we can calculate kernel

matrices but can not represent the data instances as $\boldsymbol{x}$ vectors, directly, Gönen and Alpaydın (2009) define $\Phi_{\mathcal{G}}(\boldsymbol{x})$ in terms of the kernel values:

$$\Phi_{\mathcal{G}}(\boldsymbol{x}) = [k_{\mathcal{G}}(\boldsymbol{x}_1, \boldsymbol{x}) \; k_{\mathcal{G}}(\boldsymbol{x}_2, \boldsymbol{x}) \; \ldots \; k_{\mathcal{G}}(\boldsymbol{x}_n, \boldsymbol{x})]^T$$

where the gating kernel, $k_{\mathcal{G}}$, can be one of the combined kernels $k_1, k_2, \ldots, k_p$, a combination of them, or a completely different kernel used only for determining the gating boundaries.

We can not perform the joint-optimization of the support vector coefficients and gating model parameters in (2) efficiently because of non-convexity. Lmkl uses a two-step alternate optimization procedure in order to solve (2), as also used for obtaining η_m parameters of Mkl in a previous study Rakotomamonjy et al. (2007). This procedure consists of two basic steps: (a) solving the model with a fixed gating model, and, (b) updating the gating model parameters of (4) with the gradients calculated from the current solution.

Due to strong convexity, for a given $\mathbf{V}$, the gradients of the objective functions in (2) are equal to the gradients of the objective functions in (3), respectively. These gradients are found as:

$$\begin{aligned}
\frac{\partial J(\mathbf{V})}{\partial \boldsymbol{v}_m} &= -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{h=1}^{P} \alpha_i \alpha_j y_i y_j \eta_h(\boldsymbol{x}_i|\mathbf{V}) k_h(\boldsymbol{x}_i, \boldsymbol{x}_j) \eta_h(\boldsymbol{x}_j|\mathbf{V}) \\
&\quad \left(\Phi_{\mathcal{G}}(\boldsymbol{x}_i) \left[\delta_m^k - \eta_m(\boldsymbol{x}_i|\mathbf{V}) \right] + \Phi_{\mathcal{G}}(\boldsymbol{x}_j) \left[\delta_m^h - \eta_m(\boldsymbol{x}_j|\mathbf{V}) \right] \right) \\
\frac{\partial J(\mathbf{V})}{\partial v_{m0}} &= -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{h=1}^{P} \alpha_i \alpha_j y_i y_j \eta_h(\boldsymbol{x}_i|\mathbf{V}) k_h(\boldsymbol{x}_i, \boldsymbol{x}_j) \eta_h(\boldsymbol{x}_j|\mathbf{V}) \\
&\quad \left(\delta_m^h - \eta_m(\boldsymbol{x}_i|\mathbf{V}) + \delta_m^h - \eta_m(\boldsymbol{x}_j|\mathbf{V}) \right)
\end{aligned}$$

where δ_m^h is 1 if $m = h$ and 0 otherwise. These gradients are used to update the gating model parameters at each step.

Note that any kernel machine that has a hyperplane-based decision function can be localized by replacing $\langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle$ with $\sum_{m=1}^{P} \eta_m(\boldsymbol{x}|\mathbf{V}) \langle \boldsymbol{w}_m, \Phi_m(\boldsymbol{x}) \rangle$ and deriving the corresponding update rules.

2. Face Recognition

We compare support vector machine (Svm), Mkl and Lmkl on the Olivetti face recognition data set in order to see the performance of Lmkl in a real-life scenario with a very high dimensional feature space. Olivetti data set consists of 10 different 64×64 grayscale images of 40 subjects. We construct a two-class data set by combining male subjects (36 subjects) into one class versus female subjects (4 subjects) in another class. We select two images of each subject randomly and reserve these total 80 images as the test set. Then, we apply 8-fold cross-validation on the remaining 320 images by putting one image of each subject to the validation set at each fold. The validation sets of all folds are used to optimize C by trying values 0.01, 0.1, 1, 10, and 100. The best configuration (the one that has the highest average accuracy on the validation folds) is used to train the learners on the training folds and their performance is measured over the test set. So, for each scenario, we have eight test set results; we display their averages and one standard deviations.

2.1 Combining Multiple Resolutions

In the first set of experiments, we combine 4×4, 8×8, 16×16, 32×32, and 64×64 bitmap representations of the images using the linear kernel (see Figure 2 for an example face image in different resolutions). Table 1 lists the results of the single kernel SVMs, MKL and LMKL combining kernels over multiple resolutions. We see that LMKL generally performs better than MKL, and also better than the single kernel SVMs. MKL only uses the highest resolution image (64×64) and ignores the other representations.

Figure 2: An example face image in different resolutions.

Table 1: The average accuracies and support vector percentages of the single kernel SVMs, MKL and LMKL combining multiple resolutions on the OLIVETTI data set.

	Method	Configuration	Accuracy	SV
Single Kernel	SVM	$\Phi(\boldsymbol{x}) = 4 \times 4$	93.28 ± 0.65	21.70 ± 0.93
	SVM	$\Phi(\boldsymbol{x}) = 8 \times 8$	97.50 ± 1.16	20.13 ± 1.04
	SVM	$\Phi(\boldsymbol{x}) = 16 \times 16$	97.03 ± 0.93	19.82 ± 0.94
	SVM	$\Phi(\boldsymbol{x}) = 32 \times 32$	97.97 ± 1.48	23.71 ± 1.39
	SVM	$\Phi(\boldsymbol{x}) = 64 \times 64$	97.66 ± 1.41	25.94 ± 1.01
Multiple Kernel	MKL†		97.66 ± 1.41	25.94 ± 1.01
	LMKL	$\Phi_{\mathcal{G}}(\boldsymbol{x}) = 4 \times 4$	97.03 ± 1.15	29.29 ± 2.90
	LMKL	$\Phi_{\mathcal{G}}(\boldsymbol{x}) = 8 \times 8$	99.38 ± 0.94	27.68 ± 2.95
	LMKL	$\Phi_{\mathcal{G}}(\boldsymbol{x}) = 16 \times 16$	98.59 ± 1.41	26.52 ± 2.37
	LMKL	$\Phi_{\mathcal{G}}(\boldsymbol{x}) = 32 \times 32$	99.38 ± 1.16	24.78 ± 2.57
	LMKL	$\Phi_{\mathcal{G}}(\boldsymbol{x}) = 64 \times 64$	99.53 ± 0.65	26.65 ± 4.12

†: $\eta_{4\times4} = 0, \eta_{8\times8} = 0, \eta_{16\times16} = 0, \eta_{32\times32} = 0, \eta_{64\times64} = 1$

2.2 Combining Multiple Input Patches

Instead of defining kernels over the whole input image, we can divide the image into non-overlapping patches and use linear kernels on the patches. In such a case, it is not a good idea to use softmax gating because a patch by itself does not carry enough discriminative information. We use the sigmoid function instead of softmax in gating and thereby allow multiple patches to be used in a cooperative manner:

$$\eta_m(\boldsymbol{x}|\mathbf{V}) = \frac{1}{1 + \exp(-\langle \boldsymbol{v}_m, \Phi_{\mathcal{G}}(\boldsymbol{x}) \rangle - v_{m0})} \qquad \forall m$$

and the gradients with respect to the sigmoid gating model parameters are calculated as follows:

$$\frac{\partial J(\mathbf{V})}{\partial \boldsymbol{v}_m} = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \eta_m(\boldsymbol{x}_i|\mathbf{V})k_m(\boldsymbol{x}_i,\boldsymbol{x}_j)\eta_m(\boldsymbol{x}_j|\mathbf{V}) \\ \left(\Phi_{\mathcal{G}}(\boldsymbol{x}_i)\left[1-\eta_m(\boldsymbol{x}_i|\mathbf{V})\right]+\Phi_{\mathcal{G}}(\boldsymbol{x}_j)\left[1-\eta_m(\boldsymbol{x}_j|\mathbf{V})\right]\right)$$

$$\frac{\partial J(\mathbf{V})}{\partial v_{m0}} = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \eta_m(\boldsymbol{x}_i|\mathbf{V})k_m(\boldsymbol{x}_i,\boldsymbol{x}_j)\eta_m(\boldsymbol{x}_j|\mathbf{V}) \\ \left(1-\eta_m(\boldsymbol{x}_i|\mathbf{V})+1-\eta_m(\boldsymbol{x}_j|\mathbf{V})\right) \ .$$

Table 2 shows the results of Mkl and Lmkl combining kernels calculated over non-overlapping patches of face images. The gating model looks at a low-resolution version of the image and choose a subset of the patches and only for the chosen patches, a high resolution image patch is taken as input.

Table 2: The average accuracies and support vector percentages of Mkl and Lmkl combining multiple input patches on the Olivetti data set.

Method	Configuration			Accuracy	SV
Mkl	$\Phi_m(\boldsymbol{x}) = 8\times 8$			99.38 ± 0.94	19.42 ± 0.87
Lmkl	$\Phi_m(\boldsymbol{x}) = 8\times 8$	$\Phi_{\mathcal{G}}(\boldsymbol{x}) = 8\times 8$	$\eta_m(\boldsymbol{x}\|\mathbf{V}) =$ Softmax	98.59 ± 1.24	11.74 ± 1.07
Lmkl	$\Phi_m(\boldsymbol{x}) = 8\times 8$	$\Phi_{\mathcal{G}}(\boldsymbol{x}) = 8\times 8$	$\eta_m(\boldsymbol{x}\|\mathbf{V}) =$ Sigmoid	99.84 ± 0.44	24.38 ± 1.96
Mkl	$\Phi_m(\boldsymbol{x}) = 16\times 16$			99.06 ± 0.88	22.19 ± 1.00
Lmkl	$\Phi_m(\boldsymbol{x}) = 16\times 16$	$\Phi_{\mathcal{G}}(\boldsymbol{x}) = 4\times 4$	$\eta_m(\boldsymbol{x}\|\mathbf{V}) =$ Softmax	96.56 ± 1.86	17.14 ± 2.71
Lmkl	$\Phi_m(\boldsymbol{x}) = 16\times 16$	$\Phi_{\mathcal{G}}(\boldsymbol{x}) = 4\times 4$	$\eta_m(\boldsymbol{x}\|\mathbf{V}) =$ Sigmoid	99.53 ± 0.65	23.35 ± 1.47

Figure 3 shows the combination weights found by Mkl and sample face images weighted with those. Mkl uses the same weights over the whole input space and thereby the parts whose weights are nonzero are used in the decision process for all subjects. Figure 4 illustrates an example use of Lmkl. The gating model activates important parts of each face image and these parts are used in the classifier with nonzero weights, whereas the parts whose gating model outputs are zero are not considered. That is, looking at the output of the gating model, we can skip generating or extracting the high resolution versions of these parts. This can be considered similar to a selective attention mechanism whereby the gating model between a saliency measure and drives a high resolution "eye" to consider only regions of high saliency (Alpaydın, 1995).

3. Digit Recognition

A binary classification data set is generated from the Usps data set (16×16 grayscale digit images) by putting {'3'} and {'5', '8'} into different classes. A random one-third is reserved as the test set and the remaining two-thirds is resampled using 5×2 cross-validation to generate ten training and validation sets, with stratification.

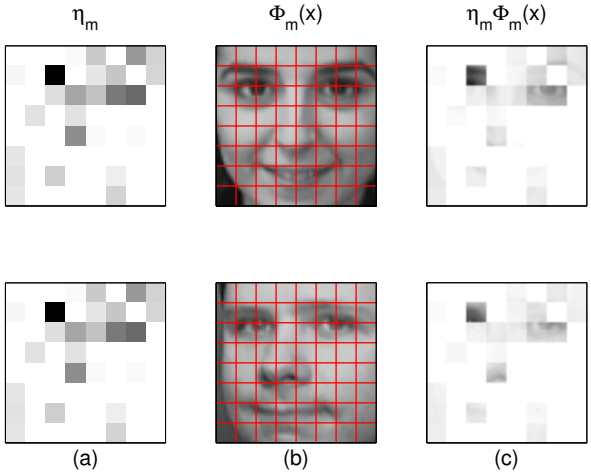


Figure 3: An example use of Mkl on the Olivetti data set: (a) combination weights, (b) features fed into kernels, and (c) features weighted with combination weights.

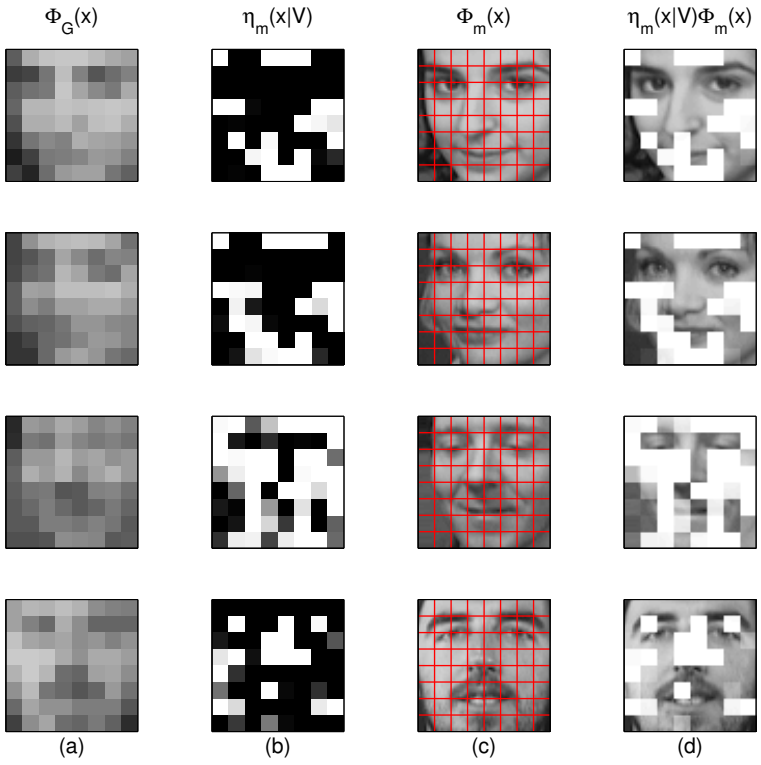


Figure 4: An example use of Lmkl on the Olivetti data set: (a) features fed into the gating model, (b) gating model outputs, (c) features fed into kernels, and (d) features weighted with gating model outputs.

3.1 Combining Multiple Resolutions

We combine 2×2, 4×4, 8×8, and 16×16 bitmap representations of the digits using the linear kernel (see Figure 5 for an example face image in different resolutions). Table 3 lists the results of Mkl and Lmkl combining kernels over multiple resolutions. We again see that Lmkl generally performs better than Mkl, and also better than the single kernel Svms. Mkl uses the highest resolution image (16×16) with a very large weight (0.87) compared to other representations.

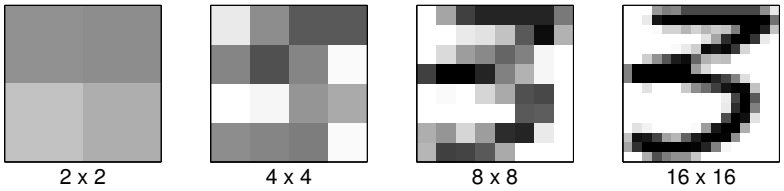


Figure 5: An example digit in different resolutions.

Table 3: The average accuracies and support vector percentages of the single kernel Svms, Mkl and Lmkl combining multiple resolutions on the Usps data set.

	Method	Configuration	Accuracy	SV
Single Kernel	Svm	$\Phi(\boldsymbol{x}) = 2 \times 2$	83.31 ± 0.31	39.19 ± 1.29
	Svm	$\Phi(\boldsymbol{x}) = 4 \times 4$	90.67 ± 0.38	27.05 ± 0.99
	Svm	$\Phi(\boldsymbol{x}) = 8 \times 8$	95.41 ± 0.39	21.49 ± 0.73
	Svm	$\Phi(\boldsymbol{x}) = 16 \times 16$	95.95 ± 0.45	23.19 ± 0.74
Multiple Kernel	Mkl†		96.09 ± 0.30	22.49 ± 0.73
	Lmkl	$\Phi_{\mathcal{G}}(\boldsymbol{x}) = 2 \times 2$	95.96 ± 0.81	15.05 ± 1.14
	Lmkl	$\Phi_{\mathcal{G}}(\boldsymbol{x}) = 4 \times 4$	96.37 ± 0.49	13.72 ± 0.65
	Lmkl	$\Phi_{\mathcal{G}}(\boldsymbol{x}) = 8 \times 8$	96.20 ± 0.53	12.31 ± 0.87
	Lmkl	$\Phi_{\mathcal{G}}(\boldsymbol{x}) = 16 \times 16$	96.28 ± 0.52	10.78 ± 0.82

†: $\eta_{2\times2} = 0.07, \eta_{4\times4} = 0.03, \eta_{8\times8} = 0.03, \eta_{16\times16} = 0.87$

3.2 Combining Multiple Input Patches

Table 4 shows the results of Mkl and Lmkl combining kernels calculated over non-overlapping patches of digits. Lmkl with both gating models works better than Mkl.

Figure 6 illustrates an example use of Lmkl. Similar to face images, the gating model chooses the important strokes of each digit image and these parts are used in the classifier with nonzero weights.

4. Conclusions

We discuss localized version of Mkl where a gating model chooses one or a subset of a set of kernels through using softmax or sigmoid gating. When given a set of kernels with different costs, instead of using the costly kernels for all cases, we can use a simple gating

Table 4: The average accuracies and support vector percentages of MKL and LMKL combining multiple input patches on the USPS data set.

Method	Configuration			Accuracy	SV
MKL	$\Phi_m(\boldsymbol{x}) = 4 \times 4$			95.69 ± 0.37	21.91 ± 1.04
LMKL	$\Phi_m(\boldsymbol{x}) = 4 \times 4$	$\Phi_g(\boldsymbol{x}) = 4 \times 4$	$\eta_m(\boldsymbol{x}\|\mathbf{V}) = \text{SOFTMAX}$	96.79 ± 0.47	9.19 ± 0.71
LMKL	$\Phi_m(\boldsymbol{x}) = 4 \times 4$	$\Phi_g(\boldsymbol{x}) = 4 \times 4$	$\eta_m(\boldsymbol{x}\|\mathbf{V}) = \text{SIGMOID}$	96.81 ± 0.64	11.92 ± 0.94
MKL	$\Phi_m(\boldsymbol{x}) = 8 \times 8$			95.72 ± 0.36	22.91 ± 0.84
LMKL	$\Phi_m(\boldsymbol{x}) = 8 \times 8$	$\Phi_g(\boldsymbol{x}) = 2 \times 2$	$\eta_m(\boldsymbol{x}\|\mathbf{V}) = \text{SOFTMAX}$	95.82 ± 0.55	14.48 ± 0.75
LMKL	$\Phi_m(\boldsymbol{x}) = 8 \times 8$	$\Phi_g(\boldsymbol{x}) = 2 \times 2$	$\eta_m(\boldsymbol{x}\|\mathbf{V}) = \text{SIGMOID}$	95.69 ± 0.48	16.87 ± 0.84

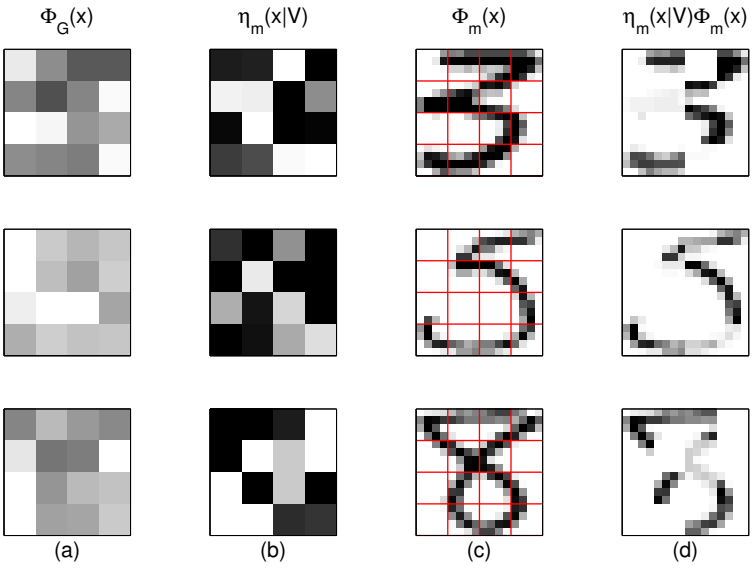


Figure 6: An example use of LMKL on the USPS data set: (a) features fed into the gating model, (b) gating model outputs, (c) features fed into kernels, and (d) features weighted with gating model outputs.

model to choose among kernels and then use the costly kernels only for cases where they are actually needed. In image processing applications such as face or handwritten digit recognition, the gating model may use a low-resolution image and acts as saliency detector driving the costly high resolution kernels only to parts of the image that are discriminative, thereby decreasing the overall complexity.

Acknowledgments

This work was supported by the Turkish Academy of Sciences in the framework of the Young Scientist Award Program under EA-TÜBA-GEBİP/2001-1-1, Boğaziçi University

Scientific Research Project 07HA101 and the Turkish Scientific Technical Research Council (TÜBİTAK) under Grant EEEAG 107E222. The work of M. Gönen was supported by the PhD scholarship (2211) from TÜBİTAK.

References

E. Alpaydın. Selective attention for handwritten digit recognition. In *Advances in Neural Information Processing Systems*, 1995.

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

M. Gönen and E. Alpaydın. Localized multiple kernel learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 352–359, 2008.

M. Gönen and E. Alpaydın. Multiple kernel machines using localized kernels. In *Supplementary Proceedings of the 4th IAPR International Conference on Pattern Recognition in Bioinformatics*, 2009.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

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

A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. In *Proceedings of the 24th International Conference on Machine Learning*, pages 775–782, 2007.