Constructing Nonlinear Discriminants from Multiple Data Views

Tom Diethe[1], David Roi Hardoon[2], and John Shawe-Taylor[1]

[1] Department of Computer Science, University College London
[2] Data Mining Department, Institute for Infocomm Research, A*Star Singapore
{t.diethe,j.shawe-taylor}@cs.ucl.ac.uk, drhardoon@i2r.a-star.edu.sg

Abstract. There are many situations in which we have more than one view of a single data source, or in which we have multiple sources of data that are aligned. We would like to be able to build classifiers which incorporate these to enhance classification performance. Kernel Fisher Discriminant Analysis (KFDA) can be formulated as a convex optimisation problem, which we extend to the Multiview setting (MFDA) and introduce a sparse version (SMFDA). We show that our formulations are justified from both probabilistic and learning theory perspectives. We then extend the optimisation problem to account for directions unique to each view (PMFDA). We show experimental validation on a toy dataset, and then give experimental results on a brain imaging dataset and part of the PASCAL 2007 VOC challenge dataset.

Keywords: Fisher Discriminant Analysis, Convex Optimisation, Multi-view Learning, Kernel methods.

1 Introduction

We consider related but subtly differing settings within the domain of supervised learning. In Multi-View Learning (MVL), we have multiple views of the same underlying semantic object, which may be derived from different sensors, or different sensing techniques. In Multi-Source Learning (MSL), we have multiple sources of data which come from different sources but whose label space is aligned. Finally, in Multiple Kernel Learning (MKL), we have multiple kernels built from different feature mappings of the same data source. In general, any algorithm built to solve any of the three problems will also solve the others, but this may not be in the most optimal or desirable manner. For example, MKL algorithms do not make any attempt to integrate the sources of information from each view, and work by simply placing weights over the kernels [1]. Anecdotally, it seems that in many practical situations in which the number of kernels is small, the performance of MKL algorithms can actually be worse than simply choosing the best kernel through a heuristic method such as cross-validation (CV)[1].

[1] Amongst others, this topic was discussed at the NIPS 2009 Workshop "Understanding Multiple Kernel Learning Methods".

J.L. Balcázar et al. (Eds.): ECML PKDD 2010, Part I, LNAI 6321, pp. 328–343, 2010.
© Springer-Verlag Berlin Heidelberg 2010

In the MVL or MSL paradigm, we are assuming that the number of views or sources is typically small (*i.e.* $2 \rightarrow 10$), and hence another viewpoint is needed in which the sources are combined more usefully. The basic idea of MVL is to introduce one function per view which only uses the features from that view, and then jointly optimize these functions such that learning is enhanced. In MVL, we are also usually interested in having weight vectors and loadings for each of the views, which we do not have when we concatenate features (or equivalently sum kernel matrices), or take convex combinations of kernels as in the MKL setting. Without loss of generality, we will assume that we are in the MVL setting for the rest of the paper.

Canonical Correlation Analysis (CCA) and Kernel Canonical Correlation Analysis (KCCA) [7] attempt to integrate two sources of information by maximising the correlations between projections of each view. They are unsupervised techniques, and as such are not ideally suited to a classification setting. A common way of performing classification on two-view data using KCCA is to use the projected data from one of the views as input to a standard classification algorithm, such as a Support Vector Machine (SVM). However, as with Principal Components Analysis (PCA), the subspace that is learnt through such unsupervised methods may not always align well with the label space.

SVM-2K [5] was an attempt to take this to its logical conclusion by combining this two stage learning into a single optimisation. The algorithm introduces the constraint of similarity between two 1-dimensional projections which identify two distinct SVMs in the two feature spaces. However SVM-2K requires extra parameters (the C-parameter for each SVM, and another mixing parameter, along with any kernel parameters) that the methods presented here will not require. In addition, it is not easy to see how the SVM-2K formulation can be generalised to more than two views. There has been one related approach that tries to find the optimum combination of Fisher classifiers [8] using the MKL architecture [1]. In its initial form this problem is non-convex, although the authors do recast the problem in terms of a semi-definite programme (SDP), at the expensive of an increase in the problem scale. In addition, the MKL architecture means that the output of the algorithm is a single weight vector for the convex combination of kernels. The formulation presented here has some similarities to that of [8], except cast here in the MVL framework and also providing additional modelling flexibility.

2 Preliminaries

We first review the convex formulation of Kernel Fisher Discriminant Analysis (KFDA) in the form given by [13]. Let $(\mathbf{x}, y) \sim S$ be an input-output pair from an m-sample S with $\mathbf{x} \in \mathbb{R}^n$ and $y \in \{-1, +1\}$. Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)'$ be the input vectors stored in matrix $\mathbf{X}$ as row vectors, and $\mathbf{y} = (y_1, \ldots, \mathbf{y}_m)'$ be a vector of outputs, where $'$ denote the transpose of vectors or matrices. For simplicity we always assume that the examples are already projected into the kernel defined feature space $\mathcal{F}$, so that the kernel matrix $\mathbf{K}$ has entries $\mathbf{K}[i, j] = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. The

explicit feature mapping is defined as $\phi : \mathbf{x} \to \phi(\mathbf{x}) \in \mathcal{F}$. Furthermore we define $\mathbf{1} \in \mathbb{R}^m$ as the vector of all ones and $\mathbf{I} \in \mathbb{R}^{m \times m}$ the $m-$dimensional identity matrix.

To proceed, we can use the fact that KFDA minimises the variance of the data along the projection whilst maximising the separation of the classes. If we characterise the variance within a vector of slack variables $\boldsymbol{\xi} \in \mathbb{R}^n$, we can directly minimise the variance as follows,

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\xi}} \quad & \|\boldsymbol{\xi}\|^2 + \mu \boldsymbol{\alpha}' \mathbf{K} \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{K}\boldsymbol{\alpha} + \mathbf{1}b = \mathbf{y} + \boldsymbol{\xi} \\ & \boldsymbol{\xi}' \mathbf{e}^c = 0 \ \text{ for } \ c = -1, +1, \qquad \text{where } \mathbf{e}_i^c = \begin{cases} 1 & \text{if } y_i = c \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{1}$$

3 Convex Multiview Fisher Discriminant Analysis

Here the convex formulation for KFDA given above will be extended to multiple views. Given p "views" of the same data source, or alternatively p aligned data sources, to form an $m-$sample S with input output $p+1$ tuples $(\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \ldots, \mathbf{x}_{(p)}, y)$. It is assumed that each view has already been projected into a feature space $\mathcal{F}_d$, so that the kernel matrix $\mathbf{K}_d$ for that view has entries $\mathbf{K}_d[i,j] = \left\langle \mathbf{x}_{(d)i}, \mathbf{x}_{(d)j} \right\rangle$. The explicit feature mapping for a each view is defined as $\phi_d : \mathbf{x}_{(d)} \to \phi_d(\mathbf{x}_{(d)}) \in \mathcal{F}_d$. Given matrices of inputs $\mathbf{X}_d = [\mathbf{x}_{(d)1}, \ldots, \mathbf{x}_{(d)m}]'$, the formulation (1) is extended to find p dual weight vectors $\boldsymbol{\alpha}_d,\ d = 1, \ldots, p$. The concatenation of these weight vectors will be denoted by $\tilde{\boldsymbol{\alpha}} = [\boldsymbol{\alpha}_1', \ldots, \boldsymbol{\alpha}_p']'$. The convex form of Multiview Fisher Discriminant Analysis (MFDA) is given in equation (2) below. The goal is now to minimise the variance of the data along the projection whilst maximising the distance between the average outputs for each class over all of the views.

$$\begin{aligned} \min_{\boldsymbol{\alpha}_d, b, \boldsymbol{\xi}} \quad & \mathcal{L}(\boldsymbol{\xi}) + \mu \mathcal{P}(\tilde{\boldsymbol{\alpha}}), \\ \text{s.t.} \quad & \sum_{d=1}^{p} (\mathbf{K}_d \boldsymbol{\alpha}_d + \mathbf{1} b_d) = \mathbf{y} + \boldsymbol{\xi}, \qquad d = 1, \ldots, p \\ & \boldsymbol{\xi}' \mathbf{e}^c = 0 \ \text{ for } \ c = 1, 2, \end{aligned} \tag{2}$$

where $\mathcal{L}(\cdot)$ and $\mathcal{P}(\cdot)$ are the loss function and regulisation function respectively, as follows,

$$\mathcal{L}(\boldsymbol{\xi}) = \|\boldsymbol{\xi}\|_2^2, \tag{3}$$

$$\mathcal{P}(\tilde{\boldsymbol{\alpha}}) = \sum_{d=1}^{p} (\boldsymbol{\alpha}_d' \mathbf{K}_d \boldsymbol{\alpha}_d). \tag{4}$$

The first constraint in (2) ensures that the average loss between the output and its class label is minimised. The second constraint ensures that the average

output for each class is each label. The classification function on a set of examples $\mathbf{x}_{(d),i}$ from views $d = 1, \ldots, p$ now becomes,

$$f(\mathbf{x}_{(d),i}) = \operatorname{sgn}\left(\sum_{d=1}^{p} f(\mathbf{x}_{(d)i})\right) = \operatorname{sgn}\left(\sum_{d=1}^{p} \mathbf{K}_d[:, i]'\boldsymbol{\alpha}_d + b_d\right). \tag{5}$$

Clearly (2) collapses to (1) for $p = 1$. Observe that the solutions given will, in the linearly separable case, be equivalent to summing kernels. Meaning that viewed in the primal form, the result is the standard criterion in the space defined by the concatenation of the features, and the norm of the full weight vector is given by (4). However this formulation leads to two main advantages. Firstly, it provides a flexible framework that allows for different noise models and regularisation functions. Secondly, explicit weight vectors are available for each view, which allows the calculation of implicit weightings over the views (see Section 3.2 below). In the non-linearly separable case, the equivalence breaks down, as the optimisation ties the views together through the shared slack variables.

Further intuition on the operation of the algorithm is as follows. Given two views $\mathbf{x}_{(1)}$ and $\mathbf{x}_{(2)}$, and using the standard ℓ_2 loss function, MFDA is trying to minimise the summed errors committed: $\left\|f_1(\mathbf{x}_{(1)}) + f(\mathbf{x}_{(2)}) - \mathbf{y}\right\|_2^2$. So if some slack is added to one of the examples, *e.g.* $\mathbf{x}_{(1)i}$, then the algorithm will try to push the corresponding example $\mathbf{x}_{(2)i}$ the other way to try to minimise the overall slack. This can be seen as "view disagreement" which means that the algorithm tries to use information from both views to aid the classification. However of course the algorithm can "give up" and allow the slack to be big for that example, meaning that $\mathbf{x}_{(1)}$ and $\mathbf{x}_{(2)}$ can be pushed the same way.

It is actually possible to state the problem as the reverse - saying that normally in MVL the goal is to search for view agreement, which would (for example) be minimising $\left\|f(\mathbf{x}_{(1)}) - f(\mathbf{x}_{(2)})\right\|_2^2$ (ignoring the labels). This is one particular form of the so-called "Co-Training" problem, which in order to work requires that each of the views are *sufficient* for classification, and methods that use this break down when there is significant view disagreement. A recent paper tried to get around this by learning separate classifiers and then looking for view agreement/disagreement between them, before combining them into a final classifier (a form of bootstrapping)[3]. MFDA should have an advantage over this as it is directly optimising the combined classifier. However, we also provide an alternative 'Private' method with separate slacks for each view as well as the overall slacks (see Section 3.4 to follow). Essentially, if there is a "trouble" point in view $\mathbf{x}_{(1)}$, but not in view $\mathbf{x}_{(2)}$, the disagreement can be soaked up by the private slack, allowing the two views to move into agreement with zero shared slack.

3.1 Probabilistic Interpretation

Following the analysis of [13], it is possible to view the KFDA algorithm from a probabilistic point of view. It is known that Fisher Discriminant Analysis (FDA)

is Bayes optimal for two Gaussian distributions with equal covariance in the input space. The data may not fall naturally into this model, but it may be the case that for certain feature spaces (*e.g.* the space defined by the Radial Basis Function (RBF) kernel), the examples projected into a manifold in this space may be well approximated by Gaussian distributions with diagonal covariance[2]. In this case KFDA would be Bayes optimal in the feature space.

Consider data generated according to a Gaussian noise model, $y_i = \text{sgn}(\mathbf{x}_i\mathbf{w} + n_i)$ where n is assumed to be an *independently and identically distributed* (i.i.d.) random variable (noise) with mean 0 and variance σ^2. If one considers KFDA as regression on to the labels, then a Gaussian noise model with known variance σ would result in the following expression for the likelihood: $\Pr(\mathbf{y}|\boldsymbol{\alpha}) = \exp(-\|\boldsymbol{\xi}\|_2^2)$. If a prior over the weights with hyperparameters μ is used, the log of the posterior is simply $\log(\Pr(\mathbf{y}|\boldsymbol{\alpha})\Pr(\boldsymbol{\alpha}|\mu)) = -\|\boldsymbol{\xi}\|_2^2 - \log(\Pr(\boldsymbol{\alpha}|\mu))$. The choice of prior then becomes equivalent to the choice of regularisation function, which will be discussed in Section 3.3. When viewed in this way the outputs produced by KFDA can be interpreted as probabilities, which in turn makes it possible to assign confidence to the final classifications.

This view of KFDA also motivates the Multiview extension of the algorithm. We can extend and combine the graphical interpretations of [2] and [6] using the above definitions as seen in Figure 1. Note that explicit mixing weights $\boldsymbol{\beta}$ paramaterised by ρ are shown (dotted). Note that due to the optimisation (which constrains the functions over each feature space with the shared slack variable) and the fact that we have separate $\boldsymbol{\alpha}$ vectors for each view, we are able to drop the mixing weights $\boldsymbol{\beta}$ from our formulation. Under the assumption that the kernels are normalised, we can calculate these weights *post-hoc* as will be shown in Section 3.2. Taking the approach of Naïve Bayes Probabilistic Label Fusion (NBF) [9], the first step is to assume conditional independence between classifiers given a class label. Suppose the set of labels $\mathbf{s} = \{s_1, \ldots, s_p\}$ are given from p classifiers for a given point $\mathbf{x}_i$. Denoting $\Pr(s_d)$ as the probability that classifier D_d labels an example $\mathbf{x}_i$ in class $\omega_c \in \Omega$, (in this case $\Omega = \{-1, +1\}$), then the likelihood of the classifiers given a label is,

$$\Pr(\mathbf{s}|\omega_c) = \Pr(s_1, \ldots, s_p|\omega_c) = \prod_{d=1}^{p} \Pr(s_d|\omega_c). \tag{6}$$

The posterior probability needed to label $\mathbf{x}_i$ is then given by,

$$\Pr(\omega_c|\mathbf{s}) = \frac{\Pr(\omega_c)\Pr(\mathbf{s}|\omega_c)}{\Pr(\mathbf{s})} = \frac{1}{Z}\Pr(\omega_c)\prod_{d=1}^{p} \Pr(s_d|\omega_c), \tag{7}$$

[2] After (empirical) whitening has been performed on the data. It may also be necessary to restrict to the main eigenvalues as the eigenvectors corresponding to smaller eigenvalues will start to be very random. In the space spanned by the top eigenvectors the data will then have diagonal covariance.

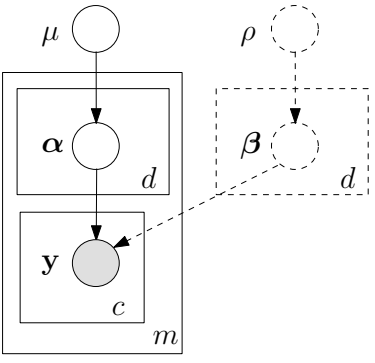


Fig. 1. Plates diagram showing the hiearchical Bayesian interpretation of MFDA. $\boldsymbol{\beta}$ are the hypothetical mixing parameters with prior weights ρ if an explicit mixing was used - in the case of MFDA these are fixed and hence can be removed, but can be calculated post-hoc.

where Z is a normalisation constant. Assume a uniform prior over labels, the log posterior is then given by,

$$\log(\Pr(\omega_c|\mathbf{s})) \propto \sum_{d=1}^{p} \log(\Pr(s_d|\omega_c)). \tag{8}$$

This implies that by directly optimising this sum, we are optimising the NBF over KFDA classifiers, which is precisely the motivation for both the objective function and the classification function for MFDA, both of which will be described in the next Section. At first glance it seems that this conditional independence assumption could be problematic, as this assumption is seldom true. However, Kuncheva made the point that despite this NBF is experimentally observed to be surprisingly accurate and efficient [9]. However, it does open the door to further possibilities for combining KFDA classifiers, but this is outside the scope of the present work.

3.2 Implicit Weighting

In order to determine the importance of each of the views after training, it is possible to calculate the implicit weighting of each view simply through the weighted sum of the absolute values of the classification functions. This is justified by the intuition made in Section 3.1 that the outputs of each classifier can be interpreted as probabilities, with the assumption that each kernel is normalised as per [16], *i.e.* $\text{trace}(\mathbf{K}_d) = m,\ d = 1, \ldots, p$. This in turn means that the overall confidence of the classifier can be calculated as the sum of the log probabilities that the function $f(\mathbf{x}_{(d)i})$ for classifier d on example i give the class label ω_c.

$$\beta_d \approx \frac{1}{Z} \sum_{c \in \Omega} \log(\Pr(s_d|\omega_c)) = \frac{\sum_{i=1}^{m} |\mathbf{K}_d[:,i]'\boldsymbol{\alpha}_d + b_d|}{\sum_{i=1}^{m}\sum_{d=1}^{p} |\mathbf{K}_d[:,i]'\boldsymbol{\alpha}_d + b_d|}. \tag{9}$$

3.3 Regularisation and Loss Functions

The natural choices for the regularisation function $\mathcal{P}(\tilde{\boldsymbol{\alpha}})$ would either be the sum of the ℓ_2-norms of the primal weight vectors (as in (4)), or the sum of the ℓ_2-norms of the dual weight vector $\mathcal{P}(\tilde{\boldsymbol{\alpha}}) = \sum_{d=1}^{p} \|\boldsymbol{\alpha}_d\|_2^2$. Potentially more interesting is the ℓ_1-norm of the dual weight vector, $\mathcal{P}(\tilde{\boldsymbol{\alpha}}) = \sum_{d=1}^{p} \|\boldsymbol{\alpha}_d\|_1$, as this choice leads to sparse solutions due to the fact that the ℓ_1-norm can be seen as an approximation to the (pseudo) ℓ_0-norm. In the rest of the chapter the ℓ_1-norm regularisation method is denoted as Sparse Multiview Fisher Discriminant Analysis (SMFDA).

In some situations these regularisation functions $\mathcal{P}(\cdot)$ may be too simplistic, in which case additional domain knowledge can be incorporated into the function. For example, there is reason to believe *a-priori* that most of the views are likely not to be useful, but the individual weights in that view are, then $\mathcal{P}(\tilde{\boldsymbol{\alpha}}) = \|\mathbf{A}\|_{2,1}$ could be used where $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p]$ is $\tilde{\boldsymbol{\alpha}}$ reshaped as a matrix of weights and the block (r, p)-norm of $\mathbf{A}$ is defined as $\|\mathbf{A}\|_{r,p} = (\sum_{i=1}^{m} \|\boldsymbol{\alpha}_i\|_p^r)^{1/p}$. Another example would be a situation it may be desirable to impose sparsity on some views but not others. For two views, this would simply be $\mathcal{P}(\tilde{\boldsymbol{\alpha}}) = \|\boldsymbol{\alpha}_1\|_2^2 + \|\boldsymbol{\alpha}_2\|_1$ in order to promote sparsity in the second view but not the first. One could also promote sparsity in the primal version of one view by passing in the explicit features for that view (if available) and penalising $\mathbf{X}_d'\boldsymbol{\alpha}_d$. In this way any mixture of linear with nonlinear features and primal with dual sparsity can be combined across the views, all in a single optimisation framework. One can also pre-specify the weights of views by parameterising them, if one has a strong prior belief that a view will be more or less useful, but it in general it is not necessary or helpful to do this.

Following [13] the assumption of a Gaussian noise model can also be removed, resulting in different loss functions on the slacks $\boldsymbol{\xi}$. For example, if a Laplacian noise model is chosen $\|\boldsymbol{\xi}\|_2^2$ can be replaced with $\|\boldsymbol{\xi}\|_1$ in the objective function. The advantage of this is if the ℓ_1-norm regulariser from above is chosen, the resulting optimisation is a linear programme, which can be solved efficiently using methods such as column generation. From a modelling perspective, it may be advantageous to choose a noise model that is robust to outliers, such as Huber's Robust loss, which can easily be used in the framework presented here.

3.4 Incorporating Private Directions

The above formulations seek to find the projection that is maximally discriminative averaged across views. However these problems are very tightly constrained, and optimisation may be difficult in situations where one or more of the views is not informative of the labels (*i.e.* is essentially noise). This leads to considering the allowance of some extra slack $\boldsymbol{\zeta}_d$ that is private to each view, which is similar to the approach taken by [11] to probabilistic latent space modelling. This leads to the following formulation which we term Private Multiview Fisher Discriminant Analysis (PMFDA),

$$\begin{aligned} \min_{\boldsymbol{\alpha}_d, b, \boldsymbol{\xi}, \boldsymbol{\zeta}_d} \quad & \mathcal{L}(\boldsymbol{\xi}, \tilde{\boldsymbol{\zeta}}, \tau) + \mu \mathcal{P}(\boldsymbol{\alpha}_d), && d = 1, \ldots, p \\ \text{s.t.} \quad & \mathbf{K}_d \boldsymbol{\alpha}_d + \mathbf{1} b = \mathbf{y} + \boldsymbol{\xi} + \boldsymbol{\zeta}_d && d = 1, \ldots, p \\ & \mathbf{1}_i' \boldsymbol{\xi} = 0 && i = 1, 2, \end{aligned} \tag{10}$$

with $\tilde{\boldsymbol{\zeta}} = [\boldsymbol{\zeta}_1', \ldots, \boldsymbol{\zeta}_p']'$. The regularisation function $\mathcal{P}(\cdot)$ is as before (4), and the loss function is updated to incorporate $\boldsymbol{\zeta}_d$ as follows,

$$\mathcal{L}(\boldsymbol{\xi}, \tilde{\boldsymbol{\zeta}}, \tau) = \|\boldsymbol{\xi}\|_2^2 + \tau \sum_{d=1}^{p} \|\boldsymbol{\zeta}_d\|_2^2. \tag{11}$$

Note the extra parameter τ which enables the tuning of the relative importance of private or shared slacks. If $\tau = 1$ the penalties of the private slack for an example i are proportional to $\boldsymbol{\xi}_i/p$, which means that the more views that are added, the less each view is allowed to dominate. In the experiments conducted here this was simply set heuristically to 0.1 to allow a small amount of leeway for each view.

3.5 Generalisation Error Bound for MFDA

We now construct a generalisation error bound for MFDA by applying the following results from [15] and [10] and extending to the Multiview case. The first bounds the difference between the empirical and true means (Theorem 3 in [15]).

Theorem 1 (Bound on the true and empirical means). *Let S_d be a view of a sample of m points drawn independently according to a probability distribution P_d, where R_d is the radius of the ball in the feature space $\mathcal{F}_d$ containing the support of the distribution. Consider the mean vector $\boldsymbol{\mu}_d$ and the empirical estimate $\hat{\boldsymbol{\mu}}_d$ defined as*

$$\begin{aligned} \boldsymbol{\mu}_d &= \mathbb{E}_{P_d}\left[\phi(\mathbf{x}_d)\right], \\ \hat{\boldsymbol{\mu}}_d &= \hat{\mathbb{E}}_{\mathbf{x}_d}\left[\phi(\mathbf{x}_d)\right] = \frac{1}{p}\sum_{d=1}^{p} \phi(\mathbf{x}_d). \end{aligned} \tag{12}$$

Then with probability at least $1 - \delta$ over the choice of S_d, we have

$$\|\hat{\boldsymbol{\mu}}_d - \mathbb{E}_{\mathbf{x}_d}[\phi(\mathbf{x}_d)]\| \leq \frac{R_d}{\sqrt{m}} \left(2 + \sqrt{2 \ln \frac{1}{\delta}}\right). \tag{13}$$

Consider the covariance matrix $\boldsymbol{\Sigma}_d$ and the empirical estimate $\hat{\boldsymbol{\Sigma}}_d$ defined as

$$\begin{aligned} \boldsymbol{\Sigma}_d &= \mathbb{E}\left[(\phi(\mathbf{x}_d) - \boldsymbol{\mu}_d)(\phi(\mathbf{x}_d) - \boldsymbol{\mu}_d)'\right], \\ \hat{\boldsymbol{\Sigma}}_d &= \hat{\mathbb{E}}\left[(\phi(\mathbf{x}_d) - \hat{\boldsymbol{\mu}}_d)(\phi(\mathbf{x}_d) - \hat{\boldsymbol{\mu}}_d)'\right]. \end{aligned} \tag{14}$$

The following corollary bounds the difference between the empirical and true covariance (Corollary 6 in [15]).

Corollary 1 (Bound on the true and empirical covariances). *Let S_d be an m sample from P_d as above, where R_d is as defined above. Provided $m \geq (2+\sqrt{2\ln 2/\delta})^2$, we have*

$$\left\|\hat{\mathbf{\Sigma}}_d - \mathbf{\Sigma}_d\right\|_F \leq \frac{2R_d^2}{\sqrt{m}}\left(2+\sqrt{2\ln\frac{2}{\delta}}\right), \tag{15}$$

The following Lemma is connected with the classification algorithm "Robust Minimax Classification" developed by [10], adapted here for MFDA.

Lemma 1. *Let $\boldsymbol{\mu}_d$ be the mean of a distribution and $\mathbf{\Sigma}_d$ its covariance matrix, $\mathbf{w}_d \neq 0$, b given, such that $\mathbf{w}_d'\boldsymbol{\mu}_d + b \leq 0$ and $\Delta \in [0,1)$, then if*

$$-\left(\mathbf{w}_d'\boldsymbol{\mu}_d + b\right) \geq \kappa(\Delta)\sqrt{\mathbf{w}_d'\mathbf{\Sigma}_d\mathbf{w}_d},$$

where $\kappa(\Delta) = \sqrt{\frac{\Delta}{1-\Delta}}$, then

$$\Pr\left(\mathbf{w}_d'\phi(\mathbf{x}_d) + b \leq 0\right) \geq \Delta$$

In order to provide a true error bound we must bound the difference between this estimate and the value that would have been obtained had the true mean and covariance been used.

Theorem 2 (Main). *Let S_d be a view of a sample of m points drawn from P_d as above, where R_d is the radius of the ball in the feature space $\mathcal{F}_d$ containing the support of the distribution. Let $\hat{\boldsymbol{\mu}}_d$ ($\boldsymbol{\mu}_d$) be the empirical (true) mean of a sample of m points from the view S_d, $\hat{\mathbf{\Sigma}}_d$ ($\mathbf{\Sigma}_d$) its empirical (true) covariance matrix, $\mathbf{w}_d \neq \mathbf{0}$, $\|\mathbf{w}\|_2 = 1$, and b given, such that $\mathbf{w}_d'\boldsymbol{\mu}_d + b \leq 0$ and $\Delta \in [0,1)$. Then with probability $1-\delta$ over the draw of the random sample, if*

$$-\left(\mathbf{w}_d'\hat{\boldsymbol{\mu}}_d + b\right) \geq \kappa(\Delta)\sqrt{\mathbf{w}_d'\hat{\mathbf{\Sigma}}_d\mathbf{w}_d} \qquad d = 1,\ldots,p,$$

then

$$\Pr\left(\left(\mathbf{w}_d'\phi_d(\mathbf{x}_d) + b\right) > 0\right) < 1 - \Delta,$$

where

$$\Delta = \frac{\left(\mathbf{w}_d'\hat{\boldsymbol{\mu}}_d + b - A_d\right)^2}{\mathbf{w}_d'\hat{\mathbf{\Sigma}}_d\mathbf{w}_d + B_d + \left(\mathbf{w}_d'\hat{\boldsymbol{\mu}}_d + b - A_d\right)^2},$$

such that $\|\hat{\boldsymbol{\mu}}_d - \boldsymbol{\mu}_d\| \leq A_d$ where $A_d = \frac{R_d}{\sqrt{m}}\left(2+\sqrt{2\ln\frac{2m}{\delta}}\right)$,
and $\left\|\hat{\mathbf{\Sigma}}_d - \mathbf{\Sigma}_d\right\|_F \leq B_d$ where $B_d = \frac{2R_d^2}{\sqrt{m}}\left(2+\sqrt{2\ln\frac{4m}{\delta}}\right)$.

Proof. (sketch). First we re-arrange $\mathbf{w}_d'\boldsymbol{\mu}_d + b \geq \kappa(\Delta)\sqrt{\mathbf{w}_d'\boldsymbol{\Sigma}_d\mathbf{w}_d}$ from Lemma 1 for each view in terms of $\kappa(\Delta)$:

$$\kappa(\Delta) = \frac{\mathbf{w}_d'\boldsymbol{\mu}_d + b}{\sqrt{\mathbf{w}_d'\boldsymbol{\Sigma}\mathbf{w}_d}}. \tag{16}$$

These quantities are in terms of the true means and covariances. In order to achieve an upper bound we need the following sample compressed results for the true and empirical means (Theorem 1) and covariances (Corollary 1):

$$\|\hat{\boldsymbol{\mu}}_d - \mathbb{E}_{\mathbf{x}_d}[\hat{\mu}_d(\mathbf{x}_d)]\| \leq A_d = \frac{R_d}{\sqrt{m}}\left(2 + \sqrt{2\ln\frac{2m}{\delta}}\right),$$

$$\left\|\hat{\boldsymbol{\Sigma}}_d - \boldsymbol{\Sigma}_d\right\|_F \leq B_d = \frac{2R_d^2}{\sqrt{m}}\left(2 + \sqrt{2\ln\frac{4m}{\delta}}\right).$$

Given equation (16) we can use the empirical quantities for the means and covariances in place of the true quantities. However, in order to derive a genuine upper bound we also need to take into account the upper bounds between the empirical and true means. Including these in the expression above for $\kappa(\Delta)$ by replacing δ with $\delta/2$, to derive a lower bound, we get:

$$\kappa(\Delta) = \frac{\mathbf{w}_d'\hat{\boldsymbol{\mu}}_{dS_d} + b - A_d}{\sqrt{\mathbf{w}_d'\hat{\boldsymbol{\Sigma}}_d\mathbf{w}_d + B_d}}.$$

Finally, making the substitution $\kappa(\Delta) = \sqrt{\frac{\Delta}{1-\Delta}}$ and solving for Δ yields the result. □

The following Proposition upper bounds the generalisation error of Multiview Fisher Discriminant Analysis (MFDA).

Proposition 1. *Let $\mathbf{w}_d$, b, be the (normalised) weight vector and associated threshold returned by the Multiview Fisher Discriminant Analysis (MFDA) when presented with a view of the training set S_d. Furthermore, let $\hat{\boldsymbol{\Sigma}}_d^+$ ($\hat{\boldsymbol{\Sigma}}_d^-$) be the empirical covariance matrices associated with the positive (negative) examples of the m training samples from S_d projected using $\mathbf{w}_d$. Then with probability at least $1-\delta$ over the draw of all the views of the random training set S_d, $d = 1, \ldots, p$ of m training examples, the generalisation error $\mathcal{R}$ is bounded by*

$$\mathcal{R} \leq \max(1 - \Delta^+, 1 - \Delta^-)$$

where Δ^j, $j = +, -$ such that

$$\Delta^j = \frac{j\left(\left(\sum_{d=1}^p(\mathbf{w}_d'\hat{\boldsymbol{\mu}}^j_{S_d} + b) - C^j\right)^2\right)}{\left(\sum_{d=1}^p \mathbf{w}_d'\hat{\boldsymbol{\Sigma}}^j_d\mathbf{w}_d\right) + D^j + \left(j(\sum_{d=1}^p \mathbf{w}_d'\hat{\boldsymbol{\mu}}^j_{S_d} + b) - C^j\right)^2},$$

where $C^j = \frac{\sum_{d=1}^p R_d}{\sqrt{m^j}}\left(2 + \sqrt{2\ln\frac{4mp}{\delta}}\right)$, $\quad D^j = \frac{2\sum_{d=1}^p R_d^2}{\sqrt{m^j}}\left(2 + \sqrt{2\ln\frac{8mp}{\delta}}\right)$.

Proof. For the negative part of the proof we require $\mathbf{w}_d'\hat{\boldsymbol{\mu}}_d^- + b \geq \kappa(\Delta)\sqrt{\mathbf{w}_d'\hat{\boldsymbol{\Sigma}}_d^-\mathbf{w}_d}$ which is a straight forward application of Theorem 2 with δ replaced with $\delta/2$. For the positive part, observe that we require $\mathbf{w}_d'\hat{\boldsymbol{\mu}}_d^+ - b \geq \kappa(\Delta)\sqrt{\mathbf{w}_d'\hat{\boldsymbol{\Sigma}}_d^+\mathbf{w}_d}$, hence, a further application of Theorem 2 with δ replaced by $\delta/2$ suffices. Finally, we take a union bound over the p views such that m is replaced by mp. □

3.6 Experiments: Toy Data

In order to show that MVL methods can be beneficial, and demonstrate the validity of the outlined methods, experiments were first conducted with simulated toy data. A data source S was created by taking two $1-$dimensional Gaussian distributions (S^+, S^-) which were well separated, which was then split into 100 train and 50 test points. The source S was embedded into $2-$dimensional views through complementary linear projections (ϕ_1, ϕ_2) to give new "views" $\mathbf{X}_1, \mathbf{X}_2$. Differing levels of independent "measurement noise" were added to each view (n_1, n_2), and identical "system noise" was added to both views (n_S). A third view was constructed of pure noise to simulate a faulty sensor $(\mathbf{X}_3)$. The labels $\mathbf{y}$ were calculated as the sign of the original data source.

$$
\begin{array}{ll}
S = \{S^+, S^-\} & \text{(source)} \\
S^+ \sim \mathcal{N}(5,1), S^- \sim \mathcal{N}(-5,1) & \\
\mathbf{y} = \operatorname{sgn}(S) & \text{(labels)} \\
\phi_1 = [1,-1], \phi_2 = [-1,1] & \text{(projections)} \\
n_1 \sim \mathcal{N}(0,5)^2, n_2 \sim \mathcal{N}(0,3)^2 & \text{(meas. noise)} \\
n_S \sim \mathcal{N}(0,2)^2 & \text{(system noise)} \\
\mathbf{X}_1 = \phi_1' S + n_1 + n_S & \text{(view 1)} \\
\mathbf{X}_2 = \phi_2' S + n_2 + n_S & \text{(view 2)} \\
\mathbf{X}_3 = n_S & \text{(view 3)}
\end{array}
$$

$\mathbf{X}_1$ and $\mathbf{X}_2$ are noisy views of the same signal, with correlated noise, which can be a typical problem in multivariate signal processing (*e.g.* sensors in close proximity). Linear kernels were used for each view. A small value for the regularisation parameter $\mu = 10^{-3}$ was chosen heuristically for all the experiments. Table 1 gives an overview of the results on the toy dataset. Comparisons were made against: KFDA on each of the views (denoted as $f(1)$, $f(2)$ and $f(3)$ respectively); summing the classification functions of these ($fsum$); summing the kernels of each view ($ksum$); followed by MFDA, PMFDA and SMFDA. Note that an unweighted sum of kernels is equivalent to concatenating the features before creating a single kernel. The table shows the test error over 10 random repeats of the experiment in first column, followed by the implicit weightings for each of the algorithms calculated via (9). Note that the $ksum$ method returns single $m-$dimensional weight vector, and unless a kernel with an explicit feature space is used it is not possible to recalculate the implicit weightings over the features. In this case, since linear kernels are used the weightings have been calculated. For the three methods outlined in this paper (MFDA, PMFDA, SMFDA),

Table 1. Test errors over ten runs on the toy dataset. Methods described in the text. $W(\cdot)$ refers to the implicit weightings given by each algorithm for each of the views. Note that the weightings closely match the actual SNR.

Method	**Test error**	$W(1)$	$W(2)$	$W(3)$
$f(a)$	0.19	1.00	0.00	0.00
$f(b)$	0.10	0.00	1.00	0.00
$f(c)$	0.49	0.00	0.00	1.00
$fsum$	0.39	0.33	0.33	0.33
$ksum$	0.04	0.29	0.66	0.05
MFDA	0.04	0.29	0.66	0.05
PMFDA	0.04	0.29	0.66	0.05
SMFDA	0.04	0.29	0.66	0.05
Actual		0.35	0.65	0.00

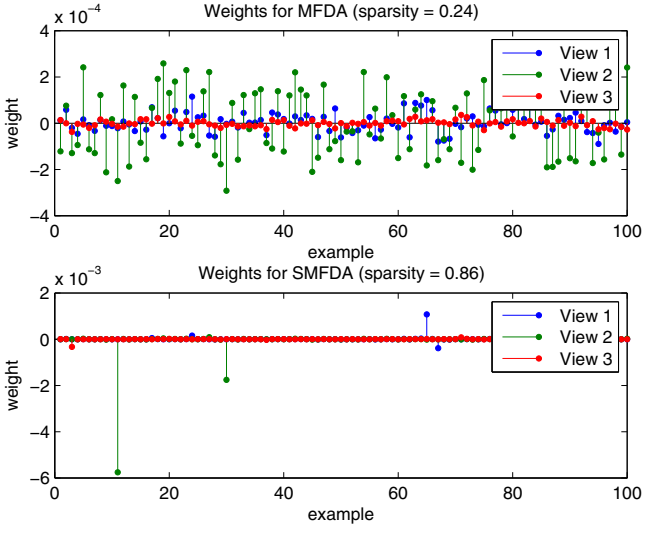


Fig. 2. Weights given by MFDA and SMFDA on the toy dataset. Notice that many of the weights for SMFDA are close to zero, indicating sparse solutions. Also notice that most of the weights for view 3 (pure noise) are close to zero.

as expected the performance is roughly equivalent to the $ksum$ method. The last row in the table (actual) is the empirical Signal to Noise Ratio (SNR) calculated as $SNR_d = \sum(\mathbf{X}'_d\mathbf{X}_d)/\text{var}(S - \mathbf{X}_d)$ for view d, which as can be seen is closely matched by the weightings given.

The sparsity of SMFDA can be seen in figure 2. The sparsity level quoted in the figure is the proportion of the weights below 10^{-5}.

4 Experiments

4.1 VOC 2007 DATASET

The sets of features ("views") used can be found in [17], with an extra feature extraction method known as Scale Invariant Feature Transformation (SIFT) [12]. RBF kernels were constructed for each of these feature sets, the RBF width parameter was set using a heuristic method[3]. The Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL) Visual Object Classes (VOC) 2007 challenge database was used which contains 9963 images, each with at least 1 object. The number of objects in each image ranges from 1 to 20, with, for instance, objects of people, sheep, horses, cats, dogs etc. For a complete list of the objects, and description of the data set see the VOC 2007 challenge website[4].

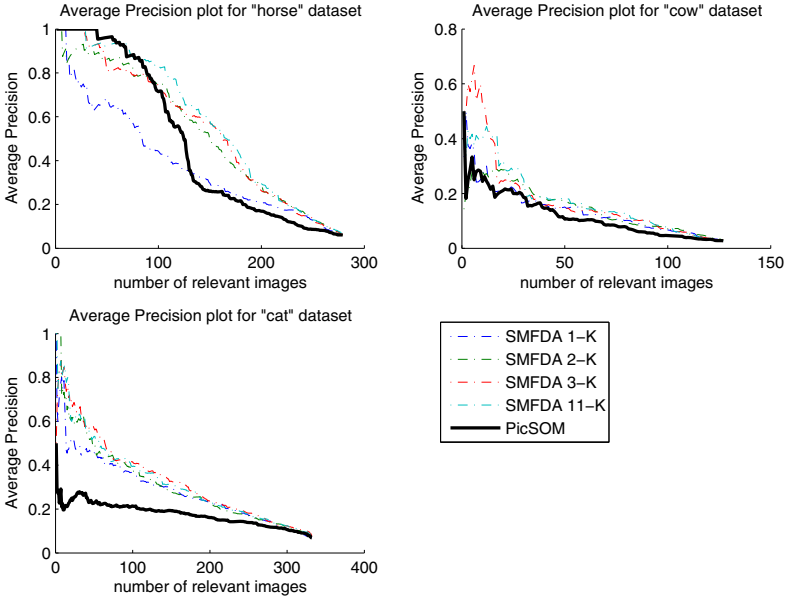


Fig. 3. Average precision recall curves for 3 VOC 2007 datasets for SMFDA plotted against PicSOM results

Figure 3 shows Recall-Precision curves for SMFDA with 1, 2, 3 or 11 kernels and PicSOM [17], and Table 2 shows the balanced error rate (the average of the errors on each class) and overall average precision for the PicSOM, KFDA

[3] For each setting of the width parameter, histograms of the kernel values were created. The chosen kernel was the one whose histogram peak was closest to 0.5 (*i.e.* furthest from 0 and 1).

[4] http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html

using cross-validation to choose the best single kernel, and SMFDA. For the purposes of training, a random subset of 200 irrelevant images was used rather than the full training set. Results for three of the object classes (cat, cow, dog) are presented. The results show that, in general, adding more kernels into the optimisation can assist in recall performance. For each object class, the subsets of kernels (*i.e.* 1,2, or 3) were chosen by the weights given by SMFDA on the 11 kernels. The best single kernel (based on SIFT features) performs well alone, yet the improvement in some cases is quite marked. Results are competitive with the PicSOM algorithm, which uses all 11 feature extraction methods, and all of the irrelevant images.

Table 2. Balanced Error Rate (BER) and Average Precision (AP) for four of the VOC challenge datasets, for four different methods: PicSOM, KFDA with cross validation (KFDA CV), and SMFDA

Dataset →	**Cat**		**Cow**		**Horse**	
Method ↓	BER	AP	BER	AP	BER	AP
PicSOM	n/a	0.18	n/a	0.12	n/a	0.48
KFDA CV	**0.26**	**0.36**	0.32	0.14	0.22	0.51
SMFDA	**0.26**	**0.36**	**0.27**	**0.15**	**0.19**	**0.58**

4.2 Neuroimaging Dataset

This section describes analysis of functional Magnetic Resonance Imaging (fMRI) data[5] that was acquired from 16 subjects who viewed image stimuli from two categories (pleasant (+ve) and unpleasant (-ve)). The images were presented in 6 blocks of 42 images (7 volumes) per category. The image stimuli are represented using SIFT features [12], and conventional pre-processing was applied to the fMRI data with linear kernels. A leave-subject-out paradigm was used where 15 subjects are combined for training and a single subject is withheld for testing. This gave a total of $42 \times 2 \times 15 = 1260$ training and $42 \times 2 = 84$ testing fMRI volumes and paired image stimuli. In the following experiment, the following comparisons were made: An SVM on the fMRI data (single view); KCCA on the fMRI + Image Stimuli (two views) followed with an SVM trained on the fMRI data projected into the learnt KCCA semantic space; MFDA on the fMRI + Image Stimuli (two views). The results are given in Table 3 where it can be observed that on average MFDA performs better than both the SVM (which is a single view approach), and the KCCA/SVM which similarly to MFDA incorporates two views into the learning process. In this case the label space is clearly not well aligned with the KCCA projections, whereas a supervised method such as MFDA is able to find this alignment.

[5] Data donated by Mourão-Miranda *et. al.* [14].

Table 3. In the following table the leave-one-out errors for each subject are presented. The following methods are compared: SVM on the fMRI data alone; KCCA analysis on the two views fMRI and Image Stimuli followed by an SVM on the projected fMRI data; the proposed MFDA on the two views fMRI+Image.

Sub.	SVM	KCCA/SVM	MFDA
1	0.1310	0.1667	**0.1071**
2	0.1905	0.2739	**0.1429**
3	0.2024	**0.1786**	0.1905
4	0.1667	0.2125	**0.1548**
5	**0.1905**	0.2977	0.2024
6	0.1667	0.1548	**0.1429**
7	**0.1786**	0.2262	0.1905
8	0.2381	0.2858	**0.2143**
9	0.3096	0.3334	**0.2619**
10	0.2977	0.3096	**0.2262**
11	**0.1191**	0.1786	0.1429
12	0.1786	0.2262	**0.1667**
13	0.2500	0.2381	**0.0714**
14	0.4405	0.4405	**0.2619**
15	**0.2500**	0.2977	0.2738
16	**0.1429**	0.1905	0.1860
Mean:	0.2158±0.08	0.2508±0.08	**0.1860±0.06**

5 Conclusions

KFDA can be formulated as a convex optimisation problem, which we extended to the Multiview setting MFDA using justifications from a probabilistic point of view. We also provide a generalisation error bound. A sparse version SMFDA was then introduced, and the optimisation problem further extended to account for directions unique to each view PMFDA. Experimental validation was shown on a toy dataset, followed by experimental results on part of the PASCAL 2007 VOC challenge dataset and a fMRI dataset, showing that the method is competitive with state-of-the-art methods whilst providing additional benefits.

Mika *et. al.* [13] demonstrate that their convex formulation of KFDA can easily be extended to both multi-class problems and regression problems, simply by updating the final two constraints. The same is also true of MFDA and its derivatives, which enhances its flexibility. The possibility of replacing the Naïve Bayes Fusion method for combining classifiers is another interesting avenue for research.

Finally, for the special case of SMFDA there is the possibility of using a stagewise optimisation procedure similar to the Least Angle Regression Solver (LARS) [4] which would have the benefit of computing the full regularisation path.

References

1. Bach, F.R., Lanckriet, G.R.G., Jordan, M.I.: Multiple kernel learning, conic duality, and the smo algorithm. In: ICML 2004: Proceedings of the twenty-first international conference on Machine learning, p. 6. ACM Press, New York (2004)
2. Centeno, T.P., Lawrence, N.D.: Optimising kernel parameters and regularisation coefficients for non-linear discriminant analysis. Journal of Machine Learning Research 7, 455–491 (2006)
3. Christoudias, C.M., Urtasun, R., Darrell, T.: Multi-view learning in the presence of view disagreement. In: Proceedings of Conference on Uncertainty in Artificial Intelligence, UAI (2008)
4. Efron, B., Hastie, T., Johnstone, L., Tibshirani, R.: Least angle regression. Annals of Statistics 32, 407–499 (2002)
5. Farquhar, J., Hardoon, D., Meng, H., Shawe-Taylor, J., Szedmak, S.: Two view learning: SVM-2k, theory and practice. In: Weiss, Y., Schölkopf, B., Platt, J. (eds.) Advances in Neural Information Processing Systems 18, pp. 355–362. MIT Press, Cambridge (2006)
6. Girolami, M., Rogers, S.: Hierarchic bayesian models for kernel learning. In: ICML, pp. 241–248 (2005)
7. Hardoon, D., Szedmak, S., Shawe-Taylor, J.: Canonical correlation analysis: An overview with application to learning methods. Neural Computation 16(12), 2639–2664 (2004)
8. Kim, S.J., Magnani, A., Boyd, S.: Optimal kernel selection in kernel fisher discriminant analysis. In: ICML 2006: Proceedings of the 23rd international conference on Machine learning. pp. 465–472. ACM, New York (2006)
9. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley Interscience, Hoboken (2004)
10. Lanckriet, G.R., Ghaoui, L.E., Bhattacharyya, C., Jordan, M.I.: A robust minimax approach to classification. J. Mach. Learn. Res. 3, 555–582 (2003)
11. Leen, G., Fyfe, C.: Learning shared and separate features of two related data sets using GPLVMs. Tech. rep., Presented at the NIPS 2008 workshop Learning from Multiple Sources (2008)
12. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the 7th IEEE International Conference on Computer vision, Kerkyra, Greece, pp. 1150–1157 (1999)
13. Mika, S., Rätsch, G., Müller, K.R.: A mathematical programming approach to the kernel Fisher algorithm. In: Leen, T., Dietterich, T., Tresp, V. (eds.) Advances in Neural Information Processing Systems, vol. 13, pp. 591–597 (2001)
14. Mourão-Miranda, J., Reynaud, E., McGlone, F., Calvert, G., Brammer, M.: The impact of temporal compression and space selection on svm analysis of single-subject and multi-subject fmri data. NeuroImage 33(4), 1055–1065 (2006)
15. Shawe-Taylor, J., Cristianini, N.: Estimating the moments of a random vector. In: Proceedings of GRETSI 2003 Conference, vol. 1, pp. 47–52 (2003)
16. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
17. Viitaniemi, V., Laaksonen, J.: Techniques for image classification, object detection and object segmentation applied to VOC challenge 2007. Tech. rep., Department of Information and Computer Science, Helsinki University of Technology, TKK (2008)