情報論的学習理論テクニカルレポート 2009
Technical Report on Information-Based Induction Sciences
2009 (IBIS2009)

Multiple Kernel Learning for Object Classification

Shinichi Nakajima [*] Alexander Binder [†] Christina Müller [‡] Wojciech Wojcikiewicz [§]
Marius Kloft [¶] Ulf Brefeld [‖] Klaus-Robert Müller [**] Motoaki Kawanabe [††]

Abstract: Combining information from various image descriptors has become a standard technique for image classification tasks. Multiple kernel learning (MKL) approaches allow to determine the optimal combination of such similarity matrices *and* the optimal classifier simultaneously. Most MKL approaches employ an ℓ^1-regularization on the mixing coefficients to promote sparse solutions; an assumption that is often violated in image applications where descriptors hardly encode orthogonal pieces of information. In this paper, we compare ℓ^1-MKL with a recently developed non-sparse MKL to object classification tasks. We show that the non-sparse MKL outperforms both the standard MKL and SVMs with average kernel mixtures on the PASCAL VOC data sets.

Keywords: multiple kernel learning, SVM, image classification, sparsity.

1 Introduction

Data fusion is an important topic in computer vision. Images can be represented by a multiplicity of features capturing certain aspects, including color, textures, and shapes. Unfortunately, the importance of different types of features varies with the tasks; color information, for instance, substantially increases the detection of stop signs while coloring is almost irrelevant for finding cars in images. Techniques for appropriately combining relevant features for a task at hand are therefore crucial for state-of-the-art object recognition systems.

From a machine learning view, different representations give rise to different kernel functions. Kernels define (possibly nonlinear) similarities between data points and allow to abstract learning algorithms from data. Thus, kernel machines have been successfully applied to many practical problems in various fields [19]. Given a task at hand, designing an appropriate kernel is essential for achieving good generalizations, for instance by incorporating prior assumptions and domain knowledge [9, 28]. However, in the absence of prior knowledge one has to resort to alternatives.

For object recognition tasks, combining information from various image descriptors into several kernels $K_1, \ldots, K_m$ has become a standard technique. Unfortunately, the choice of the right kernel mixture is often a matter of trial and error. As a remedy, uniform mixtures of normalized kernels [14, 26] or brute-force approaches [2] are employed frequently. However, the former approach may lead to suboptimal kernels and the latter is computationally infeasible if many kernels are to be combined.

Recently, multiple kernel learning (MKL) [13, 1, 20, 18, 27] was applied to object classification tasks involving various image descriptors [24]. Compared to uniform mixtures and brute-force approaches, MKL has the appealing property of always finding the optimal kernel combination and converges quickly as it can be wrapped around a regular support vector machine [20]. Multiple kernel learning aims at learning the optimal kernel mixture and the model parameters simultaneously. More specifically, MKL approaches find a linear mixture of the kernels, that is $K = \sum_j \beta_j K_j$. To support the interpretability of the solution, many MKL approaches promote sparse mixtures by incorporating an ℓ^1-norm constraint on the mixing coefficients. However, it has often been observed that ℓ^1-norm MKL is outperformed by the average-sum kernel $K = \sum_j K_j$. An explanation is that enforcing sparse mixtures may lead to degenerate models if the optimal kernel mixture is non-sparse. A remedy might be recently developed non-sparse variants of multiple kernel learning promoting non-sparse kernel mixtures [10].

In this contribution, we empirically compare sparse and

[*] Nikon Corporation, nakajima.s@nikon.co.jp,
[†] Fraunhofer Institute FIRST, binder@first.fhg.de,
[‡] Technische Universität Berlin, muechr@first.fraunhofer.de,
[§] Technische Universität Berlin, wojcikie@informatik.hu-berlin.de,
[¶] Technische Universität Berlin, mkloft@cs.tu-berlin.de,
[‖] Technische Universität Berlin, brefeld@cs.tu-berlin.de,
[**] Technische Universität Berlin, klaus-robert.mueller@tu-berlin.de,
[††] Fraunhofer Institute FIRST, nabe@first.fhg.de,

non-sparse MKL approaches to object classification tasks. We employ candidate kernels obtained from many different image descriptors including the 30 color SIFT features by the VOC2008 winner [22]. Our empirical results on image data sets from the PASCAL visual object classification (VOC) challenge 2007 and 2008 [8] show that the non-sparse MKL significantly outperforms the uniform mixture and ℓ^1-norm MKL.

This paper is organized as follows. In Section 2, we briefly review the underlying techniques, including sparse and non-sparse MKL. Section 3 discusses similarities between the prepared kernels. Based on this analysis, we precompute averages of similar kernels and apply MKL with a substantially reduced sets of kernels. We discuss our empirical results in Section 4 and Section 5 concludes.

2 Preliminaries

2.1 Support Vector Machines

In the supervised learning setting, we are given n training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathcal{X}$ is the input vector and $y_i \subseteq Y$. For instance, in object recognition, inputs $\boldsymbol{x}$ are frequently histograms of some image features and Y is a discrete set of objects that are to be identified in the images. Inputs are often annotated with several labels as different objects can occur in the same image. To account for these multi-label scenarios, we take a one-vs-all approach and focus on binary classification settings. That is, we have $y_i \in \{+1, -1\}$, where $y_i = +1$ denotes that at least one object from the actual category is included in the image and $y = -1$ otherwise.

Support vector machines originate from linear classifiers and maximize the margin between sample clouds of both classes. Introducing a feature mapping ψ from the input space $\mathcal{X}$ to a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, linear classifiers in $\mathcal{H}$ of the form

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \psi(\boldsymbol{x}) + b \tag{1}$$

provide a rich set of flexible classifiers in $\mathcal{X}$. The parameters $(\boldsymbol{w}, b)$ are determined by solving the optimization problem

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^n \xi_i, \\ \text{s.t.} \quad & \forall i, \quad y_i\left\{\boldsymbol{w}^\top\psi(\boldsymbol{x}_i) + b\right\} \geq 1 - \xi_i; \quad \xi_i \geq 0, \end{aligned} \tag{2}$$

where $\|\cdot\|_2$ denotes the ℓ^2 norm and $C > 0$ is a regularization constant. Notice that the spanned RKHS can be infinite-dimensional, however, translating the above formulation into the equivalent dual optimization problem prevents from dealing with features in $\mathcal{H}$ explicitly.

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,l=1}^n \alpha_i\alpha_l y_i y_l k(\boldsymbol{x}_i, \boldsymbol{x}_l) \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \ \forall i; \quad \sum_{i=1}^n y_i\alpha_i = 0. \end{aligned} \tag{3}$$

The above dual depends only on inner products (similarities) of inputs which can be alternatively computed by means of kernel functions k, given by

$$k(\boldsymbol{x}, \bar{\boldsymbol{x}}) = \langle\psi(\boldsymbol{x}), \psi(\bar{\boldsymbol{x}}).\rangle_{\mathcal{H}}.$$

Once, optimal parameters are found, these are used as plug-in estimates and the final decision function can be written as

$$f(\boldsymbol{x}) = \sum_{i=1}^n \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b.$$

Note that only a small fraction of the α's usually take non-zero values which are often called support vectors. The threshold b is determined by saturated support vectors with $\alpha = C$. Finally, we remark that we need to use different regularization constants C_+ and C_- for the positive and negative examples, respectively, to compensate the unbalanced sample sizes of the two classes [3].

2.2 Multiple Kernel Learning

Let $K_1, \ldots, K_m$ be m kernel matrices with $K_t = [k_t(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1,\ldots,n}$, obtained from different sources or features. The multiple kernel learning framework extends the regular SVM formulation by additionally learning a linear mixture of the kernels, i.e.

$$K_{\boldsymbol{\beta}} = \sum_{j=1}^m \beta_j K_j.$$

Thus, the model in Equation (1) is extended to

$$f(\boldsymbol{x}) = \sum_{j=1}^m \beta_j \boldsymbol{w}_j^\top \psi_j(\boldsymbol{x}) + b.$$

A common approach is to rephrase the above expression by incorporating the mixing coefficients into the parameter vector $\boldsymbol{w}_{\boldsymbol{\beta}} = (\sqrt{\beta_1}\boldsymbol{w}_1, \ldots, \sqrt{\beta_m}\boldsymbol{w}_m)^\top$ and the feature mapping $\psi_{\boldsymbol{\beta}}(\boldsymbol{x}_i) = (\sqrt{\beta_1}\psi_1(\boldsymbol{x}_i), \ldots, \sqrt{\beta_m}\psi_m(\boldsymbol{x}_i))^\top$. The corresponding optimization problem maximizes the generalization performance by simultaneously optimizing the parameters $\boldsymbol{w}, b, \boldsymbol{\xi}$, and $\boldsymbol{\beta}$. We obtain the common ℓ^1-norm

MKL for $p = 1$ [1, 20, 18, 27], and non-sparse MKL for $p > 1$ [10].

$$\begin{aligned} \min_{\boldsymbol{\beta},\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}_{\boldsymbol{\beta}}\|_2^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & \forall i: \quad y_i\left(\langle \boldsymbol{w}_{\boldsymbol{\beta}}, \psi_{\boldsymbol{\beta}}(\boldsymbol{x}_i)\rangle + b\right) \geq 1 - \xi_i \quad (4) \\ & \boldsymbol{\xi} \geq \mathbf{0}; \quad \boldsymbol{\beta} \geq \mathbf{0}; \quad \|\boldsymbol{\beta}\|_p \leq 1 \end{aligned}$$

Note that we resolve the regular SVM optimization problem in Equation (2) for learning with only a single kernel $m = 1$. Irrespectively of the actual value of p, the above optimization problem can be translated into a semi-infinite program [20, 10] which can be interpreted as a dualized variant of the optimization problem (4). We arrive at,

$$\begin{aligned} & \min_{\lambda,\boldsymbol{\beta}} \lambda \\ & \text{s.t. } \lambda \geq \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,l=1}^{n}\alpha_i\alpha_l y_i y_l \sum_{j=1}^{m}\beta_j k_j(\boldsymbol{x}_i,\boldsymbol{x}_l), \quad (5) \\ & \quad \forall \boldsymbol{\alpha} \in \mathbb{R}^n \quad (6) \\ & \quad 0 \leq \alpha_i \leq C,\ \forall i; \quad \sum_{i=1}^{n} y_i\alpha_i = 0; \\ & \quad \beta_j \geq 0,\ \forall j; \quad \|\boldsymbol{\beta}\|_p \leq 1 \end{aligned}$$

Initializing $\boldsymbol{\beta}$ with a uniform kernel mixture, the semi-infinite program can be optimized efficiently by interleaving the following two steps:

1. For the actual mixture $\boldsymbol{\beta}$, the solution of the regular SVM generates the most strongly violated constraint (Equation (6)).

2. With respect to set of active constraints, the optimal values of $\boldsymbol{\beta}$ and λ are identified by solving the corresponding optimization problem for $\boldsymbol{\beta}$.

The actual optimization problems for the mixing coefficients, however, differ with varying values of p. For instance, for $p = 1$, one obtains a linear program that can be solved with standard techniques. For $p = 2$, the ℓ^2-norm gives rise to a QCQP that can also be optimized with off-the-shelf QP-solvers. For different values of p, things get a bit tricky because there is hardly an ℓ^p-norm solver. Nevertheless, one can approximate the ℓ^p-norm constraint by a second-order Taylor expansion around the current estimates $\boldsymbol{\beta}^{\text{old}}$ given by

$$\begin{aligned} \|\boldsymbol{\beta}\|_p^p \quad \approx \quad & 1 - \frac{p(3-p)}{2} - (p^2 - 2p)\sum_j(\beta_j^{\text{old}})^{p-1}\beta_j \\ & + \frac{p(p-1)}{2}\sum_j(\beta_j^{\text{old}})^{p-2}\beta_j^2. \end{aligned}$$

Using the above approximation, one obtain a QCQP, which can again be optimized with standard techniques [10].

2.3 Kernel Alignment

In the remainder, we will need to analyze the similarity of kernel matrices. For this purpose, we now introduce kernel target alignment [5] as an adequate measure of similarity or hyper kernel [17].

Let $K_1 = [k_1(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1,\ldots,n}$ and $K_2 = [k_2(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1,\ldots,n}$ be the Gram matrices of kernel functions k_1 and k_2 for $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. The alignment between k_1 and k_2 is defined as the cosine of the angle between the two matrices K_1 and K_2 given by

$$\mathcal{A}(K_1, K_2) := \frac{\langle K_1, K_2\rangle_F}{\|K_1\|_F\|K_2\|_F}, \quad (7)$$

where $\langle K_1, K_2\rangle_F$ denotes the standard inner product $\langle K_1, K_2\rangle_F := \sum_{i,j=1}^{n} k_1(\boldsymbol{x}_i, \boldsymbol{x}_j)k_2(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\|K_1\|_F$ is the Frobenius norm in matrix space defined as $\|K_1\|_F := \langle K_1, K_1\rangle_F^{1/2}$.

It is important to center the kernels before computing the alignment as many classifiers, including support vector machines, are invariant against mean shifts in the reproducing kernel Hilbert spaces. The centering in the respective feature spaces is achieved by multiplying the matrix H, given by

$$H := I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$$

to the kernels K_1 and K_2 from both sides, where I is the identity matrix of size n and $\mathbf{1}$ is a column vector with all elements 1. Thus, the resulting alignment for centered kernels can be computed by

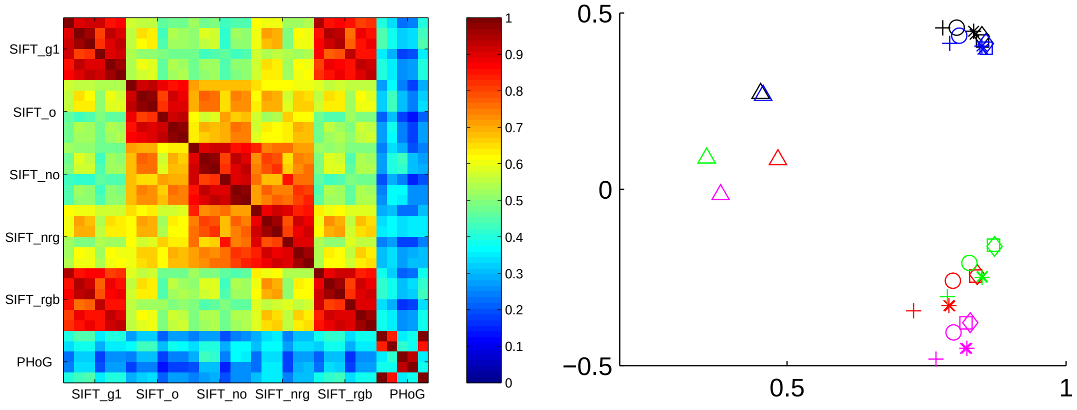
$$\mathcal{A}(HK_1H, HK_2H) = \frac{\langle HK_1H, HK_2H\rangle_F}{\|HK_1H\|_F\|HK_2H\|_F}, \quad (8)$$

where $\langle HK_1H, HK_2H\rangle_F = \text{tr}(HK_1HK_2)$, because H is a projection matrix.

3 Experiments

3.1 VOC data sets

In order to show the advantage of our procedure, we compare the performance of the different MKL procedures to



(a) (b)

図 1: Similarity between the 35 prepared kernels: (a) hyper kernel and (b) graphical representation of the similarities within the first two eigen directions. In the panel (a), 6 groups are 'SIFT_g1', 'SIFT_o', 'SIFT_no', 'SIFT_nrg', 'SIFT_rgb', and 'PHoG', while 6 elements within SIFT color channel consists of 3 pyramid levels (level 0, 1, y3) for dense grid and interest points. In the panel (b), the color channels are specified as black='g1', red='o', magenda='no', green='nrg' and blue='rgb', while the markers discriminates the pyramid levels and sampling scheme for SIFT plus PHoG (triangle), i.e. circle='dense level0', square='dense level1', diamond='dense y3', plus='interest level0', X-mark='interest points level1', star='interest points y3'.

SVMs using the average-sum kernel. We experiment on the VOC 2008 classification data set and the VOC 2007 data set [8].

The VOC 2007 data set consists of 9963 images (2501 training, 2510 validation and 4952 test) annotated with 20 object classes. The VOC 2008 data set contains 8780 images categorized into the same 20 object classes as in the VOC 2007 data. The latter is split into train, validation and test sets by the organizers (2113 for train, 2227 for validation, and 4340 for test). The ground-truth of the test set is yet disclosed by the organizers who agreed to evaluate test performance on request.

We split the multi-label problem into 20 binary classification problems using the one-vs-all strategy. That is, for each class, we define an auxiliary label $y_i = +1$ if at least one object from the actual class is included in the i-th image, and $y_i = -1$ if there is no such object in the image.[1] The evaluation is based on precision-recall (PR) curves and the principal quantitative measure is the average precision (AP) over all recall values.

We employ model selection for the SVM/MKL trade-off parameter C and for the parameter p which controls the sparseness of the multiple kernel learning. We used $p = 1 + 2^{\lambda}$, where $\lambda = \{-\infty, -5, -4, -3, -2, -1, 0, 1, \infty\}$. We resolve $p = 1$ for $\lambda = -\infty$ and obtain the unweighted-sum kernel for $p = \infty$. Furthermore, we denote ℓ^p-norm MKL with p optimized jointly for all classes as ℓ^p-joint and write ℓ^p-single for optimizing p for each class separately.

The final classifiers are obtained by re-training the respective approaches on all available data (i.e., training and holdout sets) using the previously determined optimal parameters. We report on average AP scores over 10 repetitions with different training, holdout, and test sets. The baselines SVM and ℓ^1-norm MKL are implemented using the Shogun library [20].

3.2 Image Features and Base Kernels

In our experiments, we employed the following two sets of image features. The first category contains 30 histograms of visual words (HoW) representations [6] based on color SIFT descriptors [15] which are almost the same as those applied by the winner of VOC 2008 [22]. As sampling schemes, we use a dense grid with 6 pitches and interest points from gray-scale images by the scale invariant detector [25]. For both cases, we calculated the base SIFT descriptors in 10 color channels: *g1 (grey), o1 (opponent color 1), o2, no1 (normalized o1), no2, nr (normalized red), ng (normalized green), r, g, b*. For prototype calculation and visual word assignment, the color SIFTs are combined into the following 5 groups: *g1, o=[o1,o2,g1], no=[no1,no2],*

[1] Hardly detectable objects are indicated by $y_i = 0$ by the organizers. Since these are omitted in the final evaluation we simply excluded them from the training process.

nrg=[nr,ng], rgb=[r,g,b]. For each case, we created 4000 visual words for the dense grid (800 for the interest points) by using k-means clustering. [2] Finally, we also consider three levels of the image pyramid representation [14]: for each image, its visual words are summarized into histograms for the whole image (level 0), for 4 quarter images (level 1) and for 3 horizontal stripes (y3). In total, we prepared 5 (colors) $\times 2$ (sampling) $\times 3$ (pyramid levels) $= 30$ kernels.

The second category of our image features is the pyramid histogram of oriented gradient (PHoG) [7, 2]. For each of the 5 color channels, which are same as in the first category, we compute the PHoG representations of level 2 where the 3 pyramid levels are merged by a default scheme without any adaptation. In sum, we computed 5 PHoG kernels. We used the χ^2 kernel, which has proved to be a robust similarity measure for bag of words histograms [26], where the band-width is set to the mean χ^2 distances between all pairs of training samples [12].

Although our MKL implementations are throughout efficient, simply storing all 35 kernels exceeds 1.2GB. We therefore pre-combine kernels based on a similarity analysis using kernel target alignment [5] before applying multiple kernel learning. Figure 1 (a) shows the kernel alignment score (8) between the 30 SIFT $+$ 5 PHoG kernels. We can see: (i) the kernels within the same colors are mostly similar, (ii) *g1* and *rgb* kernels are also similar and (iii) the PHoG and SIFT kernels are less similar. In order to assure our findings, we plotted the kernels in a 2-dimensional space spanned by the first and second eigenvectors of the hyper kernel obtained by a principal component analysis (PCA) and spectral clustering [16] (Figure. 1(b)). Based on this similarity analysis, we averaged 6 SIFT kernels with uniform weights within each color. By doing this, we reduced the number of base kernels to 10. We obtain 5 pre-combined SIFT and 5 PHoG kernels which are plugged into the multiple kernel learning.

3.3 Result 1: Significance Test for 10 Random Splits of VOC 2008

Before we use the official VOC 2008 data split to compare our outcomes to already published results in Section 3.4, we investigate statistical properties of the performances of the different methods. We therefore draw 2111 training, 1111 validation, and 1110 test images randomly from the labeled pool (i.e., official training and holdout split). We report on APs and standard deviations over 10 repetitions with distinct training, holdout, test sets. To test on the significance of the differences in performance, we conduct a Wilcoxon signed-ranks test for each method and class and additionally for the average AP over all classes. Table 1 shows the results.[3]

The methods whose performance are not significantly worse than the best score are marked in bold face. The ℓ^p-single MKL is always among the best performing algorithms. Its jointly-optimized counterpart ℓ^p-joint, performs similarly and attains the second best performance. Uniform weights and ℓ^1-MKL are significantly outperformed by the two non-sparse MKL variants for several object classes. The result is however not really surprising as ℓ^p-single is optimized class-wise.

Figure 2 shows the resulting kernel weights, averaged over the 10 repetitions. We see that the solutions of ℓ^p-joint distribute some weight on each kernel, achieving non-sparse solutions. The average p for ℓ^p-joint is 1.075. Furthermore, Figure 2 implies that PHoW features carry more relevant information than PHoG. Since the PHoG features do not seem to play a great role in the classification, a natural question is whether PHoG do contribute to the accuracy at all. Table 2 shows the average gain in accuracy for using PHoW kernels alone and PHoG & PHoW kernels together, respectively. The result shows that the PHoG kernels absolutely contribute to the final decision. We observe a significant gain in accuracy by incorporating PHoG kernels into the learning process for all but the average-sum kernel.

表 2: Average gain in accuracy by adding PHoG features.

	uniform	ℓ^1	ℓ^p-joint	ℓ^p-single
PHoW	**45.4±1.0**	45.6±0.8	45.5±0.8	45.5±1.0
PHoW&G	**45.2±1.0**	**46.6±0.9**	**46.9±1.0**	**46.9±1.0**

3.4 Result 2: Results for the Official Splits of VOC 2007 and VOC 2008

In our second experimental setup, we evaluated the performance of the approaches for the official splits of the VOC 2007 and 2008 challenges. The winners of VOC2008

[2] We use only 800 visual words for the interest points as about $1/5$ of the descriptors are extracted per image.

[3] Since creating a codebook and assigning descriptors to visual words is computationally demanding, we apply the codebook created with the training images of the official split. This could result in slightly better *absolute* test errors, since some information of the test images might be contained in the codebook. However, our focus in this Section lies on a *relative* comparison between different classification methods, and this computational shortcut does not favor any of these approaches.

表 1: Average precisions on the test images of our 10 splits. For each column, the best method and comparable ones based on a Wilcoxon signed-rank test at the significance level of 5% are marked in bold faces.

	average	aeroplane	bicycle	bird	boat	bottle	bus
uniform	45.2±1.0	70.4±5.3	42.5±3.6	**47.8±6.0**	**61.2±4.6**	**22.5±5.7**	**50.5±10.8**
ℓ^1	46.6±0.9	**72.8±4.7**	**44.5±5.8**	**49.3±5.4**	**61.3±4.3**	20.5±4.0	**51.5±10.0**
ℓ^p-joint	**46.9±1.0**	**72.6±5.0**	**45.1±5.0**	**49.7±5.4**	**61.9±4.4**	**22.1±4.7**	**50.5±11.2**
ℓ^p-single	**46.9±1.0**	**71.2±4.9**	**44.0±4.9**	**49.0±5.9**	**61.7±4.0**	**22.5±5.2**	**52.3±9.3**
		car	cat	chair	cow	diningtable	dog
uniform		**53.0±3.4**	52.6±3.0	42.8±3.6	**13.8±3.8**	33.1±9.4	36.1±3.0
ℓ^1		54.0±3.5	55.3±2.6	**45.9±4.4**	**13.8±4.4**	**36.7±5.1**	**38.5±4.8**
ℓ^p-joint		**54.7±3.5**	**55.7±2.5**	44.9±4.7	**13.7±4.2**	**37.8±5.5**	**38.3±4.5**
ℓ^p-single		**54.4±3.1**	**55.7±2.6**	**45.6±4.1**	**13.7±3.5**	**37.2±5.0**	**38.8±3.4**
		horse	motorbike	person	pottedplant	sheep	sofa
uniform		**48.2±8.3**	44.5±6.5	85.8±1.0	**22.2±3.7**	**23.7±6.6**	**39.6±7.4**
ℓ^1		47.1±7.9	**47.5±4.8**	86.7±1.0	**23.2±5.1**	**26.6±8.6**	**39.5±8.5**
ℓ^p-joint		**48.0±8.0**	**48.0±5.8**	**86.8±1.0**	**24.8±6.3**	**25.9±9.3**	**40.6±9.0**
ℓ^p-single		**49.3±8.2**	**47.6±4.9**	**86.8±1.0**	**24.9±6.1**	**24.7±6.8**	**40.6±9.0**
		train	tvmonitor				
uniform		**60.4±8.6**	53.4±5.9				
ℓ^1		**60.8±8.9**	**57.0±5.6**				
ℓ^p-joint		**61.6±8.2**	**56.2±6.4**				
ℓ^p-single		**61.1±8.7**	**56.0±7.3**				

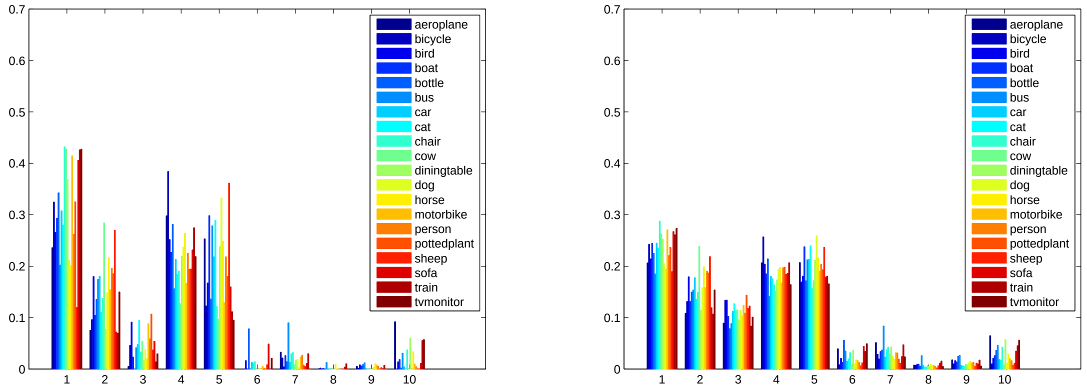


図 2: Selected weights by multiple kernel learning: ℓ^1 (left) and ℓ^p-joint (right)

[21] reported an average AP of 60.5 on VOC 2007 and achieved an AP of 54.9 on VOC2008. Their result is based on color descriptors [22], kernel codebook [23], and kernel discriminant analysis [4].

Table 3 shows the resulting average APs for our multiple kernel learning approaches.[4] The non-sparse MKL increases the accuracy of the basic color descriptors (uniform only PHoW) of about 2%. Furthermore, [21] reports a loss in accuracy of less than 1% if a support vector machine is substituted for the kernel discriminant analysis. Taking the different code books into account, we conjecture that – except for the code book – non-sparse multiple kernel learning is on par or better as the winner of last years VOC challenge. We will address the validity of our assumption in future work.

表 3: Average APs for VOC 2007/2008 using official splits.

	VOC2007	VOC2008
uniform (only PHoW)	55.0	49.0
uniform	55.0	—
ℓ^1	56.8	—
ℓ^p-joint	57.3	51.5
ℓ^p-single	57.1	50.9

[4] APs for VOC2008 have been kindly evaluated by the organizers.

4 Discussion

In contrast to anecdotal reports, we observed ℓ^1-MKL to outperform the average-sum kernel for PHoW and PHoG kernels (see Table 1). Nevertheless, carefully adjusting the norm p for boosts the performance of non-sparse MKL which performed best throughout all our experiments. The optimal choice of the norm p thereby depends on the actual set of kernels. As a rule of thumb, large values of p work out in cases where all kernels encode a similar amount of independent information while smaller values of p are best if some kernels are less informative or redundant.

As an illustrative example, consider a simple experimental setup where we deployed MKL together with the following 12 kernels: level-2 PHoW with grey and hue channels with 10 pixels pitch dense grid and 1200 vocabulary (3 pyramid levels × 2 colors), PHoG of grey channel (3 pyramid levels), and the pyramid histograms of intensity with hue channel (3 pyramid levels). Table 4 shows the results. The sparse ℓ^1-MKL yields a similar accuracy as the average-sum kernel. As suspected, both approaches are significantly outperformed by non-sparse MKL.

表 4: A simple case where the performance of ℓ^1-norm MKL deteriorates.

	uniform	ℓ^1	ℓ^p-joint	ℓ^p-single
mean AP	40.8±1.0	40.8±0.9	**42.6±0.7**	**42.3±0.9**

5 Conclusions

When measuring data with different measuring devices, it is always a challenge to combine the respective device uncertainties in order to fuse all available sensor information optimally. In this paper, we revisited this important topic and discussed machine learning approaches to adaptively combine different image descriptors in a systematic and theoretically well founded manner. While MKL approaches in principle solve this problem, it has been observed that the standard ℓ^1-norm based MKL can rarely outperform SVMs that use an average of a large number of kernels. One hypothesis why this seemingly unintuitive results may occur, is that the sparsity prior may not be appropriate in many real world problems. A close inspection reveals that most kernels contain useful structural information and should therefore not be omitted. A slightly less severe method of sparsification is to use another norm for optimization, namely the ℓ^p-norm. We tested whether this hypothesis holds true for computer vision and applied the recently developed non-sparse ℓ^p-norm MKL algorithms to object classification tasks. By choosing p as a hyperparameter which controls the degree of non-sparsity from a set of candidate values with the help of a validation data, we showed that ℓ^p-MKL significantly improves SVMs with averaged kernels and the standard sparse ℓ^1-norm MKL. Similar accuracy gain has been observed by controlling p in one-class MKL [11].

Future work will incorporate further modeling ideas of the VOC 2008 winner, e.g. the kernel code book, which we have so far not even employed. The test result with the official splits shown in this paper implied that our method is highly competitive to the winners solution. Furthermore, a combination of mid-level features by MKL will be an interesting research direction.

参考文献

[1] F. Bach, G. Lanckriet, and M. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. *International Conference on Machine Learning*, 2004.

[2] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval (CIVR '07)*, pages 401–408, 2007.

[3] U. Brefeld, P. Geibel, and F. Wysotzki. Support vector machines with example dependent costs. In *Proceedings of the European Conference on Machine Learning*, 2003.

[4] D. Cai, X. He, and J. Han. Efficient kernel discriminant analysis via spectral regression. In *ICDM*, 2008.

[5] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola. On kernel-target alignment. In *Advances in Neural Information Processing Systems*, volume 14, pages 367–373, 2002.

[6] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, Prague, Czech Republic, May 2004.

[7] N. Dalal and B. Triggs. Histograms of oriented gradientsfor human detection. In *IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition, volume 1, pages 886–893, San Diego, USA, June 2005.

[8] M. Everingham, L. Van 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/year = workshop/index.html, 2008.

[9] Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems*, volume 11, pages 487–493, 1998.

[10] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. Efficient and accurate ℓ^p-norm mkl. In *Advances in Neural Information Processing Systems 22*, 2009. to appear.

[11] M. Kloft, S. Nakajima, and U. Brefeld. Feature selection for density level-sets. In *Proc. of ECML*, 2009.

[12] C. Lampert and M. Blaschko. A multiple kernel learning approach to joint multi-class object detection. In *DAGM*, pages 31–40, 2008.

[13] Gert R.G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, pages 27–72, 2004.

[14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, New York, USA, June 2006.

[15] D. Lowe. Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[16] A.Y. Ng, M.I. Jordan, and Y.Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2002.

[17] C. Ong, A. Smola, and R. Williamson. Hyperkernels. In *NIPS*, volume 15, pages 495–502, 2002.

[18] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. In *ICML*, pages 775–782, 2007.

[19] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[20] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

[21] M. Tahir, K. van de Sande, Jasper Uijlings, Fei Yan, Xirong Li, Krystian Mikolajczyk, Josef Kittler, Theo Gevers, and Arnold Smeulders. Surreyuva srkda method. http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2008/worksh

[22] Koen E. A. van de Sande, Theo Gevers, and Cees G. M. Snoek. Evaluation of color descriptors for object and scene recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, USA, June 2008.

[23] J.C. van Gemert, J.M. Geusebroek, C.J. Veenman, and A.W.M. Smeulders. Kernel codebooks for scene categorization. In *ECCV*, 2008.

[24] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, pages 1–8, 2007.

[25] J. Zhang, M. Marszalek, S.Lazebnik, and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[26] J. Zhang, M. Marszalek, S.Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.

[27] Alexander Zien and C. Ong. Multiclass multiple kernel learning. In *ICML*, pages 1191–1198, 2007.

[28] Alexander Zien, Gunnar Rätsch, Sebastian Mika, Bernhard Schölkopf, Thomas Lengauer, and K.-R. Müller. Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, 16(9):799–807, 2000.