Combining Multiple Kernels by Augmenting the Kernel Matrix

Fei Yan Krystian Mikolajczyk Josef Kittler Atif Tahir

Centre for Vision, Speech, and Signal Processing
University of Surrey
Guildford, Surrey, GU2 7XH, UK
{f.yan,k.mikolajczyk,j.kittler,m.tahir}@surrey.ac.uk

Abstract. In this paper we present a novel approach to combining multiple kernels where the kernels are computed from different information channels. In contrast to traditional methods that learn a linear combination of n kernels of size $m \times m$, resulting in m coefficients in the trained classifier, we propose a method that can learn $n \times m$ coefficients. This allows to assign different importance to the information channel per example rather than per kernel. We analyse the proposed kernel combination in empirical feature space and provide its geometrical interpretation. We validate the approach on both UCI datasets and an object recognition dataset, and demonstrate that it leads to classification improvements.

1 Introduction

Since their introduction in the mid-1990s, kernel methods [1, 2] have proven successful for many machine learning problems, e.g., classification, regression, dimensionality reduction, clustering. Representative methods such as support vector machine (SVM) [3, 2], kernel Fisher discriminant analysis (kernel FDA) [4, 5], kernel principal component analysis (kernel PCA) [6] have been reported to produce the state-of-the-art performance in numerous applications. In a kernel method, the choice of kernel is critically important, since the kernel completely determines the embedding of the data in the feature space. In many problems, multiple kernels capturing different "views" of the problem are available. In such a situation, one naturally wants to use these kernels in an "optimal" way.

Multiple kernel learning (MKL) was pioneered by Lancriet et al. in [7], where the key idea is to learn a linear combination of a given set of base kernels by maximising the (soft) margin between two classes or by maximising the "alignment" of the base kernels. In [7], the kernel weights are regularised with an ℓ_1 norm. Following this seminal work, MKL has become one of the most active areas in the machine learning community in the past few years. Various extensions have been made to [7]. For example, the efficiency of MKL is significantly improved in [8–10]; a multiclass version and a multilabel version are proposed in [11] and [12] respectively; in [13–15], the ratio of the inter- and intra- class scatters of FDA is maximised instead of the margin and kernel alignment; while in [16–18, 15], ℓ_2 norm and even a general ℓ_p norm regularisation is considered instead of the ℓ_1 norm.

Despite the improvements achieved with these extensions both in terms of efficiency and accuracy, all these MKL methods share one limitation. To see this, let us consider an object categorisation problem as an example. Suppose the number of training samples is m and n training kernels of size $m \times m$ are available. Let these n kernels capture various aspects of the classification problem by using different features such as colour, texture, shape. Since all the MKL methods discussed above learn a linear combination of the base kernels, the learnt composite kernel also has a size $m \times m$. As a result, the learnt decision function has m coefficients, one for each training sample[1]. This means the contribution of a particular feature channel is fixed for all training samples. This is an unnecessarily strong constraint that does not allow to fully exploit the information from every sample. For example, one particular sample may carry more shape information than colour information, and vice versa for another sample. In a linear combination scheme, however, the shape information will be equally weighted in both training samples. Relaxing this constraint will allow to assign different weights to different samples depending on their importance in particular information channel. This effectively means that two different features extracted from the same sample are treated as two different samples of the same class.

In this paper, we present a learning approach that uses multiple kernels but, in contrast to existing MKL approaches, allows training samples to have different contributions in a particular feature channel. Instead of linear combination of the base kernels, we construct an $(n \times m) \times (n \times m)$ training kernel matrix. This leads to $n \times m$ coefficients in the trained decision function, in contrast to m coefficients in the linear combination scheme. As a result, the training samples contribute differently and the decision function is more flexible. We give the geometrical interpretation of our augmented kernel matrix (AKM) scheme and make comparison to that of the linear combination scheme. We show on several UCI datasets and an object recognition dataset that the AKM scheme can outperform linear combination of kernels.

The rest of this paper is organised as follows. We first introduce the concept of empirical feature space in Section 2 as it is important for understanding various kernel combination schemes. In Section 3 we briefly review the linear combination scheme. We then present our AKM scheme in Section 4 and discuss its connection to linear combination both algebraically and geometrically. Experimental results are provided in 5, which validate this new scheme. Finally conclusions are given in 6.

2 Empirical Feature Space

This section introduces the concept of empirical feature space that will be then used to discuss different methods for kernel combination. Let us for the moment

[1] More precisely, the decision function has $m + 1$ coefficients including a bias term b.

consider a single kernel case. We are given a symmetric, positive semi-definite (PSD) $m \times m$ training kernel matrix K and a corresponding $m \times l$ test kernel matrix $\dot{K}$, where K contains the pairwise dot products of the m training samples in some feature space, and $\dot{K}$ contains the pairwise dot products of the m training samples and the l test samples in the feature space. Note that this feature space usually has a very high or even infinite dimension and thus not directly tractable. However, it is shown in [19] that there exists an empirical feature space in which the intrinsic geometry of the data is identical to that in the true feature space, and for many machine learning problems, it suffices to study this empirical feature space.

To compute from K and $\dot{K}$ the training and test samples in the empirical feature space, consider the eigen decomposition of K:

$$K = V \Lambda V^T \tag{1}$$

where Λ is the $r \times r$ diagonal matrix containing the r ($r \leq m$) non-zero eigen values of K, and V is the $m \times r$ matrix containing the r associated eigen vectors. Note that since K is PSD, all the r non-negative eigenvalues of K are positive, and r is also the rank of K. It directly follows that

$$K = V \Lambda^{1/2} (\Lambda^{1/2})^T V^T = ((V \Lambda^{1/2})^T)^T (V \Lambda^{1/2})^T := X^T X \tag{2}$$

where the $r \times m$ matrix X is defined as

$$X = (V \Lambda^{1/2})^T \tag{3}$$

and its i^{th} column is the i^{th} training sample in the empirical feature space. Now let $\dot{X}$ be the $r \times l$ matrix whose i^{th} column is the i^{th} test sample in the empirical space. $\dot{X}$ is given by solving the following linear equation:

$$X^T \dot{X} = \dot{K} \tag{4}$$

We have shown in (3) and (4) given K and $\dot{K}$ how to find the training and test samples in the empirical feature space $\mathbb{R}^r$. In many practical situations, for example, in the case of Radial Basis Function (RBF) kernel, K is full rank, i.e. $r = m$. As a result, the m training samples X and l test samples $\dot{X}$ live in an m dimensional empirical feature space $\mathbb{R}^m$.

3 Linear Combination of Kernels

Now we turn to the case of multiple kernels. Assume we are given n training kernels $K_1, \cdots, K_n$ of size $m \times m$ and n corresponding test kernels $\dot{K}_1, \cdots, \dot{K}_n$ of size $m \times l$. In this section, we consider a linear combination of the base kernels:

$$K = \sum_{j=1}^{n} \beta_j K_j, \beta_j \geq 0 \tag{5}$$

Using the results from the previous section, each of these n kernels is associated with an empirical feature space:

$$
\begin{aligned}
K_j &= X_j^T X_j \\
\dot{K}_j &= X_j^T \dot{X}_j
\end{aligned}
\tag{6}
$$

where X_j and $\dot{X}_j$ are the training and test samples in the empirical feature space associated with the j^{th} kernel, respectively, and $X_j \in \mathbb{R}^{r_j}, \dot{X}_j \in \mathbb{R}^{r_j}$ for $j = 1, \cdots, n$ where r_j is the rank of K_j.

From the definition of dot product, it directly follows that taking the unweighted sum of the n base kernels is equivalent to taking the Cartesian product of the empirical feature spaces associated with the base kernels. On the other hand, taking the weighted sum of the base kernels as in Eq. 5 is equivalent to taking the Cartesian product of the base empirical feature spaces after scaling these spaces with $\sqrt{\beta_1}, \cdots, \sqrt{\beta_n}$. In this light, the goal of all MKL methods in [7–18] is to learn an optimal scaling such that some class separation criterion is maximised.

We illustrate the geometrical interpretation of taking the unweighted sum of two kernels in Fig. 1. Note that for the sake of visualisation we assume in Fig. 1 that the empirical feature spaces of both K_1 and K_2 are 1-dimensional, i.e., the ranks of both K_1 and K_2 are 1. In practice, however, both spaces can be up to m dimensional.

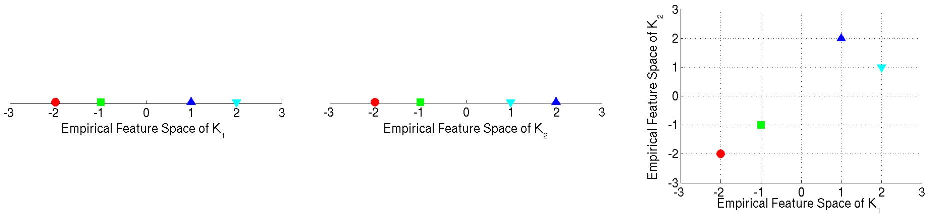

Fig. 1. Geometrical interpretation of taking the sum of two kernels. Left: the empirical feature space of K_1. Middle: the empirical feature space of K_2. Right: the empirical feature space of $K_1 + K_2$.

4 Kernel Combination with Augmented Kernel Matrix

Despite various ways of learning the optimal kernel weights, a linear combination of kernels leads to a composite kernel matrix $K = \sum_{j=1}^{n} \beta_j K_j$ which has a size $m \times m$. If SVM or kernel FDA is used as a classifier in the subsequent step, the decision function is in the form of:

$$
f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{7}
$$

where $K(\mathbf{x}, \mathbf{x}_i)$ is the dot product between a new test sample and the i^{th} training sample in the composite empirical feature space, $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_m)$ and b are learnt by maximising the margin (SVM) or by maximising the ratio between inter- and intra- class scatters (FDA). In both cases, there are m learnt coefficients $\boldsymbol{\alpha}$ if we ignore the bias term b, one for each training sample. This implies that the contribution of a given base kernel (thus a feature channel) is fixed for all training samples, which may be an unnecessarily strong constraint. For example, in an object recognition problem, one particular sample may carry more shape information than colour information and vice versa for another sample.

Instead of linear combination of kernels, we consider a different kernel combination scheme. We define an operation on two symmetric PSD training kernel matrices K_1 and K_2, $K_1 \oplus K_2$, as constructing an augmented block diagonal matrix K such that:

$$K = K_1 \oplus K_2 = \begin{pmatrix} K_1 & 0 \\ 0 & K_2 \end{pmatrix} \tag{8}$$

The zeros on the off diagonal reflect the fact that we do not have any knowledge about the cross terms between the two kernel matrices.

Let the eigen decomposition of K_1 and K_2 be:

$$K_1 = V_1 \Lambda_1 V_1^T \tag{9}$$

$$K_2 = V_2 \Lambda_2 V_2^T \tag{10}$$

where Λ_1 and Λ_2 are the diagonal matrices containing the r_1 and r_2 non-zero eigen values of K_1 and K_2 respectively, and V_1 and V_2 are the $m \times r_1$ and $m \times r_2$ matrices containing the r_1 and r_2 associated eigen vectors, respectively:

$$V_1 = \{\mathbf{v}_1^1, \mathbf{v}_2^1, \cdots, \mathbf{v}_{r_1}^1\} \tag{11}$$

$$V_2 = \{\mathbf{v}_1^2, \mathbf{v}_2^2, \cdots, \mathbf{v}_{r_2}^2\} \tag{12}$$

where the m dimensional vector $\mathbf{v}_s^j$ is the s^{th} eigen vector of kernel K_j.

On the other hand, let the eigen decomposition of $K = K_1 \oplus K_2$ be:

$$K = V \Lambda V^T \tag{13}$$

Since K is a block diagonal matrix with K_1 and K_2 on its diagonal, Λ is a diagonal matrix containing the $r_1 + r_2$ eigen values of K, and these are simply the union of the r_1 eigen values of K_1 and the r_2 eigen values of K_2. Without loss of generality, we order Λ such that its first r_1 diagonal elements are the eigen values of K_1, and the last r_2 are those of K_2. Moreover, we order the $2m \times (r_1 + r_2)$ eigen vector matrix V accordingly:

$$V = \{\tilde{\mathbf{v}}_1^1, \tilde{\mathbf{v}}_2^1, \cdots, \tilde{\mathbf{v}}_{r_1}^1, \tilde{\mathbf{v}}_1^2, \tilde{\mathbf{v}}_2^2, \cdots, \tilde{\mathbf{v}}_{r_2}^2\} \tag{14}$$

Using again the property of block diagonal matrix, the columns of V are simply the eigen vectors of K_1 and K_2 padded with m zeros:

$$\tilde{\mathbf{v}}_s^1 = ({\mathbf{v}_s^1}^T, 0, \cdots, 0)^T \quad s = 1, \cdots, r_1 \tag{15}$$

$$\tilde{\mathbf{v}}_s^2 = (0, \cdots, 0, {\mathbf{v}_s^2}^T)^T \quad s = 1, \cdots, r_2 \tag{16}$$

Now the training vectors in the empirical feature spaces associated with K_1, K_2 and K, i.e., X_1, X_2 and X, can be computed using Eq. 3. Exploiting the relation between Λ_1, Λ_2 and Λ, and that between V_1, V_2 and V, it directly follows that X is an $(r_1+r_2)\times 2m$ block diagonal matrix with X_1 and X_2 on its diagonal:

$$X = \begin{pmatrix} X_1 & 0 \\ 0 & X_2 \end{pmatrix} \tag{17}$$

where the $r_1 \times m$ matrix X_1 and $r_2 \times m$ matrix X_2 are the training vectors in the empirical feature spaces associated with K_1 and K_2, respectively.

The geometrical interpretation of this AKM scheme for kernel combination is illustrated in Fig. 2, where for the sake of visualisation we assume that the empirical feature spaces of both K_1 and K_2 are 1-dimensional. In practice, however, both spaces can be up to m dimensional. It is clear in Fig. 2 that by combining two kernels using the AKM scheme we have $2m$ training samples. This results in $2m$ coefficients in the decision function trained using the augmented kernel matrix, and as a result it allows training samples to have different contribution through the feature channels. We will show the benefit of this experimentally in the next section.

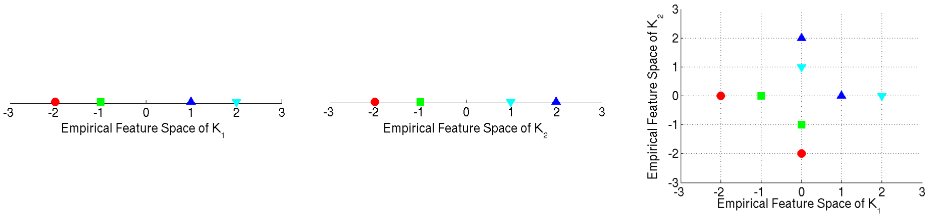

Fig. 2. Geometrical interpretation of augmenting the kernel matrix using Eq. 8. Left: the empirical feature space of K_1. Middle: the empirical feature space of K_2. Right: the empirical feature space of $K_1 \oplus K_2$.

For test kernels, the $\oplus$ operation is defined as:

$$\dot{K} = \dot{K}_1 \oplus \dot{K}_2 = \begin{pmatrix} \dot{K}_1 \\ \dot{K}_2 \end{pmatrix} \tag{18}$$

As a result the composite test kernel $\dot{K}$ has a size $2m \times l$. By applying the decision function, which has $2m$ coefficients, on $\dot{K}$, we obtain one score for each test sample.

5 Experiments

In this section, we validate the usefulness of the proposed AKM kernel combination scheme on both UCI datasets and an object recognition dataset. SVM and kernel FDA are the two most popular kernel based classification methods. It has been shown [20, 4] that SVM and kernel FDA have strong connections. In fact, the only difference between them is that SVM uses a hinge loss for computing the empirical loss while FDA uses a squared loss. In our experiments we choose kernel FDA as classifier to compare several kernel combination schemes: the ℓ_1 multiple kernel FDA (MK-FDA) of [14], the ℓ_2 MK-FDA of [15], ℓ_∞ MK-FDA where all the base kernels get equal weights, and the AKM scheme proposed in this paper.

5.1 UCI datasets

We show in this section results on four datasets from the UCI machine learning repository [21], namely, *sonar*, *heart*, *iris* and *wine*. Among these datasets, the first two are binary problems while the last two are multiclass problems. For each dataset, we first normalise each feature in the input space to between -1 and 1. We then construct 10 RBF kernels using the normalised features with the following kernel function $K_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = \exp^{-||\mathbf{x}_i - \mathbf{x}_j||^2/\sigma^2}$, where σ is set to $\{10^{-1/2}, 10^{-1/3}, 10^{-1/6}, 10^0, 10^{1/6}, 10^{1/3}, 10^{1/2}, 10^{2/3}, 10^{5/6}, 10^1\}$. All the kernels are then centred in their empirical feature spaces[6]. For each dataset, we randomly split all samples (or equivalently the kernel matrix) into a training set and a test set using a ratio of 8 : 2. We repeat experiments 1000 times using 1000 random splits and report the mean error rate and standard deviation.

The first three methods under comparison all use linear combination of kernels. In ℓ_1 MK-FDA and ℓ_2 MK-FDA, the optimal kernel weights are learnt; while in ℓ_∞ MK-FDA the kernel weights are ones for all kernels. Once the kernel weights are obtained, the composite training kernel and test kernel can be computed. For the proposed AKM scheme, augmented training kernel and test kernel are constructed using Eq. 8 and Eq. 18, respectively.

Once the training and test kernels have been obtained using the four methods, we apply FDA to find the optimal projection and compute the classification error rate. In our experiments, the spectral regression based FDA implementation in [22] is employed for its efficiency. In this implementation, a γ parameter controls the trade-off between empirical error and generalisation of the decision function. For each dataset and each of the 1000 splits, we repeat 11 times using 11 γ values: $\{0, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^{+1}\}$.

We report the error rate and standard deviation of the four kernel combination methods in Table 1 and Table 2. For each γ value, we compute the mean error rate of the 1000 runs, and report in Table 1 the smallest error rate and the associated standard deviation for each method. We also select the optimal γ for each of the 1000 runs and report the error rate and standard deviation in Table 2. Briefly speaking, results in Table 1 and Table 2 are obtained with γ optimised over the mean of all 1000 runs and over each individual run, respectively.

From both tables we can see that MK-FDAs with different regularisation norms can be advantageous on different datasets. This is because different norm tend to produce kernel weights with different levels of sparsity: the smaller the norm, the higher the sparsity. As a result, MK-FDA with different norms are suitable for kernel sets with various levels of intrinsic sparsity. On the other hand, the proposed AKM scheme outperforms all versions of MK-FDA, which are already the state-of-the-art classifiers, on two out of four datasets, and is comparable on the other two.

Table 1. Mean error rate. γ optimised over the mean of 1000 runs.

	ℓ_1 MK-FDA	ℓ_2 MK-FDA	ℓ_∞ MK-FDA	AKM FDA
sonar	13.5±5.0	14.2±5.2	13.9±5.1	$\mathbf{11.9 \pm 4.6}$
heart	17.5±4.7	$\mathbf{17.0 \pm 4.6}$	17.2±4.6	17.9±4.7
iris	5.1±3.8	4.7±3.5	4.6±3.6	$\mathbf{4.1 \pm 3.2}$
wine	5.6±9.7	$\mathbf{1.5 \pm 2.0}$	$\mathbf{1.5 \pm 2.0}$	2.5±2.6

Table 2. Mean error rate. γ optimised over each individual run.

	ℓ_1 MK-FDA	ℓ_2 MK-FDA	ℓ_∞ MK-FDA	AKM FDA
sonar	11.9±4.6	12.9±4.7	12.9±4.7	$\mathbf{9.9 \pm 4.0}$
heart	16.5±4.5	$\mathbf{16.1 \pm 4.4}$	16.3±4.4	16.6±4.4
iris	4.3±3.4	4.1±3.2	4.0±3.3	$\mathbf{2.9 \pm 2.7}$
wine	4.9±9.5	1.2±1.7	$\mathbf{1.1 \pm 1.7}$	1.9±2.3

5.2 Pascal VOC08 dataset

The Pascal visual object classes (VOC) challenge provides a yearly benchmark for comparison of object recognition methods, with one of the most challenging datasets in the object recognition / image classification community [27]. The VOC 2008 development dataset consists of 4332 images of 20 object classes such as aeroplane, cat, person, etc. The set is divided into a pre-defined training set with 2111 images and a validation set with 2221 images. In our experiments, the training set is used for training and the validation set for testing.

The classification of the 20 object classes is treated as 20 independent binary problems. Average precision (AP) [23] is used to measure the performance of each binary classifier. The mean of the APs of the 20 classes, MAP, is used as a measure of the overall performance.

SIFT descriptor [24] and codebook technique [25] are used to generate kernels. The combination of two sampling techniques (dense and Harris-Laplace), five colour variants of SIFT descriptors [26], and three ways of dividing an image into spatial location grids results in $2 \times 5 \times 3 = 30$ base kernels.

We show in Table 3 the MAPs of the four kernel combination methods. The γ parameter is set to 0, with which optimal MAPs are achieved for all four methods. The poor performance of ℓ_1 MK-FDA indicates that the base kernels carry complementary information. In such a case, non-sparse kernel selection result is

Table 3. MAPs of the four kernel combination methods with 30 base kernels.

	ℓ_1 MK-FDA	ℓ_2 MK-FDA	ℓ_∞ MK-FDA	AKM FDA
MAP	45.1	46.3	46.2	**46.4**

favoured since it does not lead to information loss. The proposed AKM scheme outperforms ℓ_2 and ℓ_∞ MK-FDAs by seemingly small margins. However, it is worth noting that a difference of 0.1 in MAP is more significant than it may appear to be. For example, the leading methods in PASCAL VOC classification competitions typically differ only by a few tenths of a percent in MAP. Moreover, uniform FDA was used by the method that produced the highest MAP in PASCAL VOC 2008 classification challenge [27]. This means the proposed AKM scheme improves over the state-of-the-art classifier for object recognition.

In both the experiments on UCI datasets and on VOC08 dataset, ℓ_1 and ℓ_2 MK-FDAs are implemented in Matlab and the associated optimisation problems are solved with the Mosek optimisation software [2]. The stopping threshold ϵ in ℓ_1 and ℓ_2 MK-FDAs is set to 5×10^{-4}.

6 Conclusions

In this paper we have presented a novel approach to combining multiple kernels where the kernels are computed from different information channels. In contrast to traditional methods that learn a linear combination of n kernels of size $m \times m$, resulting in m coefficients in the trained classifier, we propose a method that can learn $n \times m$ coefficients. This allows to assign different importance to the information channel per example rather than per kernel. We analyse the proposed kernel combination in empirical feature space and provide its geometrical interpretation. We validate the approach on both UCI datasets and an object recognition dataset, and demonstrate that it leads to classification improvements.

Acknowledgements

This work has been supported by EU IST-2-045547 VIDI-Video Project.

References

1. B. Scholkopf and A. Smola, *Learning with Kernels.* MIT Press, 2002.
2. J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.
3. V. Vapnik, *The Nature of Statistical Learning Theory.* Springer-Verlag, 1999.
4. S. Mika, “Kernel fisher discriminants,” PhD Thesis, University of Technology, Berlin, Germany, 2002.

[2] http://www.mosek.com

5. G. Baudat and F. Anouar, “Generalized discriminant analysis using a kernel approach,” *Neural Computation*, vol. 12, pp. 2385–2404, 2000.
6. B. Scholkopf, A. Smola, and K. Muller, “Kernel principal component analysis,” *Advances in Kernel Methods: Support Vector Learning*, pp. 327–352, 1999.
7. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. Jordan, “Learning the kernel matrix with semidefinite programming,” *JMLR*, vol. 5, pp. 27–72, 2004.
8. F. Bach and G. Lanckriet, “Multiple kernel learning, conic duality, and the smo algorithm,” in *ICML*, 2004.
9. S. Sonnenburg, G. Ratsch, C. Schafer, and B. Scholkopf, “Large scale multiple kernel learning,” *JMLR*, vol. 7, pp. 1531–1565, 2006.
10. A. Rakotomamonjy, F. Bach, Y. Grandvalet, and S. Canu, “Simplemkl,” *JMLR*, vol. 9, pp. 2491–2521, 2008.
11. A. Zien and C. Ong, “Multiclass multiple kernel learning,” in *ICML*, 2007, pp. 1191–1198.
12. S. Ji, L. Sun, R. Jin, and J. Ye, “Multilabel multiple kernel learning,” in *NIPS*, 2008.
13. S. Kim, A. Magnani, and S. Boyd, “Optimal kernel selection in kernel fisher discriminant analysis,” in *ICML*, 2006.
14. J. Ye, S. Ji, and J. Chen, “Multi-class discriminant kernel learning via convex programming,” *JMLR*, vol. 9, pp. 719–758, 2008.
15. F. Yan, J. Kittler, K. Mikolajczyk, and A. Tahir, “Non-sparse multiple kernel learning for fisher discriminant analysis,” in *ICDM*, 2009.
16. M. Kloft, U. Brefeld, P. Laskov, and S. Sonnenburg, “Non-sparse multiple kernel learning,” in *NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, 2008.
17. M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, “Efficient and accurate lp-norm mkl,” in *NIPS*, 2009.
18. C. Cortes, M. . Mohri, and A. Rostamizadeh, “L2 regularization for learning kernels,” in *UAI*, 2009.
19. B. Scholkopf, S. Mika, C. Burges, P. Knirsch, K. Muller, G. Ratsch, and A. Smola, “Input space versus feature space in kerenl-based methods,” *IEEE TranSactions on Neural Networks*, vol. 10(5), pp. 1000–1017, 1999.
20. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer, 2002.
21. D. Newman, S. Hettich, C. Blake, and C. Merz, “Uci repository of machine learning databases,” http://www.ics.uci.edu/ mlearn/MLRepository.html, 1998.
22. D. Cai, X. He, and J. Han, “Efficient kernel discriminant analysis via spectral regression,” in *ICDM*, 2007.
23. C. Snoek, M. Worring, J. Gemert, J. Geusebroek, and A. Smeulders, “The challenge problem for automated detection of 101 semantic concepts in multimedia,” in *ACM Multimedia Conference*, 2006, pp. 421–430.
24. K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *PAMI*, vol. 27(10), pp. 1615–1630, 2005.
25. J. Gemert, J. Geusebroek, C. Veenman, and A. Smeulders, “Kernel codebooks for scene categorization,” in *ECCV*, 2008.
26. K. Sande, T. Gevers, and C. Snoek, “Evaluation of color descriptors for object and scene recognition,” in *CVPR*, 2008.
27. M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results,” http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.