Augmented Kernel Matrix vs Classifier Fusion for Object Recognition

Muhammad Awais
m.rana@surrey.ac.uk
Fei Yan
f.yan@surrey.ac.uk
Krystian Mikolajczyk
k.mikolajczyk@surrey.ac.uk
Josef Kittler
j.kittler@surrey.ac.uk

Centre for Vision,
Speech and Signal
Processing (CVSSP)
University of Surrey, UK

Abstract

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 Multiple Kernel Learning (MKL) that assigns the same weight to all examples in one feature space. However, the AKM approach is limited to small datasets due to its memory requirements. An alternative way to fuse information from different feature channels is classifier fusion (ensemble methods). There is a significant amount of work on linear programming formulations of classifier fusion (CF) in the case of binary classification. In this paper we derive primal and dual of AKM to draw its correspondence with CF. We propose a multiclass extension of binary ν-LPBoost, which learns the contribution of each class in each feature channel. Existing approaches of CF promote sparse features combinations, due to regularization based on ℓ_1-norm, and lead to a selection of a subset of feature channels, which is not good in case of informative channels. We also generalize existing CF formulations to arbitrary ℓ_p-norm for binary and multiclass problems which results in more effective use of complementary information. We carry out an extensive comparison and show that the proposed nonlinear CF schemes outperform its sparse counterpart as well as state-of-the-art MKL approaches.

1 Introduction

Due to the importance of complementary information in feature combination [9, 10, 12, 16, 18, 21], much research has been done in the field of feature design [11, 20] to diversify kernels. Proper selection and fusion of these kernels is, therefore, crucial. The key idea of MKL, is to learn a linear combination of base kernels by maximizing soft margin between classes [10]. Alternatively, AKM [23] is proposed arguing that in MKL a single kernel corresponding to a particular feature space is attributed a single weight. Therefore, MKL does not exploit information from individual samples in different feature spaces, e.g., in the context of object recognition, some samples can carry more shape information while others may carry more texture information for the same object category. In contrast to MKL, the

© 2011. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.
BMVC 2011 http://dx.doi.org/10.5244/C.25.60

main idea of CF [8] is to construct a set of base classifiers and then classify a new sample by a weighted combination of their predictors. CF attracted much attention, after the success of AdaBoost [4] in particular, in many practical applications [2, 7, 16]. This led to linear programming (LP) formulations of AdaBoost [7, 16]. The fundamental problem with AKM is its large augmented matrix which makes it inapplicable to large datasets. We derive primal and dual of AKM and draw a comparison between AKM and CF, by carefully analysing the dual and the feature space of AKM.

We present a novel multiclass CF scheme (NLP-νMC) based on binary $\nu-LPBoost$ [16], which incorporates arbitrary norms $\{\ell_p, p \geq 1\}$ and optimizes the contribution from each class in each feature channel. We also incorporate nonlinear constraints in previously proposed binary $\nu-LPBoost$ and multiclass LPBoost [5] and show empirically that the nonlinear variants perform consistently better than their sparse counterparts, and baseline methods. The proposed optimization problems are nonlinear separable convex problem which can be solved using off-the-shelf solvers. It is important to note that both LP-β and LP-B [5] are different than NLP-νMC. In particular, the number of constraints in the optimization problems and the concept of margin are significantly different (see Section 3.1). For example, LP-B is not applicable to large multiclass datasets due to large number of constraints. We consider our extensive evaluation and comparison to the state-of-the-art fusion approaches as another important contribution of the paper. We perform experiments on multilabel and multiclass problems using standard benchmarks including Pascal VOC 2007, Flower 17, Flower 102 and Caltech101. Our multiclass formulation and nonlinear extensions of CF consistently outperforms the state-of-the-art MKL and sparse CF schemes. Note that we use object recognition datasets for evaluation, however, the proposed fusion schemes can be applied to any underlying pattern recognition problems provided that we have multiple feature channels.

The rest of paper is organized as follows. In Section 2 we present AKM structure and drive its primal and dual to draw its correspondence with classifier fusion. We then discuss LP formulation of CF in Section 3 which also present proposed multiclass CF scheme and extend LP formulation of binary and multiclass classifier fusion. In Section 4 we present the evaluation results and conclude in Section 5.

2 AKM and its Correspondence to Classifier Fusion

In this section, we first present the structure of AKM and give its primal and dual formulations for binary classification. We then draw the correspondence between AKM and classifier fusion by analysing the dual of AKM. 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 and n test kernels $\dot{K}_p$. Each kernel $K_p = \langle \Phi_p(x_i), \Phi_p(x_j) \rangle$ implicitly maps samples x_i from the input space to a feature space with a mapping function $\Phi_p(x_i)$. In MKL the aim is to find a linear combination $\sum_{p=1}^{n} \beta_p K_p$, normal vector $\mathbf{w}$ and bias b of separating hyperplane simultaneously such that the soft margin between classes is maximized [10]. The primal and its corresponding dual for a linear combination of kernels are derived for various formulations in [1, 9, 10, 19]. In contrast, in AKM [23], given a set of base training kernels (K_p) the 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 the diagonal. The zeros on off diagonal reflect that there are no cross terms between different kernel matrices, hence, the feature spaces of base kernels in AKM do not interfere with each other [1]. This fact is important and can be used to show the relationship between AKM and classifier fusion. Note that all base kernels are of size $m \times m$ while the AKM is of size $(n \times m) \times (n \times m)$, thus it uses $n \times m$ training samples instead of m. The primal of AKM scheme is then given by:

$$\min_{w,\xi,b} \quad \frac{1}{2}\sum_{p=1}^{n}\langle w_p, w_p\rangle + C\sum_{i=1}^{n\times m}\xi_i \tag{2}$$
$$s.t. \quad y_i(\sum_{p=1}^{n}\langle w_p, \Phi_p(x_i)\rangle + b) \geq 1-\xi_{pi}, \;\; \xi_{pi} \geq 0, \;\; i = 1,\ldots,m, \;\; p = 1,\ldots,n$$

The dual of Eq. (2) can be derived using Lagrange multiplier techniques. Note that the same samples from different feature channels are added as separate examples of the same class, therefore, one Lagrange multiplier α_{pi} is learned for each sample from each feature channel. The dual of AKM is given as follows:

$$\max_{\alpha} \sum_{i=1}^{m}\alpha_{1i} - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_{1i}\alpha_{1j}y_iy_j\langle\Phi_1(x_i),\Phi_1(x_j)\rangle + \ldots + \tag{3}$$
$$\sum_{i=1}^{m}\alpha_{ni} - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_{ni}\alpha_{nj}y_iy_j\langle\Phi_n(x_i),\Phi_n(x_j)\rangle$$
$$s.t. \sum_{i=1}^{m}\alpha_{1i}y_i + \ldots + \sum_{i=1}^{m}\alpha_{ni}y_i = 0, \;\; 0 \preceq \alpha \preceq C,$$

By comparing Eq. (3) to standard dual formulation of single kernel SVM it can be seen that the AKM dual consists of sum of n duals problems corresponding to each base kernel. Also there are no cross term between different feature spaces in objective which points to the fact that feature channels are independent. Therefore, solving optimization problem of Eq. (3) is similar to solving the dual problem of each base kernel and then sum them together. In case of test pattern the solution will be the unweighted sum of output from each base classifier which in fact is classifier fusion with unweighted sum.

3 Classifier Fusion with Non-Linear Constraints

In this section we review the LP formulation of CF (ensemble methods) based on boosting. We also extend the ν-LP-AdaBoost [16] formulation for binary classification with nonlinear constraints to avoid discarding of channels with complementary information while keeping it robust to noisy feature channels. We focus on the LP formulations of AdaBoost [4] and its soft margin formulations [16]. It has been argued that AdaBoost tries to maximizes smallest margin ρ on the training set [7, 16]. Based on this and the idea of soft margin SVM, ν-LP-AdaBoost has been proposed in [16]. Roughly speaking the minimum margin of an ensemble on a training set is the smallest confidence it gives to a training example. We define margin (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, $\rho := \min_{1\leq i\leq m} y_i \sum_{r=1}^{n}\beta_r g_r(x_i)$.

[1] It can also be seen by analysing the Empirical feature space X for training kernel K of size $m \times m$ which can be derived by eigen value decomposition as shown in [23].

The ν-LP-AdaBoost performs a sparse selection of feature channels due to ℓ_1 regularization, which is suboptimal if all feature channels carry complementary information. Similarly, in the case of ℓ_∞ norm, noisy features channels may have significant impact on results. To address these problems we generalize binary classifier fusion for arbitrary norms $\{\ell_p, p \geq 1\}$. In contrast to AdaBoost, we consider n fixed number of base classifiers $\{g_r, \forall r = 1,\ldots,n\}$ which are independently trained. A feature channel gives rise to a kernel K_r, which is used to train a base classifier, $g_r(x)$. As number of base hypothesis is fixed, the aim of ensemble learning in this case is to find the optimal weight vector β for a linear combination of the base classifiers, $f(x) = \sum_{r=1}^{n} \beta_r g_r(x)$. Given base classifiers, we learn the optimal weights β_r by maximizing the smallest margin ρ in the following optimization problem:

$$\begin{aligned} &\max_{\beta,\xi,\rho} \quad \rho - \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i \\ &s.t. \quad y_i \sum_{r=1}^{n} \beta_r f_r(x_i) \geq \rho - \xi_i\,, \quad \|\beta\|_p^p \leq 1, \quad \beta \succeq 0, \xi \succeq 0, \rho \geq 0 \;\; \forall\, i = 1,...,m \end{aligned} \tag{4}$$

where ξ_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 (4) is a nonlinear separable convex optimization problem and can be solved efficiently for global optimal solution by standard optimization toolboxes[2].

3.1 Multiclass Classifier Fusion with Non-Linear Constraints

In this section we propose a novel multiclass extension of ν-LP-AdaBoost and compare it with other existing multiclass variants. We also incorporate nonlinear constraints in two existing multiclass classifier fusion schemes: LP-β [5] and LP-B [5]. The empirical results show that the nonlinear constraints improve the performance of these methods.

Nonlinear Programming ν-Multiclass (NLP-νMC): 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, each base classifiers now maps into an N_C dimensional space, $g_r(x) \mapsto \mathbb{R}^{N_C}$, and the output corresponding to c'th class is denoted by $g_{r,c}(x)$. We train all base classifiers individually as in the case of binary classifier fusion. 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 $\beta \in \mathbb{R}^n$ as in case of binary classifier fusion, we have an $n \times N_C$ dimensional weight vector $\beta \in \mathbb{R}^{n \times N_C}$. After finding the optimal weights, the decision function for a test sample x is given by selecting maximum response class among weighted sum of classes across all channels. We extend the definition of margin for binary CF to multiclass CF as follows:

$$\rho(x_i,\beta) := \sum_{r=1}^{n} \beta_{(N_C(r-1)+y_i)} g_{r,y_i}(x_i) - \sum_{r=1}^{n}\sum_{j=1,j\neq i}^{N_C} \beta_{(N_C(r-1)+y_j)} g_{r,y_j}(x_i) \tag{5}$$

The classification confidence for examples x_i depends upon β and scores from base classifiers. The main difference between the two margins is that here, we are taking the class

[2]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.

confidence of true target class and subtracts the combined effect of all the non-target classes from it, this difference is then summed over all feature channels. This is done for all n feature channels. The normalized (smallest) margin can then be defined as $\rho := \min_{1\leq i\leq m} \rho(x_i,\beta)$. Inspired by the soft margin LP formulations of AdaBoost we propose to maximize the normalized margin ρ to learn linear combination of base classifiers. 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 optimization problem is given by:

$$\max_{\beta,\xi,\rho} \quad \rho - \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i \tag{6}$$

$$\begin{aligned} s.t. \quad & \sum_{r=1}^{n}\beta_{(N_C(r-1)+y_i)}g_{r,y_i}(x_i) - \sum_{r=1}^{n}\sum_{j=1,j\neq i}^{N_C}\beta_{(N_C(r-1)+y_j)}g_{r,y_j}(x_i) \\ & \geq \rho - \xi_i \;\; i=1,...,m, \\ & \|\beta\|_p^p \leq 1, \rho \geq 0, \beta \succeq 0 \quad \xi \succeq 0 \quad \forall i = 1,...,m \end{aligned} \tag{7}$$

where $\frac{1}{\nu m}$ is the regularization constant and gives a trade-off between minimum classification confidence ρ and the margin errors. The main difference between this formulation and the formulation in Eq. (4) is the definition of margin used in the constraints in Eq. (7), in which the difference between the classification confidence of the true class and the joint confidence of all other classes is lower bounded. 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 variables positivity constraints). Therefore, the difference in complexity, compared to the binary classifier fusion, is the increased number of variables in weight vector β, while having the same number of constraints. In the rest of this section, we extend two multiclass CF schemes: LP-β and LP-B, proposed in [5] 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. The optimization problems of NLP-νMC, NLP-β and NLP-B, are nonlinear separable convex and can be solved using MOSEK.

Nonlinear Programming-β (NLP-β): We generalize LP-β [5] by incorporating $\ell_p, \forall p \geq 1$ norm constraints. The optimization problem is given by:

$$\min_{\beta,\xi,\rho} \quad -\rho + \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i \tag{8}$$

$$s.t. \quad \sum_{r=1}^{n}\beta_r g_{r,y_i}(x_i) - max \sum_{y_j\neq y_i, r=1}^{n}\beta_r g_{r,y_j}(x_i) \geq \rho - \xi_i, \;\; \forall \;\; i=1,...,m \tag{9}$$

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

Note that weight vector β lies in an n dimensional space $\beta \in \mathbb{R}^n$ as in classifier fusion. After finding the weight vector β, the decision function of generalized LP-β 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 [5] with arbitrary regularization norms $\ell_p, \forall p \geq 1$. Instead of having a weight vector β, 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:

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

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

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

The constraints in Eq. (11) gives a lower bound on the pairwise difference between classification confidences 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 presented in this section is in the definition of the feasible region which is defined by Eq. (7), Eq. (9) and Eq. (11) for NLP-νMC, NLP-β and NLP-B respectively. In NLP-β 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 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 numbers of constraints are $m \times (N_C - 1) + 1$. The large number of constraints makes this approach less attractive for datasets with a large number of classes. For example, for Caltech101 [3] with only 15 images per class for training, the number of constraints for LP-B is more than 150 thousand ($15 \times 101 \times 100 + 1 \cong 1.5 \times 10^5$). The feasible region of NLP-νMC, depends upon the joint classification confidence of all the non-target classes subtracted from classification confidence of the true class. Thus, the feasible region of NLP-νMC is much smaller than the feasible region of NLP-B. Due to these joint constraints the total number of constraints for NLP-νMC is $m+1$, e.g., for Caltech101 with 15 images per class for training, the number of constraints for NLP-νMC is only 1516 (15*101+1) which is only 1% of the constraints in NLP-B. We, therefore, can apply NLP-νMC to large multiclass datasets, as opposed to NLP-B, especially for norms greater than 1. Note that the difference in complexity between NLP-νMC and NLP-β or binary CF is the extended weight vector β.

4 Experiments and Discussion

This section presents the experimental evaluation of the methods investigated in this paper on a multi-label dataset, namely, Pascal VOC 2007 and three multiclass datasets, namely, Flower17, Flower102 and Caltech101. For multi-label dataset we use binary relevance [17] as it is recommended by the organizers of Pascal VOC challenge [7]. The MKL results on Pascal VOC 2007 are reported using binary MKL from SHOGUN toolbox[3], and for CF we use ν-LP-AdaBoost given in Eq. (4). For multiclass dataset we have used multiclass MKL from the SHOGUN. For CF we use three CF schemes proposed in this paper namely, NLP-νMC, NLP-β and NLP-B given in Section 3.1. We do not have results for higher values of norms in case of NLP-B, and for some norms in case of MKL because their optimization

[3] http://www.shogun-toolbox.org/

Fusion Methods \ norms	1	$1+2^{-3}$	$1+2^{-2}$	$1+2^{-1}$	2	3	4	8	ℓ_∞
MKL	55.4	56.4	58.5	61.1	621.0	62.5	62.6	62.8	62.9
CF	63.7	63.9	**64.0**	**64.0**	**64.0**	**64.0**	63.8	63.7	63.1

Table 1: Mean Average Precision of PASCAL VOC 2007.

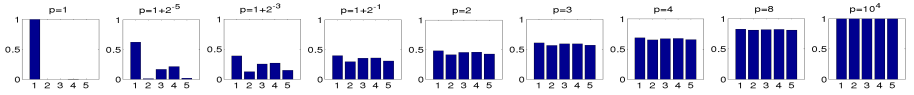

Figure 1: Pascal VOC 2007. Feature channels weights learned with various ℓ_p for CF(ℓ_p)

problems take several days. On the other hand NLP-β and NLP-νMC are very fast as compared to multiclass MKL and NLP-B and take few seconds and few minutes, respectively. We have verified equivalence of AKM and CF (ℓ_∞) empirically and got the same results (up to 4^{th} significant figure), therefore, we are only presenting results in term of CF.

4.1 Pascal VOC 2007

Pascal VOC 2007 [7] is one of the most challenging object recognition dataset consisting of 20 object classes with 9963 image examples. Classification of 20 object categories is handled as 20 independent binary classification problems. We present results using average precision (AP) [7] and mean average precision (MAP).

We combined 5 base kernels to produce state-of-the-art results on this dataset by using descriptors introduced in [11, 20]. We use RBF kernel [19] based on χ^2 distance matrix. We apply SVM as base classifiers, for CF schemes proposed in this paper, with the regularization parameter C in the set $\{2^{(-2,0,3,7,10,15)}\}$. The regularization parameter ν 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 MKL and CF 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. Figure 1 shows learned weights on the training set of aeroplane category of Pascal VOC 2007 for several values of p using CF. The plotted weights are corresponding to the optimal value of C. The sparsity of learned weights can be observed easily for low values of p. The sparsity decreases with increased p, up to uniform weights (corresponding to ℓ_∞) achieved at $p = 10000$. Weights can also be learned corresponding to best performing p on validation set. The results for several fusion methods are given in Table 1. Low performance of MKL-ℓ_1-norm, which leads to sparse selection, indicates that base kernels carry complementary information. Therefore, the non-sparse MKL or CF methods, 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 CF schemes outperform the state-of-the-art MKL (ℓ_2-norm, ℓ_∞-norm) by 2 % and 1.1% respectively.

4.2 Oxford Flower 17

Oxford Flower 17 [14] consists of 17 categories with 80 images in each category. The dataset is split into training, validation and test using 3 predefined random splits by the authors of the dataset. For experiments we have used the 7 χ^2 distance matrices provided online[4]. RBF kernels are computed using these distance matrices. We have used SVM as a base

[4] http://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html

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-β	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-νMC	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

Comparison with State-of-the-Art	
MKL-prod [8]	85.5 ± 1.2
MKL-avg (ℓ_∞) [8]	84.9 ± 1.9
CF (ℓ_∞) / AKM	86.7 ± 2.7
CG-Boost [8]	84.8 ± 2.2
MKL (SILP or Simple) [8] or OBSCURE [13]	85.2 ± 1.5
LP-β [8]	85.5 ± 3.0
LP-B [8]	85.4 ± 2.4
MKL-FDA (ℓ_p) [2]	86.7 ± 1.2

Table 2: Classification Rate on Flower17.

ML-Methods	1	$1+2^{-3}$	$1+2^{-1}$	2	3	4	8	ℓ_∞
MKL	69.9	64.7	65.3	65.9	65.7	-	-	73.4
NLP-β	61.2	**75.7**	73.5	74.7	73.0	73.9	74.6	73.0
NLP-νMC	72.6	73.1	73.2	73.3	73.4	73.4	73.4	73.0
NLP-B	73.6	-	-	-	-	-	-	73.0

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

Table 3: Mean accuracy on Oxford Flower 102 dataset.

classifier and its regularization parameter is in the range $\{10^{(-2,-1,\ldots,3)}\}$. The Regularization parameter for different CF is in the range $\nu \in \{0.05, 0.1, \ldots, 0.95\}$. Both SVM and CF 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 [8].

The results are given in Table 2 and compared with the state-of-the-art. The baselines for MKL and CF are MKL-avg(ℓ_∞) and CF(ℓ_∞). Nonlinear versions of classifier fusion perform better than their sparse counterparts as well as state-of-the-art MKL. The best result in CF is obtained by the proposed NLP-νMC (ℓ_2) and NLP-β (ℓ_4). They outperform the MKL baseline by more than 2.5% and multiclass MKL by 0.6%. The second half of Table 2 shows comparison with published state-of-the-art results. According to our knowledge the best performing method published, using the 7 distance matrices provided online, is giving 86.7% which is similar to the CF baseline. Our best CF method outperforms it by 1.2%.

4.3 Oxford Flower 102

Oxford Flower 102 [13] is an extended multiclass dataset containing 102 flower categories. The dataset is split into training, validation and test using a split predefined by the authors of the dataset. For the experiments we use RBF kernels using 4 χ^2 distance matrices provided online[5]. The experimental setup is the same as for Oxford Flower 17.

The results are given in Table 3. We have not reported the variance of the results as the authors have given only 1 split online and for a fair comparison with previously published results we use the same split as used by other authors. Multiclass MKL doesn't perform well on this dataset and performs significantly lower that its baseline (MKL ℓ_∞). The best among classifier fusion is the NLP-β ($\ell_{1+2^{-3}}$) scheme. It performs 5.8% better than multiclass MKL and 2.3%, 2.7% better than MKL and CF baselines, respectively. Note that NLP-νMC performs slightly worse than NLP-β as it has to estimate N_C times more parameter than NLP-

[5] http://www.robots.ox.ac.uk/~vgg/data/flowers/102/index.html

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-β	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-νMC	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	-	-	-	-	-	-

MKL-prod	62.2 ± 0.6
MKL-avg (ℓ_∞)	67.4 ± 1.1
CF (ℓ_∞)	68.4 ± 0.7

Table 4: Mean accuracy on Caltech101 dataset.

β in the presence of few training example per category. For example, compared to Flower17 it has 2 times less training data per class while numbers of classes are 6 times more. We expect NLP-νMC to perform better in the presence of more training data. The results for MKL are reported from [13] for comparison. In comparison to the published results, our best method has an improvement of 3.4% which is a significant gain.

4.4 Caltech101

Caltech101 [8] 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 $\times$ 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 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 [11, 20] with dense sampling strategies. We use RBF kernel function to compute kernel matrices from the χ^2 distance matrices. The experimental setup is the same as for Oxford Flower 17.

The results of the proposed methods are presented in Table 4. Classifier fusion achieves best results on this dataset (NLP-$\beta\ell_3$)). It performs 1.8% and 0.7% better than MKL and CF baselines and performs 0.6% better than multiclass MKL. NLP-νMC performs slightly worse than NLP-β as it has to estimate N_C times more parameter than NLP-β in the presence of few training example per category. For example, compared to Flower17 it has 3 times less training data per class while numbers of classes are 6 times more. It is well known that the type and 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 [6] and [5] using a single kernel are giving 60% and 61% respectively. The performance in [5] using 8 feature channels is close to 63% while the performance using 39 kernels is 70.4%. Similarly, performance in [15] using 39 kernels is approximately 69%, while picking the best 5 kernels out of 39 is giving approximately 70%. Note that our best method is giving 69.2% using 10 kernels only.

5 Conclusions

In this paper we draw a correspondence between AKM and CF with unweighted sum of ensembles. We also proposed a nonlinear separable convex optimization formulation for multiclass classifier fusion (NLP-νMC) which learn the weight for each class in every feature channel. We have also extended linear programming for binary and multiclass classifier fusion 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 have performed comparative experiments on challenging object recognition benchmarks for both multi-label and multiclass cases. The proposed methods perform better than the state-of-the-art MKL methods. In addition to this, the non-sparse version of the CF is performing better than sparse selection of feature channels.

The two step training of CF 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 several times in α-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 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] F.R. Bach, G.R.G. Lanckriet, and M.I. Jordan. Multiple Kernel Learning, Conic Duality, and the SMO Algorithm. In *International Conference on Machine Learning*, 2004.

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2): 303–338, June 2010.

[3] L. Fei-Fei, R. Fergus, and P. Perona. One-shot Learning of Object Categories. *PAMI*, pages 594–611, 2006.

[4] Y. Freund and R. Schapire. A Desicion-Theoretic Generalization of On-Line Learning and an Application to Boosting. In *Computational Learning Theory*, 1995.

[5] P. Gehler and S. Nowozin. On Feature Combination for Multiclass Object Classification. In *International Conference on Computer Vision*, 2009.

[6] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[7] A.J. Grove and D. Schuurmans. Boosting in the Limit: Maximizing the Margin of Learned Ensembles. In *National Conference on Artificial Intelligence*, 1998.

[8] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[9] M. Kloft, U. Brefeld, S. Sonnenburg, A. Zien, P. Laskov, and KR Müller. Efficient and Accurate lp-norm MKL. In *Neural Information Processing Systems*, 2009.

[10] G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L.E. Ghaoui, and M.I. Jordan. Learning the Kernel Matrix with Semidefinite Programming. *The Journal of Machine Learning Research*, 5:27–72, 2004.

[11] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[12] K.R. Muller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

[13] M-E. Nilsback and A. Zisserman. Automated Flower Classification over a Large Number of Classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

[14] M.E. Nilsback and A. Zisserman. A visual Vocabulary for Flower Classification. In *Computer Vision and Pattern Recognition*, 2006.

[15] F. Orabona, L. Jie, and B. Caputo. Online-batch strongly convex multi kernel learning. 2010.

[16] G. Rätsch, B. Schölkopf, A. Smola, S. Mika, K.R. Müller, and T. Onoda. Robust Ensemble Learning for Data Analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2000.

[17] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning and Knowledge Discovery in Databases*, pages 254–269, 2009.

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

[19] S. Sonnenburg, G. Rätsch, C. Schafer, and B. Schölkopf. Large Scale Multiple Kernel Learning. *The Journal of Machine Learning Research*, 7:1531–1565, 2006.

[20] K. van de Sande, T. Gevers, and C. Snoek. Evaluation of color descriptors for object and scene recognition. In *Computer Vision and Pattern Recognition*, 2008.

[21] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 2000.

[22] F. Yan, K. Mikolajczyk, M. Barnard, H. Cai, and J. Kittler. Lp norm multiple kernel fisher discriminant analysis for object and image categorisation. In *Computer Vision and Pattern Recognition*, 2010.

[23] F. Yan, K. Mikolajczyk, J. Kittler, and A. Tahir. Combining Multiple Kernels by Augmenting the Kernel Matrix. In *International Workshop on Multiple Classifier Systems*, 2010.