Journal of Machine Learning Research 7 (2006) 455–491 Submitted 12/04; Revised 10/05; Published 2/06

Optimising Kernel Parameters and Regularisation Coefficients for Non-linear Discriminant Analysis

Tonatiuh Peña Centeno TPENA@DCS.SHEF.AC.UK
Neil D. Lawrence NEIL@DCS.SHEF.AC.UK
Department of Computer Science
The University of Sheffield
Regent Court, 211 Portobello Street
Sheffield, S1 4DP, U.K.

Editor: Greg Ridgeway

Abstract

In this paper we consider a novel Bayesian interpretation of Fisher's discriminant analysis. We relate Rayleigh's coefficient to a noise model that minimises a cost based on the most probable class centres and that abandons the 'regression to the labels' assumption used by other algorithms. Optimisation of the noise model yields a direction of discrimination equivalent to Fisher's discriminant, and with the incorporation of a prior we can apply Bayes' rule to infer the posterior distribution of the direction of discrimination. Nonetheless, we argue that an additional constraining distribution has to be included if sensible results are to be obtained. Going further, with the use of a Gaussian process prior we show the equivalence of our model to a regularised kernel Fisher's discriminant. A key advantage of our approach is the facility to determine kernel parameters and the regularisation coefficient through the optimisation of the marginal log-likelihood of the data. An added bonus of the new formulation is that it enables us to link the regularisation coefficient with the generalisation error.

1. Introduction

Data analysis typically requires a preprocessing stage to give a more parsimonious representation of data, such preprocessing consists of selecting a group of characteristic features according to an optimality criterion. Tasks such as data description or discrimination commonly rely on this preprocessing stage. For example, Principal Component Analysis (PCA) describes data more efficiently by projecting it onto the principal components and then by minimising the reconstruction error, see e.g. (Jolliffe, 1986). In contrast, Fisher's linear discriminant (Fisher, 1936) separates classes of data by selecting the features[1] that maximise the ratio of projected class means to projected intraclass variances.

The intuition behind Fisher's linear discriminant (FLD) consists of looking for a vector of compounds $\mathbf{w}$ such that, when a set of training samples are projected on to it, the class centres are far apart while the spread within each class is small, consequently producing a small overlap between classes (Schölkopf and Smola, 2002). This is done by maximising a cost function known in some contexts as Rayleigh's coefficient, $J(\mathbf{w})$. Kernel Fisher's discriminant (KFD) is a nonlinearisation

1. In Fisher's terminology the features are grouped into a vector of 'compounds'.

that follows the same principle but in a typically high-dimensional feature space $\mathcal{F}$. In this case, the algorithm is reformulated in terms of $J(\alpha)$, where α is the new direction of discrimination. The theory of reproducing kernels in Hilbert spaces (Aronszajn, 1950) gives the relation between vectors $\mathbf{w}$ and α, see Section 5.1. In either case, the objective is to determine the most 'plausible' direction according to the statistic J.

Mika et al. (1999) demonstrated that KFD can be applied to classification problems with competitive results. KFD shares many of the virtues of other kernel based algorithms: the appealing interpretation of a kernel as a mapping of an input to a high dimensional space and good performance in real life applications, among the most important. However, it also suffers from some of the deficiencies of kernelised algorithms: the solution will typically include a regularisation coefficient to limit model complexity and parameter estimation will rely on some form of cross validation. Unfortunately, there is no principled approach to set the former, while the latter precludes the use of richer models.

In this paper we introduce a novel probabilistic interpretation of Fisher's discriminant. Classical FLD is revised in Section 2 while an alternative noise model is outlined in Section 3. We build on the model in Section 4 by first applying priors over the direction of discrimination to develop a *Bayesian* Fisher discriminant and later we use a Gaussian process prior to reformulate the problem. In Section 5, we compare our model to other approaches. We explore the connections of our model to the expected generalisation error in Section 6. Section 7 details an EM-based algorithm for estimating the parameters of the model (kernel and regularisation coefficients) by optimising the marginal log likelihood. We present the results of our approach by applying it on toy data and by classifying benchmark data sets, in Section 8. Finally we address future directions of our work in Section 9.

2. Fisher's Discriminant Analysis

As mentioned above, discriminant analysis involves finding a vector of compounds $\mathbf{w} \in \mathbb{R}^{d\times 1}$ for which class separation will be maximised according to some defined statistic. Considering a set of training data and labels, $\mathcal{D} = (\mathbf{X}, \mathbf{y}) = \left\{\mathbf{x}^{(n)}, y^{(n)}\right\}_{n=1}^{N} \in \mathbb{R}^{N\times(d+1)}$, the discriminant reduces the dimensionality of the data through a linear combination, such that a set of single variates $\left\{\left(\mu_1, \sigma_1^2\right), \left(\mu_0, \sigma_0^2\right)\right\}$ is produced; where we define $\left(\mu_q, \sigma_q^2\right)$ as the sample mean and variance of each projected group. The hope is that both groups will be distinguished from one another by using this new set. Fisher was the first to conclude that the compounds should be given by maximising the ratio of between to within class variances,

$$J = \frac{\left(\mu_1 - \mu_0\right)^2}{\sigma_1^2 + \sigma_0^2}. \tag{1}$$

We will use the following definitions. A vector of projections is generated by taking the product $\mathbf{f} = \mathbf{X}\mathbf{w} \in \mathbb{R}^{N\times 1}$ and the sample means for each class are $\mathbf{m}_q = N_q^{-1}\sum_{n\in N_q}\mathbf{x}_q^{(n)}$, hence the projected mean and variance are given by

$$\begin{aligned} \mu_q &= N_q^{-1}\mathbf{w}^T\mathbf{m}_q \\ &= N_q^{-1}\mathbf{f}^T\mathbf{y}_q, \end{aligned} \tag{2}$$

and

$$\begin{aligned}\sigma_q^2 &= \sum_{n\in N_q}\left(\mathbf{w}^T\mathbf{x}_q^{(n)}-\mu_q\right)^2 \\ &= \sum_{n\in N_q}\left(f^{(n)}-\mu_q\right)^2, \end{aligned} \tag{3}$$

respectively. Abusing the notation, we have split the training data into two disjoint groups $(\mathbf{X},\mathbf{y}) = (\mathbf{X}_0,\mathbf{y}_0)\cup(\mathbf{X}_1,\mathbf{y}_1)$, with $y_q^{(n)}\in\{0,1\}$. The coefficient N_q is the cardinality of each group, $q\in\{0,1\}$.

Modern texts on pattern recognition and machine learning (Fukunaga, 1990; Duda and Hart, 1973; Bishop, 1995; Ripley, 1996) prefer to make explicit the dependence of this statistic on the vector of compounds. Hence, with some manipulation and the introduction of a couple of matrices we arrive at

$$J(\mathbf{w}) = \frac{\mathbf{w}^T\Sigma_B\mathbf{w}}{\mathbf{w}^T\Sigma_w\mathbf{w}}, \tag{4}$$

where $\Sigma_B = (\mathbf{m}_1-\mathbf{m}_0)(\mathbf{m}_1-\mathbf{m}_0)^T$ and $\Sigma_w = \sum_{q\in\{0,1\}}\sum_{n=1}^{N_q}\left(\mathbf{x}_q^{(n)}-\mathbf{m}_q\right)\left(\mathbf{x}_q^{(n)}-\mathbf{m}_q\right)^T$, are between and within covariance matrices respectively. Matrix Σ_B measures the separation between class means while Σ_w gives an estimation of the spread around them. A solution for this problem consists of taking the derivative of Equation 4 w.r.t. $\mathbf{w}$ and solving. This leads to a generalised eigenvalue problem of the form $\Sigma_w^{-1}\Sigma_B\mathbf{w} = \lambda\mathbf{w}$, with λ being the eigenvalues. A solution for the discriminant can also be derived from geometric arguments. Given a test point $\mathbf{x}^\star$, the discriminant is a hyperplane $D(\mathbf{x}^\star) = \mathbf{w}^T\mathbf{x}^\star + b$, that outputs a number according to the class membership of the test point, where b is a bias term. In this context $\mathbf{w}$ is a vector that represents the direction of discrimination. Following this line, the solution $\mathbf{w}\propto\Sigma_w^{-1}(\mathbf{m}_0-\mathbf{m}_1)$ is sometimes easier to interpret than the eigenvalue problem.

As it was demonstrated by Mika (2001), a more detailed analysis of FLD allows it to be cast as a quadratic programming problem. In order to do so, we observe that the magnitude of the solution is not relevant, so for example, the numerator of Equation 1 can be fixed to an arbitrary scalar while the denominator is minimised. In other words, the variance of the projections is minimised while the distance between projected means is kept at, say $d = \mu_0-\mu_1$. Rayleigh's statistic can then be written as $J = d^2/\left(\sigma_1^2+\sigma_0^2\right)$. The subsequent discussion will make use of this 'average distance' constraint to reformulate the discriminant problem.

3. Probabilistic Interpretation

We introduce some notation that will be used throughout the rest of the paper. The set of variables $\mathcal{D} = (\mathbf{X},\mathbf{y})\in\mathbb{R}^{N\times(d+1)}$ is observed or instantiated, $\mathbf{f}\in\mathbb{R}^{N\times1}$ is a dependent or latent variable and $\mathbf{t}\in\mathbb{R}^{N\times1}$ is a vector of targets that have been observed as well. The random variables will follow some probability law and in this model, in particular, we study the relationship between observed and latent variables: the noise model. From Section 2, we know that every observation in $\mathcal{D}$ is projected into a single variate that ideally can take only two values which are the projected class centres, where the variance around the projections tries to be minimised. We define the parameters c_0 and c_1 as the true class centres in the projected space. Additionally, we introduce a precision β that corresponds to the variance around the projected data. Because of the nature of the mapping

process, it is convenient to define some auxiliary variables as well, $\mathbf{t}_1$ is a vector filled with c_1's whenever $y^{(n)} = 1$ and filled with zeros otherwise; $\mathbf{t}_0$ is a vector filled with c_0's whenever $y^{(n)} = 0$ and with zeros otherwise. We also take $\mathbf{y}_1 = \mathbf{y}$ and $\mathbf{y}_0 = \mathbf{1} - \mathbf{y}$ and denote by $\hat{\mathbf{v}}$ the maximum likelihood estimate of a vector/scalar $\mathbf{v}$.

3.1 The Noise Model

Figure 1 models the causal relationship between the observations $\mathcal{D}$ and the variables $\mathbf{f}$ and $\mathbf{t}$, such that the distribution $p(\mathbf{f},\mathbf{t}|\mathcal{D},)$ can be decomposed into noise model $p(\mathbf{t}|\mathbf{y},\mathbf{f})$ and prior $p(\mathbf{f}|\mathbf{X})$, disregarding the parameter β. For the moment, we will ignore the prior and consider only the noise model. In graphical notation every fully shaded circle corresponds to an observed variable and a blank circle indicates a latent variable. We make use as well of partially shaded circles to indicate the binary nature of the discriminant, that is, that targets should only take one of two different values. In Figure 1 the variable $t_1^{(n)}$ is observed whenever $y^{(n)} = 1$; and $t_0^{(n)}$, whenever $y^{(n)} = 0$. Both variables $\mathbf{t}_0$ and $\mathbf{t}_1$ are discrete, with each of their elements being given by the class centres c_0 and c_1, nevertheless, we will make a Gaussian[2] approximation such that every element $t_q^{(n)} \sim \mathcal{N}\left(f^{(n)}, \beta^{-1}\right)$. From this approximation the noise model can be defined as

$$p(\mathbf{t}|\mathbf{y},\mathbf{f},\beta) = \frac{\beta^{\frac{N}{2}}}{(2\pi)^{\frac{N}{2}}} \exp\left\{-\frac{\beta}{2}\sum_{q\in\{0,1\}} (\mathbf{t}_q - \mathbf{f})^T \operatorname{diag}(\mathbf{y}_q)(\mathbf{t}_q - \mathbf{f})\right\}. \tag{5}$$

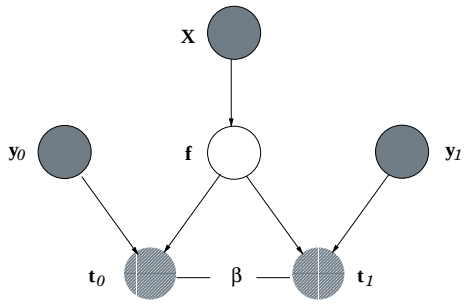

Figure 1: The proposed graphical model for discriminant analysis. The graph models the joint distribution over the latent variables $\mathbf{f}$ and the targets $\mathbf{t} = \mathbf{t}_0 \cup \mathbf{t}_1$, which have been decomposed into their two possible types. Disregarding the parameter β, the joint probability is factorised as $p(\mathbf{f},\mathbf{t}|\mathcal{D}) = p(\mathbf{t}|\mathbf{y},\mathbf{f})\,p(\mathbf{f}|\mathbf{X})$, where the noise model is given by $p(\mathbf{t}|\mathbf{y},\mathbf{f})$ and the prior by $p(\mathbf{f}|\mathbf{X})$. Note that we express the labels into two different groups $\mathbf{y}_0$ and $\mathbf{y}_1$. Shaded nodes indicate instantiated variables, blank ones correspond to latent variables and partially shaded ($\mathbf{t}_0$ and $\mathbf{t}_1$) nodes are only observed according to the values of the labels ($\mathbf{y}_0$ and $\mathbf{y}_1$, respectively). We assume that every observed target is distributed according to $t_q^{(n)} \sim \mathcal{N}\left(f^{(n)}, \beta^{-1}\right)$, where β is the precision parameter.

As it can be observed from both the figure and Equation 5, there is a conditional independence assumption on the observed targets given $\mathbf{y}$ and $\mathbf{f}$; in other words, the noise model can be further

2. We use the notation $\mathcal{N}(\mathbf{x}|\mathbf{m},\Sigma)$ to indicate a multivariate Gaussian distribution over $\mathbf{x}$ with mean $\mathbf{m}$ and covariance Σ.

decomposed as $p(\mathbf{t}|\mathbf{y},\mathbf{f}) = p(\mathbf{t}_0|\mathbf{y}_0,\mathbf{f})\,p(\mathbf{t}_1|\mathbf{y}_1,\mathbf{f})$, where we have disregarded the dependence on β.

We can substitute every element $t_q^{(n)}$ by its class centre c_q and take the log of (5) to obtain

$$\mathcal{L}(\mathbf{f},\beta) \quad = \quad -\frac{\beta}{2}\sum_{n=1}^{N}\left[y^{(n)}\left(c_1 - f^{(n)}\right)^2 + \left(1-y^{(n)}\right)\left(c_0 - f^{(n)}\right)^2\right] + \mathrm{C}, \tag{6}$$

where $\mathrm{C} = \dfrac{N}{2}\log\dfrac{\beta}{2\pi}$.
Note that the class centres can be made to coincide with the labels. In such a 'regression to the labels' scheme, FLD can be recovered in a straightforward manner.

3.1.1 MAXIMUM LIKELIHOOD

Parameter estimates can be found by zeroing the gradient of $\mathcal{L}$ with respect to each $f^{(n)}$ and β and solving the resulting expressions for each parameter. This leads to the fixed point equations

$$\hat{f}^{(n)} = \left(1-y^{(n)}\right)c_0 + y^{(n)}c_1 \tag{7}$$

and

$$\hat{\beta} = \frac{N}{\sum_{n=1}^{N} y_n\left(c_1 - f^{(n)}\right)^2 + \sum_{n=1}^{N}(1-y_n)\left(c_0 - f^{(n)}\right)^2}. \tag{8}$$

However, the values of the class centres c_0 and c_1 are not known, so $\mathcal{L}$ can also be maximised w.r.t. them to obtain

$$\hat{c}_q = \frac{1}{N_q}\sum_{n=1}^{N_q} y_q^{(n)} f^{(n)} \text{ for} q \in \{0,1\}. \tag{9}$$

The results $\hat{f}^{(n)}$ and $\hat{c}_q$ suggest applying an iterative scheme to find the maximum. This can be done by substituting $\hat{f}^{(n)}$ and $\hat{c}_q$ on the right hand sides of Equations 9 and 7, respectively, initialising one of the variables to an arbitrary value and updating all of them until convergence.

3.2 Model Equivalence

We now turn to the connections between Rayleigh's statistic and the proposed noise model. In particular, we want to show that maximum likelihood learning in our framework is equivalent to maximisation of Rayleigh's coefficient. In order to do so, we back substitute the values $\hat{c}_q$ into $\mathcal{L}$ (Equation 6) compute the gradient w.r.t β and solve the resulting expression for β. The substitution of each class centre by their most probable values is indispensable and central to our framework. As a result of this substitution we can create a cost function that reduces the error around the most probable class centres. The solution for β leads to an expression of the form

$$\hat{\beta} = \frac{N}{\sigma_1^2 + \sigma_0^2},$$

with σ_q^2 defined in Equation 3, for $q \in \{0,1\}$, and where we have recognised that Equation 2 is equivalent to Equation 9. The result above is proportional to the constrained version of Rayleigh's

quotient mentioned before, $J = d^2 / \left(\sigma_1^2 + \sigma_0^2\right)$, hence we can write

$$J(\mathbf{f}) = \frac{d^2 \hat{\beta}}{N}. \tag{10}$$

It is clear that this quantity monotonically increases over the domain $\mathbb{R}^+$ because $\hat{\beta}$ can only take positive values. Meanwhile the likelihood, the exponential of Equation 6, expressed in terms of the estimate $\hat{\beta}$ takes the form

$$L(\mathbf{f}) = \frac{\hat{\beta}^{N/2}}{(2\pi)^{N/2}} \exp\left\{-\frac{N}{2}\right\}, \tag{11}$$

which is monotonic as well on this estimate.

Therefore, as Equations 10 and 11 are monotonic in $\hat{\beta}$, their maximisation with respect to this parameter must yield equivalent results.

3.3 Parametric Noise Model

In this section we make two modifications to Equations 5 and 6 in order to parameterise the noise model. First, the vector of targets $\mathbf{t}$ is replaced by a new vector filled with the estimates $\hat{c}_q$ such that $\hat{\mathbf{t}} = \hat{\mathbf{t}}_0 \cup \hat{\mathbf{t}}_1$ is generated. Second, every latent variable is related to the observations via a vector of parameters $\mathbf{w}$. In a linear relation this is expressed by the inner product $f^{(n)} = \mathbf{w}^T \mathbf{x}^{(n)}$. Therefore after making these changes the log-likelihood becomes

$$\mathcal{L} \;=\; -\frac{\beta}{2} \sum_{n=1}^{N} \left[y^{(n)} \left(\hat{c}_1 - \mathbf{w}^T \mathbf{x}^{(n)} \right)^2 + \left(1 - y^{(n)} \right) \left(\hat{c}_0 - \mathbf{w}^T \mathbf{x}^{(n)} \right)^2 \right] + \mathrm{C}. \tag{12}$$

Thus a new probabilistic model is obtained, which is depicted in Figure 2.

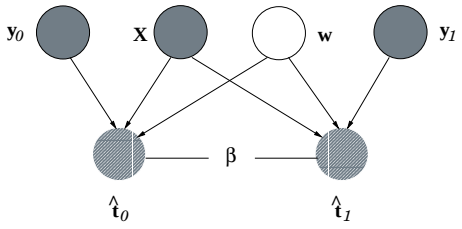

Figure 2: Partially modified graphical model for discriminant analysis. In comparison with Figure 1, the latent variable $\mathbf{f}$ has been replaced by a vector of parameters $\mathbf{w}$. Ignoring the parameter β, the graph factorises the joint distribution $p\left(\hat{\mathbf{t}}, \mathbf{w} \middle| \mathcal{D}\right)$ with the product $p\left(\hat{\mathbf{t}} \middle| \mathcal{D}, \mathbf{w}\right) \times p(\mathbf{w})$, where $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ is the training data; $\hat{\mathbf{t}} = \hat{\mathbf{t}}_1 \cup \hat{\mathbf{t}}_0$, the modified targets and $\mathbf{y}_0$ and $\mathbf{y}_1$ are the class labels. The log of the noise model $p\left(\hat{\mathbf{t}} \middle| \mathcal{D}, \mathbf{w}\right)$ is expressed in Equation 12 while the prior $p(\mathbf{w})$ is specified in Section 4.

Furthermore, we look not only to parameterise the latent variables, but the class centres as well. Equation 9 can be used to this purpose, substituting every $f^{(n)}$ in it with their parametric versions $\mathbf{w}^T \mathbf{x}^{(n)}$ leads to $\hat{c}_q = \frac{1}{N_q} \sum_{n=1}^{N_q} y_q^{(n)} \mathbf{w}^T \mathbf{x}^{(n)}$. The vector of parameters can be pulled out of

the summation and leave a quantity that we recognise to be the sample mean for class q, which we express as $\mathbf{m}_q$. Hence we can write $\hat{c}_q = \mathbf{w}^T \mathbf{m}_q$. Therefore the log of the new noise model can be expressed as

$$\mathcal{L} = -\frac{\beta}{2} \sum_{n=1}^{N} \left[y^{(n)} \left(\mathbf{w}^T \mathbf{m}_1 - \mathbf{w}^T \mathbf{x}^{(n)} \right)^2 + \left(1 - y^{(n)} \right) \left(\mathbf{w}^T \mathbf{m}_0 - \mathbf{w}^T \mathbf{x}^{(n)} \right)^2 \right] + \text{C}. \tag{13}$$

As it will be seen in Section 5, most models make the assumption that class centres and class labels coincide, that is $c_q = y_q$; including the least squares support vector machine of Suykens and Vandewalle (1999). However this approach is suboptimal because there is no guarantee that class centres should map perfectly with the labels. Instead of following this 'regression to the labels' assumption, we have preferred to make use of the maximum likelihood estimates of the class centres. As we saw above, by taking this step, the class centres can be parameterised as well.

3.3.1 MAXIMUM LIKELIHOOD

Maximisation of this new form of $\mathcal{L}$ (Equation 13) has to be carried out in a slightly different way to the one presented in Section 3.1.1. Previously, the class centres were parameters which we knew beforehand were separated by some given distance. However, their parameterisation implies that the separation constraint must be considered explicitly. We therefore introduce a Lagrange multiplier to force the projected class centres to lie at a distance d, leading to the following function

$$\begin{aligned} \Lambda(\mathbf{w}, \lambda) = & \\ -\frac{\beta}{2} \sum_{n=1}^{N} & \left[y^{(n)} \left(\mathbf{w}^T \mathbf{m}_1 - \mathbf{w}^T \mathbf{x}^{(n)} \right)^2 + \left(1 - y^{(n)} \right) \left(\mathbf{w}^T \mathbf{m}_0 - \mathbf{w}^T \mathbf{x}^{(n)} \right)^2 \right] \\ & + \lambda \left[\mathbf{w}^T (\mathbf{m}_0 - \mathbf{m}_1) - d \right] + \text{C}. \end{aligned}$$

A solution for this constrained optimisation problem is given by

$$\hat{\mathbf{w}} = \frac{\lambda}{\beta} \Sigma_w^{-1} (\mathbf{m}_0 - \mathbf{m}_1),$$

with

$$\lambda = d\beta \left[(\mathbf{m}_0 - \mathbf{m}_1)^T \Sigma_w^{-1} (\mathbf{m}_0 - \mathbf{m}_1) \right]^{-1}.$$

Therefore, by letting $\Delta\mathbf{m} = \mathbf{m}_0 - \mathbf{m}_1$, we can express the solution as

$$\hat{\mathbf{w}} = \frac{d\Sigma_w^{-1} \Delta\mathbf{m}}{\Delta\mathbf{m}^T \Sigma_w^{-1} \Delta\mathbf{m}}, \tag{14}$$

which is equivalent to that produced by FLD up to a constant of proportionality (see Section 2).

This completes the discussion of an alternative noise model for FLD. The new probabilistic formulation is based on a noise model that reduces the error around the class centres, instead of the class labels. Furthermore, we were interested on parameterising not only the latent variables in the model but also the centres themselves. Through the introduction of a Lagrange multiplier we saw that a constrained maximisation of the new likelihood was equivalent to standard FLD.
In this section we made use only of one part of the graphical models presented in Figures 1 and 2. In the next section we complete the analysis by including the prior distributions that were left

unattended. First we complete the study of Figure 2 by incorporating a prior over the parameters, $p(\mathbf{w})$, and later study the model of Figure 1 under the assumption that the prior, $p(\mathbf{f}|\mathbf{X})$, is a Gaussian process.

4. Bayesian Formulation

One of the aims of discriminant analysis is to determine the group membership of an input $\mathbf{x}^\star$ outside the training set. From a probabilistic perspective this process is only possible if a noise model and a prior distribution have been identified. Then the posterior over the parameters $p(\mathbf{w}|\mathcal{D})$ can be found as well as the corresponding predictive distribution. The posterior distribution is important because it summarises the knowledge gained after having observed the training set. The application of a Bayesian probabilistic approach offers some intrinsic advantages over other methods, for example the ability to compute 'error bars' and, in the context of our model, the possibility to introduce Gaussian process priors in a natural way.

This section will show that the introduction of a separable Gaussian prior over $\mathbf{w}$ leads to a posterior distribution that is not enough to recover FLD's solution. Later on, it will be argued that an additional step is required to ensure the equivalence is achieved. This additional step will also include the distance constraint previously implemented through a Lagrange multiplier.

4.1 Weight Space Formulation

So far we have found a maximum likelihood estimate of the parameters' vector (see Equation 14). Now what we seek is a distribution over this vector which is obtained by combining the noise model with a prior distribution through Bayes' rule,

$$p\left(\mathbf{w}|\hat{\mathbf{t}},\mathcal{D}\right)=\frac{p\left(\hat{\mathbf{t}}|\mathcal{D},\mathbf{w}\right)p(\mathbf{w})}{p\left(\hat{\mathbf{t}}|\mathcal{D}\right)},$$

where we have used $\mathcal{D}$ to indicate the training set $(\mathbf{X},\mathbf{y})$ and have omitted the dependence on β.

A common choice of prior is a separable Gaussian, $p(\mathbf{w})=\mathcal{N}\left(\mathbf{w}|\mathbf{0},\mathbf{A}^{-1}\right)$, whith zero mean and diagonal covariance $\mathbf{A}^{-1}$. The combination of this prior with the parametric noise model of Equation 13 gives a posterior of the form

$$\begin{aligned}p(\mathbf{w}|\hat{\mathbf{t}},\mathcal{D}) \quad\propto\quad & \exp\left\{-\frac{\beta}{2}\sum_{n=1}^{N}\left[y^{(n)}\left(\mathbf{w}^T\mathbf{m}_1-\mathbf{w}^T\mathbf{x}^{(n)}\right)^2+\ldots\right.\right.\\ & \left.\left.\left(1-y^{(n)}\right)\left(\mathbf{w}^T\mathbf{m}_0-\mathbf{w}^T\mathbf{x}^{(n)}\right)^2\right]-\frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w}\right\}. \qquad (15)\end{aligned}$$

In order to obtain a complete expression for $p(\mathbf{w}|\mathcal{D})$ it is necessary to define the normalisation constant. As the expression is quadratic in $\mathbf{w}$ we know the posterior distribution will be Gaussian. However, it is still necessary to specify the mean and covariance of the distribution. In order to do so, Bayesian methods take advantage of an important property of Gaussians: if two sets of variables are Gaussian, like $\hat{\mathbf{t}}$ and $\mathbf{w}$, then the conditional distribution of one set conditioned on the other is Gaussian as well. On the RHS of (15), we look to condition variable $\mathbf{w}$ on $\hat{\mathbf{t}}$. The process simply consists of considering the variable $\hat{\mathbf{t}}$ as being given and on grouping terms in $\mathbf{w}$. This leads to a Gaussian posterior of the form

$$p(\mathbf{w}|\mathcal{D})=\mathcal{N}\left(\mathbf{w}|\mathbf{0},\mathbf{B}^{-1}\right),$$

with **zero** mean and covariance matrix $\mathbf{B} = \beta \mathbf{X}^T \mathbf{L} \mathbf{X} + \mathbf{A}$, where

$$\mathbf{L} = \mathbf{I} - N_1^{-1} \mathbf{y}_1 \mathbf{y}_1^T - N_0^{-1} \mathbf{y}_0 \mathbf{y}_0^T. \tag{16}$$

The posterior obtained is not equivalent to FLD because the mean of $\mathbf{w}$ is zero. In consequence, the posterior mean projection of any $\mathbf{x}^\star$ will collapse to the origin. Nonetheless, this formulation yields a consistent result if we consider that standard discriminant analysis exhibits a sign symmetry for the vector $\mathbf{w}$, hence the average is zero. What our new model is missing is the incorporation of the distance constraint. In Section 3.3.1, knowledge about the variable d was incorporated to the noise model in the form of a Lagrange multiplier. We look to do the same again but in a Bayesian approach this requires that we deal with every variable in terms of probability distributions.

We propose to use the posterior $p(\mathbf{w}|\mathcal{D})$ as the prior for a new model that is depicted in Figure 3. In the new formulation, d is considered an extra random variable that has been observed and that depends on the distribution over $\mathbf{w}|\mathcal{D}$. From the figure we can deduce that the joint factorises as $p(d,\mathbf{w}|\mathcal{D},\gamma) = p(d|\mathcal{D},\mathbf{w},\gamma)\, p(\mathbf{w}|\mathcal{D})$, with γ being a positive parameter. Note that this time we have made $\mathcal{D} = \left(\hat{\mathbf{t}}, \mathbf{X}, \mathbf{y}\right)$.

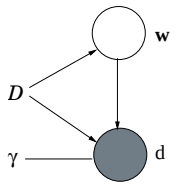

Figure 3: Graphical model to constrain the projected distance d. The graph specifies the distribution $p(d,\mathbf{w}|\mathcal{D},\gamma)$ which is composed by the distributions $p(\mathbf{w}|\mathcal{D})$ and $p(d|\mathcal{D},\mathbf{w},\gamma)$. The former is the posterior over the direction of discrimination, described in Section 4.1, and the latter is the constraining distribution, defined in Equation 17.

One of our main concerns is to keep the model tractable at all stages, but we are also interested in having a realistic representation of the discriminant. In order to guarantee both conditions we assume d is Gaussian with infinite precision γ,

$$p(d|\mathcal{D},\mathbf{w},\gamma) = \lim_{\gamma\to\infty} \frac{\gamma^{\frac{1}{2}}}{\sqrt{2\pi}} \exp\left(-\frac{\gamma}{2}\left(d - \mathbf{w}^T \Delta\mathbf{m}\right)^2\right). \tag{17}$$

We can see that this distribution introduces the same effect as the Lagrangian of Section 3.3.1 by placing all its mass at the point $d = \mu_0 - \mu_1$ when the limit $\gamma \to \infty$ is taken.

The process to determine a posterior $p(\mathbf{w}|\mathcal{D},d)$ is based on combining $p(\mathbf{w}|\mathcal{D})$ with $p(d|\mathcal{D},\mathbf{w},\gamma)$ and then conditioning $\mathbf{w}$ on d. However, a final step needs to be added to work out the limit to eliminate the dependence over γ. As a partial result, the conditional distribution $p(\mathbf{w}|\mathcal{D},d,\gamma)$ will be $\mathcal{N}(\mathbf{w}|\bar{\mathbf{w}},\Sigma)$ with mean

$$\bar{\mathbf{w}} = \lim_{\gamma\to\infty} \gamma d \Sigma \Delta\mathbf{m},$$

and covariance

$$\Sigma = \lim_{\gamma\to\infty} \left(\mathbf{B} + \gamma \Delta\mathbf{m}\Delta\mathbf{m}^T\right)^{-1}.$$

With some algebraic manipulations and the application of the Morrison-Woodbury formula (Golub and Van Loan, 1996) we can arrive to the desired result. See Appendix A for the detailed derivation. After taking the limit, the resulting distribution will be a Gaussian

$$p(\mathbf{w}|\,\mathcal{D},d) = \mathcal{N}(\mathbf{w}|\bar{\mathbf{w}},\Sigma)$$

with parameters

$$\bar{\mathbf{w}} \;=\; \frac{d\mathbf{B}^{-1}\Delta\mathbf{m}}{\Delta\mathbf{m}^T\mathbf{B}^{-1}\Delta\mathbf{m}}$$

and

$$\Sigma \;=\; \mathbf{B}^{-1} - \frac{\mathbf{B}^{-1}\Delta\mathbf{m}\Delta\mathbf{m}^T\mathbf{B}^{-1}}{\Delta\mathbf{m}^T\mathbf{B}^{-1}\Delta\mathbf{m}}.$$

Noticing that $\mathbf{B} = \beta\mathbf{X}^T\mathbf{L}\mathbf{X} + \mathbf{A}$, the mean of the new posterior coincides with the maximum likelihood solution of Section 3.3 when an improper prior is used (i.e. $\mathbf{A} = \lim_{\alpha\to\infty}\alpha\mathbf{I}$). Note that the matrix Σ is positive semidefinite and therefore not invertible, this is a consequence of the fact that any vector $\mathbf{w}$ which does not satisfy the constraint imposed by the distribution $p(\,d|\,\mathcal{D},\mathbf{w},\gamma)$ has a posterior probability of zero. Nevertheless, variances associated with the posterior projections can still be computed by applying

$$\mathrm{var}\left(\mathbf{w}^T\mathbf{x}\right) = \mathbf{x}^T\mathbf{B}^{-1}\mathbf{x} - \frac{\mathbf{x}^T\mathbf{B}^{-1}\Delta\mathbf{m}\Delta\mathbf{m}^T\mathbf{B}^{-1}\mathbf{x}}{\Delta\mathbf{m}^T\mathbf{B}^{-1}\Delta\mathbf{m}},$$

which will be zero if the point $\mathbf{x}$ is on the direction of $\Delta\mathbf{m}$.

The Bayesian approach we have outlined leads to a posterior distribution over the direction of discrimination which can be used to compute expected outputs and their associated variances for any given input $\mathbf{x}$. However, the limitation imposed by applying a linear model is a strong one. There is an extensive amount of literature explaining why linear models are not always convenient. A common solution is to use a set of nonlinear basis functions ϕ such that the new function is linear in the parameters but nonlinear in the input space $f = \mathbf{w}^T\phi(\mathbf{x})$, see for example (Ruppert et al., 2003) and (Bishop, 1995). However the problem is shifted to that of specifying which and what number of basis functions to use. In the next section we shall consider the alternative approach of placing a prior directly over the vector of projections $\mathbf{f}$, such that we will be working with a possibly infinite amount of basis functions. This approach will lead to a regularised version of kernel Fisher's discriminant and ultimately to an alternative strategy to select model parameters.

4.2 Gaussian Process Formulation

The choice of a Gaussian probability measure over functions has been justified by the study of the limiting prior distribution in the neural network case when the number of hidden units 'reaches' infinity, (Neal, 1996). A Gaussian process (GP) is a type of stochastic process that is defined by a mean and a covariance function. By stochastic process we understand that a countable infinite set of observations $\{f_1,\ldots,f_N\}$ has been sampled from a common probability distribution.

In GP's (O'Hagan, 1978) a prior is placed directly over the latent variables such that a posterior distribution over them can be inferred. Although there are many GP's with an equivalent 'weight space' prior, there exists a large class of them for which no finite dimensional expansion exists. In

this regard, a covariance function (or kernel) measures *a priori* the expected correlation between any two pair of points $\mathbf{x}^{(n)}$ and $\mathbf{x}^{(m)}$ in the training set. For example, in a function parameterised as

$$f^{(n)} = \mathbf{w}^T \phi\left(\mathbf{x}^{(n)}\right),$$

with a prior over $\mathbf{w}$ specified by a spherical Gaussian with zero mean, $p(\mathbf{w}) = \mathcal{N}\left(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}\right)$, the implied correlation between two points is

$$E\left[f^{(n)}, f^{(m)} \middle| \mathbf{w}\right] = \alpha^{-1}\phi\left(\mathbf{x}^{(n)}\right)^T \phi\left(\mathbf{x}^{(m)}\right).$$

In other words, provided that the product is positive and symmetric, the correlation between the two points will lead to a Mercer kernel; see (Schölkopf and Smola, 2002). However, under these circumstances it no longer makes sense to talk about a prior over the vector $\mathbf{w}$, but rather a prior over instantiations of the functions is considered.

4.2.1 PREDICTION OVER A TEST POINT

In order to adopt GP's we need to go back to the formulation of the discriminant presented in Figure 1. In this figure the graph models the joint distribution $p(\mathbf{f}, \mathbf{t}|\mathcal{D})$ with the product of noise model $p(\mathbf{t}|\mathbf{y}, \mathbf{f})$ and prior $p(\mathbf{f}|\mathbf{X})$. In this section we need to make two assumptions before doing any kind of prediction. First of all, the joint distribution over every instance f belonging to the training set or not will be a multivariate Gaussian, that is a GP. Secondly, we will continue to work with the maximum likelihood estimates of the class centres, which were denoted $\hat{c}_q$. In other words, if we use Equation 9 to form a vector $\hat{\mathbf{t}}$ and substitute it into Equation 5 we will obtain the distribution $p\left(\hat{\mathbf{t}}|\mathbf{y}, \mathbf{f}\right)$.

Following the steps of the previous section, we could work out the posterior distribution $p\left(\mathbf{f}|\hat{\mathbf{t}}, \mathcal{D}\right)$. However, this is not what we are looking for because what we truly want is to make predictions out of new test data. Therefore, what we seek ultimately is the distribution $p(f^\star|\mathcal{D}, d)$, where the distance variable d has been included. In order to do so, first we propose to compute the joint distribution $p\left(\hat{\mathbf{t}}, d, \mathbf{f}_+ \middle| \mathbf{y}, \gamma\right)$, where the variable $\mathbf{f}_+$ is given by an extended vector of the form $\mathbf{f}_+ = \left[\mathbf{f}^T, f^\star\right]^T$, with $f^\star$ being a point outside the training set. Second, the distribution $p(f^\star|\mathcal{D}, d)$ can be found from $p\left(\hat{\mathbf{t}}, d, \mathbf{f}_+ \middle| \mathbf{y}, \gamma\right)$ by marginalising out the variables $\mathbf{f}$ and conditioning the resulting distribution on the variables $\hat{\mathbf{t}}$ and d. Lastly, the dependence on the parameter γ can be eliminated by taking the limit $\gamma \to \infty$.

This process is facilitated if the joint distribution is factorised into well known factors. For example, $p\left(\hat{\mathbf{t}}, d, \mathbf{f}_+ \middle| \mathbf{y}, \gamma\right)$, can be given by the product of noise model, $p\left(\hat{\mathbf{t}}|\mathbf{y}, \mathbf{f}\right)$; Gaussian process prior $p(\mathbf{f}_+)$; and constraining distribution $p(d|\mathbf{y}, \mathbf{f}, \gamma)$. Firstly, the modified noise model is defined in terms of $\mathbf{f}$ by applying the values of $\hat{c}_q$ and rearranging, (see Appendix B). The result is

$$p\left(\hat{\mathbf{t}}|\mathbf{y}, \mathbf{f}\right) \quad \propto \quad \exp\left(-\frac{\beta}{2}\mathbf{f}^T\mathbf{L}\mathbf{f}\right), \tag{18}$$

with $\mathbf{L}$ defined in Equation 16. Secondly, let the augmented vector $\mathbf{f}_+$ be correlated with a covariance matrix $\mathbf{K}_+ \in \mathbb{R}^{(n+1)\times(n+1)}$, then the prior is a GP of the form

$$p(\mathbf{f}_+) \propto \exp\left(-\frac{1}{2}\mathbf{f}_+^T\mathbf{K}_+^{-1}\mathbf{f}_+\right). \tag{19}$$

For future reference, the inverse of $\mathbf{K}_+$ is partitioned as

$$\mathbf{K}_+^{-1} = \begin{pmatrix} \mathbf{C} & \mathbf{c} \\ \mathbf{c}^T & c_\star \end{pmatrix},$$

with

$$\begin{aligned} c_\star &= \left(k_\star - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k}\right)^{-1}, \\ \mathbf{c} &= -c_\star\mathbf{K}^{-1}\mathbf{k}, \\ \mathbf{C} &= \mathbf{K}^{-1} + c_\star\mathbf{K}^{-1}\mathbf{k}\mathbf{k}^T\mathbf{K}^{-1}. \end{aligned}$$

Note that the vector $\mathbf{k} \in \mathbb{R}^{N\times 1}$ is filled with scalars $k_{(n)} = \mathbf{K}\left(\mathbf{x}^{(n)}, \mathbf{x}\right)$ for $\mathbf{x} \in \mathcal{X}$.
Finally, the model still needs to consider that projected class means must be separated by the distance d. The introduction of a constraining distribution of the form of Equation 17 is what is needed. We can express this distribution in terms of $\mathbf{f}$ by replacing the term $\mathbf{w}^T\Delta\mathbf{m}$ inside the exponential by $\mathbf{f}^T\Delta\hat{\mathbf{y}}$, where $\Delta\hat{\mathbf{y}} = N_0^{-1}\mathbf{y}_0 - N_1^{-1}\mathbf{y}_1$. Therefore the constraint becomes

$$p\left(d|\mathbf{y},\mathbf{f},\gamma\right) = \lim_{\gamma\to\infty}\frac{\gamma^{\frac{1}{2}}}{\sqrt{2\pi}}\exp\left(-\frac{\gamma}{2}\left(d - \mathbf{f}^T\Delta\hat{\mathbf{y}}\right)^2\right). \tag{20}$$

Hence we can write the marginal distribution (after marginalisation of $\mathbf{f}$) as

$$p\left(f^\star, \hat{\mathbf{t}}, d\middle|\mathbf{y},\gamma\right) = \int p\left(\hat{\mathbf{t}}\middle|\mathbf{y},\mathbf{f}\right)p\left(d|\mathbf{y},\mathbf{f},\gamma\right)p\left(\mathbf{f}_+\right)\partial\mathbf{f}.$$

This is a Gaussian integral that can be solved straightforwardly by applying (for example) the material on exponential integrals (Bishop, 1995) that we present in Appendix C. After conditioning $f^\star$ on both $\hat{\mathbf{t}}$ and d, the solution is a Gaussian of the form

$$p\left(f^\star|\mathcal{D},d,\gamma\right) \propto \exp\left\{-\frac{1}{2\left(\sigma^\star\right)^2}\left(f^\star - \bar{f}^\star\right)^2\right\}$$

with mean

$$\bar{f}^\star = \lim_{\gamma\to\infty} -\gamma d\left(\sigma^\star\right)^2\mathbf{c}^T\mathbf{Q}^{-1}\Delta\hat{\mathbf{y}}.$$

and variance

$$\left(\sigma^\star\right)^2 = \lim_{\gamma\to\infty}\left(c_\star - \mathbf{c}^T\mathbf{Q}^{-1}\mathbf{c}\right)^{-1},$$

where we have defined the matrix $\mathbf{Q} = \beta\mathbf{L} + \mathbf{C} + \gamma\Delta\hat{\mathbf{y}}\Delta\hat{\mathbf{y}}^T$.

Just as in Section 4.1, the dependence on γ is eliminated by taking the limit as $\gamma\to\infty$. This procedure is detailed in Appendix C. The parameters of the distribution are

$$\bar{f}^\star = \frac{d\mathbf{k}^T\mathbf{A}^{-1}\mathbf{K}\Delta\hat{\mathbf{y}}}{\Delta\hat{\mathbf{y}}^T\mathbf{K}\mathbf{A}^{-1}\mathbf{K}\Delta\hat{\mathbf{y}}}, \tag{21}$$

and

$$(\sigma^\star)^2 = k_\star - \mathbf{k}^T \left(\mathbf{K}^{-1} - \mathbf{D}^{-1}\right) \mathbf{k}, \tag{22}$$

with the matrices

$$\mathbf{D} = \left(\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{K}\Delta\hat{\mathbf{y}} \left(\Delta\hat{\mathbf{y}}^T\mathbf{K}\mathbf{A}^{-1}\mathbf{K}\Delta\hat{\mathbf{y}}\right)^{-1} \Delta\hat{\mathbf{y}}^T\mathbf{K}\mathbf{A}^{-1}\right)^{-1}$$

and

$$\mathbf{A} = \beta\mathbf{K}\mathbf{L}\mathbf{K} + \mathbf{K}. \tag{23}$$

The predictive mean is given by a linear combination of the observed labels, in this case expressed by $\Delta\hat{\mathbf{y}}$. Additionally, the predictive variance is composed by two terms, one representing the test point and the other representing the observed data. These results are similar to those of typical GP regression, described in (Williams, 1999). The scheme proposed above will be termed Bayesian Fisher's discriminant (BFD) to facilitate its referencing.

5. Relationship with Other Models

There are several well known connections between discriminant analysis and other techniques. In the statistics community, FLD is equivalent to a t-test or F-test for significant difference between the mean of discriminants for two sampled classes, in fact, the statistic is designed to have the largest possible value (Michie et al., 1994). In this section, however, we prefer to explore the connections of our approach to some algorithms that have been applied to machine learning problems, namely kernel Fisher's discriminant and the least-squares and proximal support vector machines.

5.1 Kernel Fisher's Discriminant

The algorithm known as kernel Fisher's discriminant consists of a two stage procedure. The first consists of embedding the data space $\mathcal{X}$ into a possibly infinite dimensional reproducing kernel Hilbert space $\mathcal{F}$ via a kernel function k. The second simply consists of applying FLD in this new data space. As the second stage is exactly the same as standard linear discriminant, many of the properties for FLD observed in $\mathcal{X}$ will hold also in $\mathcal{F}$; for example, some form of regularisation needs to be included. However there is an extra effort involved in preparing the original data for a new data representation in the induced space, namely in terms of the kernel function.

Data embedding is carried out by applying a non-linear transformation $\phi : \mathcal{X} \to \mathcal{F}$ that induces a positive definite kernel function. From the theory of reproducing kernels (Aronszajn, 1950) it is well known that the vector of compounds is a weighted combination of the training samples, such that $\mathbf{w} = \sum_{i=1}^{N} \alpha^{(i)} \phi\left(\mathbf{x}^{(i)}\right)$. The application of this property plus the decomposition of the kernel into its spectrum:

$$k(\mathbf{x}, \mathbf{x}') \quad = \quad \sum_{i=1}^{d} \lambda_i \phi_i(\mathbf{x}) \phi_i\left(\mathbf{x}'\right)$$

leads to the formulation of the Rayleigh coefficient in the feature space. Following the path of other kernel methods, the novelty in (Mika et al., 1999) resides in defining the kernel function directly and working without any reference to the spectral-based formulation.

A direct implication of working in an infinite dimensional space is that there is no form to express directly the matrices Σ_w and Σ_B. Nonetheless, the discriminant function can still be written

as the rule $D(\mathbf{x}^\star) = \sum_{i=1}^{N} \alpha^{(i)} k\left(\mathbf{x}^\star, \mathbf{x}^{(i)}\right) + b$ with the coefficients $\alpha^{(i)}$'s being obtained as the solution of maximizing a new form of the statistic

$$J(\alpha) = \frac{\alpha^T \mathbf{M} \alpha}{\alpha^T \mathbf{N} \alpha}.$$

Where $\mathbf{M} = \left(\mathbf{m}_0^{\mathcal{F}} - \mathbf{m}_1^{\mathcal{F}}\right)\left(\mathbf{m}_0^{\mathcal{F}} - \mathbf{m}_1^{\mathcal{F}}\right)^T$, $\mathbf{N} = \mathbf{KLK}$ and $\mathbf{m}_q^{\mathcal{F}} = N_q^{-1}\mathbf{K}\mathbf{y}_q$. Just as in FLD, in KFD the 'within scatter' matrix is not full rank. This implies that some form of regularisation will need to be applied when inverting $\mathbf{N}$ and this will generally be done by applying $\mathbf{N}_\delta = \mathbf{N} + \delta\mathbf{C}$, with $\mathbf{C}$ being the identity or the kernel matrices. Therefore the solution can be computed by either solving a generalised eigenproblem or by taking

$$\alpha_{KFD} \propto (\mathbf{N} + \delta\mathbf{C})^{-1}\left(\mathbf{m}_0^{\mathcal{F}} - \mathbf{m}_1^{\mathcal{F}}\right). \tag{24}$$

We are now in position to show the equivalence of KFD and our scheme, BFD.

Demonstration Disregarding the bias term, the projection of a new test point under KFD will be

$$\bar{f}^\star = \alpha_{KFD}^T \mathbf{k}. \tag{25}$$

Our claim is that Equation 21 is equivalent to Equation 25. In other words, that the projection of a new test point in KFD is equal to the mean of the predictive distribution for a test point under BFD. As in both equations the vector $\mathbf{k}$ is the same, we can write Equation 21 as

$$\bar{f}^\star = \alpha_{BFD}^T \mathbf{k},$$

with the vector

$$\alpha_{BFD} \propto d\mathbf{A}^{-1}\mathbf{K}\Delta\hat{\mathbf{y}} \tag{26}$$

and the constant of proportionality being given by the denominator of (21). Then our proof reduces to showing that the coefficients α_{KFD} and α_{BFD} are the same.

On one hand, we start by analysing KFD's main result which is given by Equation 24. From the definition of $\mathbf{m}_q^{\mathcal{F}}$, the difference $\left(\mathbf{m}_0^{\mathcal{F}} - \mathbf{m}_1^{\mathcal{F}}\right)$ can be written as $\mathbf{K}\Delta\hat{\mathbf{y}}$, with $\Delta\hat{\mathbf{y}} = \left(N_0^{-1}\mathbf{y}_0 - N_1^{-1}\mathbf{y}_1\right)$, and by regularising $\mathbf{N}$ with a multiple of the kernel matrix we obtain

$$\alpha_{KFD} \propto \left(\mathbf{KLK} + \beta^{-1}\mathbf{K}\right)^{-1}\mathbf{K}\Delta\hat{\mathbf{y}},$$

where β^{-1} is the regularisation coefficient.
On the other hand, substituting the value of $\mathbf{A}$ (Equation 23) into Equation 26, premultiplying by β and ignoring d we get

$$\alpha_{BFD} \propto \left(\mathbf{KLK} + \beta^{-1}\mathbf{K}\right)^{-1}\mathbf{K}\Delta\hat{\mathbf{y}},$$

which clearly is the regularised version of KFD that we were talking about.

As an additional insight, we observe that the coefficients α_{BFD} have an equivalent α_{KFD} if and only if KFD uses a regularisation based on a multiple of the kernel matrix. This equivalence is lost if the regulariser is based on the identity matrix.

5.2 Least Squares Support Vector Machines

A least squares support vector machine (LS-SVM) implements a two-norm cost function[3] and uses equality constraints instead of the inequalities present in the standard SVM (Vapnik, 1995). This greatly simplifies the way to obtain the solution as the resulting system of equations is linear. Unfortunately the sparseness which is characteristic of the SVM is lost. LS-SVM's have been related to ridge regression with modified targets, discriminant analysis in the feature space (KFD) and, as many other kernelised algorithms, to GP's.

Given a set of training data $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ with labels $y^{(i)} \in \{-1, 1\} \forall i$, the primal optimisation problem for an LS-SVM is expressed as

$$\begin{aligned} \min \quad & C = \frac{\mu}{2}\mathbf{w}^T\mathbf{w} + \frac{\zeta}{2}\sum_{n=1}^{N}\left(e^{(n)}\right)^2 \\ s.t. \quad & e^{(n)} = \left(y^{(n)} - \mathbf{w}^T\mathbf{x}^{(n)}\right) \quad \forall n, \end{aligned}$$

with μ and ζ being positive coefficients. This formulation in particular was given by Van Gestel et al. (2002) to elaborate the Bayesian framework of the LS-SVM. Such framework is nothing else but the recognition that the primal problem implements a regularised least squares cost function with regression to the labels. This cost function arises from the model depicted in Figure 4. In this figure, the joint distribution over labels and parameters factorises as $p(\mathbf{y}, \mathbf{w}|\mathbf{X}) = p(\mathbf{y}|\mathbf{X}, \mathbf{w}) \times p(\mathbf{w})$, with noise model $p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \mathcal{N}\left(\mathbf{X}\mathbf{w}, \zeta^{-1}\mathbf{I}\right)$ and prior $p(\mathbf{w}) = \mathcal{N}\left(\mathbf{0}|\mu^{-1}\mathbf{I}\right)$.

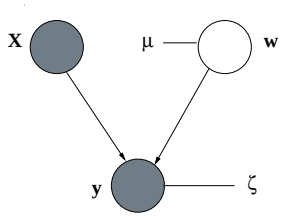

Figure 4: LS-SVM noise model assumes a regularised least squares cost function. The model depicted can be interpreted as the joint distribution $p(\mathbf{y}, \mathbf{w}) = p(\mathbf{y}|\mathbf{X}, \mathbf{w})\, p(\mathbf{w})$, whereby the noise is Gaussian, $p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \mathcal{N}\left(\mathbf{X}\mathbf{w}, \zeta^{-1}\mathbf{I}\right)$, as is the prior $p(\mathbf{w}) = \mathcal{N}\left(\mathbf{0}|\mu^{-1}\mathbf{I}\right)$. In this model the targets and the labels are the same $\mathbf{t} \equiv \mathbf{y}$.

It is clear from the figure that LS-SVM employs a different noise model than BFD. In practice, the regression to the labels assumption can work well. However, it suffers from the fundamental missconception that the class labels ± 1 have to coincide with the projected class centres c_q. The main difference with our algorithm is that the LS-SVM assumes that targets and labels are the same, $\mathbf{t} \equiv \mathbf{y}$, but we do not.

Van Gestel et al. (2002) were aware of this limitation[4] and relaxed the assumption $\mathbf{t} \equiv \mathbf{y}$ by modelling the distribution $p\left(\hat{\mathbf{t}}_q \middle| \mathbf{X}, \mathbf{w}\right)$ by application of Bayes' rule. In other words, they computed $p\left(\hat{\mathbf{t}}_q \middle| \mathbf{X}, \mathbf{w}\right) \propto p\left(\mathbf{X}|\hat{\mathbf{t}}_q, \mathbf{w}\right) p\left(\hat{\mathbf{t}}_q\right)$. This is in marked contrast with the strategy adopted in this paper. As is shown by Equation 12, in BFD we model directly the distribution $p\left(\hat{\mathbf{t}}_q \middle| \mathbf{X}, \mathbf{y}, \mathbf{w}\right)$. Hence it can

3. This is instead of the traditional ℓ_1.
4. See Section 3.2 of their paper.

be seen that in our approach $\mathbf{y}$ is used as a conditioning value whereas in Van Gestel's paper it is not.

5.2.1 PROXIMAL SUPPORT VECTOR MACHINES

Another related approach is known as the proximal support vector machine or P-SVM, proposed by Fung and Mangasarian (2001). A P-SVM is very close to LS-SVM in the sense that both of them consider equality constraints and implement regularised least squares cost functions. However, P-SVM's have been interpreted from the point of view of classifying points by clustering data around two parallel hyperplanes; whereas LS-SVM's have been interpreted from the more classical point of view of maximising the margin around a single hyperplane. P-SVM's have also been approached from a probabilistic point of view by Agarwal (2002). Indeed, by following Agarwal's work it is possible to see that they also implement the graphical model depicted in Figure 4, except for a few changes in parameters. Ignoring the bias term, in P-SVM's the joint distribution $p(\mathbf{y}, \mathbf{w}|\mathbf{X})$ is factorised according to the noise model $p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \mathcal{N}\left(\mathbf{X}\mathbf{w}, \sigma^2\mathbf{I}\right)$ and the prior distribution $p(\mathbf{w}) = \mathcal{N}\left(\mathbf{0}, \nu\sigma^2\mathbf{I}\right)$. The parameter σ^2 is the variance of the residuals[5] while ν is known as ridge parameter. In many applications, such as data mining, the ridge parameter is chosen by cross-validation. It is clear that this task becomes unfeasible if the ridge parameter is taken to the extreme of considering one parameter for every 'predictor', in other words, if we take as rigde parameter a matrix of the form $diag(\nu_1, \ldots, \nu_d)$.

In (Agarwal, 2002) the problem of tuning the ridge parameter is addressed by studying its effects on ridge regression. This can be observed by writing up the regularised P-SVM cost function

$$C_{PSVM} = \frac{1}{\sigma^2}\left[(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \frac{1}{\nu}\mathbf{w}^T\mathbf{w}\right].$$

Whenever ν becomes small, the ridge part takes over, but if it becomes large the 'noise' part will dominate. Nevertheless, it is clear that BFD implements a different type of noise model when compared to LS-SVM's and P-SVM's.

6. Connections with the Generalisation Error

In Section 3.1.1 we saw that optimisation of the proposed noise model and that of Rayleigh's coefficient give equivalent results. In both cases the solution to the discriminant problem was given by adjusting the level of β. In order to understand better the physical significance that this represents, it is useful to analyse the problem from the point of view of classification of two populations. Specifically, during this section we will refer to the plot in Figure 5 and always assume that both classes have the same cost of misclassification.

In Figure 5, it can be observed that both mapping distributions share the same precision. Under this assumption, for fixed d, we can see that the generalisation error will decrease as β increases, *i.e.* as $\beta^{-1/2}$ decreases. From the point of view of projected data, the problem has shifted from computing the direction of discrimination to that of minimising the generalisation error through the adjustment of the variable β.

The likelihood function $L(\mathbf{f})$ defined in Equation 11 allows us to think of β as an extra random variable. Hence placing a prior over it not only places a prior over the generalisation error but on

5. The residuals are defined as $e^{(n)} = y^{(n)} - \mathbf{w}^T\mathbf{x}^{(n)}, \forall n$.

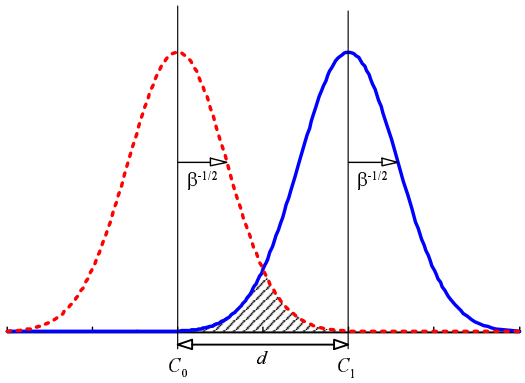

Figure 5: Generalisation error as it relates to β and d. The shaded area gives the generalisation error if the true densities conform to those given by two Gaussians with equal precision β. The class centres have been denoted by c_q with $q \in \{1,0\}$.

Rayleigh's coefficient as well. Consider, for example, the case where $d = 2$ and the class priors are equal: if the data does truly map to the mixture distribution, then the generalisation error will be

$$E_{\text{eq}} = \frac{1}{2} - \frac{1}{2}\text{erf}\left(\sqrt{\frac{\beta}{2}}\right).$$

Let Equation 11 be a 'likelihood function', then by considering a gamma distribution $\mathcal{G}(\beta|a,b)$ as a prior,

$$p(\beta) = \frac{b^a}{\Gamma(a)}\beta^{a-1}\exp(-b\beta),$$

the MAP solution for β will be (see Appendix D)

$$\hat{\beta}_{\text{MAP}} = \frac{N + 2a - 2}{\sigma_1^2 + \sigma_0^2 + 2b}. \tag{27}$$

By setting $a = b = 0.5$ we indirectly obtain a uniform distribution over E_{eq}, which is also a chi-square distribution with one degree of freedom. This special case leads to a new expression of the form

$$\hat{\beta}_{\text{MAP}} = \frac{N-1}{\sigma_1^2 + \sigma_0^2 + 1}, \tag{28}$$

which can be viewed as a regularised version of Equation 8. The prior could also be used to bias β towards low or high generalisation errors if this is thought appropriate.

From the discussion of Section 5.1, taking the limit as $\beta \to \infty$ leads to the standard kernel Fisher's discriminant. From Figure 5 it can be seen that an *a priori* setting of β^{-1} to zero is equivalent to assuming that we can achieve a generalisation error of zero.

OTHER SPECIAL CASES

Taking the limit as $\beta \rightarrow 0$ causes the mean prediction for $f^\star$ and its variance to take on a much simpler form,

$$\bar{f}^\star = \alpha_\beta^T \mathbf{k}$$

where

$$\alpha_\beta = \frac{d\Delta\hat{\mathbf{y}}}{\Delta\hat{\mathbf{y}}^T \mathbf{K} \Delta\hat{\mathbf{y}}},$$

and

$$(\sigma^\star)^2 = k_\star - \mathbf{k}^T \frac{\Delta\hat{\mathbf{y}}^T \Delta\hat{\mathbf{y}}}{\Delta\hat{\mathbf{y}}^T \mathbf{K} \Delta\hat{\mathbf{y}}} \mathbf{k}.$$

This result is remarkable for the absence of any requirement to invert the kernel matrix, which greatly reduces the computational requirements of this algorithm. In fact, driving β to zero leads to the well known Parzen windows classifier, sometimes known as probabilistic neural network, (Duda and Hart, 1973). See the work of Schölkopf and Smola (2002) or Roth (2005) for some related studies in limiting cases.

7. Optimising Kernel Parameters

One key advantage of our formulation is that it leads to a principled approach for determining all the model parameters. In the Bayesian formalism it is quite common to make use of the marginal likelihood to reach this purpose, therefore we look to optimise

$$\mathcal{L}(\Theta_t) = \log p(\mathbf{t}|\mathcal{D}, \Theta_t),$$

with respect to the model parameters Θ_t. Recall in Section 3.1 that we optimised the likelihood with respect to the parameters c_0 and c_1 leading to a new encoding of the targets

$$\hat{\mathbf{t}}_q = \left(\frac{\mathbf{f}^T \mathbf{y}_q}{N_q}\right) \mathbf{y}_q.$$

We back substituted these values into the likelihood in order to demonstrate the equivalence with maximisation of Rayleigh's coefficient. Unfortunately, one side effect of this process is that it makes the new targets $\hat{\mathbf{t}}$ dependent on the inputs. As a consequence, the targets will shift whenever the kernel parameters are changed. As expressed in Section 3.1.1, one solution could be to iterate between determining $\mathbf{t}_0$, $\mathbf{t}_1$ and optimising the rest of the parameters. This approach is simple, but it may be difficult to prove convergence properties. We therefore prefer to rely on an expectation-maximisation (EM) algorithm (Dempster et al., 1977) which finesses this issue and for which convergence is proved.

7.1 EM Algorithm

We denote the parameters of the prior as Θ_k and the complete set of model parameters as $\Theta_t = \{\Theta_k, \beta\}$. Then the goal is to solve the problem $\arg\max_{\Theta_t} \log p\left(\hat{\mathbf{t}} \middle| \mathbf{X}, \Theta_t\right)$, where we have made use again of the modified targets $\hat{\mathbf{t}}$. In order to solve the problem, a variational lower bound on the marginal log-likelihood is imposed

$$\mathcal{L}(\Theta_t) \geq \int q(\mathbf{f}) \log \frac{p\left(\hat{\mathbf{t}} \middle| \mathbf{y}, \mathbf{f}, \beta\right) p(\mathbf{f}|\mathbf{X}, \Theta_k)}{q(\mathbf{f})} d\mathbf{f}, \tag{29}$$

where $q(\mathbf{f})$ is a distribution over the latent variables that is independent on the current value Θ_t. EM consists of the alternation of the maximisation of $\mathcal{L}$ with respect to $q(\mathbf{f})$ and Θ_t, respectively, by holding the other fixed. This procedure repeated iteratively guarantees a local maxima for the marginal likelihood will be found. Thus our algorithm will be composed of the alternation of the following steps:

E-step Given the current parameters Θ_t^{it}, approximate the posterior with

$$q^{it}(\mathbf{f}) \propto \exp\left(-\frac{1}{2}\mathbf{f}^T\Sigma_p^{-1}\mathbf{f}\right),$$

where

$$\Sigma_p = \left(\mathbf{K}^{-1} + \beta\mathbf{L}\right)^{-1}. \tag{30}$$

M-step Fix $q^{it}(\mathbf{f})$ to its current value and make the update

$$\Theta_t^{it+1} = \arg\max_{\Theta_t} \mathcal{L}, \tag{31}$$

where the evidence is computed as $\mathcal{L} = \left\langle \log p\left(\hat{\mathbf{t}}|\,\mathbf{y},\mathbf{f},\beta\right) p(\mathbf{f}|\mathbf{X},\Theta_k)\right\rangle_{q(\mathbf{f})}$. We have used the notation $\langle\cdot\rangle_{p(x)}$ to indicate an expectation under the distribution $p(x)$.

Maximisation with respect to Θ_k, the kernel parameters, cannot be done in closed form and has to rely on some optimisation routine, for example gradient descent, therefore it is necessary to specify the gradients of Equation 29 w.r.t. Θ_k. An update for β can be worked out quite easily because the maximisation of $\mathcal{L}$ with respect to this parameter has a closed form solution. The expression obtained is of the form

$$\hat{\beta}^{it} = \frac{N}{\overline{\sigma}_1^2 + \overline{\sigma}_0^2},$$

where $\overline{\sigma}_1^2 = \sum y_n \left\langle (f_n - \mu_1)^2 \right\rangle$ and the expectation $\langle\cdot\rangle$ is computed under the predictive distribution for the nth training point, see Equation 21. An expression for $\overline{\sigma}_0^2$ is given in a similar way.

7.2 Updating β

Following our discussion in Section 6, we propose (and in fact used) Equation 28 to update the value of β at every iteration. We repeat the expression here

$$\hat{\beta}_{\text{MAP}}^{it} = \frac{N-1}{\overline{\sigma}_1^2 + \overline{\sigma}_0^2 + 1}. \tag{32}$$

The resulting optimisation framework is outlined in Algorithm 1.

8. Experiments

In this section we report the results of experiments that we carried out to test our algorithmic approach. A first batch of experiments was carried out on classification of synthetic data with the purpose of understanding better the behaviour of the algorithm, and in order to test it more realistically, a second batch of experiments was carried out on benchmark data.

Algorithm 1 A possible ordering of the updates.

Select Convergence tolerances η_β and η_{Θ_k}.
Set Initial values $\Theta_k^{(0)}$ and $\hat{\beta}^{(0)}$.
Require Data-set $\mathcal{D} = (\mathbf{X}, \mathbf{y})$.
while change in $\hat{\beta}^{(it)} < \eta_\beta$ and change in $\Theta_k^{(it)} < \eta_{\Theta_k}$ **do**

- Compute kernel matrix $\mathbf{K}$ using $\Theta_k^{(it)}$.
- Update Σ_p with Equation 30
- Use scale conjugate gradients to maximise $\mathcal{L}$ with respect to $\Theta_k^{(it)}$. Apply Equation 31
- Update $\hat{\beta}^{(it)}$, use Equation 32.

end

8.1 Toy Data

As a first experiment, we compared the KFD, LS-SVM and BFD algorithms on four synthetic data sets using an RBF kernel. Additionally, as a second experiment, we used BFD with an ARD prior on the same data sets to observe some of the capabilities of our approach. In order to facilitate further reference, each data set will be named according to its characteristics. Firstly, **Spiral**[6] can only be separated by highly non-linear decision boundaries. **Overlap** comes from two Gaussian distributions with equal covariance, and is expected to be separated by a linear plane. **Bumpy** comes from two Gaussians but by being rotated at 90 degrees, quadratic boundaries are called for. Finally, **Relevance** is a case where only one dimension of the data is relevant to separate the data.

We hypothesized BFD would perform better than LS-SVM and KFD in all the cases because it models directly the class conditional densities. In order to compare the three approaches, we trained KFD, LS-SVM and BFD classifiers with a standard RBF kernel, as specified in Appendix E. Model parameters for KFD and LS-SVM were selected by 10-fold cross-validation whereas BFD was trained by maximising the evidence, using Algorithm 1.

In Figure 6 we present a comparison of the three algorithms. We can observe a similar performance in the case of **Spiral**; however it is encouraging to observe that BFD gives more accurate results in the rest of the cases. Despite not producing a straight line, KFD and BFD give accurate results in **Overlap**, whereas LS-SVM overfits. If none of the algorithms separates this data set with a line it is because obtaining a linear boundary from an RBF kernel is extremely difficult (see Gramacy and Lee, 2005). In **Bumpy**, the three algorithms give arguably the same solution, with BFD having the smoothest boundary. Lastly, in **Relevance** all the algorithms provide accurate results, with BFD giving the smoothest solution. In all these experiments we set the initial $\Theta_t = \mathbf{1}$ for BFD and furthermore, observed that BFD did not present any initialisation problems. In all our simulations, we let the algorithm stop whenever $\eta_\beta < 1 \times 10^{-6}$ or the change in $\eta_{\Theta_k} < 1 \times 10^{-6}$.

As a second experiment, we were interested in training BFD to test the different facets of the following kernel

$$k\left(\mathbf{x}^i, \mathbf{x}^j\right) = \theta_1 \exp\left(-\frac{\theta_2}{2}\left(\mathbf{x}^i - \mathbf{x}^j\right)^T \Theta_{ard}\left(\mathbf{x}^i - \mathbf{x}^j\right)\right) + \theta_3\left(\mathbf{x}^i\right)^T \Theta_{ard}\mathbf{x}^j + \theta_4 + \theta_5\delta_{ij}, \tag{33}$$

6. This was first used by Lang and Witbrock (1988).

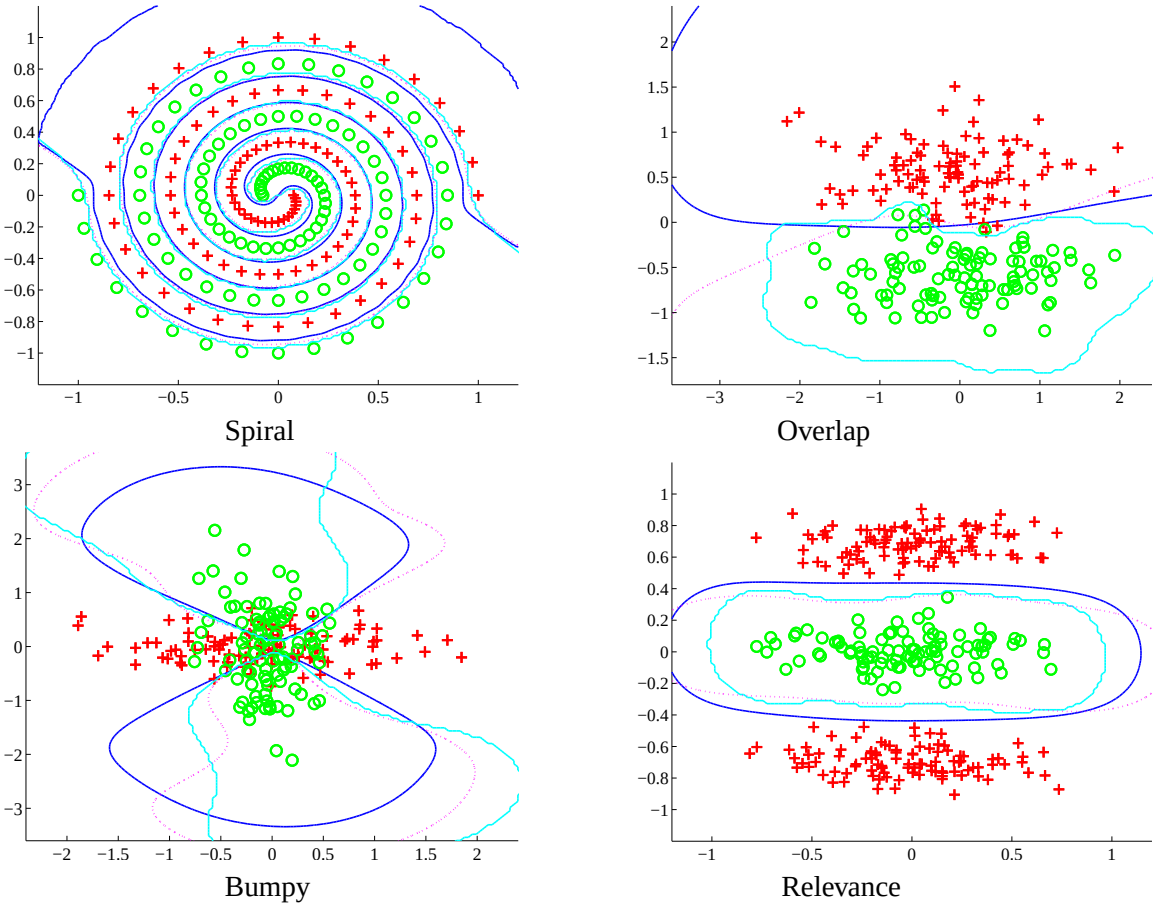

Figure 6: Comparison of classification of synthetic data sets using an RBF kernel. Two classes are shown as pluses and circles. The separating lines were obtained by projecting test data over a grid. The lines in blue (dark), magenta (dashed) and cyan (gray) were obtained with BFD, KFD and LS-SVM respectively. Kernel and regularisation parameters for KFD and LS-SVM were obtained by 10-fold cross validation, whereas BFD related parameters were obtained by evidence maximisation. We trained BFD using Algorithm 1; details of our implementations are given in Appendix E.

where δ_{ij} is the Kronecker delta and the matrix $\Theta_{ard} = diag(\theta_6, \ldots, \theta_{6+d-1})$ with d being the dimension of $\mathbf{X}$. This kernel has four components: an RBF part composed of $(\theta_1, \theta_2, \Theta_{ard})$; a linear part, composed of (θ_3, Θ_{ard}); a bias term given by θ_4 and the so-called 'nugget' term θ_5 which, for a large enough value θ_5, ensures that $\mathbf{K}$ is positive definite and therefore invertible at all times. Therefore, the parameters of the model are $\Theta_t = (\Theta_k, \beta)$, with $\Theta_k = (\theta_1, \ldots, \theta_{6+d-1})$.

On this occassion, BFD got stuck into local minima so we resorted to do model selection to choose the best solution. This process was carried out by training each data set with three different initial values for θ_2 while the remaining $\theta_{i\neq 2}$ were always initialised to 1. In the cases of **Bumpy** and **Relevance** we made the initial $\theta_2 = \left[10^{-2}, 10^{-1}, 1\right]$, for **Spiral** we made it equal to $[1, 10, 100]$ and for **Overlap**, $\left[1.5 \times 10^{-2}, 10^{-1}, 1\right]$. From the resulting solutions (three per data set), we selected the model that produced the highest marginal likelihood $\mathcal{L}$. In all our simulations, we let the algorithm

stop whenever $\eta_\beta < 1 \times 10^{-6}$ or the change in $\eta_{\Theta_k} < 1 \times 10^{-6}$. The parameter β was always initialised to 1. The selected models for each set are summarised in Figure 7.

The results are promising. In **Spiral**, the separating plane is highly non-linear as expected. Meanwhile, we observe in **Overlap** that the predominating decision boundary in the solution is linear. In **Bumpy**, the boundary starts to resemble a quadratic and, finally, for **Relevance**, only one dimension of the data is used to classify the data. Note that the values for Θ_k, summarised in Table 1, go in accordance with these observations. For example, in **Overlap** and **Relevance**, the value of θ_6 is significantly lower than θ_7, indicating that only one dimension of the data is relevant for the solution. This is markedly different to the cases of **Spiral** and **Bumpy**, where both dimensions (θ_6 and θ_7) have been given relatively the same weights. Hence, for every case we have obtained sensible solutions. All the kernel parameters determined by the algorithm, for the four experiments, are given in Table 1.

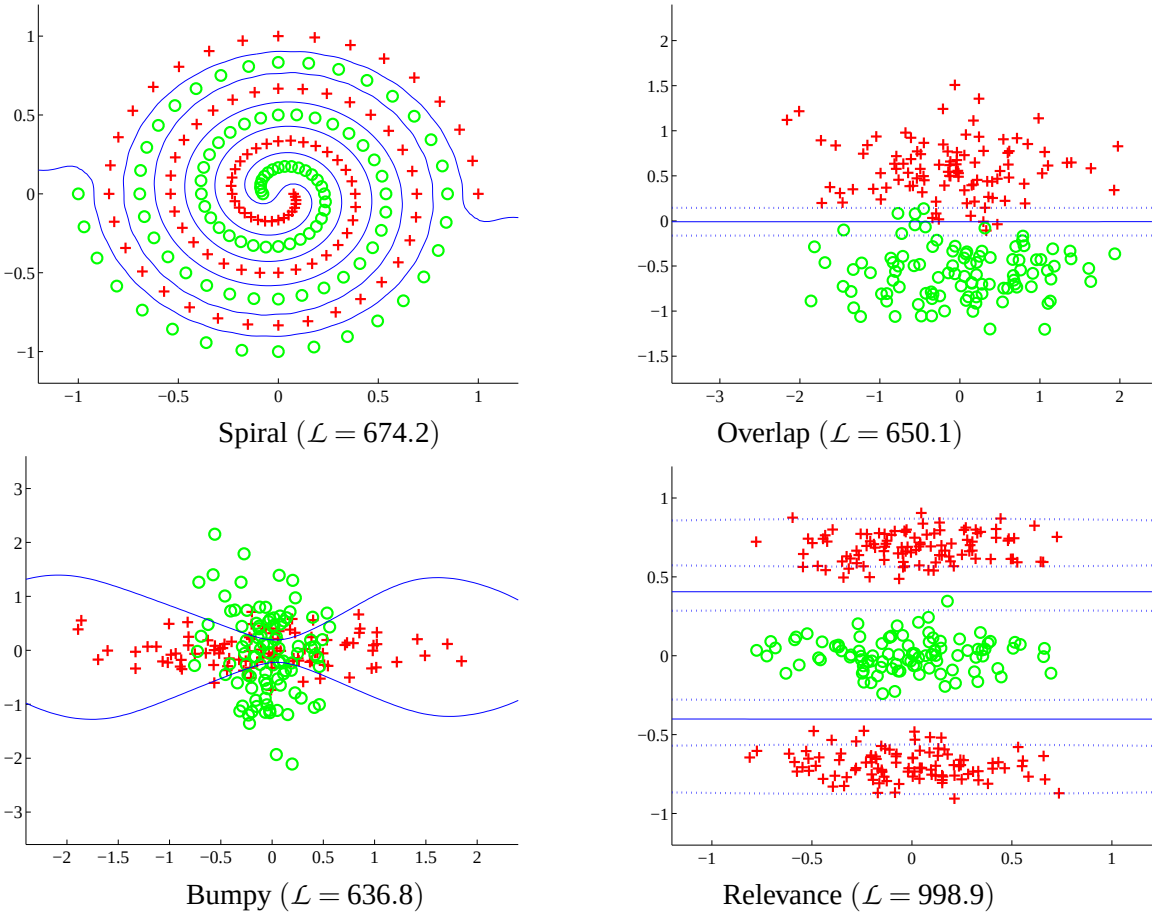

Figure 7: Classification results on toy data sets using an ARD prior. Two classes are shown as pluses and circles. The decision boundary is given by the solid line. Dotted lines indicate points at $1/4$ of the distance (as measured in the projected space) from the decision boundary to the class mean. Log-likelihood values appear enclosed by brackets.

Figure 8 shows an example of the result of training **Spiral** with a poor initialisation. It can be seen that the value of the marginal likelihood in this case is smaller to the one presented in Figure 7. However, this behaviour is not exclusive of BFD, indeed we observed a very similar situation with a poorly initialised Bayesian LS-SVM and with KFD cross-validated with a badly selected grid.

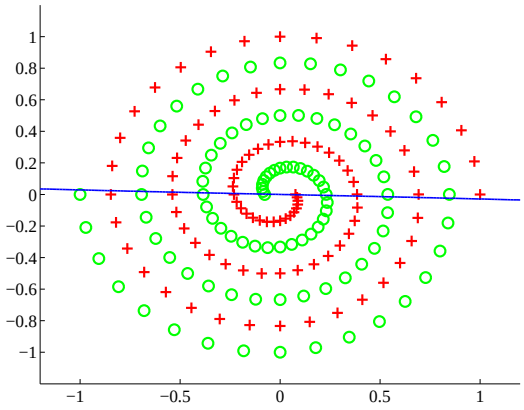

Figure 8: The solution for the spiral data with a poor initialisation $\theta_2 = 1$. Associated log-likelihood $\mathcal{L} = 562.7$.

Experiment	$\ln\theta_1$	$\ln\theta_2$	$\ln\theta_3$	$\ln\theta_4$	$\ln\theta_5$	$\ln\theta_6$	$\ln\theta_7$
Spiral	8.5015	−9.5588	1.0139	−4.9759	−10.6373	−2.78	−2.9609
Overlap	0.5011	−7.9801	1.1455	−4.8319	−8.5990	−6.9953	−0.1026
Bumpy	4.9836	−10.8222	1.1660	−4.7495	−13.5996	−3.9131	−3.7030
Relevance	4.6004	−9.5036	1.2734	−4.9351	−13.8155	−6.9968	−1.5386

Table 1: *log*-values of the parameters learnt with BFD for the different toy experiments. In **Overlap** and **Relevance**, the weights of the feature θ_6 are low if compared with the feature θ_7. This is in contrast with **Spiral** and **Bumpy**, where both features have been given relatively the same weights.

8.2 Benchmark Data Sets

In order to evaluate the performance of our approach, we tested five different algorithms on well known problems. The algorithms used were: linear and quadratic discriminants (LDA and QDA), KFD, LS-SVM and BFD. The last two algorithms provided the opportunity to use ARD priors so they were reported as well. We used a synthetic set (**banana**) along with 12 other real world data sets coming from the **UCI**, **DELVE** and **STATLOG** repositories.[7] In particular, we used instances of these data that had been preprocessed an organised by Rätsch et al. (1998) to do binary classification tests. The main difference between the original data and Rätsch's is that he converted every problem

7. The **breast cancer** domain was obtained from the University Medical Center, Institute of Oncology, Ljubljana, Yugoslavia. Thanks to M. Zwitter and M. Soklic for the data.

into binary classes and randomly partitioned every data set into 100 training and testing instances.[8] In addition, every instance was normalised to have zero mean and unit standard deviation. More details can be found at (Rätsch et al., 1998).

Mika et al. (1999) and Van Gestel et al. (2002) have given two of the most in depth comparisons of algorithms related to FLD. Unfortunately, the reported performance in both cases is given in terms of test-set accuracy (or error rates), which implied not only the adjustment of the bias term but also the implicit assumption that the misclassification costs of each class were known. Given that discriminant methods operate independently of the method of bias choice, we felt it more appropriate to use a bias independent measure like the area under the ROC curve (AUC).

The LDA and QDA classifiers were provided by the Matlab function `classify` with the options 'linear' and 'quadratic', respectively. In both cases, no training phase was required, as described by Michie et al. (1994). The output probabilities were used as latent values to trace the curves. Meanwhile, for KFD's parameter selection we made use of the parameters obtained previously by Mika et al. (1999) and which are available at `http://mlg.anu.edu.au/~raetsch`. The ROC curves for KFD were thus generated by projecting every instance of the test set over the direction of discrimination.

Mika trained a KFD on the first five training partitions of a given data set and selected the model parameters to be the median over those five estimates. A detailed explanation of the experimental setup for KFD and related approaches can be found in Rätsch et al. (1998) and Mika et al. (1999). In the case of LS-SVM, we tried to follow a similar process to estimate the parameters, hence we trained LS-SVM's on the first five realisations of the training data and then selected the median of the resulting parameters as estimates. In the same way, projections of test data were used to generate the ROC curves.

Finally, for BFD we also tried to follow the same procedure. We trained a BFD model with $N_x = 8$ different initialisations over the first five training instances of each data set. Hence we obtained an array of parameters of dimensions 8×5 where the rows were the initialisations, the columns were the partitions and each element a vector Θ_t. For each column, we selected the results that gave the highest marginal likelihood, so that the array reduced from 40 to only 5 elements. Then we followed the KFD procedure of selecting the median over those parameters. In these experiments, we used the tolerances η_β and η_{Θ_k} to be less than 1×10^{-6}. More details of the experimental setup are given in Appendix E.

In Table 2 we report the averages of the AUC's over all the testing instances of a given data set. In the cases of KFD, LS-SVM and BFD we used the RBF kernel of Appendix E. Computation of the ROC curves were done with the function `ROC` provided by Pelckmans et al. (2003) and Suykens et al. (2002) and no further processing of the curves was required, for instance removing convexities was unnecessary.

It can be observed that BFD outperforms all the other methods in $6/13$ data sets, comes second in 3 cases and third in the remaining 4. In particular, it is remarkable to see BFD performing consistently better than KFD across most of the problem domains. It seems that leaving the 'regression to the labels' assumption pays-off in terms of areas under the ROC curves. It is also interesting to observe that LDA performs well in almost all the problems (except **banana**) and it thus indicates that most of these data sets could be separated with a linear hyperplane with acceptable results. From these results we can conclude that the better designed noise model in BFD allows it to outperform 'similar'

8. Data sets can be obtained from `http://mlg.anu.edu.au/~raetsch`.

state of the art approaches. The P-SVM was not included in the experiments because it is a 'type-of' LS-SVM.

RBF	Banana	Breast	Diabetis	German	Heart	Image
LDA	53.7 (1.3)	71.2 (5.2)	**82.7** (**1.6**)	78.5 (2.5)	90.1 (2.6)	87.9 (0.7)
QDA	64.7 (2.5)	70.8 (5.5)	80.3 (2.0)	76.5 (2.6)	86.9 (3.1)	91.2 (1.5)
KFD	**96.1** (**0.4**)	70.9 (5.8)	76.7 (2.3)	69.9 (4.7)	88.8 (3.0)	**99.5** (**0.1**)
LS-SVM	95.5 (0.4)	61.1 (5.2)	73.7 (2.3)	74.0 (2.8)	89.9 (2.8)	98.8 (0.3)
BFD	95.1 (0.6)	**73.4** (**5.3**)	81.1 (1.9)	**79.0** (**2.5**)	**90.9** (**2.7**)	98.2 (0.4)

RBF	Ringnorm	Flare S.	Splice	Thyroid	Titanic	Twonorm	Waveform
LDA	80.0 (0.8)	**73.9** (**1.9**)	91.8 (0.4)	86.6 (5.8)	70.8 (1.0)	99.7 (0.0)	**92.5** (**0.7**)
QDA	99.8 (0.0)	61.6 (1.8)	**93.0** (**0.4**)	97.5 (1.7)	71.4 (2.0)	99.5 (0.0)	91.2 (0.4)
KFD	99.8 (0.0)	65.6 (2.5)	91.3 (0.5)	97.4 (3.6)	70.9 (1.0)	**99.8** (**0.0**)	88.6 (0.5)
LS-SVM	99.6 (0.1)	73.8 (1.6)	88.2 (0.7)	97.8 (1.4)	**73.8** (**2.4**)	93.8 (0.8)	83.3 (1.2)
BFD	**99.9** (**0.0**)	72.9 (2.0)	91.6 (0.5)	**98.5** (**1.1**)	71.6 (0.6)	**99.8** (**0.0**)	91.6 (0.8)

Table 2: Average classification results of benchmark data. We report mean and standard deviations (within brackets) of the AUC over all testing instances. The compared algorithms are: linear discriminant (LDA), quadratic discriminant (QDA), kernel Fisher's discriminant (KFD), least squares support vector machine (LS-SVM) and Bayesian Fisher's discriminant (BFD). In all the experiments an RBF kernel was used. It can be observed that BFD performs better in 6 out of 13 problem domains.

The BFD framework allows for the inclusion of some type of ARD priors. Incorporation of this type of prior performs feature selection by assigning very high weights to some of the posterior values of the hyperparameters and hence prunning out features, (see Mackay, 1995). We were interested in comparing our approach with the Bayesian version of the LS-SVM, which can also make use of ARD priors. Our results are presented in Table 3. In this case, however, the comparison is tighter with LS-SVM performing narrowly better than BFD in 7 out of the 13 problems. The EM algorithm we proposed is slower to converge than direct optimisation of the marginal likelihood as can be applied to the LS-SVM. Our use of the EM algorithm is necessary due to the nature of the moving targets, this is a disadvantage of our approach. Hence to obtain a solution in a reasonable time, we were obliged to reduce the number of initialisations to $N_x = 3$ and to increase the tolerances η_β and η_{Θ_k} to be less than 1×10^{-5} and 1×10^{-6}, respectively.

In Figure 9 we show a comparison of the weights assigned to each feature in two data sets, Ringnorm and Splice. We were interested on showing if there was any correlation on the degree of importance assigned to each feature by the two algorithms. Ringnorm and Splice were specially selected because they were examples in which BFD performed better (marginally) and worse than LS-SVM, respectively. In the figure, we report the values of the inverse weights, Θ_{ard}, for BFD while for LS-SVM we report the 'ranking' coefficients produced by the LS-SVM implementation we used (see Appendix E). These coefficients are related to the values Θ_{ard}. For identical results we expected to observe a high degree of correlation between BFD weights and LS-SVM rankings. As a first example, in Figure 9 we observe that Ringnorm is assigned varied values by BFD and LS-SVM. In fact, for features $[3-6]$ and 18, there is a reasonable degree of overlap between assigned values, and this could help to explain why both algorithms performed similarly well. This observation goes

in accordance with our intuition. It is noticeable that none of the features in BFD has been driven to zero, which indicates that this algorithm required all of the features to learn a solution. The second case corresponds to Splice. In this data set, LS-SVM performed better than BFD by a wide margin and this could well be explained by the aggressive pruning of features that BFD performed; as it is shown in the figure.

ARD	Banana	Breast	Diabetis	German	Heart	Image
LS-SVM	91.6 (1.0)	72.3 (5.4)	**83.3** (**1.7**)	**79.5** (**2.5**)	90.5 (2.6)	**98.9** (**0.6**)
BFD	**95.1** (0.6)	**74.2** (**5.1**)	81.1 (1.6)	77.9 (2.6)	**90.9** (**2.7**)	76.8 (0.9)

ARD	Ringnorm	Flare S.	Splice	Thyroid	Titanic	Twonorm	Waveform
LS-SVM	99.8 (0.0)	66.0 (3.3)	**95.7** (**0.3**)	**99.5** (**0.5**)	**73.6** (**2.6**)	99.6 (0.0)	**96.4** (**0.2**)
BFD	**99.9** (**0.0**)	**73.2** (**1.7**)	88.8 (0.5)	98.9 (0.8)	71.9 (1.1)	**99.7** (**0.0**)	94.0 (0.1)

Table 3: Average classification results of benchmark data. We report mean and standard deviations (within brackets) of the AUC over all testing instances. This table compares the BFD algorithm against an LS-SVM, both employing ARD based kernels. In this case Bayesian LS-SVM outperforms BFD in 7 out of 13 cases.

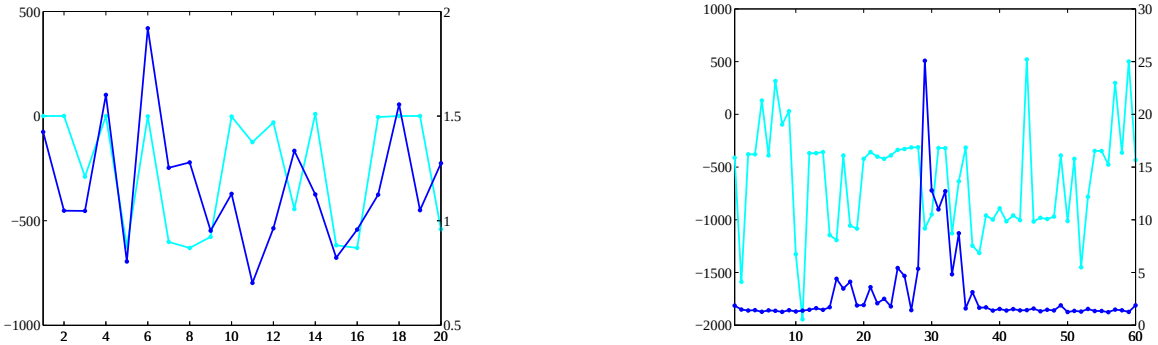

Figure 9: Plot of the feature values learned by BFD and LS-SVM on Ringnorm and Splice data sets. For both algorithms we used an ARD kernel. The weights learned in LS-SVM are plotted in cyan (gray) and correspond to the left *y*-axis, whereas the weights learned in BFD are plotted in blue (dark) and correspond to the right *y*-axis. We report weight values for BFD while for LS-SVM, 'ranking' coefficients. These coefficients are related to the weights Θ_{ard}. Ideally we would expect to see a perfect match between feature values of BFD and rankings in LS-SVM. As it is shown in Table 3, BFD performed better in Ringnorm while LS-SVM did better in Splice.

8.2.1 HISTOGRAMS OF PROJECTED DATA

A good way to visualize whether an FLD-based algorithm is performing adequately consists of generating histograms of projected data. We briefly compare the output distributions generated by BFD and KFD on training and test instances of the Waveform and Twonorm data sets. We used the data produced from the experiments with an RBF kernel to generate Figures 10 and 11.

In the Figure 10, BFD produced very consistent outputs between training and test sets for both Twonorm and Waveform. In particular, it is encouraging to see that Twonorm projects very close to two Gaussian distributions because this data set comes from two multivariate Gaussians. Meanwhile, in Figure 11, KFD produced very consistent output distributions in Twonorm but failed to do so in Waveform. Following similar arguments to those of Mika (2002), we believe this is one of the reasons for BFD performing better than KFD, in terms of AUC.

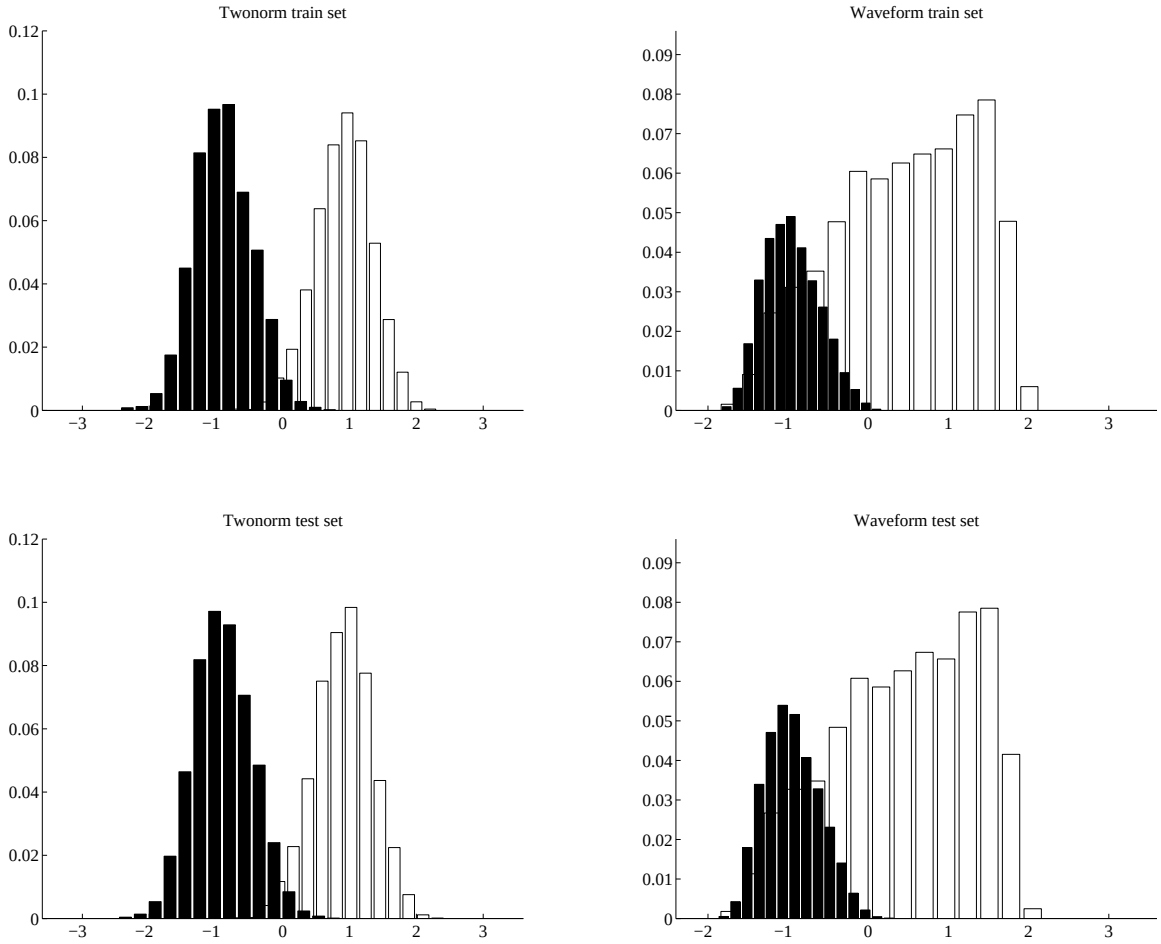

Figure 10: Comparison of output distributions on training and test sets for BFD. The data sets depicted are Twonorm and Waveform, respectively. It is clearly observable that training and test set distributions for BFD are quite consistent.

9. Conclusions and Future Work

We have presented a Bayesian probabilistic approach to discriminant analysis that can correspond to kernel Fisher's discriminant. Regularisation of the discriminant arises naturally in the proposed framework and through maximisation of the marginal likelihood we were able to determine kernel

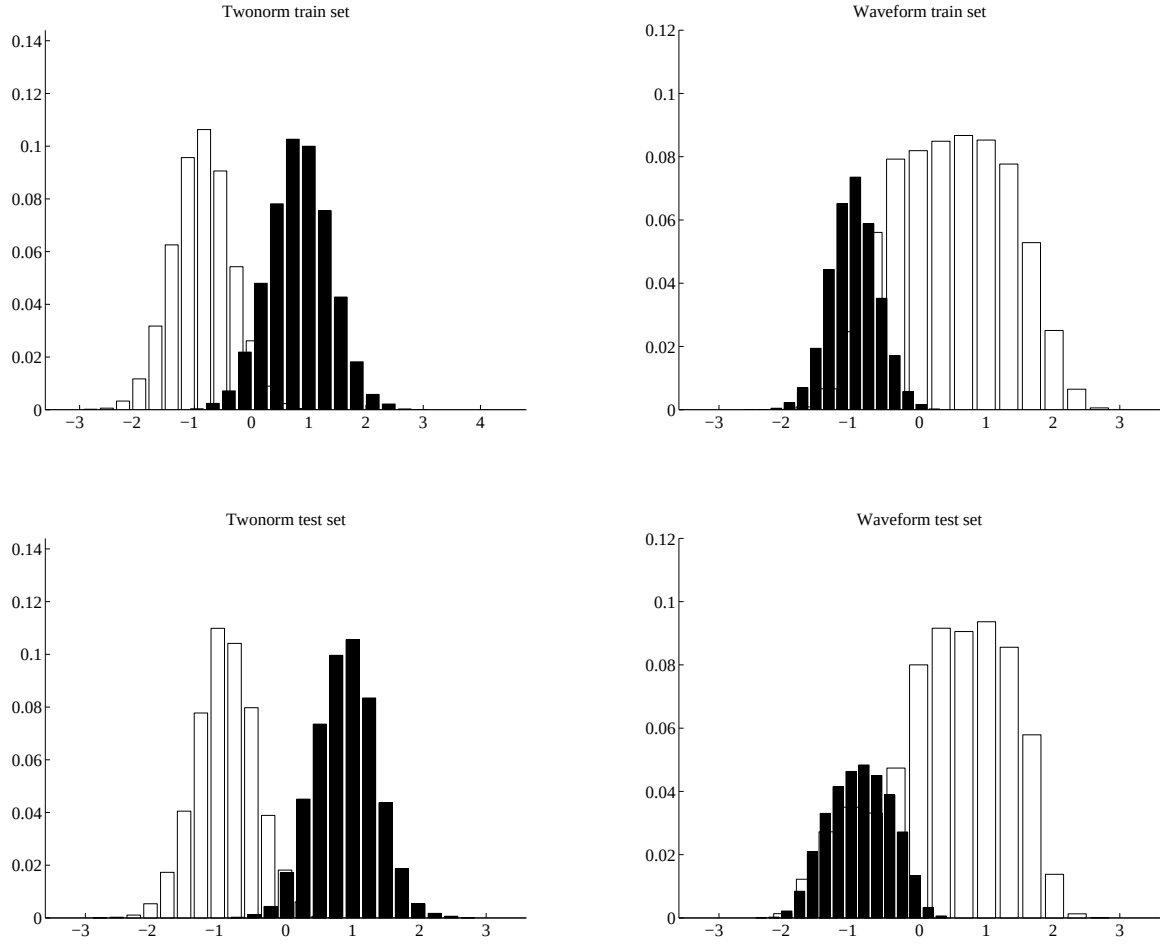

Figure 11: Comparison of output distributions on training and test sets for KFD. The data sets depicted are Twonorm and Waveform. Observe that training and test distributions for Waveform are noticeably different; this might explain KFD's lower classification performance if compared to BFD, see Table 2.

parameters. This paper has established the theoretical foundations of the approach and has shown that for a range of simple toy problems the methodology does discover sensible kernel parameters. The optimisation is only guaranteed to find a local minimum and therefore the quality of the solution can be sensitive to the initialisation. We performed experiments on real world data obtaining results which are competitive with the state of the art, moreover, we were able to do some relevance determination on the data set features.

Future directions of this work will be centred on sparsifying the kernel matrix. We intend to adapt the informative vector machine model to our framework (Lawrence et al., 2003). This should make larger data sets practical because at present, we are restricted by the $O\left(N^3\right)$ complexity associated with inverting the kernel matrix. Another direction of research will consist of allowing

the model to learn in the presence of label noise, building on work by Lawrence and Schölkopf (2001).

Acknowledgements

Both authors gratefully acknowledge support from the EPSRC grant number GR/R84801/01 'Learning Classifiers from Sloppily Labelled Data'. T.P.C. wishes to thank the continuous support provided by Banco de México and helpful discussions with Guido Sanguinetti and Gavin C. Cawley. Finally, we thank Jonathan Laidler for comments on the final version of this manuscript.

Appendix A. Weight Space Approach

In order to derive the distribution of $\mathbf{w}$ under the constraint d, we first realise that the combination of $p(\mathbf{w}|\mathcal{D})$ and $p(d|\mathcal{D},\mathbf{w},\gamma)$ yields a Gaussian distribution. Therefore, after conditioning on d, the resulting distribution will be Gaussian with the form $p(\mathbf{w}|\mathcal{D},d,\gamma) = \lim_{\gamma\to\infty} \mathcal{N}(\bar{\mathbf{w}},\Sigma)$ and parameters

$$\bar{\mathbf{w}} = \lim_{\gamma\to\infty} \gamma d \Sigma \Delta\mathbf{m} \tag{34}$$

and

$$\Sigma = \lim_{\gamma\to\infty} \left(\mathbf{B} + \gamma\Delta\mathbf{m}\Delta\mathbf{m}^T\right)^{-1}. \tag{35}$$

Inversion of Σ through Morrison-Woodbury formula allows us to take the limit, such as is shown below

$$\Sigma = \lim_{\gamma\to\infty} \left(\mathbf{B}^{-1} - \frac{\mathbf{B}^{-1}\Delta\mathbf{m}\Delta\mathbf{m}^T\mathbf{B}^{-1}}{\gamma^{-1} + \Delta\mathbf{m}^T\mathbf{B}^{-1}\Delta\mathbf{m}}\right),$$

hence

$$\Sigma = \mathbf{B}^{-1} - \frac{\mathbf{B}^{-1}\Delta\mathbf{m}\Delta\mathbf{m}^T\mathbf{B}^{-1}}{\Delta\mathbf{m}^T\mathbf{B}^{-1}\Delta\mathbf{m}}.$$

The mean $\bar{\mathbf{w}}$ can be obtained by substituting Equation 35 (without evaluating the limit) into Equation 34. Then, the application of Morrison-Woodbury formula and some extra manipulations will lead to a form suitable for taking the limit.

Appendix B. Expressing $p\left(\hat{\mathbf{t}}\middle|\mathbf{f},\mathbf{y}\right)$ in terms of f and L

Disregarding an additive constant, the log of the modified noise model $p\left(\hat{\mathbf{t}}\middle|\mathbf{f},\mathbf{y}\right)$ is

$$\mathcal{L}(\hat{c}_0,\hat{c}_1) = -\frac{\beta}{2}\sum_{n=1}^{N}\left[y_n(\hat{c}_1 - f_n)^2 + (1-y_n)(\hat{c}_0 - f_n)^2\right],$$

where we have used Equation 6 as base. From Equation 9, we substitute the values of each estimate $\hat{c}_q$ so that

$$\begin{aligned}\mathcal{L} &= -\frac{\beta}{2}\sum_{n=1}^{N}\left[y_n\left(\frac{1}{N_1}\mathbf{y}_1^T\mathbf{f}-f_n\right)^2+(1-y_n)\left(\frac{1}{N_0}\mathbf{y}_0^T\mathbf{f}-f_n\right)^2\right]\\ &= -\frac{\beta}{2}\left(\mathbf{f}\mathbf{f}^T-\frac{1}{N_1}\mathbf{y}_1^T\mathbf{f}\mathbf{f}^T\mathbf{y}_1-\frac{1}{N_0}\mathbf{y}_0^T\mathbf{f}\mathbf{f}^T\mathbf{y}_0\right)\\ &= -\frac{\beta}{2}\left(\mathbf{f}^T\mathbf{L}\mathbf{f}\right).\end{aligned}$$

Appendix C. Gaussian Process Approach

In this section we find the parameters of a new projected data point, namely its mean $\bar{f}^\star$ and variance $(\sigma^\star)^2$, as specified in Equations 21 and 22. In the model we have three types of distributions: $p\left(\hat{\mathbf{t}}\middle|\mathbf{y},\mathbf{f}\right)$, the noise model; $p(\mathbf{f}_+)$, an extended GP prior that includes the point $f^\star$ and the constraint $p(d|\mathbf{y},\mathbf{f},\gamma)$. In order to derive the distribution $p(f^\star|\mathcal{D},d,\gamma)$ we first compute the joint distribution

$$p\left(\mathbf{f}_+,\hat{\mathbf{t}},d\middle|\mathbf{y},\gamma\right)=p\left(\hat{\mathbf{t}}\middle|\mathbf{y},\mathbf{f}\right)p(d|\mathbf{y},\mathbf{f},\gamma)\,p(\mathbf{f}_+),$$

and then take advantage of the separability of $\mathbf{f}_+$ into $\left[\mathbf{f}^T,f^\star\right]^T$ to be able to marginalise the latent variables $\mathbf{f}$, which are associated with the training set. In other words we do

$$p\left(f^\star,\hat{\mathbf{t}},d\middle|\mathbf{y},\gamma\right) = \int p\left(f^\star,\mathbf{f},\hat{\mathbf{t}},d\middle|\mathbf{y},\gamma\right)\partial\mathbf{f} \tag{36}$$

The rest of the process consists of conditioning $f^\star$ on the targets $\hat{\mathbf{t}}$ and the distance d and on taking the limit $\gamma\to\infty$. In the remaining part of this section we detail this process.

C.1 Derivations

Grouping Equations 18, 19 and 20 gives

$$p(\mathbf{f}_+,\hat{\mathbf{t}},d\big|\mathbf{y},\gamma) \propto \exp\left\{-\frac{\beta}{2}\mathbf{f}^T\mathbf{L}\mathbf{f}-\frac{1}{2}\mathbf{f}_+\mathbf{K}_+^{-1}\mathbf{f}_+-\frac{\gamma}{2}\left(d-\mathbf{f}^T\Delta\hat{\mathbf{y}}\right)^2\right\}.$$

The idea consists of expanding and collecting terms in $\mathbf{f}$ and $f^\star$. In order to do so, we partition the inverse of the extended kernel by making $\mathbf{K}_+^{-1}=\begin{bmatrix}\mathbf{C} & \mathbf{c}\\ \mathbf{c}^T & c_\star\end{bmatrix}$. Thus we know that the product

$$\mathbf{f}_+^T\mathbf{K}_+^{-1}\mathbf{f}_+ = \mathbf{f}^T\mathbf{C}\mathbf{f}+2f^\star\mathbf{c}^T\mathbf{f}+c_\star(f^\star)^2.$$

Hence we get

$$p(f^\star,\mathbf{f},\hat{\mathbf{t}},d\big|\mathbf{y},\gamma) \propto \exp\left\{-\tfrac{1}{2}\mathbf{f}^T\mathbf{Q}\mathbf{f}-\left(f^\star\mathbf{c}-\gamma d\Delta\hat{\mathbf{y}}\right)^T\mathbf{f}-\tfrac{1}{2}c_\star(f^\star)^2\right\} \tag{37}$$

with

$$\mathbf{Q}=\left(\beta\mathbf{L}+\mathbf{C}+\gamma\Delta\hat{\mathbf{y}}\Delta\hat{\mathbf{y}}^T\right). \tag{38}$$

MARGINALISING LATENT VARIABLES

We are now in position to determine the distribution for a new mapping $f^\star$. The marginalisation of $\mathbf{f}$ is done by computing the integral of Equation 36 using as integrand the expression in (37). First we observe that the term $\left(-\frac{1}{2}c_\star (f^\star)^2\right)$ will be required when computing the distribution over $f^\star$ and when taking the limit, so it will be kept apart from the integral and used at a later stage.

The integral in Equation 36 is of exponential form, so we know how to solve it in a straightforward way. See below that

$$\int \exp\left(-\frac{1}{2}\mathbf{f}^T\mathbf{Q}\mathbf{f}+\mathbf{h}^T\mathbf{f}\right)\partial\mathbf{f} \propto \exp\left(\frac{1}{2}\mathbf{h}^T\mathbf{Q}^{-1}\mathbf{h}\right),$$

where we recognise that

$$\mathbf{h} \quad = \quad -(f^\star\mathbf{c}-\gamma d\Delta\hat{\mathbf{y}})^T \ .$$

Therefore the result, after incorporating $\left(-\frac{1}{2}c_\star (f^\star)^2\right)$, is

$$p(\, f^\star, \hat{\mathbf{t}}, d \,|\, \mathbf{y}, \gamma) \quad \propto \quad \exp\left\{\frac{1}{2}\mathbf{h}^T\mathbf{Q}^{-1}\mathbf{h}-\frac{1}{2}c_\star (f^\star)^2\right\} \ . \tag{39}$$

OBTAINING CONDITIONAL DISTRIBUTION

The distribution $p(\, f^\star | \, \mathcal{D}, d, \gamma)$, with $\mathcal{D} = \left(\hat{\mathbf{t}}, \mathbf{y}\right)$, is obtained by conditioning the expression in (39) on $\hat{\mathbf{t}}$ and d. Therefore we will work with the argument inside the *exponential* of (39) and group terms in $f^\star$, ignoring the rest. We begin by substituting $\mathbf{h}$ and expanding

$$\frac{1}{2}(f^\star\mathbf{c}-\gamma d\Delta\hat{\mathbf{y}})^T\mathbf{Q}^{-1}(f^\star\mathbf{c}-\gamma d\Delta\hat{\mathbf{y}})-\frac{1}{2}c_\star (f^\star)^2 =$$

$$-\frac{1}{2}\left(c_\star-\mathbf{c}^T\mathbf{Q}^{-1}\mathbf{c}\right)\left[(f^\star)^2+\frac{2\gamma d\mathbf{c}^T\mathbf{Q}^{-1}\Delta\hat{\mathbf{y}}}{c_\star-\mathbf{c}^T\mathbf{Q}^{-1}\mathbf{c}}f^\star\right]+\frac{1}{2}\left[(\gamma d)^2\Delta\hat{\mathbf{y}}^T\mathbf{Q}^{-1}\Delta\hat{\mathbf{y}}\right].$$

Completing the squares on $f^\star$ gives a Gaussian

$$p(\, f^\star | \, \mathcal{D}, d, \gamma) \quad \propto \quad \exp\left\{-\tfrac{1}{2(\sigma^\star)^2}\left(f^\star-\bar{f^\star}\right)^2\right\}$$

where the variance is

$$(\sigma^\star)^2 = \lim_{\gamma\to\infty}\left(c_\star-\mathbf{c}^T\mathbf{Q}^{-1}\mathbf{c}\right)^{-1} \tag{40}$$

and the mean,

$$\bar{f^\star} = -\lim_{\gamma\to\infty}\gamma d\,(\sigma^\star)^2\,\mathbf{c}^T\mathbf{Q}^{-1}\Delta\hat{\mathbf{y}}. \tag{41}$$

WORKING OUT $(\sigma^\star)^2$

We now determine the limit $\gamma\to\infty$ in Equation 40. First, we express each of the terms that form the inverse of the kernel matrix:

$$c_\star = \left(k_\star-\mathbf{k}^T\mathbf{K}^{-1}\mathbf{k}\right)^{-1}, \tag{42}$$

$$\mathbf{c} = -c_\star \mathbf{K}^{-1}\mathbf{k}, \tag{43}$$

and

$$\mathbf{C} = \mathbf{K}^{-1} + c_\star \mathbf{K}^{-1}\mathbf{k}\mathbf{k}^T\mathbf{K}^{-1}. \tag{44}$$

Partitioning the kernel matrix implies that

$$\mathbf{K}_+ \quad = \quad \begin{pmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^T & k_\star \end{pmatrix}.$$

Substituting (38) and (43) into Equation 40 gives

$$(\sigma^\star)^2 = \lim_{\gamma\to\infty} \left(c_\star - c_\star^2 \mathbf{k}^T \left[\beta\mathbf{KLK} + \gamma\mathbf{K}\Delta\hat{\mathbf{y}}\Delta\hat{\mathbf{y}}^T\mathbf{K} + \mathbf{KCK} \right]^{-1} \mathbf{k} \right)^{-1}.$$

The product **KCK** can be worked out by using Equation 44. Therefore

$$(\sigma^\star)^2 = \lim_{\gamma\to\infty} \left(c_\star - c_\star^2 \mathbf{k}^T \left[\beta\mathbf{KLK} + \gamma\mathbf{K}\Delta\hat{\mathbf{y}}\Delta\hat{\mathbf{y}}^T\mathbf{K} + \mathbf{K} + c_\star\mathbf{k}\mathbf{k}^T \right]^{-1} \mathbf{k} \right)^{-1}.$$

Defining

$$\mathbf{D}_\gamma = \gamma\mathbf{K}\Delta\hat{\mathbf{y}}\Delta\hat{\mathbf{y}}^T\mathbf{K} + \mathbf{A}, \tag{45}$$

with $\mathbf{A} = \beta\mathbf{KLK} + \mathbf{K}$ leads to

$$\begin{aligned} (\sigma^\star)^2 &= \lim_{\gamma\to\infty} \left[c_\star - c_\star^2 \mathbf{k}^T \left(\mathbf{D}_\gamma + c_\star \mathbf{k}\mathbf{k}^T \right)^{-1} \mathbf{k} \right]^{-1}, \\ &= c_\star^{-1} + \mathbf{k}^T \mathbf{D}_\gamma^{-1} \mathbf{k}. \end{aligned}$$

Using Equation 42, we arrive to an expression for which $(\sigma^\star)^2$ only depends on γ by the term $\mathbf{D}_\gamma$,

$$(\sigma^\star)^2 = \lim_{\gamma\to\infty} \left[k_\star - \mathbf{k}^T \left(\mathbf{K}^{-1} - \mathbf{D}_\gamma^{-1} \right) \mathbf{k} \right]. \tag{46}$$

Working with $\mathbf{D}_\gamma^{-1}$ Inversion of $\mathbf{D}_\gamma$ (Equation 45) through the Morrison-Woodbury lemma allows us to obtain **D** by taking the limit. See below.

$$\mathbf{D}_\gamma^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{K}\Delta\hat{\mathbf{y}} \left(\frac{1}{\gamma} + \Delta\hat{\mathbf{y}}^T\mathbf{K}\mathbf{A}^{-1}\mathbf{K}\Delta\hat{\mathbf{y}} \right)^{-1} \Delta\hat{\mathbf{y}}^T\mathbf{K}\mathbf{A}^{-1}.$$

Therefore, by taking $\gamma \to \infty$ we obtain

$$\mathbf{D} = \left(\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{K}\Delta\hat{\mathbf{y}} \left(\Delta\hat{\mathbf{y}}^T\mathbf{K}\mathbf{A}^{-1}\mathbf{K}\Delta\hat{\mathbf{y}} \right)^{-1} \Delta\hat{\mathbf{y}}^T\mathbf{K}\mathbf{A}^{-1} \right)^{-1}.$$

Substituting this expression into (46) gives the desired result,

$$(\sigma^\star)^2 = k_\star - \mathbf{k}^T \left(\mathbf{K}^{-1} - \mathbf{D}^{-1} \right) \mathbf{k}.$$

WORKING OUT THE MEAN

Substituting Equations 38 and 43, the values of $\mathbf{Q}$ and $\mathbf{c}$, into Equation 41 leads to

$$\bar{f}^{\star} = \lim_{\gamma\to\infty} \gamma d \left(\sigma^{\star}\right)^2 c_{\star} \mathbf{k}^T \left(\mathbf{D}_{\gamma} + c_{\star}\mathbf{k}\mathbf{k}^T\right)^{-1} \mathbf{K}\Delta\hat{\mathbf{y}}.$$

Inverting the matrix $\left(\mathbf{D}_{\gamma} + c_{\star}\mathbf{k}\mathbf{k}^T\right)$ and substituting the value of $c_{\star}$ gives

$$\bar{f}^{\star} = \lim_{\gamma\to\infty} \gamma d \left(\sigma^{\star}\right)^2 \left[k_{\star} - \mathbf{k}^T \left(\mathbf{K}^{-1} - \mathbf{D}_{\gamma}^{-1}\right)\mathbf{k}\right]^{-1} \mathbf{k}^T \mathbf{D}_{\gamma}^{-1} \mathbf{K}\Delta\hat{\mathbf{y}}.$$

Using (46) implies that

$$\bar{f}_{\star} = \lim_{\gamma\to\infty} \gamma d \mathbf{k}^T \mathbf{D}_{\gamma}^{-1} \mathbf{K}\Delta\bar{\mathbf{y}}.$$

Substituting (45) and inverting gives

$$\bar{f}^{\star} = \lim_{\gamma\to\infty} d \left(\frac{1}{\gamma} + \Delta\hat{\mathbf{y}}^T \mathbf{K}\mathbf{A}^{-1}\mathbf{K}\Delta\hat{\mathbf{y}}\right)^{-1} \mathbf{k}^T \mathbf{A}^{-1}\mathbf{K}\Delta\hat{\mathbf{y}}.$$

Taking the limit gives the desired result

$$\bar{f}_{\star} = \frac{d\mathbf{k}^T \mathbf{A}^{-1}\mathbf{K}\Delta\hat{\mathbf{y}}}{\Delta\hat{\mathbf{y}}^T \mathbf{K}\mathbf{A}^{-1}\mathbf{K}\Delta\hat{\mathbf{y}}}.$$

Appendix D. Obtaining MAP Solution for β

Making

$$\begin{aligned} V &= \Sigma_{n=1}^{N} y_n \left(\hat{c}_1 - f_n\right)^2 + \Sigma_{n=1}^{N} \left(1 - y_n\right) \left(\hat{c}_0 - f_n\right)^2, \\ &= \sigma_1^2 + \sigma_0^2. \end{aligned}$$

the modified noise model[9] becomes

$$p\left(\hat{\mathbf{t}}|\mathbf{f},\beta\right) = \frac{\beta^{N/2}}{(2\pi)^{N/2}} \exp\left\{-\frac{\beta}{2}V\right\}.$$

Then combining it with a gamma prior $G\left(\beta|a,b\right)$ gives a posterior of the form $G\left(\beta|N/2+a,(V/2+b)\right)$, that is

$$p\left(\beta|\hat{\mathbf{t}},\mathbf{f}\right) \propto \beta^{N/2+a-1} \exp\left\{-\beta\left(\frac{V}{2}+b\right)\right\}.$$

Taking the derivative of the log of this distribution, equating the result to zero and solving will give Equation 27.

Appendix E. Experimental Setup

In this section we give more details on the way we carried out our experiments on toy and real data.

9. Use Equation 6 and substitute the class centres c_q by their estimates $\hat{c}_q$.

E.1 Toy Data

In all our experiments with synthetic data we used an RBF kernel of the form

$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 \exp\left(-\frac{\theta_2}{2}\left|\left|\mathbf{x}^i - \mathbf{x}^j\right|\right|^2\right) + \theta_3 \delta_{ij}. \tag{47}$$

For KFD and LS-SVM we worked with the bandwidth of the kernel $\sigma = 1/\theta_2$, whereas for BFD we used θ_2 itself. The nugget parameter θ_3 ensures that $\mathbf{K}$ can be inverted at all times.

Regarding model training, we used the matlab implementation of Baudat and Anouar (2000) to solve the generalized eigenvalue problem, see their function `BuildGDA`. Furthermore, we used the function `crossvalidate`, provided by Pelckmans et al. (2003) and Suykens et al. (2002), to cross-validate the values of σ and of the 'threshold' of the mininum accepted eigenvalue. The latter was used instead of the regularisation coefficient because of the way the eigenvalue problem is solved, see Baudat's implementation for more details. In the LS-SVM case, we used the toolbox LS-SVMlab of Pelckmans et al. (2003) and Suykens et al. (2002) to do the classifications. The values of σ and C were cross-validated, with the latter being the coefficient associated with the support vector formulation. Lastly, in BFD, we used our own implementation which is available at

`http://www.dcs.shef.ac.uk/~neil/bfd`.

In this case, the function `scg` provided in Netlab's toolbox (Nabney, 2002) was used to adapt kernel parameters.

E.2 Benchmark Data Sets

In the BFD experiments with an RBF kernel, we used $N_x = 8$ different initialisations of the parameter θ_2 (Equation 47) during the training phase. The initialisations selected ranged from 1×10^{-4} to 1×10^4; initialisations that produced numerical errors were ignored. We trained on the first 5 realisations (partitions) of each data set and computed their marginal loglikelihood. In this way, an array of $N_x \times 5$ elements was obtained. See below.

$$\begin{array}{cccc} \mathit{Init}. & \mathit{Part}.1 & \dots & \mathit{Part}.5 \\ \theta_2^1 & \Theta_t^{(11)} & \dots & \Theta_t^{(15)} \\ \vdots & \vdots & \vdots & \vdots \\ \theta_2^{N_x} & \Theta_t^{(N_x 1)} & \dots & \Theta_t^{(N_x 5)} \end{array}$$

For each partition, we selected the vector $\Theta_t^{(ip)}$ with highest associated marginal likelihood, with $p \in [1,5]$ and $i \in [1,8]$. Hence the original array of 40 elements Θ_t^{ip} was reduced to an array of 5 elements. The final vector of parameters was determined by extracting the trained values θ_2 from each element of the reduced array and taking the median over them. The selected vector Θ_t^{sel} was the one associated with the median value of θ_2.

For the ARD experiments, we changed $N_x = 3$. The initialisations were given by $\left[1 \times 10^{-3}, 1, 10\right]$.

The Bayesian LS-SVM experiments were carried out using the toolbox LS-SVMLab (see Suykens et al., 2002). We trained LS-SVM's on the first five realisations of the training data and selected the median of the parameters. We observed that this algorithm was susceptible to fall into local minima so in order to avoid this problem, we used the function `bay_initlssvm` to have good initialisations. The ARD rankings were obtained by applying the function `bay_lssvmARD`.

References

Deepak K. Agarwal. Shrinkage estimator generalizations of proximal support vector machines. In *KDD '02: Proceedings of the eighth ACM SIGKDD International conference on Knowledge Discovery and Data Mining*, pages 173–182, New York, NY, USA, 2002. ACM Press.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, May 1950.

Gaston Baudat and Fatiha Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000.

Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39:1–38, 1977.

Richard O. Duda and Peter E. Hart. *Pattern Recognition and Scene Analysis*. John Wiley, 1973.

Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179, 1936.

Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press Inc., Boston, Massachusetts, 2nd edition, 1990.

Glenn Fung and Olvi L. Mangasarian. Proximal support vector machine classifiers. In *KDD '01: Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 77–86, New York, NY, USA, 2001. ACM Press.

Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, USA, 3rd edition, 1996.

Robert B. Gramacy and Herbert K. H. Lee. Gaussian processes and limiting linear models. Technical Report ams2005-01, Department of Applied Mathematics and Statistics, University of California, Santa Cruz., 2005.

Ian T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1st edition, 1986.

Kevin J. Lang and Michael J. Witbrock. Learning to tell two spirals apart. In David S. Touretzky, Geoff E. Hinton, and Terrence J. Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School*. Morgan Kauffman, 1988.

Neil D. Lawrence and Bernhard Schölkopf. Estimating a kernel Fisher discriminant in the presence of label noise. In Carla E. Brodley and Andrea P. Danyluk, editors, *Proceedings of the 18th International Conference on Machine Learning*, Williamstown, MA, July 2001. Morgan Kauffman.

Neil D. Lawrence, Matthias Seeger, and Ralf Herbrich. Fast sparse Gaussian process methods: the informative vector machine. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 609–616, Cambridge, MA, 2003. MIT Press.

David J. C. Mackay. Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3):469–505, 1995.

Donald Michie, David J. Spiegelhalter, and Charles C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.

Sebastian Mika. A mathematical approach to kernel Fisher algorithm. In Todd K. Leen and Thomas G. Dietterich Völker Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 591–597, Cambridge, MA, 2001. MIT Press.

Sebastian Mika. *Kernel Fisher Discriminants*. PhD thesis, Technischen Universität, Berlin, Germany, 2002.

Sebastian Mika, Gunnar Rätsch, Jason Weston, Bernhard Schölkopf, and Klaus-Robert Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, E. Wilson J. Larsen, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

Ian T. Nabney. *Netlab: Algorithms for Pattern Recognition*. Springer-Verlag, 2002.

Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, 1996.

Anthony O'Hagan. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society, Series B (Methodological)*, 40(1):1–42, 1978.

Kristiaan Pelckmans, Johan A. K. Suykens, Tony Van Gestel, Jos De Brabanter, Lukas Lukas, Bart Hamers, Bart De Moor, and Joos Vandewalle. *LS-SVMlab Toolbox User's Guide*. Katholieke Universiteit, Leuven. ESAT-SCD-SISTA, 2003.

Gunnar Rätsch, Takashi Onoda, and Klaus-Robert Müller. Soft margins for AdaBoost. Technical Report NC-TR-98-021, Royal Holloway College, University of London, U. K., 1998.

Brian D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

Volker Roth. Outlier detection with one-class kernel Fisher discriminants. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1169–1176, Cambridge, MA, 2005. MIT Press.

David Ruppert, Matthew P. Wand, and Raymond J. Carroll. *Semiparametric Regression*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, U. K., 2003.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.

Johan A. K. Suykens, Tony Van Gestel, Jos De Brabanter, Bart De Moor, and Joos Vandewalle. *Least Squares Support Vector Machines*. World Scientific, 2002.

Johan A. K. Suykens and Joos Vandewalle. Least squares support vector machines. *Neural Processing Letters*, 9(3):293–300, 1999.

Tony Van Gestel, Johan A. K. Suykens, Gert Lanckriet, Annemie Lambrechts, Bart de Moor, and Joos Vandewalle. Bayesian framework for least squares support vector machine classifiers, Gaussian processes and kernel discriminant analysis. *Neural Computation*, 14(5):1115–1147, 2002.

Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

Christopher K. I. Williams. Prediction with Gaussian processes: from linear regression to linear prediction and beyond. In Michael I. Jordan, editor, *Learning in Graphical Models*, D, Behavioural and social sciences 11. Kluwer, Dordrecht, The Netherlands, 1999.