Multiple Kernel Learning and Feature Space Denoising

FEI YAN   JOSEF KITTLER   KRYSTIAN MIKOLAJCZYK

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

Abstract

We review a multiple kernel learning (MKL) technique called ℓ_p regularised multiple kernel Fisher discriminant analysis (MK-FDA), and investigate the effect of feature space denoising on MKL. Experiments show that with both the original kernels or denoised kernels, ℓ_p MK-FDA outperforms its fixed-norm counterparts. Experiments also show that feature space denoising boosts the performance of both single kernel FDA and ℓ_p MK-FDA, and that there is a positive correlation between the learnt kernel weights and the amount of variance kept by feature space denoising. Based on these observations, we argue that in the case where the base feature spaces are noisy, linear combination of kernels cannot be optimal. An MKL objective function which can take care of feature space denoising automatically, and which can learn a truly optimal (non-linear) combination of the base kernels, is yet to be found.

Keywords:

Kernel Methods, Multiple Kernel Learning, Kernel FDA, Kernel PCA

1 Introduction

Kernel methods [11, 13] have proven successful for many machine learning problems since their introduction in the mid-1990s. Representative methods such as support vector machine (SVM) [16, 13], kernel Fisher discriminant analysis (kernel FDA) [9, 2], kernel principal component analysis (kernel PCA) [12] have been reported to produce the state-of-the-art performance in numerous applications. Kernel methods work by embedding data items in an input space (vector, graph, string, etc.) into a vector space called a feature space, and applying linear methods in such a feature space. This embedding is defined implicitly by specifying an inner product for the feature space via a positive semidefinite (PSD) kernel function: $k(\mathbf{x}_i, \mathbf{x}_j) =< \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) >$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two data items in the input space, and $\phi(\cdot)$ is the (implicit) mapping function.

In kernel methods, the choice of kernel function is critically important, since it completely determines the embedding of the data in the feature space. In many problems, multiple kernels capturing different "views" of the problem are available. In such a situation, one naturally wants to combine these kernels in an "optimal" way. This multiple kernel learning (MKL) problem was pioneered by Lancriet et al. in [8], where the key idea is to learn a linear combination of a given set of base kernels by maximising the (soft) margin between two classes or by maximising kernel alignment. Following this seminal work, MKL has become one of the most active areas in the machine learning community in the past few years. Various extensions have been made to [8]. For example, the efficiency of MKL is significantly improved in [1, 14, 10]; multiclass and multilabel MKL are proposed in [20, 5]; in [6, 19, 17, 18], the ratio of the inter- and intra- class scatters of FDA is maximised instead of the margin and kernel alignment. All these MKL methods learn a linear combination of base kernels, which corresponds to concatenation of base feature spaces. We argue that in the case where the base feature spaces are noisy, linear combination of kernels cannot be optimal, since the noisy dimensions of a base feature space can not be eliminated completely as long as the weight assigned to this kernel is not zero. In such a situation, a better strategy would be to denoise each base kernel before applying MKL.

In this paper, we present an approach that combines denoising of feature space and multiple kernel Fisher discriminant analysis (MK-FDA). In Section 2 we review an ℓ_p MK-FDA method that we recently proposed. We then introduce in Section 3 feature space denoising by means of kernel PCA. In Section 4, we show the effect of feature space denoising on MKL, with experiments on a challenging object recognition dataset, and provides some insights on the connection between feature space denoising and MKL. Finally conclusions are given in Section 5.

2 ℓ_p Norm Multiple Kernel Fisher Discriminant Analysis

For the sake of completeness, in this section we review an ℓ_p regularised MK-FDA method we recently proposed [18]. Most MKL techniques learn kernel weights by maximising some measure of class separation. The kernel weights must be regularised in order to make sure this measure of separation remains meaningful and does not become arbitrarily large. It is known that in an optimisation problem, the regularisation norm controls the sparsity of the solution. For example, an ℓ_1 norm regularisation promotes sparse solution, while ℓ_2 regularisation tends to produce non-sparse solution. In our ℓ_p MK-FDA, the kernel weights can be regularised with a general ℓ_p norm for any $p \geq 1$. This allows to learn the intrinsic sparsity of the given set of base kernels by tuning the regularisation norm p on an independent validation set. As a result, using the learnt optimal norm p in the proposed ℓ_p MK-FDA offers better performance than ℓ_1, ℓ_2, or ℓ_∞ MK-FDAs. In the following, we first formulate the associated optimisation problem, then solve it with semi-infinite programming.

2.1 Problem formulation

We consider a binary classification problem. Suppose we are given n $m \times m$ training kernel matrices $K_j, j = 1, \cdots, n$ and m class labels $y_i \in \{1, -1\}, i = 1, \cdots, m$, where m is the number of training samples. Our goal is to learn optimal kernel weights $\boldsymbol{\beta} \in \mathbb{R}^n$ for the linear combination of n kernels under the ℓ_p constraint: $K = \sum_{j=1}^n \beta_j K_j, \beta_j \geq 0, ||\boldsymbol{\beta}||_p^p \leq 1$ for any $p \geq 1$, such that the ratio criterion of FDA is maximised. The $p \geq 1$ requirement is to ensure that the triangle inequality is satisfied and that $|| \cdot ||_p$ defines a norm.

Let m^+ be the number of positive training samples, and $m^- = m - m^+$ the number of negative training samples. For a given kernel K, we assume it has been centred in its feature space [12]. Let μ^+ and μ^- be the centroids of the positive and negative samples in the feature space, respectively, and C^+ and C^- be the covariance matrices of the two classes, respectively. The between class scatter S_B and within class scatter S_w are defined as:

$$S_B = \tfrac{m^+ m^-}{m}(\mu^+ - \mu^-)(\mu^+ - \mu^-)^T \tag{1}$$

$$S_W = m^+ C^+ + m^- C^- \tag{2}$$

The objective of single kernel FDA is to find the projection direction $\mathbf{w}$ in the feature space that maximises $\frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$, or equivalently, $\frac{\mathbf{w}^T \frac{m}{m^+ m^-} S_B \mathbf{w}}{\mathbf{w}^T S_T \mathbf{w}}$, where $S_T = S_B + S_W$ is the total scatter matrix. In practice a regularised version,

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \frac{m}{m^+ m^-} S_B \mathbf{w}}{\mathbf{w}^T (S_T + \lambda I)\mathbf{w}} \tag{3}$$

is maximised to improve generalisation and numerical stability [9], where λ is a small positive number.

Exploring the link between FDA and regularised least squares (RLS) and using the duality theory of optimisation, it is proved in [19] that the maximal value of (3) is given by (up to an additive constant determined by the labels):

$$J^* \sim \min_{\boldsymbol{\alpha}} (\frac{1}{4}\boldsymbol{\alpha}^T (I + \frac{1}{\lambda} K)\boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{a}) \tag{4}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^m$ and $\mathbf{a} = (\frac{1}{m^+}, \cdots, \frac{1}{m^+}, \frac{-1}{m^-}, \cdots, \frac{-1}{m^-})^T \in \mathbb{R}^m$ contains the centred labels. Now consider the case where the kernel K can be chosen from linear combinations of a set of base kernels. The kernel weights must be regularised somehow to make sure (4) remains meaningful. We impose a general ℓ_p regularisation $||\boldsymbol{\beta}||_p^p \leq 1$ for any $p \geq 1$, and use second order Taylor expansion to approximate this constraint [7]:

$$\begin{aligned} ||\boldsymbol{\beta}||_p^p \quad &\approx \quad \frac{p(p-1)}{2} \sum_{j=1}^n \tilde{\beta}_j^{p-2} \beta_j^2 - \sum_{j=1}^n p(p-2)\tilde{\beta}_j^{p-1} \beta_j \\ &+ \frac{p(p-3)}{2} + 1 \quad := \quad \nu(\boldsymbol{\beta}) \end{aligned} \tag{5}$$

where $\tilde{\beta}_j$ is the current estimate of β_j in an iterative process, which will be explained in more detail in the next section. After some arrangements, we arrive at the binary ℓ_p MK-FDA optimisation problem. Under the ℓ_p constraint, the optimal K maximising (4) is found by solving:

$$\max_{\boldsymbol{\beta}} \min_{\boldsymbol{\alpha}} S(\boldsymbol{\alpha}, \boldsymbol{\beta}) \quad \text{s.t.} \;\; \boldsymbol{\beta} \geq \mathbf{0}, \;\; \nu(\boldsymbol{\beta}) \leq 1 \tag{6}$$

where

$$S(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{4\lambda}\boldsymbol{\alpha}^T \sum_{j=1}^n \beta_j K_j \boldsymbol{\alpha} + \frac{1}{4}\boldsymbol{\alpha}^T \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{a} \tag{7}$$

2.2 Solving the optimisation problem with SIP

A semi-infinite program (SIP) is an optimisation problem with finite number of variables $\mathbf{x} \in \mathbb{R}^{\mathbf{d}}$ on a feasible set described by infinitely many constraints [4]. It is straightforward to show that (6) is equivalent to a SIP:

$$\begin{aligned} &\max\nolimits_{\theta, \boldsymbol{\beta}} \quad \theta \\ \text{s.t.} \quad &\boldsymbol{\beta} \geq \mathbf{0}, \quad \nu(\boldsymbol{\beta}) \leq 1, \quad S(\boldsymbol{\alpha}, \boldsymbol{\beta}) \geq \theta \quad \forall \boldsymbol{\alpha} \in \mathbb{R}^m \end{aligned} \tag{8}$$

We adapt the wrapper algorithm proposed in [14] to solve (8). The basic idea is to divide a SIP into an inner sub-problem and an outer sub-problem. The algorithm alternates between solving the two sub-problems until convergence. At step t, assuming the current optimal $(\theta^{(t)}, \boldsymbol{\beta}^{(t)})$

Table 1. An iterative algorithm for solving the SIP problem (8)

```
• Initialisation: $S^{(0)} = 1$, $\theta^{(1)} = -\infty$, $\beta_j^{(1)} = n^{-1/p}$ for $j = 1, \cdots, n$
• for $t = 1, 2, \cdots$ do
    – Compute $\boldsymbol{\alpha}^{(t)} = \arg\min_{\boldsymbol{\alpha}} S(\boldsymbol{\alpha}, \boldsymbol{\beta}^{(t)})$ using (10)
    – Compute $S^{(t)} := S(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)})$
    – if $|1 - \frac{S^{(t)}}{\theta^{(t)}}| \leq \epsilon$ break
    – Compute $(\theta^{(t+1)}, \boldsymbol{\beta}^{(t+1)}) = \arg\max_{\theta,\boldsymbol{\beta}} \theta$ in (11),
      where $\nu(\boldsymbol{\beta})$ is defined as in (5) with $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(t)}$.
• end for
```

have been obtained in the outer sub-problem, the inner sub-problem identifies the constraint that maximises the constraint violation for $(\theta^{(t)}, \boldsymbol{\beta}^{(t)})$:

$$\boldsymbol{\alpha}^{(t)} := \arg\min_{\boldsymbol{\alpha}} S(\boldsymbol{\alpha}, \boldsymbol{\beta}^{(t)}) \tag{9}$$

Observing that (9) is an unconstrained quadratic program, $\boldsymbol{\alpha}^{(t)}$ is obtained by solving the following linear system [19]:

$$(\frac{1}{2}I + \frac{1}{2\lambda}\sum_{j=1}^{n}\beta_j^{(t)}K_j)\boldsymbol{\alpha}^{(t)} = \mathbf{a} \tag{10}$$

If $\boldsymbol{\alpha}^{(t)}$ satisfies constraint $S(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}) \geq \theta^{(t)}$ then solution $(\theta^{(t)}, \boldsymbol{\beta}^{(t)})$ is optimal. Otherwise, the constraint is added to the set of constraints and the algorithm proceeds to the outer sub-problem of step $t + 1$.

At step t, the outer sub-problem computes the optimal $(\theta^{(t+1)}, \boldsymbol{\beta}^{(t+1)})$ in (8) for a restricted subset of constraints:

$$(\theta^{(t+1)}, \boldsymbol{\beta}^{(t+1)}) = \arg\max\nolimits_{\theta,\boldsymbol{\beta}} \theta \tag{11}$$
$$\text{s.t.} \quad \boldsymbol{\beta} \geq \mathbf{0},\ \nu(\boldsymbol{\beta}) \leq 1,\ S(\boldsymbol{\alpha}^{(r)}, \boldsymbol{\beta}) \geq \theta\ \forall r = 1, \cdots, t$$

When $p = 1$, $\nu(\boldsymbol{\beta}) \leq 1$ reduces to a linear constraint. When $p > 1$, (11) is a quadratically constrained linear program (QCLP) with one quadratic constraint $\nu(\boldsymbol{\beta}) \leq 1$ and $t + n$ linear constraints. This can be solved by off-the-shelf optimisation tools such as Mosek [1]. Note that at time $t + 1$, $\nu(\boldsymbol{\beta})$ is defined as in (5) with $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(t)}$, i.e., the current estimate of $\boldsymbol{\beta}$.

Normalised maximal constraint violation is used as a convergence criterion. The algorithm stops when $|1 - \frac{S^{(t)}}{\theta^{(t)}}| \leq \epsilon$, where $S^{(t)} := S(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)})$ and ϵ is a predefined accuracy parameter. This iterative algorithm for solving the ℓ_p binary MK-FDA SIP problem is summarised in Table 1, and it is guaranteed to converge [4].

3 Feature Space Denoising with Kernel PCA

Principal Component Analysis (PCA) is an orthogonal basis transformation that transforms a number of correlated variables into uncorrelated ones called principal components. When used as an unsupervised dimensionality reduction technique, PCA retains as much variance as possible while reducing the dimensionality of the data.

Given a set of m vectors $\mathbf{x}_i \in \mathbb{R}^d, i = 1, \cdots, m$, and assuming the vectors are centred: $\sum_{i=1}^{m}\mathbf{x}_i = \mathbf{0}$, the orthogonal basis of PCA can be found by diagonalising the covariance matrix of the data. More precisely, let $X = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m)$ be the $d \times m$ data matrix. Consider the eigen decomposition $C = V\Omega V^T$, where $C = XX^T$ is the sample covariance matrix. The $d \times d$ diagonal matrix Ω contains the d eigenvalues of C, and the orthogonal basis sought is given by the eigenvectors of C, which are contained in the columns of the $d \times d$ matrix V. The data now can be decorrelated by projecting onto the orthogonal basis. If we sort the eigenvalues in descending order, and project only onto the eigenvectors that are associated with the leading eigenvalues, dimensionality reduction is achieved with a minimum loss of variance.

Now consider the kernel case. If we knew the mapping function $\phi(\cdot)$, we could then map the data into the feature space explicitly, compute the sample covariance (after centring): $\tilde{C} = \phi(X)\phi^T(X)$, and diagonalise $\tilde{C}$ to obtain explicitly the orthogonal basis in the feature space:

$$\tilde{C} = \tilde{V}\tilde{\Omega}\tilde{V}^T \tag{12}$$

where the $\tilde{d} \times \tilde{d}$ diagonal matrix $\tilde{\Omega}$ contains the eigenvalues of $\tilde{C}$ and the $\tilde{d} \times \tilde{d}$ matrix $\tilde{V} = (\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \cdots, \tilde{\mathbf{v}}_{\tilde{d}})$ contains the orthogonal basis in its columns, and $\tilde{d}$ is the dimensionality of the feature space. However, the mapping function is specified only implicitly through a kernel function, and the kernel matrix is all we have to work with.

Kernel PCA applies orthogonal basis transformation implicitly in the feature space. We first centre implicitly the data in the feature space by $K = PK'P$, where P is the $m \times m$ centring matrix defined as $P = I - \frac{1}{m}\mathbf{1} \cdot \mathbf{1}^T$, and K' is the uncentred kernel matrix [12]. Now we consider the eigen decomposition of the centred kernel matrix:

$$K = U\Delta U^T \tag{13}$$

where the $m \times m$ diagonal matrix Δ contains the eigenvalues of K and the $m \times m$ matrix $U = (\mathbf{u}_1, \mathbf{u}_1, \cdots, \mathbf{u}_m)$ contains in its columns the eigenvectors of K. Using the connection between $\tilde{C}$ and K ($\tilde{C} = \phi(X)\phi^T(X)$ and $K = \phi^T(X)\phi(X)$), it is shown in [12] that the non-zero eigenvalues of $\tilde{C}$ are the same as those of K, and for the i^{th} non-zeros eigenvalue, the corresponding eigenvectors $\tilde{\mathbf{v}}_i$ and $\mathbf{u}_i$ are related by:

$$\tilde{\mathbf{v}}_i = \phi(X)\mathbf{u}_i \tag{14}$$

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

Note that (14) only shows that each of the orthogonal basis vectors in the feature space lies in the span of the training samples in the feature space, and the orthogonal basis is not explicitly available. However, we are interested not in the orthogonal basis itself, but instead in the projection onto the basis. It directly follows from (14) that the projection onto the i^{th} basis vector $\tilde{\mathbf{v}}_i$ is given by $\phi^T(X)\tilde{\mathbf{v}}_i = \phi^T(X)\phi(X)\mathbf{u}_i = K\mathbf{u}_i$.

We have shown, given the kernel matrix, how kernel PCA can be used to find the projection onto any basis vector in the feature space. Similarly as in PCA, if we sort the eigenvalues of K (also the eigenvalues of $\widetilde{C}$), and project only onto the basis vectors that are associated with the leading eigenvalues, dimensionality reduction in the feature space is achieved with a minimum loss of variance. This can be considered as a denoising process in the feature space, where the level of denoising is controlled by the proportion of retained variance. In the next section we show the effect of feature space denoising especially in the context of multiple kernel learning.

4 Experiments

In this section we present experimental results showing the effect of feature space denoising, and discuss its connection to multiple kernel learning.

4.1 Setup

We carry out experiments on a challenging object recognition dataset, namely, Pascal visual object classes (VOC) 2007 dataset. Pascal VOC challenge provides a yearly benchmark for comparison of object recognition methods [3]. The VOC 2007 dataset consists of 9963 images of 20 object classes such as aeroplane, cat, person, etc. The set is divided into pre-defined training, validation, and test sets, with 2501, 2510, and 4952 images, respectively.

The classification of the 20 object classes is treated as 20 independent binary problems. Average precision (AP) is used to measure the performance of each binary classifier. The mean of the APs over the 20 classes, MAP, is used as a measure of the overall performance. Features provided by various detectors and descriptors are used to compute 33 kernels, and the computed kernels serve as base kernels in our experiments. A detailed description of the features and the kernel construction process can be found online [2].

4.2 Results with original kernels

In the first set of experiments we apply the proposed ℓ_p MK-FDA to the 33 original kernels. We learn the regularisation norm p on the validation set from 12 values: $\{1, 1+2^{-6}, 1+2^{-5}, 1+2^{-4}, 1+2^{-3}, 1+2^{-2}, 1+2^{-1}, 2, 3, 4, 8, 10^6\}$, and then apply the learnt optimal p to test set. We compare this ℓ_p MK-FDA scheme with fixed-norm MK-FDA, where the regularisation norm is fixed to ℓ_1, ℓ_2, and ℓ_∞.

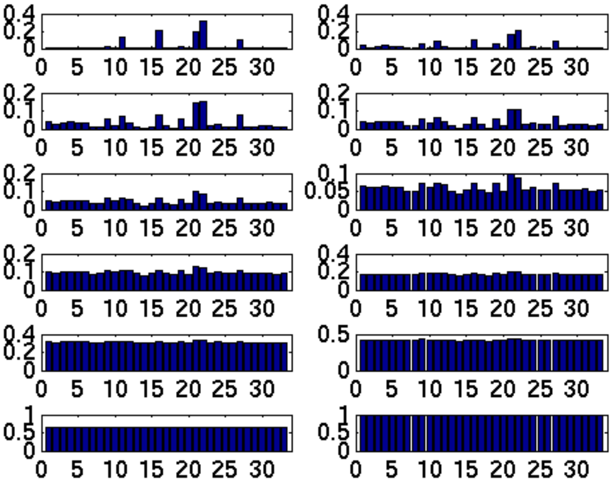

Figure 1. Kernel weights learnt on the training set with various p values. "dog" class. From left to right, top to bottom: $p = \{1, 1+2^{-6}, 1+2^{-5}, 1+2^{-4}, 1+2^{-3}, 1+2^{-2}, 1+2^{-1}, 2, 3, 4, 8, 10^6\}$.

Table 2. MK-FDAs with original kernels

	ℓ_1 MK-FDA	ℓ_2 MK-FDA	ℓ_∞ MK-FDA	ℓ_p MK-FDA
MAP	54.85	54.79	54.64	**55.61**

Shown in Fig. 1 are the kernel weights learnt on the training set with various regularisation norms for the "dog" class. This figure confirms that the norm p controls the sparsity of the learnt weights: the smaller the value, the more sparse the weights. When $p = 10^6$ (practically infinity), the kernels weights become ones, i.e., ℓ_∞ MK-FDA is equivalent to single kernel FDA with uniformly weighted sum of the base kernels.

In Table 2, we show MAPs of the four MK-FDA methods. By tuning the regularisation norm p using the validation set, the intrinsic sparsity of the kernel set can be learnt. As a result, ℓ_p MK-FDA outperforms its fixed norm counterparts.

4.3 Results with denoised kernels

In this section we show the effect of feature space denoising using kernel PCA. We again use the "dog" class as an example. Fig. 2 plots the APs obtained with single kernel FDA on the validation set and test set when keeping various

[2] http://www.featurespace.org/

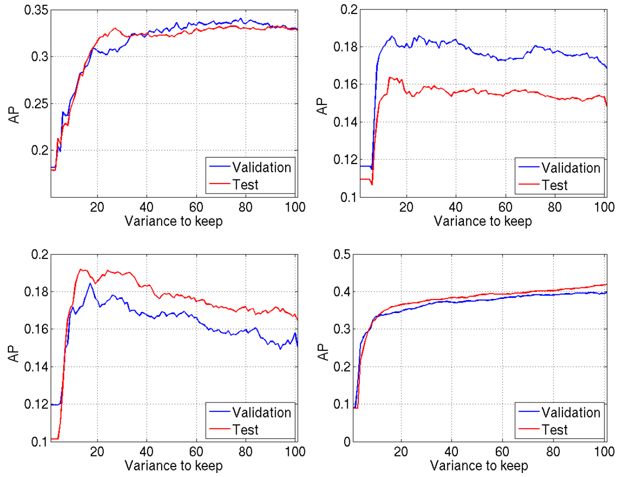

Figure 2. Feature space denoising with kernel PCA. “dog” class. Top left: kernel 21. Top right: kernel 26. Bottom left: kernel 30. Bottom right: sum of all 33 base kernels.

Table 3. MK-FDAs with denoised kernels

	ℓ_1 MK-FDA	ℓ_2 MK-FDA	ℓ_∞ MK-FDA	ℓ_p MK-FDA
MAP	54.26	56.06	55.82	**56.17**

amount of variance in kernel PCA. The first three subplots are with three base kernels, and the bottom left one is with the sum of all 33 base kernels.

Fig. 2 top and bottom left show that the performance curve on the validation set is a good indicator of the amount of noise in the feature space. In other words, one can determine how much variance to retain, or equivalently how much noise to remove, based on this curve. The optimal amount of variance to keep is kernel dependent ($\sim 20\%$ for kernel 26 and kernel 30, $\sim 80\%$ for kernel 21), but feature space denoising consistently boosts the performance on the test set compared to using the whole feature space without denoising.

Another interesting observation is that when applying feature space denoising to the sum of all 33 base kernels, we do not obtain any improvement (Fig. 2, bottom right). In this case, The best performance on both validation and test sets are achieved when all dimensions of the feature space are used. The MAP of this “summing + denoising” strategy is 54.37: compared to the “summing only” strategy (ℓ_∞ MK-FDA in Table 2), the performance even drops slightly.

Considering these observations, a more reasonable strategy would be to first denoise each base kernel, and then apply ℓ_p MK-FDA. The results of such a strategy are shown in Table 3. Note that the ℓ_∞ MK-FDA in the table is simply a “denoising + summing” scheme. Its advantage over the “summing + denoising” scheme is evident (55.82 vs. 54.37).

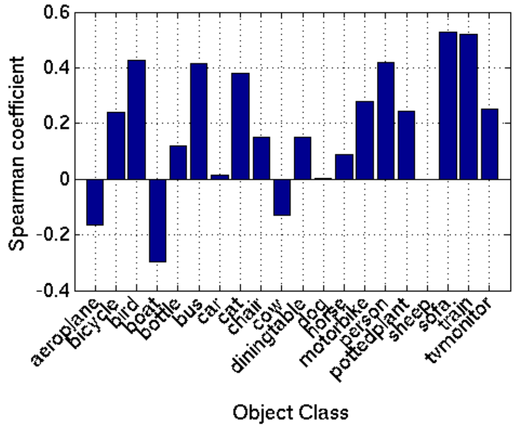

Figure 3. Spearman’s coefficient between the learnt kernel weights and percentage of variance to keep.

With the denoised kernels, the ℓ_p MK-FDA again outperforms the fixed-norm versions. However, the margin between it and its competitors is smaller this time. For example, it outperforms the ℓ_2 version only by 0.11. This seems to suggest that much of the benefit of MKL comes from its tendency to assign small weights to noisy kernels and vice versa. In order to test this hypothesis, we rank the 33 base kernels according to the kernel weights learnt with the optimal p value. We then rank the base kernels again according to the amount of variance kept by the feature space denoising process. If our hypothesis is correct, the two rankings should show some consistency.

We use Spearman’s rank correlation coefficient [15] to measure the similarity between the two rankings. A coefficient of +1 indicates identical rankings, while a coefficient of -1 means the two rankings are reversed of each other. The Spearman’s coefficients for the 20 object classes are shown in Fig. 3. Out of 20, positive coefficients are observed on 16 object classes, and negative coefficients are observed only on 3 classes. On the “sheep” class the optimal kernel weights learnt are uniform, for which case the Spearman’s coefficient is not defined. The mean of the 19 Spearman’s coefficients is 0.1917, which indicates there is indeed some correlation between the kernel weights learnt in MKL and the noise level in the base kernels.

In the experiments, the stopping threshold ϵ in ℓ_p MK-FDA is set to 10^{-4}, and λ is also set to 10^{-4}. The kernels used in the paper and a Matlab implementation of ℓ_p MK-

FDA are available online [3].

5 Conclusions

In this paper, we have reviewed an MKL technique, namely, ℓ_p regularised MK-FDA, and have investigated the effect of feature space denoising by means of kernel PCA. Experiments show that with both the original base kernels or denoised base kernels, by learning their intrinsic sparsity using a validation set, the ℓ_p MK-FDA we recently proposed outperforms its fixed-norm counterparts. Experiments also show that feature space denoising boosts the performance of both single kernel FDA and the ℓ_p MK-FDA. This observation, together with the one that there is in general a positive correlation between the learnt kernel weights in ℓ_p MK-FDA and the amount of variance kept by feature space denoising, seems to suggest that MKL should be performed on a per dimension basis instead of per kernel basis. However, this is not possible with MKL techniques that learn linear combinations of base kernels. An MKL objective function which can take care of the feature space denoising automatically, and which can learn a truly optimal (non-linear) combination of the base kernels, is yet to be found.

Acknowledgement

This work has been supported by EPSRC ACASVA Grant EP/F069421/1.

References

[1] F. Bach and G. Lanckriet. Multiple kernel learning, conic duality, and the smo algorithm. In *ICML*, 2004.

[2] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12:2385–2404, 2000.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[4] R. Hettich and K. Kortanek. Semi-infinite programming: Theory, methods, and applications. *SIAM Review*, 35(3):380–429, 1993.

[5] S. Ji, L. Sun, R. Jin, and J. Ye. Multilabel multiple kernel learning. In *NIPS*, 2008.

[6] S. Kim, A. Magnani, and S. Boyd. Optimal kernel selection in kernel fisher discriminant analysis. In *ICML*, 2006.

[7] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. Efficient and accurate lp-norm mkl. In *NIPS*, 2009.

[8] G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *JMLR*, 5:27–72, 2004.

[9] S. Mika. Kernel fisher discriminants. PhD Thesis, University of Technology, Berlin, Germany, 2002.

[10] A. Rakotomamonjy, F. Bach, Y. Grandvalet, and S. Canu. Simplemkl. *JMLR*, 9:2491–2521, 2008.

[11] B. Scholkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[12] B. Scholkopf, A. Smola, and K. Muller. Kernel principal component analysis. *Advances in Kernel Methods: Support Vector Learning*, pages 327–352, 1999.

[13] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[14] S. Sonnenburg, G. Ratsch, C. Schafer, and B. Scholkopf. Large scale multiple kernel learning. *JMLR*, 7:1531–1565, 2006.

[15] C. Spearman. The proof and measurement of association between two things. *Ameriman Journal of Psychology*, 15:72–101, 1904.

[16] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1999.

[17] F. Yan, J. Kittler, K. Mikolajczyk, and A. Tahir. Non-sparse multiple kernel learning for fisher discriminant analysis. In *International Conference on Data Mining*, 2009.

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

[19] J. Ye, S. Ji, and J. Chen. Multi-class discriminant kernel learning via convex programming. *JMLR*, 9:719–758, 2008.

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

[3] http://www.featurespace.org