Two-stage Augmented Kernel Matrix for Object Recognition

Muhammad Awais, Fei Yan, Krystian Mikolajczyk, and Josef Kittler

Centre for Vision, Speech and Signal Processing (CVSSP),
University of Surrey, UK
`{m.rana,f.yan,k.mikolajczyk,j.kittler}@surrey.ac.uk`

Abstract. Multiple Kernel Learning (MKL) has become a preferred choice for information fusion in image recognition problem. Aim of MKL is to learn optimal combination of kernels formed from different features, thus, to learn importance of different feature spaces for classification. Augmented Kernel Matrix (AKM) has recently been proposed to accommodate for the fact that a single training example may have different importance in different feature spaces, in contrast to MKL that assigns same weight to all examples in one feature space. However, AKM approach is limited to small datasets due to its memory requirements. We propose a novel two stage technique to make AKM applicable to large data problems. In first stage various kernels are combined into different groups automatically using kernel alignment. Next, most influential training examples are identified within each group and used to construct an AKM of significantly reduced size. This reduced size AKM leads to same results as the original AKM. We demonstrate that proposed two stage approach is memory efficient and leads to better performance than original AKM and is robust to noise. Results are compared with other state-of-the art MKL techniques, and show improvement on challenging object recognition benchmarks.

1 Introduction

Object and image recognition has undergone a rapid progress in last decade due to advances in both features design and kernel methods [15] in machine learning. In particular, recent introduction of multiple kernel learning methods set a new direction of research. The state-of-the-art object and image recognition algorithms use multiple kernel learning based methods for classification, dimensionality reduction and clustering in a wide range of applications [15], [18]. Due to importance of complementary information in MKL, much research was done in field of feature design [9], [13] to diversify kernels, leading to large number of kernels in typical visual classification tasks. Kernels are often computed independently of each others thus may be highly informative, noisy or redundant. Proper selection and fusion of kernels is therefore crucial to maximize performance and to address the efficiency issues in large scale visual recognition applications.

MKL was first proposed by Lancriet et al. [6] using semi-definite programming, where kernel weights were learned by maximizing soft margin between two classes. Since algorithm proposed in [6] was limited to small kernel sizes and low number of kernels, a number of other methods were proposed to address these problems [1], [16]. All these MKL methods focus on linear combination of kernels, in which a single kernel corresponding to a particular feature space is attributed a single weight. This is a strong constraint as it does not exploit information from individual samples in different feature spaces, e.g., in context of object recognition, some samples can carry more shape information while others may carry more texture information for same object category. To address this problem AKM was proposed [19] in which different features extracted form same sample are treated as different samples of same class. Fundamental problem with AKM is its large augmented matrix which requires a lot of memory and makes it inapplicable to large datasets. In this paper we derive primal and dual of AKM, discuss its empirical feature space and address its issues with a two stage architecture. In the first stage, groups are formed from a set of base kernels based on similarity between kernels. Next, a representative kernel for each group is learned by a linear combination of within group kernels. These representative kernels are highly informative containing most of information from each group. Our grouping approach is also useful for methods proposed in [17], [10], which assumed that kernel groups are available. We further reduce complexity of AKM by exploiting independence of empirical feature spaces of representative kernels in augmented kernel matrix. Due to independence, only most influential training examples from the representative kernels can be used to build an AKM of a reduced size without compromising its performance. In second stage, AKM scheme is used to include contribution of most influential samples from all representative kernels in final classifier. Our experiments show that proposed strategy of grouping kernels and selecting subsets of training examples makes approach efficient and improves classifier performance. AKM results are compared to other MKL techniques, using different regularization, ℓ_1, ℓ_2, and ℓ_∞ norms. We demonstrate significant improvement on challenging object recognition benchmark Pascal VOC 2007 [3] and multiclass flower datasets [12], [11]. Moreover, proposed memory efficient learning strategy is also applicable in other MKL techniques which is particularly important in large scale data scenario.

Rest of paper is organized as follows. In section 2 we discuss the structure of AKM matrix and derive its primal and dual for SVM. We then compare empirical feature spaces of a linear combination MKL and AKM schemes. Our proposed two stage multiple kernel learning for AKM is presented in section 3. In section 4 we present the result and compare with other state-of-art MKL methods for object recognition.

2 Linear Combination vs Augmented Kernel Matrix

We first present structure of AKM and give primal formulations for a binary classification. We then present concept of empirical feature space for AKM scheme.

Consider we are given m training samples (x_i, y_i), where x_i is a sample in input space and $y_i \in \pm 1$ is its label. Feature extraction results in n training kernels (K_p) of size $m \times m$ and corresponding n test kernels ($\dot{K_p}$) of size $m \times l$. Each kernel $K_p = \langle \Phi_p(x_i), \Phi_p(x_j) \rangle$ implicitly maps samples x_i from input space to feature space with mapping function ($\Phi_p(x_i)_{p=1,...,n}$). In MKL aim is to find linear combination $\sum_{p=1}^{n} \beta_p K_p$, normal vector $\mathbf{w}$ and bias b of separating hyperplane simultaneously such that soft margin between two classes is maximized. Primal and its corresponding dual for a linear combination of kernels are derived for various formulations in [5], [6], [1], [16]. The decision function is then $f(x) = sign(\sum_{i=1}^{m} \alpha_i y_i k(x_i, x) + b)$, where $k(x_i, x)$ is dot product of test sample x with i^{th} training sample in feature space, $\alpha \in \mathbb{R}^m$, and b are Lagrange multiplier and bias. Contribution of a given feature channel is fixed by β_p, which may be suboptimal, as in a particular feature channel one example can carry more shape information than texture or vice versa. In contrast, in AKM [19], given the set of base training kernels augmented kernel is defined as follows:

$$K = K_1 \oplus \cdots \oplus K_n = \begin{bmatrix} K_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & K_n \end{bmatrix} \tag{1}$$

where base kernels are on diagonal. Zeros on off diagonal reflect that there is no cross terms between different kernel matrices. Note that all base kernels are of size $m \times m$ while AKM is of size $(n \times m) \times (n \times m)$, thus it uses $n \times m$ training samples instead of m. The SVM primal of AKM scheme is then given:

$$\min_{\boldsymbol{w},\boldsymbol{\xi},b} \frac{1}{2} \sum_{p=1}^{n} \langle \boldsymbol{w_p}, \boldsymbol{w_p} \rangle + C \sum_{i=1}^{n \times m} \xi_i \tag{2}$$

$$\text{s.t. } y_i(\sum_{p=1}^{n} \langle \boldsymbol{w_p}, \Phi_p(x_i) \rangle + b) \geq 1 - \xi_{pi},\ \ \xi_{pi} \geq 0,\ \ i = 1, ..., m,\ \ p = 1, ..., n$$

The dual of Eq. (2) can be derived using Lagrange multiplier techniques:

$$\max_{\boldsymbol{\alpha}} \sum_{p=1}^{n} \sum_{i=1}^{m} \alpha_{pi} - \frac{1}{2} \sum_{p=1}^{n} \sum_{i,j=1}^{m} \alpha_{pi} \alpha_{pj} y_i y_j k_p(x_i, x_j) \tag{3}$$

$$s.t. \sum_{p=1}^{n} \sum_{i=1}^{m} \alpha_{pi} y_i = 0,\ \ 0 \leq \alpha \leq C,$$

Decision function of AKM is $f(x) = sign(\sum_{p=1}^{n} \sum_{i=1}^{m} \alpha_{pi} y_i k_p(x_i, x) + b$, where α_{pi} are Lagrange multipliers and x is test sample. Note that same samples from different feature channels are added as separate examples of same class, therefore one Lagrange multiplier is learnt for each sample from each feature channel.

The concept of empirical feature space is crucial to analyze spread and shape of data. Kernel matrices consist of dot products between samples in some feature spaces. These feature spaces are usually very high or even infinite dimensional. However, in [14] it is shown that there exists an empirical feature space in which the intrinsic geometry of data is identical to true feature space, thus, in many problems it is sufficient to study empirical feature space. Empirical feature spaces X and $\dot{X}$ for training kernel K of size $m \times m$ and test kernel $\dot{K}$ of size $m \times l$ can be derived by eigen value decomposition as shown in [19].

Consider a linear combination of two training kernels K_1, K_2 with sample points in r_1, r_2 dimensional empirical feature space given by matrices X_1, X_2 of sizes $r_1 \times m$ and $r_2 \times m$, respectively. By definition of a dot product, computing weighted sum of base kernels is equivalent to computing cartesian product of associated empirical feature spaces, after scaling them with $\sqrt{\beta_p},\ p = 1, ..n$. An illustration of empirical feature space is given in figure 1. K_1, K_2 are two base kernels with rank $r_1 = r_2 = 1$ i.e., the samples live in one dimensional empirical feature space as shown in figure 1(a) and (b). Note, this toy example is for illustration purpose, whereas, in practice the empirical feature spaces can be up to m dimensional. Figure 1(c) shows the empirical feature space of a sum of two kernels. Note that the number of samples in figure 1(c) is equal to m which is the same as the number of samples in K_1 and K_2.

Let K be AKM of two training kernels K_1, K_2. The matrix X of training vectors in empirical feature space associated with K can be computed by eigen value decomposition [19]. However, by exploiting property of block diagonal augmented matrix K, its associated matrix X is directly given by:

$$X = \begin{pmatrix} X_1 & 0 \\ 0 & X_2 \end{pmatrix} \tag{4}$$

where X is a block diagonal matrix of size $(r_1 + r_2) \times 2m$, with matrix X_1 and X_2 on its diagonal. The empirical feature space for augmented kernel matrix from two one-dimensional kernels K_1 and K_2 is shown in figure 1(d). Note that there are now total of $2m$ training examples in the empirical feature space of AKM.

3 Two-Stage Multiple Kernel Learning

In this section we present a two stage architecture for multiple kernel learning which combines the MKL and AKM schemes. Kernel matrix of AKM needs large amount of memory and is very slow in training of classifier. For example, the extra memory required by cross terms in a large augmented kernel matrix of n base kernels is $n(n-1)$ times larger than linear combination of these base kernels. This makes AKM less inapplicable to large datasets especially when n is large. We address this problem by introducing grouping of base kernels followed by a selection of training samples. Two stage approach serves two goals. It addresses the memory problems of AKM but also filters out noisy and redundant feature channels. Adding redundant feature channels as separate examples increases the memory requirements in AKM and adding noisy feature channel as separate examples leads to a significant performance loss. These two problems are alleviated by applying the grouping stage.

3.1 Kernel Grouping

We define multiple groups of base kernels using a similarity criterion. One such grouping criterion can be based on the modality of features or their extraction technique. For example, feature channels based on colour can belong to one group, texture based feature channels to another group and shape based ones to yet another group. However, this kind of grouping is not automatic and needs prior information about input spaces of kernel which may not be available. We exploit Kernel Alignment [2] as a measure of similarity between kernels to group then in unsupervised manner. Given an unlabeled sample set $S = \{x_i\}_{i=1}^{m}$, we use the Frobenius inner product between kernel matrices i.e., $\langle K_1, K_2 \rangle_F = \sum_{i,j=1}^{m} K_1(x_i, x_j) K_2(x_i, x_j)$. The empirical alignment between kernels with respect to the set S is defined as:

$$\hat{A}(S, K_1, K_2) = \frac{\langle K_1, K_2 \rangle_F}{\sqrt{\langle K_1, K_1 \rangle_F \langle K_2, K_2 \rangle_F}} \tag{5}$$

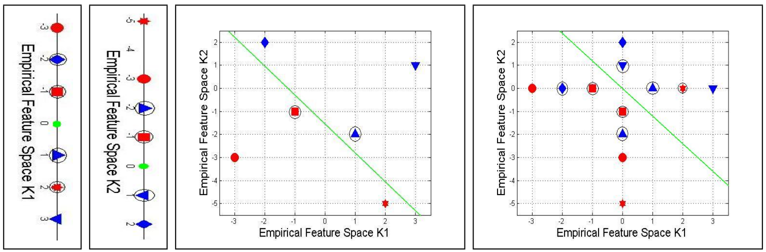
where K_i is the kernel matrix for the sample S. In [2] concentration and generalization of kernel alignment was introduced and proved. Concentration means that the probability of an empirical estimate deviating from its mean can be bounded as an exponentially decaying function of that deviation. In other words, the alignment is little dependent on the training set S as shown by theorem 3 in [2]. Generalization (test error) of a simple classification function is related to the value of the alignment as shown by theorem 4 in [2].

Using kernel alignment $\hat{\boldsymbol{A}}(\boldsymbol{S}, k_1, k_2)$ defined in Eq. (5) as a similarity measure we preform agglomerative clustering to find g groups of kernels. We initialize all kernels as clusters and merge two most similar clusters at a time. Similarity between two clusters is defined as largest distance between all possible pairs of clusters members. This continues until g groups re obtained. We used agglomerative as opposed to k-means to make it independent to initialization. Kullback-Leibler divergence can also be used as a similarity criterion between kernels [7].

Learning a linear combination of kernels within a group can discard or downweight redundant or noisy kernels thus result in a better kernel. Moreover, linear combination leads to more compact representation without loss of information. Therefore, for each group, MKL-SVM methods using ℓ_1, ℓ_2 and ℓ_∞ norms are applied to obtain the representative kernels. The kernel that obtains the highest score on the validation data is used as group representant. Thus, the grouping and within group combination results in a set of representative kernels containing most of the information from various feature channels.

3.2 Selection of Training Samples

Kernel grouping partially addresses the issue of large AKM matrix. However, the matrix can be further reduced without compromising the performance by selecting only the samples from representative kernels which are crucial for classification. The decision function of SVM is determined by the α_i, one for each



(a) (b) (c) (d)

Fig. 1. Empirical feature spaces for Multiple kernels: (a) empirical feature space for K_1; (b) empirical feature space for K_2; (c) empirical feature space for $K_1 + K_2$; (d) empirical feature space for $K_1 \oplus K_2$.

training sample. The α_i are non-zero for the support vectors only. Hence, for a single kernel, support vectors are sufficient for classification and all other samples can be discarded without performance degradation. This is supported by the fact that the feature spaces do not interfere with each other due to the structure of the augmented kernel matrix (cf. Eq. (1)). It can be proved by considering the dual of AKM, Eq. (3), which can be rewritten as follows:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \sum_{i=1}^{m} \alpha_{1i} - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_{1i}\alpha_{1j} y_i y_j \langle \Phi_1(x_i), \Phi_1(x_j) \rangle + \ldots + \\ \sum_{i=1}^{m} \alpha_{ni} - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_{ni}\alpha_{nj} y_i y_j \langle \Phi_n(x_i), \Phi_n(x_j) \rangle \\ s.t. \sum_{i=1}^{m} \alpha_{1i} y_i + \ldots + \sum_{i=1}^{m} \alpha_{ni} y_i = 0, \;\; 0 \le \alpha \le C, \end{aligned} \tag{6}$$

The first constraint in Eq. (6) is the sum of constraints for n kernels. The support vectors for all individual kernels together satisfy this constraint and thus lie in the feasible set of the optimization problem in Eq. (6). This is also illustrated by a toy example of a binary classification in figure 1. All the support vectors in empirical feature space for base kernels K_1 and K_2 are shown by the enclosing black circles and the hyperplane is represented in green at origin in figure 1(a) and (b), respectively. Figure 1(c) shows the empirical feature space of unweighted linear combination of base kernel. There are only two support vectors in figure 1(c), and the classes are separated by the hyperplane. However, the separability of the training set does not necessarily guarantee better performance as it depends upon the generalization to the test set [15]. Figure 1(d) is the empirical feature space of AKM combination of base kernels. Feature spaces of two base kernels are orthogonal to each other. There are $2m$ training samples and all the support vectors of kernel K_1 and K_2 are support vectors of AKM due to the orthogonality of their feature space. It is clear from Eq. (6) and figure 1, that the support vectors of representative kernels from each group are sufficient to construct the

AKM matrix as the Lagrange multipliers of support vectors lie in the feasible set of Eq. (6). The use of support vectors only for different combinations of kernels is validated empirically in section 4.

4 Experiments and Discussion

This section presents experimental results on challenging binary and multiclass object recognition datasets Pascal VOC 2007 [3], Oxford Flower 17 [12] and Oxford Flower 102 [11].

Pascal VOC 2007 [3] consists of 20 object classes with 9963 image examples. The classification of 20 object categories is handled as 20 independent binary classification problems (as recommended by organizers of Pascal challenge). We present results using average precision (AP) [3], which is proportional to area under precision recall curve. Mean average precision (MAP) is computed by averaging scores for all 20 classes.

We compute 20 kernels by combining features introduced in [9], [13] with 2 sampling strategies (dense, interest points) and spatial location grids [8]: whole image (1x1), horizontal bars (1x3), vertical bars (3x1) and image quarters (2x2). In experiments we use SVM to compare several kernel combination schemes and two stage AKM scheme proposed in this paper. The multiple kernel SVM (MK-SVM) schemes differ by regularization norms used during learning, which include ℓ_1 [16], ℓ_2 [5], and ℓ_∞ (equal weights). We divide 20 kernels into 4 groups as discussed in section 3.1. For each group, MKL-SVM methods using ℓ_1, ℓ_2 and ℓ_∞ norms are applied to obtain representative kernels. Results for various learning techniques are presented in table 1.

Consistently lower performance of ℓ_1-norm, which typically leads to sparsely selected kernels, indicates that most of base kernels carry complementary information. Therefore, non-sparse multiple kernel methods, ℓ_2-norm and ℓ_∞-norm, give better results. Proposed two stage AKM scheme outperforms other MKL combination schemes. In case of ℓ_2 within group and AKM between groups, (AKM, ℓ_2), we obtain an improvement of 0.6%, and in case of ℓ_∞ within group and AKM between groups, an improvement of 0.7% over all linear combinations of MKL-SVM. In case of informative kernels, use of kernel grouping achieves comparable performance to corresponding non-grouping schemes. The best performance of state-of-the-art multiple kernel learning for these kernels is 62.1% , as shown in table 1 while performance of winning method for this challenge is 59.4% [3]. We beat winning method by 3.4%, moreover, 0.7% improvement by proposed two stage AKM over state-of-the-art MKL is still significant given that all kernels are highly informative due to carefully designed features. For example, leading methods in PASCAL VOC often differ by a fraction of a percent in MAP. It is important to note that AKM on its own is giving 61.0%, however, when it is used together with grouping stage it is performing 1.8% better. It is because linear combination within grouping stage gives good representative kernel with less noisy or redundant data. These highly informative representative kernel should be combined with AKM scheme so that information in each

Table 1. MAP of PASCAL VOC 2007 with various MKL and AKM approaches.

within group between groups	no grouping	linear ℓ_1	linear ℓ_2	linear ℓ_∞
linear ℓ_1	56.0	55.3	56.5	56.6
linear ℓ_2	61.4	60.8	61.3	56.5
linear ℓ_∞	62.1	61.1	62.1	62.0
AKM	61.0	60.8	**62.7**	**62.8**

example of these kernels is exploited. We expect grouping scheme to show better performance if there are noisy or redundant kernels in set as shown by noisy feature channels experiment in next section.

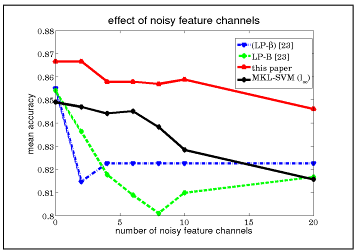
We have also validated empirically the selection of support vectors for AKM on 20 binary classification problems of the Pascal 2007 [3] dataset. Only 0.3% to 0.5% of the support vectors of AKM differs from the union of individual support vectors of representative kernels, while the MAP results are same up to sixth decimal place. However, due to use of the significant examples only, we are using 3 to 4 times less samples per base kernel. Hence, size of AKM matrix is 60% to 70% less than original size without compromising performance. It is important to note that it is not possible to apply AKM without selection of significant examples in this benchmark due to memory requirements. We have used 4 groups of kernels thus AKM kernel is even smaller than original kernel of size 5011 × 5011. Note that for each group a classifier has to be trained i.e. 4 in this experiment. This is however done efficiently on small kernels and acceptable considering performance gain achieved over other multiple kernel learning methods. Moreover, in α-step of alternative MKL techniques [16], [5] we have to train linear combination of base kernels for different regularization norms several times before obtaining optimal weights values β for base kernels. All results presented for AKM in this paper are obtained using "support vectors only scheme".

Oxford Flower 17 [12] dataset consists of 17 categories with 80 images in each category. Dataset is split into training, validation and test using 3 predefined random splits. We have used used seven distance matrices provided online. Features used to compute these distance matrices include different types of shape, texture and color based descriptors [12]. We have used SVM as classifier and follow one-vs-all setup for multiclass classification [12]. We train an AKM classifier for each category and use the maximum response of the classifiers for each example to obtain the label and score for evaluation. Regularization parameter for the SVM is in the range $C \in \{10^{(-2,-1,\ldots,3)}\}$.

Results are given in figure 2(a). For comparison we use recent evaluation results from [4] of state-of-the-art feature fusion techniques including MKL and boosting based classifier fusion. There are two baseline techniques, MKL-prod-SVM and MKL-avg-SVM, which are obtained from element wise product and averaging of base kernels and classifying with SVM. MKL baseline for kernel product gives the highest score of 85.5%. Moreover, it is very simple and fast in comparison to other MKL methods in figure 2(a). Our proposed scheme based on AKM gives 86.7%, which is better than all MKL and Boosting based methods.

ML methods	Accuracy
MKL-prod-SVM [4]	85.5 ± 1.2
MKL-avg-SVM [4]	84.9 ± 1.9
CG-Boost [4]	84.8 ± 2.2
MKL (Simple) [4]	85.2 ± 1.5
LP-β [4]	85.5 ± 3.0
LP-B [4]	85.4 ± 2.4
AKM	**86.7 ± 2.7**

(a)



(b)

ML methods	Accuracy
MKL [11]	72.8
AKM	**73.0**

(c)

Fig. 2. (a) Mean accuracy on Oxford Flower17 and comparision with different machine learning methods; (b) Oxford Flower17. Mean Accuracy of diffierent fusion methods under noisy feature channels; (c) Mean accuracy on Oxford Flower 102 dataset.

We also investigate effect of adding random feature channels on different fusion schemes. In addition to 7 informative kernels of Flower17 dataset we have generated 20 RBF kernels from 20 sets of random vectors. We started with all informative kernels, i.e., zero noisy kernels, then we added different number of noisy kernel. Mean accuracy of different state-of-the-art methods under noisy channels is presented in figure 2(b). MKL baseline drops down significantly with the number of noisy kernels while two-stage AKM is robust to noisy feature channels and perform significantly better than MKL or boosting based approaches.

Oxford Flower 102 [11] is an extended multi-class dataset containing 102 flower categories. The dataset is split into training, validation and test using predefined splits. For experiments we have used 4 χ^2 distance matrices provided online. The details of the features used to compute these distance matrices can be found in [11]. The experimental setup is the same as for Oxford Flower 17. AKM is performing comparable to MKL as shown in figure 2(c).

5 Conclusions

In this paper we have presented a novel two stage multiple kernel learning approach for augmented kernel matrix. The proposed method addresses the complexity problems of AKM and makes it robust to redundant and noisy kernels. We propose automatic grouping of kernels based on kernel alignment by agglomerative clustering of kernels. Learning representative kernels for each group results in a small set of highly informative kernels. Learning a combination within a group discards or downweights redundant and noisy kernels thus results in an optimal kernel from a set of informative base kernels. The complexity is further reduced by exploiting the property of independence of empirical feature spaces in the AKM scheme. It allows to use only the most influential examples from each representative kernel to construct the AKM matrix. We perform experiments on challenging object recognition datasets and the results validate our technique. The proposed approach makes it possible to use the AKM method for 20 kernels with several thousands of training examples. A performance increase is observed compared to MKL based on a linear combination of all base kernels. This observation is significant as it suggests that the information in the kernels can be exploited more effectively and the classification rate increases without using additional features.

Acknowledgements. This research was supported by UK EPSRC EP/F0034 20/1 and the BBC R&D grants.

References

1. Bach, F., Lanckriet, G., Jordan, M.: Multiple Kernel Learning, Conic Duality, and the SMO Algorithm. In: ICML (2004)
2. Cristianini, N., Shawe-Taylor, J., Elisseeff, A., Kandola, J.: On Kernel-Target Alignment. In: NIPS (2001)
3. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
4. Gehler, P., Nowozin, S.: On Feature Combination for Multiclass Object Classification. In: ICCV (2009)
5. Kloft, M., Brefeld, U., Laskov, P., Sonnenburg, S.: Nonsparse Multiple Kernel Learning. In: NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels (2008)
6. Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L., Jordan, M.: Learning the Kernel Matrix with Semidefinite Programming. JMLR 5, 27–72 (2004)
7. Lawrence, N., Sanguinetti, G.: Matching Kernel through Kullback-Leibler Divergence Minimisation. In: Technical Report CS-04-12, Department of Computer Science, University of Sheffield (2005)
8. Lazebnik, S., Schmid, C., Ponce, J.: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: CVPR (2006)
9. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI 27(10), 1615–1630 (2005)
10. Nath, J., Dinesh, G., Raman, S., Bhattacharyya, C., Ben-Tal, A., Ramakrishnan, K.: On the Algorithmics and Applications of a Mixed-norm Based Kernel Learning Formulation. In: NIPS (2009)
11. Nilsback, M.E., Zisserman, A.: Automated Flower Classification over a Large Number of Classes. In: ICCVGIP (2008)
12. Nilsback, M., Zisserman, A.: A visual Vocabulary for Flower Classification. In: CVPR (2006)
13. van de Sande, K., Gevers, T., Snoek, C.: Evaluation of color descriptors for object and scene recognition. In: CVPR (2008)
14. Schölkopf, B., Mika, S., Burges, C., Knirsch, P., Müller, K., Rätsch, G., Smola, A.: Input Space Versus Feature Space in Kernel-Based Methods. NN 10(5), 1000–1017 (1999)
15. Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press (2002)
16. Sonnenburg, S., Rätsch, G., Schafer, C., Schölkopf, B.: Large Scale Multiple Kernel Learning. JMLR 7, 1531–1565 (2006)
17. Szafranski, M., Grandvalet, Y., Rakotomamonjy, A.: Composite Kernel Learning. ML 79(1), 73–103 (2010)
18. Vapnik, V.: The Nature of Statistical Learning Theory. Springer Verlag (2000)
19. Yan, F., Mikolajczyk, K., Kittler, J., Tahir, A.: Combining Multiple Kernels by Augmenting the Kernel Matrix. In: International Workshop on MCS (2010)