# Novel Fusion Methods for Pattern 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.** Over the last few years, several approaches have been proposed for information fusion including different variants of classifier level fusion (ensemble methods), stacking and multiple kernel learning (MKL). MKL has become a preferred choice for information fusion in object recognition. However, in the case of highly discriminative and complementary feature channels, it does not significantly improve upon its trivial baseline which averages the kernels. Alternative ways are stacking and classifier level fusion (CLF) which rely on a two phase approach. There is a significant amount of work on linear programming formulations of ensemble methods particularly in the case of binary classification.

In this paper we propose a multiclass extension of binary $\nu$-LPBoost, which learns the contribution of each class in each feature channel. The existing approaches of classifier fusion promote sparse features combinations, due to regularization based on $\ell_1$-norm, and lead to a selection of a subset of feature channels, which is not good in the case of informative channels. Therefore, we generalize existing classifier fusion formulations to arbitrary $\ell_p$-norm for binary and multiclass problems which results in more effective use of complementary information. We also extended stacking for both binary and multiclass datasets. We present an extensive evaluation of the fusion methods on four datasets involving kernels that are all informative and achieve state-of-the-art results on all of them.

## 1 Introduction

The goal of this paper is to investigate machine learning methods for combining different feature channels for pattern recognition. Due to the importance of complementary information in feature combination, much research has been undertaken in the field of low level feature design to diversify kernels, leading to a large number of feature channels (kernels) in typical pattern recognition tasks. Kernels are often computed independently of each other, thus may be highly redundant. On the other hand, different kernels capture different aspects of intraclass variability while being discriminative at the same time. Proper selection and fusion of kernels is, therefore, crucial to optimizing the performance and to addressing the efficiency issues in large scale pattern recognition applications.

The key idea of MKL [10,15,20], in the case of SVM, is to learn a linear combination of given base kernels by maximizing the soft margin between classes using $\ell_1$-norm regularization on weights. In contrast to MKL, the main idea of classifier level fusion [8] is to construct a set of base classifiers and then classify

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part I, LNAI 6911, pp. 140–155, 2011.

a new test sample by a weighted combination of their predictors. CLF methods attracted much attention, with AdaBoost [5] in particular, after being successful in many practical applications; this led to linear programming (LP) formulation of AdaBoost [16]. Inspired by the soft margin SVM, a soft margin LP for boosting, $\nu - LPBoost$, was proposed in [16]. Similar to ensemble methods, the aim of stacking [2] is to combine the prediction labels of multiple base classifiers using another classifier often referred as meta-level classifier.

Information fusion methods for MKL and CLF favor sparse feature/kernel selection due to $\ell_1$-norm regularization, arguing that the sparse models have intuitive interpretation [10] as a method of filtering out irrelevant information. However, in practical applications sparse models do not always perform well (c.f. [9] and references therein). In fact ,$\ell_1$ regularization hardly outperforms trivial baselines, such as average of kernels. Furthermore, sparseness may lead to poor generalization due to discarding useful information, especially in case of features encoding orthogonal characteristics of a problem. On the other hand $\ell_\infty$ regularization promotes combinations with equal emphasis on all feature channels, which leads to poor performance in case of noisy channels. To address these problems, different regularization norms [9] are considered for MKL. Similarly, among the classifier fusion approaches, $\nu$-LPBoost with $\ell_1$ regularization favors sparse solutions, or suffers from noisy channels in the case of $\ell_\infty$ regularization. In contrast to MKL, there is a lack of intermediary solutions with different regularization norms in ensemble methods.

In this paper, we present a novel multiclass classifier fusion scheme (NLP-$\nu$MC) based on binary $\nu - LPBoost$, which incorporates arbitrary norms $\{\ell_p, p \geq 1\}$ and optimizes the contribution from each class in each feature channel. The proposed optimization problem is a nonlinear separable convex problem which can be solved using off-the-shelf solvers. We also incorporate nonlinear constraints in previously proposed binary $\nu - LPBoost$ and multiclass LPBoost [6] and show empirically that nonlinear variants perform consistently better than their sparse counterparts, as well as baseline methods. It is important to note that both LP-$\beta$ and LP-B [6] are different from NLP-$\nu$MC. In Particular, the number of constraints in the optimization problems and the concept of margin are significantly different (see Section 3.1 for more details). For example, LP-B is not applicable to large multiclass datasets due to large number of constraints.

We use SVM as a base classifier in stacking and instead of using prediction labels from the base classifier we propose to use its real valued output. We also incorporate SVM as a base learner for stacking in case of multiclass datasets. We finally use SVM with RBF kernel as a meta-level classifier. The last contribution is an extensive evaluation and comparison of state-of-the-art fusion approaches. We perform experiments on multi-label and multiclass problems using standard benchmarks. Our multiclass formulation and nonlinear extensions of CLF consistently outperforms the state-of-the-art MKL and sparse CLF schemes. The best results are achieved with stacking, especially when the stacking kernel is combined with base kernels using CLF. Note that the datasets used for evaluation are visual category recognition datasets, however, the proposed fusion schemes

can be applied to any underlying pattern recognition problems provided that we have multiple feature channels. The proposed methods can also be applied to multi-model pattern recognition problems.

The remainder of this paper is organized as follows. We start with a review of two widely used information fusion schemes, the multiple kernel learning in Section 2 and linear programming (LP) formulation of ensemble methods for classifier fusion in Section 3 which also extends LP formulation of binary classifier fusion to incorporate arbitrary norms. Our proposed multiclass classifier fusion and schemes are presented in Section 3.1 and Section 4. In Section 5 we present the evaluation results and conclude in Section 6.

## 2 Multiple Kernel Learning

In this section, we review state-of-the-art MKL methods for classification. Consider $m$ training samples $(x_i, y_i)$, where $x_i$ is a sample in input space and $y_i$ is its label, $y_i \in \pm 1$ for binary classification and $y_i \in \{1, \ldots, N_C\}$, for multiclass classification. We are given $n$ training kernels (one kernel corresponding to each feature channel) $K_r$ of size $m \times m$ and corresponding $n$ test kernels $\dot{K}_r$ of size $m \times l$, with $l$ being the number of test samples. Each kernel, $K_r = \langle \Phi_r(x_i), \Phi_r(x_j) \rangle$, implicitly maps samples from the input space to a feature space with mapping function $\Phi_r(x_i)$ and gives similarity between corresponding samples $x_i$ and $x_j$ in the feature space. In the case of the SVM decision function for a single kernel is the sign of real valued output $g_r(x)$:

$$g_r(x) = \dot{K}_r(x)^T Y \boldsymbol{\alpha} + b, \tag{1}$$

where $\dot{K}_r(x)$ is the column corresponding to test sample $x$, $Y$ is an $m \times m$ matrix with labels $y_i$ on the diagonal and $\boldsymbol{\alpha}$ is a vector of lagrangian multipliers.

In MKL, the aim is to find a convex combination of kernels $K = \sum_{r=1}^{n} \beta_r K_r$ by maximizing the soft margin [1,10,15,20,25] using the following program:

$$\min_{\boldsymbol{w_r}, \boldsymbol{\xi}, b, \boldsymbol{\beta}} \frac{1}{2} \sum_{r=1}^{n} \boldsymbol{w_r}^T \boldsymbol{w_r} + C \sum_{i=1}^{m} \xi_i \tag{2}$$

$$\text{s.t. } y_i \Big( \sum_{r=1}^{n} \langle \boldsymbol{w_r}, \sqrt{\beta_r} \Phi_r(x_i) \rangle + b \Big) \geq 1 - \xi_i, \quad \boldsymbol{\xi} \succeq 0, \boldsymbol{\beta} \succeq 0, \ \ \|\boldsymbol{\beta}\|_p^p \leq 1$$

The dual of Eq. (2) can be derived easily using Lagrange multiplier techniques. The MKL primal for linear combination and its corresponding dual are derived for different formulations in [1,9,10,15,20,24] and compared in [25] which also extended MKL to the multiclass case. The dual problem can be solved by using several existing MKL approaches, e.g, SDP [10], SMO [1], SILP [20] and simpleMKL [15]. The decision function for MKL SVM is the sign of $f(x)$:

$$f(x) = \sum_{r=1}^{n} \beta_r \dot{K}_r(x)^T Y \boldsymbol{\alpha} + b. \tag{3}$$

The weight vector $\boldsymbol{\beta} \in \mathbb{R}^n$, Lagrange multiplier $\boldsymbol{\alpha} \in \mathbb{R}^m$, and bias $b$ are learnt together by maximizing the soft margin. We can consider $f(x)$ as a linear combination of real valued output $g_r(x)$ of the base classifier with the same $\alpha$ and $b$ shared across all base classifiers.

## 3 Classifier Fusion with Non-Linear Constraints

In this section we review the linear programming formulation of ensemble methods for classifier level fusion (CLF) based on boosting. We also extend the $\nu$-LP-AdaBoost [16] formulation for binary classification with nonlinear constraints. This is a significant extension as it avoids discarding channels with complementary information while keeping it robust to noisy feature channels.

The empirical work has shown that boosting, and other related ensemble methods [5,6,16] for combining predictors, can lead to a significant reduction in the generalization error and, hence, improves performance. Our focus is on the linear programming (LP) formulations of AdaBoost [5] and its soft margin LP formulations [16] over a set of base classifiers $G = \{g_r : x \mapsto \pm 1, \forall r = 1, \ldots, n\}$. For a test example $x$, the output label generated by such ensemble is a weighted majority vote and is given by the sign of $f(x)$:

$$f(x) = \sum_{r=1}^{n} \beta_r g_r(x). \tag{4}$$

Note that for the SVM, $f(x)$ is a linear combination of the real valued output of $n$ SVMs, where $g_r(x)$ is given by Eq. (1). The decision function of MKL in Eq. (3) shows that the same set of parameters $\{\boldsymbol{\alpha}, b\}$ is shared by all participating kernels. In contrast to MKL, the decision function of CLF methods in Eq. (4) uses separate sets of SVM parameters, since different $\{\boldsymbol{\alpha}, b\}$ embedded in $g_r(x)$ can be used for each base learner. In that sense, MKL can be considered as a restricted version of CLF [6]. The aim of the ensemble learning is to find optimal weight vector $\boldsymbol{\beta}$ for the linear combination of base classifiers given by Eq. (4).

We define the margin (or classification confidence) for an example $x_i$ as $\rho_i := y_i f(x_i) = y_i \sum_{r=1}^{n} \beta_r g_r(x_i)$ and the normalized (smallest) margin as:

$$\rho := \min_{1 \le i \le m} y_i f(x_i) = \min_{1 \le i \le m} y_i \sum_{r=1}^{n} \beta_r g_r(x_i). \tag{5}$$

It has been argued that AdaBoost maximizes the smallest margin $\rho$ on the training set [16]. Based on this idea and the idea of soft margin SVM formulations, the $\nu$-LP-AdaBoost formulation has been proposed in [16]. The $\nu$-LPBoost performs a sparse selection of feature channels due to $\ell_1$ regularization, which is suboptimal if all feature channels carry complementary information. Similarly, in the case of $\ell_\infty$ norm, noisy features channels may have significant impact on the results. To address these problems, we generalize binary classifier fusion for arbitrary norms $\{\ell_p, p \ge 1\}$.

The input to classifier fusion are predictions corresponding to each feature channel, which are real valued outputs of base classifiers. To obtain these predictions for a training set we can use leave one out or $v$-fold cross validation. In contrast to AdaBoost, we consider $n$ to be a fixed number of base classifiers $\{g_r, \forall r = 1, \ldots, n\}$ which are independently trained. Given the base classifiers, we learn the optimal weights $\beta_r$ for their linear combination (Eq. (4)) by maximizing the smallest margin $\rho$ in the following optimization problem:

$$\begin{aligned} \max_{\boldsymbol{\beta},\boldsymbol{\xi},\rho} \quad & \rho - \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i \\ s.t. \quad & y_i \sum_{r=1}^{n}\beta_r f_r(x_i) \geq \rho - \xi_i \quad \forall\ i = 1, ..., m \\ & \|\boldsymbol{\beta}\|_p^p \leq 1, \quad \boldsymbol{\beta} \succeq 0, \boldsymbol{\xi} \succeq 0, \rho \geq 0 \end{aligned} \tag{6}$$

where $\xi_i$ are slack variables which accommodate negative margins. The regularization constant is given by $\frac{1}{\nu m}$, which corresponds to the $C$ constant in SVM. Problem (6) is a nonlinear separable convex optimization problem and can be solved efficiently for global optimal solution by standard optimization toolboxes[1].

### 3.1 Multiclass Classifier Fusion with Non-Linear Constraints

In this section we propose a novel multiclass extension of $\nu$-LP-AdaBoost and compare it with other existing multiclass variants. We also incorporate nonlinear constraints in two existing multiclass classifier fusion schemes: LP-$\beta$ [6] and LP-B [6]. The empirical results show that the nonlinear constraints improve the performance of these methods.

**Nonlinear Programming $\nu$-Multiclass (NLP-$\nu$MC):** We consider one-vs-all formulation for multiclass case with $N_C$ classes, i.e., for each feature channel we solve $N_C$ binary problems, one corresponding to each class. Therefore, the set of base classifiers $G = \{g_r : x \mapsto \mathbb{R}^{N_C}, \forall r = 1, \ldots, n\}$ consists of $n$ base hypotheses (weak learners) $g_r$, where each base classifier maps into an $N_C$ dimensional space $g_r(x) \mapsto \mathbb{R}^{N_C}$. The output of $g_r$ corresponding to $c$'th class is denoted by $g_{r,c}(x)$. Recently it has been shown that One-vs-All is as good as any other approach [18], moreover it fits naturally to the proposed CF and computational complexity for other methods are higher, even prohibitive in case of many classes. Note that in practice the predictions for all base classifiers can be computed in parallel as they are independent of each other, which makes this approach appealing. We learn the weights for every class in each feature channel and, therefore, instead of $n$ dimensional weight vector $\boldsymbol{\beta} \in \mathbb{R}^n$ as in case of binary classifier fusion, we have an $n \times N_C$ dimensional weight vector $\boldsymbol{\beta} \in \mathbb{R}^{n \times N_C}$. The first $N_C$ entries of vector $\boldsymbol{\beta}$ correspond to weights of classes

[1] We have used MATLAB and MOSEK (`http://www.mosek.com`) and found that interior-point based separable convex solver in MOSEK is faster by an order of magnitude of time.

in first feature channel and last $N_C$ entries correspond to weights in feature channel $n$. After finding the optimal weights, the decision function for a test sample $x$ corresponding to each class is given by weighted sum and the overall decision function of multiclass classifier fusion is obtained by picking the class with maximum response.

We extend the definition of margin (classification confidence) for binary classifier fusion given in Eq. (5) to multiclass case as follows.

$$\rho_i(x_i,\beta) := \sum_{r=1}^{n} \beta_{(N_C(r-1)+y_i)} g_{r,y_i}(x_i) - \sum_{r=1}^{n} \sum_{j=1,j\neq i}^{N_C} \beta_{(N_C(r-1)+y_j)} g_{r,y_j}(x_i) \quad (7)$$

The classification confidence for examples $x_i$ depends upon $\beta$ and scores from base classifiers. The main difference between the two margins is that here, we are taking responses (scores multiplied with corresponding weights) from all negative classes, sum them and subtract this sum from the response of positive class. This is done for all n feature channels. Normalized (smallest) margin can then be defined as $\rho := \min_{1\leq i\leq m} \rho(x_i,\beta)$. Inspired by LP formulations of AdaBoost (cf. [16] and references therein) we propose to maximize the normalized margin $\rho$ to learn linear combination of base classifiers. However, generalization performance of LP formulation of AdaBoost based on maximizing only normalized margin is inferior to AdaBoost for noisy problems [16]. Moreover, theorem 2 in [16] highlights the fact that minimum bound on generalization error is not necessarily achieved with a maximum margin. To address these issues, soft margin SVM based formulation with slack variable is introduced in Eq. (8). This formulation does not force all the margins to be greater than zero. To avoid penalization of informative channels and to gain robustness against noisy feature channels, we change the regularization norm to handle any arbitrary norm $\ell_p, \forall p \geq 1$. The final optimization problem is (replacing $\rho_i$ with Eq. (7)):

$$\max_{\boldsymbol{\beta},\xi,\rho} \quad \rho - \frac{1}{\nu m} \sum_{i=1}^{m} \xi_i \quad (8)$$

$$s.t. \quad \sum_{r=1}^{n} \beta_{(N_C(r-1)+y_i)} g_{r,y_i}(x_i) - \sum_{r=1}^{n} \sum_{j=1,j\neq i}^{N_C} \beta_{(N_C(r-1)+y_j)} g_{r,y_j}(x_i)$$
$$\geq \rho - \xi_i \;\; i = 1,...,m, \quad (9)$$
$$\|\boldsymbol{\beta}\|_p^p \leq 1, \quad \rho \geq 0, \boldsymbol{\beta} \succeq 0 \;\; \boldsymbol{\xi} \succeq 0 \;\; \forall i = 1,...,m$$

where $\frac{1}{\nu m}$ is the regularization constant and gives a trade-off between minimum classification confidence $\rho$ and the margin errors. This formulation looks similar to Eq. (6), in fact we are using the same objective function but the main difference is the definition of margin which is used in the constraints in Eq. (9). Eq. (9) employs a lower bound on the differences between the classification confidence (margin) of the true class and the joint confidence of all other classes. It is important to note that the total number of constraints is equivalent to the number of training examples $m$ plus one regularization constraint for $l_p$-norm

(ignoring the positivity constraints on variables). Therefore, the difference in complexity, compared to the binary classifier fusion, is the increased number of variables in weight vector $\boldsymbol{\beta}$, while having the same number of constraints. Note that the problem in Eq. (8) is a nonlinear separable convex optimization problem and can be solved efficiently using MOSEK. We now extend LP-$\boldsymbol{\beta}$ and LP-B by introducing arbitrary regularization norms $\ell_p, \forall p \geq 1$, which avoids rejection of informative feature channels while being robust against noisy features channels. Generalized optimization problems for, LP-$\boldsymbol{\beta}$ and LP-B, are separable convex programs and can be solved efficiently by MOSEK.

**Nonlinear Programming-$\boldsymbol{\beta}$ (NLP-$\boldsymbol{\beta}$):** We generalize LP-$\beta$ [6] by incorporating $\ell_p, \forall p \geq 1$ norm constraints. The optimization problem is given by:

$$\min_{\boldsymbol{\beta},\xi,\rho} \quad -\rho + \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i \tag{10}$$

$$s.t. \quad \sum_{r=1}^{n}\beta_r g_{r,y_i}(x_i) - max \sum_{y_j \neq y_i, r=1}^{n} \beta_r g_{r,y_j}(x_i) \geq \rho - \xi_i, \ \forall \ i = 1, ..., m \tag{11}$$

$$\|\boldsymbol{\beta}\|_p^p \leq 1, \quad \beta_r \geq 0, \ \xi_i \geq 0, \rho \geq 0, \quad \forall r = 1, \ldots, n, \forall \ i = 1, \ldots, m.$$

Note that weight vector $\beta$ lies in an $n$ dimensional space $\boldsymbol{\beta} \in \mathbb{R}^n$ as in binary classifier fusion. After finding the weight vector $\boldsymbol{\beta}$, the decision function of generalized LP-$\boldsymbol{\beta}$ is simply the maximum response of the weighted sum of all classes in all feature channels.

**Nonlinear Programming-B (NLP-B):** We also propose an extension of multiclass LP-B [6] with arbitrary regularization norms $\ell_p, \forall p \geq 1$. Instead of having a weight vector $\boldsymbol{\beta}$, LP-B has a weight matrix $B \in \mathbb{R}^{n \times N_C}$. For learning weights in matrix $B$, we propose the following convex optimization problem:

$$\min_{B,\xi,\rho} \quad -\rho + \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i \tag{12}$$

$$s.t. \quad \sum_{r=1}^{n} B_r^{y_i} g_{r,y_i}(x_i) - \sum_{y_j \neq y_i, r=1}^{n} B_r^{y_j} g_{m,y_j}(x_i) \geq \rho - \xi_i \ \ i = 1, ..., m, \tag{13}$$

$$\|B\|_p^p \leq 1, \ \ B_r^c \geq 0, \quad \boldsymbol{\xi} \succeq 0, \rho \geq 0, \ \forall \ r = 1, ..., n, c = 1, ..., N_C$$

The first set of constraints (Eq. (13)) gives a lower bound on the pairwise difference between classification confidences (margins) of the true class and non-target class. Note that in this formulation $N_C - 1$ constraints are added for every training example and the total number of constraints is $m \times (N_C - 1) + 1$.

**Discussion:** The main difference between the three multiclass approaches discussed in this section is in the definition of the feasible region which is defined by Eq. (9), Eq. (11) and Eq. (13) for NLP-$\nu$MC, NLP-$\boldsymbol{\beta}$ and NLP-B respectively. In NLP-$\boldsymbol{\beta}$ and Lp-$\boldsymbol{\beta}$ [6] the feasible region depends on the difference between the classification confidence of the true class and the closest non-target

class only. The total number of constraints in this case is $m + 1$. The feasible region of NLP-B and LP-B [6] is defined by the pairwise difference between class confidence of the true class and non-target class added as one constraint at a time. In other words each difference pair is added as an independent constraint without having any interaction among each other. There are $N_C$ constraints for each example and the total number of constraints is $m \times (N_C - 1) + 1$. The large number of constraints makes this approach less attractive for datasets with a large number of classes. For example, for Caltech101 [4] with only 15 images per class for training, the number of constraints for LP-B is more than 150 thousand ($15 \times 101 \times 100 + 1 \cong 1.5 \times 10^5$). In case of our NLP-$\nu$MC, the feasible region depends upon the joint classification confidence of all the non-target classes subtracted from the class confidence of the true class. Thus, the feasible region of NLP-$\nu$MC is much smaller than the feasible region of NLP-B. Due to these joint constraints the total number of constraints for NLP-$\nu$MC is $m+1$, e.g., for Caltech101 [4] with 15 images per class for training, the number of constraints for NLP-$\nu$MC is only 1516 (15*101+1) which is only 1% of the constraints in NLP-B. We, therefore, can apply NLP-$\nu$MC to large multiclass datasets, as opposed to NLP-B, especially for norms greater than 1. Note that the difference in complexity between NLP-$\nu$MC and NLP-$\boldsymbol{\beta}$ or binary classifier fusion is the extended weight vector $\boldsymbol{\beta}$.

## 4 Extended Stacking

In this section we give a brief overview of stacking proposed in [21]. We then present an extension to the stacking framework. The main aim of stacking [2] is to combine the prediction labels of multiple base classifiers $C_r$ using another classifier, often referred to as meta-level classifier. In the first phase, prediction labels $y_i^r$ for example $x_i$ of base classifiers are obtained by leave-one-out or by $v$-fold cross validation on the training set. The input to the meta-level classifier are these prediction labels together with the output label for example $i$ and form a tuple of the form $((y_i^1, \ldots, y_i^n), y_i)$. By the use of meta-level classifier, stacking tries to infer reliable and unreliable base classifiers. By using output probabilities corresponding to each label, the performance of stacking can be improved. The size of the meta-level training tuple is multiplied by the number of classes in this case. It has been shown empirically that stacking does not perform better than selecting the best classifier in ensemble by cross validation [2]. To improve the performance of stacking they replaced meta-level classifier by a new multi-response model tree and empirically showed enhancement in performance as compared to stacking or selecting the best base classifier by cross validation.

We have used SVM as a base classifier. Instead of using the prediction labels, we use the real valued outputs $g_r(x_i)$ of the SVM classifier. The input training tuple for meta-level classifier is of the form $(g_1(x_i), \ldots, g_n(x_i), y_i)$. For multiclass case we use one vs all formulation within base classifiers, therefore $g_r$ maps into an $N_C$ dimensional space, $g_r(x) \mapsto \mathbb{R}^{N_C}$. The input tuple in this case is multiplied by the number of classes. We concatenate the outputs of all base SVM classifiers

corresponding to example $x_i$ and consider it as an input feature vector for a meta-level SVM classifier. We build an RBF kernel by using euclidean distance between these feature vectors. We refer to this as stacking kernel. To the best of our knowledge the use of real valued SVM output for base classifiers in stacking is novel, for both binary and multiclass datasets. We consider the stacking kernel as a separate feature channel and can then apply MKL or any proposed CLF scheme, discussed in Section 3, to combine it with base kernels.

## 5 Experiments and Discussion

This section presents the experimental evaluation of the methods investigated in this paper on different object recognition benchmarks. These datasets include a large variety of objects under different poses, scale and lighting condition with cluttered background in real world scenario. We first discuss the results of the multi-label datasets, namely, Pascal VOC 2007 and then present the results for three multiclass datasets, namely, Flower17, Flower102 and Caltech101. In multi-label, classification each example can be associated with a set of labels as opposed to a single label. We use binary relevance [17], a well know method for multi-label classification, as it is recommended by the organizers of Pascal VOC challenge [3]. The MKL results on Pascal VOC 2007 are reported using binary MKL from SHOGUN toolbox[2], and for CLF we have used $\nu$-LP-AdaBoost given in Eq. (6). For multiclass dataset we have used multiclass MKL from the SHOGUN toolbox. For classifier level fusion we use three CLF schemes proposed in this paper namely, NLP-$\nu$MC, NLP-$\boldsymbol{\beta}$ and NLP-B given by Eq.(8), Eq.(10) and Eq.(12), respectively. We do not have results for higher values of norms in case of NLP-B, and for some values of norms in case of MKL because their optimization problems take several days. On the other hand NLP-$\boldsymbol{\beta}$ and NLP-$\nu$MC are very fast as compared to multiclass MKL and NLP-B and take few seconds and few minutes, respectively. Stacking results are presented using the approach described in section 4. Finally, we present results by combining the stacking kernel with the base kernels using MKL, NLP-$\boldsymbol{\beta}$ and NLP-$\nu$MC.

### 5.1 Pascal VOC 2007

Pascal VOC 2007 [3] is a challenging dataset consisting of 20 object classes with 9963 image examples (2501 training, 2510 validation, and 4952 testing images). Images include indoor and outdoor scenes, truncated and occluded objects at various scales and different lighting conditions. Classification of 20 object categories is handled as 20 independent binary classification problems. We present results using average precision (AP) [3] and mean average precision (MAP).

In general, kernels can be obtained from various feature extractors. To produce state-of-the-art results we use 5 kernels from various descriptors introduced in [12,19] computed for 2 sampling strategies (i.e., dense and interest points) and spatial location grids [11]: entire image (1x1), horizontal bars (1x3), vertical bars

[2] `http://www.shogun-toolbox.org/`

(3x1) and image quarters (2x2). The descriptors are clustered using k-means to form a codebook of 4000 visual words. Each spatial grid is then represented by histograms of codebook occurrences and a separate kernel matrix is computed for each grid. The kernel function to compute entry $(i, j)$ of the kernel matrix is based on $\chi^2$ distance between features $F_i$ and $F_j$.

$$K(F_i, F_j) = e^{-\frac{1}{A} dist(F_i, F_j)} \tag{14}$$

where, A is a scalar for normalizing the distance, and is set to average $\chi^2$ distance between all features.

We apply Support Vector Machine (SVM) as base classifiers for nonlinear classifier level fusion schemes and the stacking proposed in this paper and compare them with MKL schemes. The regularization parameter for SVM is in the set $\{2^{(-2,0,3,7,10,15)}\}$. The regularization parameter $\nu$ for different CF methods is in the range $\nu \in [.05, .95]$ with the step size of 0.05. Both SVM and CF regularization parameters are selected on the validation set. The values for norms for generalized classifier fusion are in the range $p \in \{1, 1 + 2^{-5,-3,-1}, 2, 3, 4, 8, 10^4\}$. We consider each value of $p$ as a separate fusion scheme. Note for $p = 10000$ we get uniform weights which corresponds to unweighted sum or $\ell_\infty$. Figure 1 shows learnt weights on the training set of aeroplane category of Pascal VOC 2007 for several values of $p$ using CLF. The plotted weights are corresponding to the optimal value of regularization parameter $C$ of SVM. The sparsity of learnt weights can be observed easily for low values of $p$. The sparsity decreases with increased $p$, up to uniform weights (corresponding to $\ell_\infty$) achieved at $p = 10000$. Weights can also be learnt corresponding to best performing $p$ on validation set.

The mean average precision for several fusion methods are given in Table 1. Row MKL shows the results for nine MKL methods with different regularization norms applied to 5 base kernels. Note that MAP increases with the decrease in sparsity at higher values of norms. Similar trend can be found in CLF. Low performance of MKL-$\ell_1$-norm, which leads to sparse selection, indicates that base kernels carry complementary information. Therefore, the non-sparse MKL or CLF methods such as $\ell_2$-norm and $\ell_\infty$-norm, give better results as reported in Table 1. Unweighted sum in the case of MKL is performing better than any other MKL methods which reflects that in case of all informative channels, learning the weights for MKL does not improve much on this dataset. The proposed non-sparse CLF ($\ell_2$) schemes outperform the state-of-the-art MKL ($\ell_2$-norm, $\ell_\infty$-norm) by 2 % and 1.1% respectively. The stacking is performing the best among all the methods and outperforms MKL by 1.5%. Further improvements can be gained by fusing the stacking kernel together with 5 base kernels in case of both MKL and CLF. The combination of base plus the stacking kernel under MKL produced state-of-the-art result on this dataset with a MAP of 66.24%, and outperforms MKL and CLF by 3.3% and 2.3% respectively.

### 5.2 Flower 17

Flower 17 [14] consists of 17 categories of flowers common in UK with 80 images in each category. The dataset is split into training (40 images per class),

**Table 1.** Mean Average Precision of PASCAL VOC 2007

| Fusion Methods \ norms | 1 | $1+2^{-3}$ | $1+2^{-2}$ | $1+2^{-1}$ | 2 | 3 | 4 | 8 | $\ell_\infty$ |
|---|---|---|---|---|---|---|---|---|---|
| MKL | 55.42 | 56.42 | 58.53 | 61.07 | 61.98 | 62.45 | 62.61 | 62.81 | 62.93 |
| CLF | 63.71 | 63.94 | 63.97 | **63.98** | 63.97 | 63.97 | 63.77 | 63.69 | 63.11 |
| Stacking | **64.44** | | | | | | | | |
| MKL (Base + Stacking) | 64.39 | 64.55 | 65.06 | 65.75 | 66.06 | 66.23 | **66.24** | 66.09 | 65.93 |
| CLF (Base + Stacking) | 65.18 | 65.20 | 65.45 | 65.57 | 65.65 | 65.63 | 65.59 | 65.54 | 65.48 |

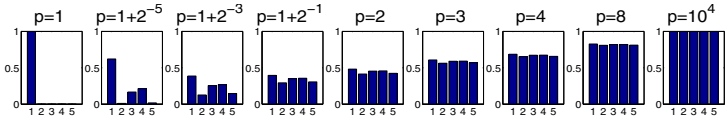


**Fig. 1.** Pascal VOC 2007. Feature channels weights learned with various $\ell_p$ for CLF($\ell_p$).

validation (20 images per class) and test (20 images per class) using 3 predefined random splits by the authors of the dataset. There are large appearance variations within each category and similarities with other categories. For experiments we have used 7 RBF kernels from the 7 $\chi^2$ distance matrices provided online[3]. The features used to compute these distance matrices include different types of shape, texture and color based descriptors whose details can be found in [14]. We have used SVM as a base classifier and its regularization parameter is in the range $\{10^{(-2,-1,\ldots,3)}\}$. The Regularization parameter for different CLF is in the range $\nu \in \{0.05, 0.1, \ldots, 0.95\}$. Both SVM and CLF regularization parameters are selected on the validation set. To carry out a fair comparison, the regularization parameters and other setting are the same as in [6].

The results given in Table 2, show that the baseline for MKL, i.e., MKL-avg($\ell_\infty$) gives 84.9% [6], and baseline for classifier level fusion, i.e., CLF($\ell_\infty$) gives 86.7%. The MKL results are obtained using the SHOGUN multiclass MKL implementation for different norms. Nonlinear versions of classifier fusion perform better than their sparse counterparts as well as state-of-the-art MKL. The best result in CLF is obtained by the proposed NLP-$\nu$MC ($\ell_2$) and NLP-$\boldsymbol{\beta}$ ($\ell_4$). They outperform the MKL baseline by more than 2.5% and multiclass MKL by 0.6%. Stacking yields the best results on this dataset, outperforming MKL baseline by more than 4.5%, MKL by more than 2% and the best CLF method by more than 1.5%. Combining the stacking kernel with the 7 base kernels using multiclass MKL also shows similar results. Note that the performance drops when the stacking kernel is combined with the 7 base kernels using MKL ($\ell_\infty$) or CLF ($\ell_\infty$). This highlights the importance of learning in fusion methods. However, when the stacking kernel is combined with the 7 base kernels using classifier fusion, it produces state-of-the-art results on this dataset, and outperforms MKL, the best in CLF and stacking by 3%, 2.3% and 0.8%, respectively.

The second half of Table 2 shows comparison with published state-of-the-art results. According to our knowledge the best performing method using the 7

[3] `http://www.robots.ox.ac.uk/~{}vgg/data/flowers/17/index.html`

**Table 2.** Classification Rate on Flower17

| ML-Methods | 1 | $1+2^{-3}$ | $1+2^{-1}$ | 2 | 3 | 4 | 8 |
|---|---|---|---|---|---|---|---|
| MKL | 87.2±2.7 | 74.9±1.7 | 72.2±3.6 | 71.2±2.7 | 70.6±3.8 | 73.1±3.9 | 81.0±4.0 |
| NLP-$\boldsymbol{\beta}$ | 86.5±3.3 | 86.6±3.4 | 86.6±1.1 | 86.7±1.2 | 87.4±1.5 | **87.9±1.8** | 87.8±2.1 |
| NLP-$\nu$MC | 85.5±1.3 | 86.6±2.0 | 87.6±2.2 | 87.7±2.6 | **87.8±2.1** | 87.7±2.0 | 87.8±1.9 |
| NLP-B | 84.6±2.5 | 84.6±2.4 | 84.8±2.6 | 84.8±2.5 | 85.5±3.7 | 86.9±2.7 | 87.3±2.7 |
| Stacking | **89.4 ± 0.5** | | | | | | |
| MKL(Base +Stacking) | 89.3±0.9 | 79.7±2.7 | 77.6±1.2 | 74.7±2.4 | 73.8±2.6 | 77.8±4.3 | 86.3±1.9 |
| NLP-$\boldsymbol{\beta}$(Base +Stacking) | **90.2±1.5** | 89.3±0.7 | 89.6±0.5 | 89.2±1.6 | 89.3±1.2 | 89.1±1.4 | 89.0±1.0 |
| NLP-$\nu$MC(Base +Stacking) | 86.1±2.5 | 87.3±1.4 | 88.5±0.5 | 88.6±0.9 | 88.6±0.9 | 88.8±1.1 | 88.9±1.2 |

| Comparison with State-of-the-Art | |
|---|---|
| MKL-prod [6] (7 kernels) | 85.5 ± 1.2 |
| MKL-avg ($\ell_\infty$) [6] (7 kernels) | 84.9 ± 1.9 |
| CLF ($\ell_\infty$) (7 kernels) | 86.7 ± 2.7 |
| MKL-avg ($\ell_\infty$) (7 kernels + Stacking kernel) | 88.5 ± 1.1 |
| CLF ($\ell_\infty$) (7 kernels+ Stacking kernel) | 88.8 ± 1.4 |
| CG-Boost [6] (7 kernels) | 84.8 ± 2.2 |
| MKL (SILP or Simple) [6] (7 kernels) | 85.2 ± 1.5 |
| LP-$\beta$ [6] (7 kernels) | 85.5 ± 3.0 |
| LP-B [6] (7 kernels) | 85.4 ± 2.4 |
| MKL-FDA ($\ell_p$) [23] (7 kernels) | 86.7 ± 1.2 |
| $L_1$-BRD [22] (30 kernels) | 89.0 ± 0.6 |

distance matrices provided by the authors is giving 86.7% which is similar to the CLF baseline. Our best CLF method outperforms it by 1.2% while our stacking approach outperforms it by 2.7% and our CLF combination of base plus stacking outperforms it by 3.5%. It is important to note that while comparing fusion methods, the base feature channels (kernels) must be the same across different schemes. For example, the comparison of Flower 17 with state-of-the-art in [22] is not justified as it uses 30 kernels while normally the results are reported using the 7 kernels provided online. Nevertheless, our best method outperforms this by 1.2% which can be considered as a significant improvement in spite of using 4 times fewer feature channels.

### 5.3 Flower 102

Flower 102 [13] is an extended multiclass dataset containing 102 flower categories commonly present in UK. It consists of 8189 images with 40 to 250 images in each class. The dataset is split into training (10 images per class), validation (10 images per class) and test (with a minimum of 20 images per class) using a split predefined by the authors of the dataset. For the experiments we have used the 4 $\chi^2$ distance matrices provided online[4]. The details of the features used to compute these distance matrices can be found in [13]. RBF kernels are computed using Eq. (14) and these four distance matrices. The experimental setup is the same as for Flower 17.

The results are given in Table 3. We have not reported the variance of the results as the authors of the dataset have given only 1 split online and for a fair

[4] `http://www.robots.ox.ac.uk/~{}vgg/data/flowers/102/index.html`

**Table 3.** Mean accuracy on Flower 102 dataset

| ML-Methods | 1 | $1+2^{-3}$ | $1+2^{-1}$ | 2 | 3 | 4 | 8 | $\ell_\infty$ |
|---|---|---|---|---|---|---|---|---|
| MKL | 69.9 | 64.7 | 65.3 | 65.9 | 65.7 | - | - | 73.4 |
| NLP-$\boldsymbol{\beta}$ | 61.2 | **75.7** | 73.5 | 74.7 | 73.0 | 73.9 | 74.6 | 73.0 |
| NLP-$\nu$MC | 72.6 | 73.1 | 73.2 | 73.3 | 73.4 | 73.4 | 73.4 | 73.0 |
| NLP-B | 73.6 | - | - | - | - | - | - | 73.0 |
| Stacking | **77.7** | | | | | | | |
| MKL(Base+ Stacking) | 79.8 | 65.9 | 66.2 | 65.8 | 65.5 | - | 68.9 | 76.4 |
| NLP-$\boldsymbol{\beta}$(Base +Stacking) | 79.2 | 77.8 | 77.8 | 78.3 | 79.0 | 79.4 | **80.3** | 77.2 |
| NLP-$\nu$MC(Base +Stacking) | 77.6 | 77.3 | 77.1 | 77.2 | 77.2 | 77.2 | 77.2 | 77.2 |

| Comparison with State-of-the-Art | |
|---|---|
| MKL-prod | 73.8 |
| MKL-avg | 73.4 |
| MKL [13] | 72.8 |

comparison with previously published results we use the same split as used by other authors. The baseline for MKL gives 73.4%, and baseline for CLF gives 73.0%. Multiclass MKL is not performing well on this dataset with the best result achieved by MKL ($\ell_1$) and performs 3.5% lower than the trivial baseline. The best among classifier level fusion is the NLP-$\boldsymbol{\beta}$ ($\ell_{1+2^{-3}}$) scheme. It performs 5.8% better than multiclass MKL and 2.3%, 2.7% better than MKL and CLF baselines, respectively. Note that NLP-$\nu$MC is performing worse than NLP-$\boldsymbol{\beta}$ as it has to estimate $N_C$ times more parameter than NLP-$\boldsymbol{\beta}$ in the presence of few training example per category. We expect NLP-$\nu$MC to perform better in the presence of more training data. Stacking achieves the best results on this dataset and it performs 7.8% better than multiclass MKL and 4.3%, 4.7% better than MKL and CLF baselines, respectively. The results can be further improved by combining the stacking kernel with the 4 base kernels by using MKL or CLF. However, the performance drops when the stacking kernel is combined with the 4 base kernels using MKL ($\ell_\infty$) or CLF ($\ell_\infty$). This highlights the importance of learning in fusion methods. We achieve state-of-the-art results on this dataset by combining the stacking kernel with the 4 base kernels using CLF. This combination performs 10% better than multiclass MKL and 6.6%, 7% and 2.3% better than MKL baseline, CLF baseline and stacking, respectively. Note that we are unable to compute the mean accuracy for NLP-B, especially for $\ell_p$-norm greater than 1, due to a large number of constraints in the optimization problem. The results for MKL are reported from [13] for comparison. In comparison to the published results, our best method has an improvement of 7.2% which is a significant gain. given that we are not using any new information.

### 5.4 Caltech101

Caltech101 [4] is a multiclass dataset consisting of 101 object categories and a background category. There are 31 to 800 images per category of medium resolution (200 × 300). We follow the common practice used on this dataset, i.e., use 15 randomly selected images per category for training and validation, while up to

**Table 4.** Mean accuracy on Caltech101 dataset

| ML-Methods | 1 | $1+2^{-3}$ | $1+2^{-1}$ | 2 | 3 | 4 | 8 |
|---|---|---|---|---|---|---|---|
| MKL | 68.6±2.2 | 61.2±1.1 | 58.1±0.8 | 57.4±0.7 | 57.0±0.6 | - | 63.9±0.9 |
| NLP-$\boldsymbol{\beta}$ | 69.0±1.8 | 68.6±2.2 | 69.1±1.2 | 69.0±1.4 | **69.2±1.5** | 69.0±1.3 | 69.0±1.3 |
| NLP-$\nu$MC | 67.4±2.4 | 68.7±1.8 | 68.4±1.0 | 68.5±0.8 | 68.4±0.7 | 68.4±0.7 | 68.4±0.7 |
| NLP-B | 64.1±0.7 | - | - | - | - | - | - |
| Stacking | 68.0 ± 2.4 | | | | | | |
| MKL(Base+ Stacking) | 68.6±2.2 | 68.9±2.4 | 68.5±2.5 | 68.5±2.6 | 68.5±2.5 | - | 69.6±2.2 |
| NLP-$\boldsymbol{\beta}$(Base +Stacking) | 69.7±1.7 | 69.3±2.3 | 70.0±1.7 | 70.6±1.8 | 70.4±1.4 | **70.7±1.9** | 70.6±1.9 |
| NLP-$\nu$MC(Base +Stacking) | 68.1±3.0 | 69.0±1.3 | 69.4±1.3 | 69.5±1.4 | 69.6±1.4 | 69.6±1.3 | 69.7±1.3 |

| | |
|---|---|
| MKL-prod | 62.2 ± 0.6 |
| MKL-avg ($\ell_\infty$) | 67.4 ± 1.1 |
| CLF ($\ell_\infty$) | 68.4 ± 0.7 |
| MKL-avg ($\ell_\infty$) (Base + Stacking) | 69.0 ± 1.3 |
| CLF ($\ell_\infty$) (Base + Stacking) | 69.7 ± 1.3 |

50 images per category are randomly selected for testing. The average accuracy is computed over all 101 object classes. This process is repeated 3 times and the mean accuracy over 3 splits is reported for each method. In this experiment, we combine 10 features channels based on the features introduced in [12,19] with dense sampling strategies. The RBF kernel function to compute kernel matrices from the $\chi^2$ distance matrices is given in Eq. (14). The experimental setup is the same as for Flower 17.

The results of the proposed methods are presented in Table 4 and compared with other techniques. The baseline for MKL gives 67.4% and the baseline for CLF gives 68.5%. The best result among MKL is achieved by multiclass MKL ($\ell_1$). It performs 1.2% better than the MKL baseline and performs similar to CLF baseline. Stacking does not perform well on this dataset. It performs 0.6% better than the MKL baseline, however, it performs worse than both CLF baseline and multiclass MKL. Classifier level fusion achieves best results on this dataset (NLP-$\boldsymbol{\beta}\ell_3$)). It performs 1.8% and 0.7% better than MKL and CLF baselines and performs 0.6% better than multiclass MKL. The results can be further improved by using the stacking kernel with the 10 base kernels. We achieve state-of-the-art results on this dataset by combining the stacking kernel with the 10 base kernels using CLF. This combination performs 3.3%, 2.7%, 2.2% and 2.1% better than the MKL baseline, stacking, the CLF baseline and multiclass MKL. Note that we are unable to compute the Mean accuracy for NLP-B, especially for $\ell_p$-norm greater than 1, due to a large number of constraints in the optimization problem.

It is well known that the type and the number of kernels have a large impact on the overall performance. Therefore, a direct comparison of scores with the published methods is not entirely fair. Nonetheless, it can be noted that the best performing methods on Caltech101 in [7] and [6] using a single kernel are giving 60% and 61% respectively. The performance in [6] using 8 kernels is close to 63% while the performance using 39 feature channels is 70.4%. Note that our best method gives 70.7% using 10 feature channels only, which can be considered as a significant improvement, given that we have used 4 times fewer feature channels.

## 6 Conclusions

In this paper we proposed a nonlinear separable convex optimization formulation for multiclass classifier fusion (NLP-$\nu$MC) which learns the weight for each class in every feature channel. We have also extended linear programming for binary and multiclass classifier fusion (ensemble methods) to nonlinear separable convex classifier fusion by incorporating arbitrary norms. Unlike the existing methods, these formulations do not reject informative feature channels and make the classifier fusion robust to both noisy and redundant feature channels which results in an improved performance.

We also extended stacking in the case of both binary and multiclass datasets. By considering stacking as a separate feature channel, we can combine the stacking kernel with base kernels using any proposed fusion method. We have performed comparative experiments on challenging object recognition benchmarks for both multi-label and multiclass cases. Our results show that optimal $p$ is an intrinsic property of kernels set and can be different for different datasets. It can be learnt systematically using validation set. In general if some channels are noisy $\ell_1$-norm is better (sparse weights). For carefully designed features non-sparse solutions, e.g., $\ell_2$-norm, are better. Note that both are special cases of our approaches. The proposed methods perform better than the state-of-the-art MKL methods. In addition to this, the non-sparse version of the classifier fusion is performing better than sparse selection of feature channels. We achieve state-of-the-art performance on all datasets by combining the stacking kernel with base kernels using classifier level fusion.

The two step training of classifier fusion may seem as an overhead. However, the first step is independent for each feature channel as well as each class and can be performed in parallel. Independent training also makes the systems applicable to large datasets. Moreover, in MKL one has to train an SVM classifier in $\alpha$-step before getting the optimal weights. As MKL is optimizing parameters jointly, one may argue that the independent optimization of weights in case of classifier fusion is less effective. However, as our consistently better results show, these schemes seem to be more suitable for visual recognition problems. The proposed classifier fusion schemes seem to be attractive alternatives to the state-of-the-art MKL approaches for both binary and multiclass problems and address the complexity issues of the MKL.

**Acknowledgements.** This research was supported by UK EPSRC EP/F0034 20/1, EP/F0694 21/1 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. Džeroski, S., Ženko, B.: Is combining classifiers with stacking better than selecting the best one? ML 54(3), 255–273 (2004)
3. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV 88(2), 303–338 (2010)

4. Fei-Fei, L., Fergus, R., Perona, P.: One-shot Learning of Object Categories. PAMI, 594–611 (2006)
5. Freund, Y., Schapire, R.: A Desicion-Theoretic Generalization of On-Line Learning and an Application to Boosting. In: CLT (1995)
6. Gehler, P., Nowozin, S.: On Feature Combination for Multiclass Object Classification. In: ICCV (2009)
7. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Tech. Rep. 7694, California Institute of Technology (2007)
8. Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. PAMI 20(3), 226–239 (1998)
9. Kloft, M., Brefeld, U., Sonnenburg, S., Zien, A., Laskov, P., Müller, K.: Efficient and Accurate lp-norm MKL. In: NIPS (2009)
10. Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L., Jordan, M.: Learning the Kernel Matrix with Semidefinite Programming. JMLR 5, 27–72 (2004)
11. Lazebnik, S., Schmid, C., Ponce, J.: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: CVPR (2006)
12. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI 27(10), 1615–1630 (2005)
13. Nilsback, M.E., Zisserman, A.: Automated Flower Classification over a Large Number of Classes. In: ICCVGIP (2008)
14. Nilsback, M., Zisserman, A.: A visual Vocabulary for Flower Classification. In: CVPR (2006)
15. Rakotomamonjy, A., Bach, F., Canu, S., Grandvalet, Y.: SimpleMKL. JMLR 9, 2491–2521 (2008)
16. Rätsch, G., Schölkopf, B., Smola, A., Mika, S., Müller, K., Onoda, T.: Robust Ensemble Learning for Data Analysis. In: PACKDDM (2000)
17. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. In: MLKDD, pp. 254–269 (2009)
18. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. JMLR 5, 101–141 (2004)
19. van de Sande, K., Gevers, T., Snoek, C.: Evaluation of color descriptors for object and scene recognition. In: CVPR (2008)
20. Sonnenburg, S., Rätsch, G., Schafer, C., Schölkopf, B.: Large Scale Multiple Kernel Learning. JMLR 7, 1531–1565 (2006)
21. Wolpert, D.: Stacked generalization. Neural Networks 5(2), 241–259 (1992)
22. Xie, N., Ling, H., Hu, W., Zhang, Z.: Use bin-ratio information for category and scene classification. In: CVPR (2010)
23. Yan, F., Mikolajczyk, K., Barnard, M., Cai, H., Kittler, J.: Lp norm multiple kernel fisher discriminant analysis for object and image categorisation. In: CVPR (2010)
24. Ying, Y., Huang, K., Campbell, C.: Enhanced protein fold recognition through a novel data integration approach. BMCB 10(1), 267 (2009)
25. Zien, A., Ong, C.: Multiclass Multiple Kernel Learning. In: ICML (2007)