Manuscript no. 2281

Bayesian Framework for Least Squares Support Vector Machine Classifiers, Gaussian Processes and Kernel Fisher Discriminant Analysis

Tony Van Gestel, Johan A.K. Suykens[*], Gert Lanckriet, Annemie Lambrechts, Bart De Moor and Joos Vandewalle

Katholieke Universiteit Leuven
Dept. of Electrical Engineering ESAT-SISTA
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
Tel. +32/16/32.17.09, Fax. +32/16/32.19.70
E-mail: {tony.vangestel, johan.suykens}@esat.kuleuven.ac.be

August 2001

[*]Corresponding author.

Abstract

The Bayesian evidence framework has been successfully applied to the design of multilayer perceptrons (MLPs) in the work of MacKay. Nevertheless, the training of MLPs suffers from drawbacks like the non-convex optimization problem and the choice of the number of hidden units. In Support Vector Machines (SVMs) for classification, as introduced by Vapnik, a nonlinear decision boundary is obtained by mapping the input vector first in a nonlinear way to a high dimensional kernel-induced feature space in which a linear large margin classifier is constructed. Practical expressions are formulated in the dual space in terms of the related kernel function and the solution follows from a (convex) quadratic programming (QP) problem. In Least Squares SVMs (LS-SVMs), the SVM problem formulation is modified by introducing a least squares cost function and equality instead of inequality constraints and the solution follows from a linear system in the dual space. Implicitly, the least squares formulation corresponds to a regression formulation and is also related to kernel Fisher Discriminant Analysis. The least squares regression formulation has advantages for deriving analytic expressions in a Bayesian evidence framework in contrast with the classification formulations, e.g., used in Gaussian Processes (GPs). The LS-SVM formulation has clear primal-dual interpretations and without the bias term one explicitly constructs a model which yields the same expressions as have been obtained with Gaussian Processes (GPs) for regression. In this paper, the Bayesian evidence framework is combined with the LS-SVM classifier formulation. Starting from the feature space formulation, analytic expressions are obtained in the dual space on the different levels of Bayesian inference, while posterior class probabilities are obtained by marginalizing over the model parameters. Empirical results obtained on ten public domain datasets show that the LS-SVM classifier designed within the Bayesian evidence framework consistently yields good generalization performances.

1 Introduction

Bayesian probability theory provides a unifying framework to find models that are well-matched to the data and to use these models for making optimal decisions. Multilayer perceptrons (MLPs) are popular nonlinear parametric models for both regression and classification. In (MacKay, 1992; 1995; 1999) the evidence framework was successfully applied to the training of MLPs using three levels of Bayesian inference: the model parameters, regularization hyperparameters and network structure are inferred on the first, second and third level, respectively. The moderated output is obtained by marginalizing over the model- and hyperparameters using a Laplace approximation in a local optimum.

Whereas MLPs are flexible nonlinear parametric models that can approximate any continuous nonlinear function over a compact interval (Bishop, 1995), the training of an MLP suffers from drawbacks like the non-convex optimization problem and the choice of the number of hidden units. In Support Vector Machines (SVMs) the classification problem is formulated and represented as a convex Quadratic Programming (QP) problem (Cristianini & Shawe-Taylor, 2000; Vapnik, 1995; 1998). A key idea of the nonlinear SVM classifier is to map the inputs to a high dimensional feature space where the classes are assumed to be linearly separable. In this high dimensional space a large margin classifier is constructed. By applying the Mercer condition, the classifier is obtained by solving a finite dimensional QP problem in the dual space, which avoids the explicit knowledge of the high dimensional mapping and uses only the related kernel function. In (Suykens & Vandewalle, 1999), a Least Squares type of SVM classifier (LS-SVM) was introduced by modifying the problem formulation so as to obtain a linear set of equations in the dual space. This is done by taking a least squares cost function, with equality instead of inequality constraints.

The training of MLP classifiers is often done by using a regression approach with binary targets for solving the classification problem. This is also implicitly done in the LS-SVM formulation and has the advantage of deriving analytic expressions within a Bayesian evidence framework in contrast with classification approaches used, e.g., in GPs. Like in ordinary ridge regression (Brown, 1977) no regularization is applied on the bias term in SVMs and LS-SVMs which results into a centering in the kernel-induced feature space and allows to relate the LS-SVM formulation to kernel Fisher discriminant analysis (Baudat & Anouar, 2000; Mika *at al.*, 2001). The corresponding eigenvalues of the centered kernel matrix are obtained from kernel PCA (Schölkopf *et al.*, 1998). When no bias term is used in the LS-SVM formulation, similar expressions are obtained as with kernel ridge regression and Gaussian Processes (GPs) for regression (Gibbs, 1997; Neal, 1997; Rasmussen, 1996; Williams, 1998). In this paper, a Bayesian framework is derived for the LS-SVM formulation starting from the SVM and LS-SVM feature space formulation, while the corresponding analytic expressions in the dual space are similar, up to the centering, to the expressions obtained for GP. The primal-dual interpretations and equality constraints of LS-SVMs have also allowed to extend the LS-SVM framework to recurrent networks and optimal control (Suykens & Vandewalle, 2000; Suykens *et al.*, 2001). The regression formulation allows to derive analytic expressions in order to infer the model-, hyper- and kernel parameters on the corresponding three levels of Bayesian inference, respectively. Posterior class probabilities of the LS-SVM classifier are obtained by marginalizing over the model parameters within the evidence framework.

This paper is organized as follows. In Section 2 links between kernel-based classification techniques are discussed. The three levels of inference are described in Sections 3, 4 and 5. The design strategy is explained in Section 6. Empirical results are discussed in Section 7.

2 Kernel-based Classification Techniques

Given a binary classification problem with classes $\mathcal{C}_+$ and $\mathcal{C}_-$, with corresponding class labels $y = \pm 1$, the classification task is to assign a class label to a given new input vector $x \in \mathbb{R}^n$. Applying Bayes' formula, one can calculate the posterior class probability:

$$P(y|x) = \frac{p(x|y)P(y)}{p(x)}, \tag{1}$$

where $P(y)$ is the (discrete) a priori probability of the classes and $p(x|y)$ is the (continuous) probability of observing x when corresponding to class label y. The denominator $p(x)$ follows from normalization. The class label is then assigned to the class with maximum posterior probability

$$y(x) = \text{sign}[g_0(x)] \triangleq \text{sign}[P(y{=}{+}1|x) - P(y{=}{-}1|x)] \tag{2}$$

or

$$\begin{aligned} y(x) &= \text{sign}[g_1(x)] \\ &\triangleq \text{sign}[\log(p(x|y{=}{+}1)P(y{=}{+}1)) - \log(p(x|y{=}{-}1)P(y{=}{-}1))]. \end{aligned} \tag{3}$$

Given $g_0(x)$ one obtains the posterior class probabilities from $P(y = +1|x) = \frac{1}{2}(1 + g_0(x))$ and $P(y = -1) = \frac{1}{2}(1 - g_0(x))$ (Duda & Hart, 1973).

When the densities $p(x|y = +1)$ and $p(x|y = -1)$ have a multivariate normal distribution with the same covariance matrix Σ and corresponding mean m_+ and m_-, respectively, the Bayesian decision rule (3) becomes the linear discriminant function

$$y(x) = \text{sign}[w^T x + b], \tag{4}$$

with $w = \Sigma^{-1}(m_+ - m_-)$ and $b = -w^T(m_+ + m_-)/2 + \log(P(y = +1)) - \log(P(y = -1))$ (Bishop, 1995; Duda & Hart, 1973).

In practice, the class covariance matrix Σ and the mean m_+ and m_- are not known and the linear classifier $w^T x + b$ has to be estimated from given data $D = \{(x_i, y_i)\}_{i=1}^N$ that consist of N_+ positive and N_- negative labels. The corresponding sets of indices with positive and negative labels are denoted by $\mathcal{I}_+$ and $\mathcal{I}_-$ with the full index set $\mathcal{I}$ equal to $\mathcal{I} = \mathcal{I}_+ \cup \mathcal{I}_- = \{1, \ldots, N\}$. Some well-known algorithms to estimate the discriminant vector w and bias term b are Fisher Discriminant Analysis, Support Vector Machine (SVM) classifier and a regression approach with binary targets $y_i = \pm 1$. However, when the class densities are not normally distributed with the same covariance matrix, the optimal decision boundary typically is no longer linear (Bishop, 1995; Duda & Hart, 1973). A nonlinear decision boundary in the input space can be obtained by applying the kernel trick: the input vector $x \in \mathbb{R}^n$ is mapped in a nonlinear way to the high (possibly infinite) dimensional feature vector $\varphi(x) \in \mathbb{R}^{n_f}$, where the nonlinear function $\varphi(\cdot)$: $\mathbb{R}^n \rightarrow \mathbb{R}^{n_f}$ is related to the symmetric, positive definite kernel function

$$K(x_1, x_2) = \varphi(x_1)^T \varphi(x_2) \tag{5}$$

from Mercer's theorem (Cristianini & Shawe-Taylor, 2000; Smola *et al.*, 1998; Vapnik, 1995; Vapnik, 1998). In this high dimensional feature space a linear separation is made. For the kernel function K one typically has the following choices: $K(x, x_i) = x_i^T x$ (linear kernel); $K(x, x_i) = (x_i^T x + 1)^d$ (polynomial kernel of degree $d \in \mathbb{N}$); $K(x, x_i) = \exp\{-\|x - x_i\|_2^2/\sigma^2\}$ (RBF kernel); $K(x, x_i) = \tanh(\kappa\, x_i^T x + \theta)$ (MLP kernel). Notice that the Mercer condition holds for all $\sigma \in \mathbb{R}$ and d values in the RBF, resp. the polynomial case, but not for all possible choices of $\kappa, \theta \in \mathbb{R}$ in the MLP case. Combinations of kernels can be obtained by stacking the corresponding feature vectors.

The classification problem now is assumed to be linear in the feature space and the classifier takes the form

$$y(x) = \mathrm{sign}[w^T \varphi(x) + b], \tag{6}$$

where w and b are obtained by applying the kernel version of the above-mentioned algorithms, where typically a regularization term $w^T w/2$ is introduced in order to avoid overfitting (large margin $2/w^T w$) in the high (and possibly infinite) dimensional feature space. On the other hand, the classifier (6) is never evaluated in this form and the Mercer condition (5) is applied instead. In the remainder of this Section the links between the different kernel-based classification algorithms are discussed.

2.1 SVM Classifiers

Given the training data $\{(x_i, y_i)\}_{i=1}^N$ with input data $x_i \in \mathbb{R}^n$ and corresponding binary class labels $y_i \in \{-1, +1\}$, the SVM classifier, according to Vapnik's original formulation (Vapnik, 1995; Vapnik, 1998), incorporates the following constraints ($i = 1, \ldots, N$):

$$\begin{cases} w^T\varphi(x_i) + b \geq +1, & \text{if} \quad y_i = +1 \\ w^T\varphi(x_i) + b \leq -1, & \text{if} \quad y_i = -1 \end{cases} \tag{7}$$

which is equivalent with $y_i[w^T\varphi(x_i) + b] \geq 1$, ($i = 1, \ldots, N$). The classification problem is formulated as follows:

$$\min_{w,b,\xi} \mathcal{J}_1(w, \xi) = \tfrac{1}{2} w^T w + C \textstyle\sum_{i=1}^N \xi_i \tag{8}$$

$$\text{subject to} \begin{cases} y_i[w^T\varphi(x_i) + b] \geq 1 - \xi_i, & i = 1, \ldots, N \\ \xi_i \geq 0, & i = 1, \ldots, N. \end{cases} \tag{9}$$

This optimization problem is solved in its dual form and the resulting classifier (6) is evaluated in its dual representation. The variables ξ_i are slack variables which are needed in order to allow misclassifications in the set of inequalities (e.g., due to overlapping distributions). The positive real constant C should be considered as a tuning parameter in the algorithm. More details on SVMs can be found in (Cristianini & Shawe-Taylor, 2000; Smola *et al.*, 1998; Vapnik, 1995; Vapnik, 1998). Observe that no regularization is applied on the bias term b.

2.2 LS-SVM Classifiers

In (Suykens & Vandewalle, 1999) the SVM classifier formulation was modified basically as follows:

$$\min_{w,b,e_c} \mathcal{J}_{2c}(w, e_c) = \frac{\mu}{2} w^T w + \frac{\zeta}{2} \sum_{i=1}^{N} e_{c,i}^2 \tag{10}$$

$$\text{subject to} \quad y_i \left[w^T \varphi(x_i) + b \right] = 1 - e_{c,i}, \qquad i = 1, \ldots, N. \tag{11}$$

Besides the quadratic cost function, an important difference with standard SVMs is that the formulation consists now of equality instead of inequality constraints.

The LS-SVM classifier formulation (10)-(11) implicitly corresponds to a regression interpretation with binary targets $y_i = \pm 1$. By multiplying the error $e_{c,i}$ with y_i and using $y_i^2 = 1$, the sum squared error term $\sum_{i=1}^{N} e_{c,i}^2$ becomes:

$$\sum_{i=1}^{N} e_{c,i}^2 = \sum_{i=1}^{N} (y_i e_{c,i})^2 = \sum_{i=1}^{N} e_i^2 = (y_i - (w^T \varphi(x) + b))^2, \tag{12}$$

with

$$e_i = y_i - (w^T \varphi(x) + b). \tag{13}$$

Hence, the LS-SVM classifier formulation is equivalent to

$$\mathcal{J}_2(w, b) = \mu E_W + \zeta E_D, \tag{14}$$

with

$$E_W = \frac{1}{2} w^T w, \tag{15}$$

$$E_D = \frac{1}{2} \sum_{i=1}^{N} e_i^2 = \frac{1}{2} \sum_{i=1}^{N} (y_i - \ [w^T \varphi(x_i) + b])^2. \tag{16}$$

Both μ and ζ should be considered as hyperparameters in order to tune the amount of regularization versus the sum squared error. The solution of (14) does only depend on the ratio $\gamma = \zeta/\mu$. Therefore, the original formulation

(Suykens & Vandewalle, 1999) used only γ as tuning parameter. The use of both parameters μ and ζ will become clear in the Bayesian interpretation of the LS-SVM cost function (14) in the next Sections. Observe that no regularization is applied to the bias term b, which is the preferred form for ordinary ridge regression (Brown, 1977).

The regression approach with binary targets is a common approach for training MLP classifiers and also for the simpler case of linear discriminant analysis (Bishop, 1995). Defining the MSE error between $w^T\varphi(x)+b$ and the Bayes discriminant $g_0(x)$ from (2)

$$\mathrm{MSE} = \int [w^T\varphi(x) + b - g_0(x)]^2 p(x)dx, \tag{17}$$

it has been shown (Duda & Hart, 1973) that minimization of E_D in (14) is asymptotically ($N \to \infty$) equivalent to minimizing (17). Hence, the regression formulation with binary targets yields asymptotically the best approximation to the Bayes discriminant (2) in least squares sense (Duda & Hart, 1973). Such an approximation typically gives good results, but may be suboptimal since the misclassification risk is not directly minimized.

The solution of the Least Squares Support Vector Machine (LS-SVM) regressor is obtained after constructing the Lagrangian $\mathcal{L}(w, b, e; \alpha) = \mathcal{J}_2(w, e) - \sum_{i=1}^N \alpha_i\{y_i - [w^T\varphi(x_i) + b] - e_i\}$, where $\alpha_i \in \mathbb{R}$ are the Lagrange multipliers. The conditions for optimality are:

$$\left\{ \begin{array}{lllr} \frac{\partial \mathcal{L}}{\partial w} = 0 & \rightarrow & w = \sum_{i=1}^N \alpha_i \varphi(x_i) & \\ \frac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow & \sum_{i=1}^N \alpha_i = 0 & \\ \frac{\partial \mathcal{L}}{\partial e_i} = 0 & \rightarrow & \alpha_i = \gamma e_i, & i = 1, \ldots, N \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 & \rightarrow & y_i = w^T\varphi(x_i) + b + e_i = 0, & i = 1, \ldots, N. \end{array} \right. \tag{18}$$

As in standard SVMs, we never calculate w nor $\varphi(x_i)$. Therefore, we eliminate w and e yielding a linear Karush-Kuhn-Tucker system instead of a QP problem:

$$\left[\begin{array}{c|c} 0 & 1_v^T \\ \hline 1_v & \Omega + \gamma^{-1} I_N \end{array} \right] \left[\begin{array}{c} b \\ \hline \alpha \end{array} \right] = \left[\begin{array}{c} 0 \\ \hline Y \end{array} \right] \tag{19}$$

with

$$Y = [y_1; \ldots; y_N], \quad 1_v = [1; \ldots; 1], \quad e = [e_1; \ldots; e_N], \quad \alpha = [\alpha_1; \ldots; \alpha_N], \tag{20}$$

and where Mercer's condition (5) is applied within the kernel matrix $\Omega \in \mathbb{R}^{N\times N}$

$$\Omega_{ij} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j). \tag{21}$$

The LS-SVM classifier is then constructed as follows:

$$y(x) \quad = \quad \text{sign}[\textstyle\sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b]. \tag{22}$$

In numerical linear algebra, efficient algorithms exist for solving large scale linear systems (Golub & Van Loan, 1989). The system (19) can be reformulated into two linear systems with positive definite data matrices, so as to apply iterative methods such as the Hestenes-Stiefel conjugate gradient algorithm (Suykens, 2000). LS-SVM classifiers can be extended to multiple classes by defining additional output variables. Although sparseness is lost due to the use of a 2-norm, a sparse approximation of the LS-SVM can be obtained by sequentially pruning the support value spectrum (Suykens, 2000), without loss of generalization performance.

2.3 Gaussian Processes for Regression

When one uses no bias term b in the regression formulation (Cristianini & Shawe-Taylor, 2000; Saunders *et al.*, 1998) the support values α^* are obtained from the linear system

$$(\Omega + \tfrac{\mu}{\zeta} I_N)\alpha^* = Y. \tag{23}$$

The output of the LS-SVM regressor for a new input x is given by

$$\hat{y}(x) = \textstyle\sum_{i=1}^{N} \alpha_i^* K(x, x_i) = \theta(x)^T \alpha^*, \tag{24}$$

with $\theta(x) = [K(x, x_1); ...; K(x, x_N)]$. For classification purposes one can use the interpretation of an optimal least squares approximation (17) to the Bayesian decision rule and the class label is assigned as follows $y = \text{sign}[\hat{y}(x)]$.

Observe that the result of (24) is equivalent with the Gaussian Process (GP) formulation (Gibbs, 1997; Neal, 1997; Rasmussen, 1996; Sollich, 2000; Williams, 1998; Williams & Barber, 1998) for regression. In GPs one assumes that the data are generated as $y_i = \hat{y}(x)+e_i$. Given N data points $\{(x_i, y_i)\}_{i=1}^N$, the predictive mean for a new input x is given by $\hat{y}(x) = \theta(x)^T C_N^{-1} Y_R$, with $\theta(x) = [C(x, x_1); ...; C(x, x_N)]$ and the matrix $C_N \in \mathbb{R}^{N\times N}$ with $C_{N,ij} = C(x_i, x_j)$, where $C(x_i, x_j)$ is the parameterized covariance function

$$C(x_i, x_j) = \tfrac{1}{\mu} K(x_i, x_j) + \tfrac{1}{\zeta}\delta_{ij}, \tag{25}$$

with δ_{ij} the Kronecker delta and $i, j = 1, ..., N$. The predictive mean is obtained as:

$$\hat{y}(x) = \tfrac{1}{\mu}\theta(x)^T (\tfrac{1}{\mu}\Omega + \tfrac{1}{\zeta} I_N)^{-1} Y. \tag{26}$$

By combination of (23) and (24), one also obtains (26). The regularization term E_W is related to the covariance matrix of the inputs, while the error term E_D yields a ridge regression estimate in the dual space (Saunders *et al.*, 1999; Suykens & Vandewalle, 1999; Suykens 2000). With the results of the next Sections, one can also show that the expression for the variance in GP is equal to the expressions for the LS-SVM without the bias term. Compared with the GP classifier formulation, the regression approach allows to derive analytical expressions on all three levels of inference.

In GPs one typically uses combinations of kernel functions (Gibbs, 1997; Neal, 1997; Rasmussen, 1996; Williams & Barber, 1998), while a positive constant is added when there is a bias term in the regression function. In (Neal, 1997), the hyperparameters of the covariance function C and the variance $1/\zeta$ of the noise e_i are obtained from a sampled posterior distribution of the hyperparameters. Evidence maximization is used in (Gibbs, 1997) to infer the hyperparameters on the second level. In this paper, the bias term b is inferred on the first level, while μ and ζ are obtained from a scalar optimization problem on the second level. Kernel parameters are determined on the third level of inference.

Although the results from the LS-SVM formulation without bias term and Gaussian Processes are identical, LS-SVMs explicitly formulate a model in the primal space. The resulting support values α_i of the model give further insight in the importance of each data point and can be used to obtain sparseness and to detect outliers. The explicit use of a model also allows to define, in a straightforward way, the effective number of parameters γ_{eff} in Section 4. In the LS-SVM formulation, the bias term is considered as a model parameter and is obtained on the first level of inference. As in ordinary ridge regression (Brown, 1977) no regularization is applied on the bias term b in LS-SVMs and a zero mean training set error is obtained from (18): $\sum_{i=1}^{N} e_i = 0$. It will become clear that the bias term also results in a centering of the Gram matrix in the feature space as is done in kernel PCA (Schölkopf *et al.*, 1998). The corresponding eigenvalues can be used to derive improved generalization bounds for SVM classifiers (Schölkopf *et al.*, 1999). The use of the unregularized bias term also allows to derive explicit links with kernel Fisher Discriminant Analysis (Baudat & Anouar, 2000; Mika *et al.*, 2001).

2.4 Regularized Kernel Fisher Discriminant Analysis

The main concern in Fisher discriminant analysis (Bishop, 1995; Duda & Hart, 1973) is to find a linear discriminant w that yields an optimal discrimination between the two classes $\mathcal{C}_+$ and $\mathcal{C}_-$ depicted in Figure 1. A good discriminant maximizes the distance between the projected class centers and minimizes the overlap between both distributions. Given the estimated class centers $\hat{m}_+ = \sum_{i \in \mathcal{I}_+} \varphi(x_i)/N_+$ and $\hat{m}_- = \sum_{i \in \mathcal{I}_-} \varphi(x_i)/N_-$, one maximizes the squared distance $(w^T(\hat{m}_+ - \hat{m}_-))^2$ between the projected class centers and minimizes the regularized scatter s around the class centers

$$s=\sum_{i\in\mathcal{I}_+}(w^T(\varphi(x_i)-\hat{m}_+))^2+\sum_{i\in\mathcal{I}_-}(w^T(\varphi(x+i)-\hat{m}_-))^2+\gamma^{-1}w^Tw, \quad (27)$$

where the regularization term $\gamma^{-1}w^Tw$ is introduced so as to avoid overfitting in the high-dimensional feature space. The scatter s is minimized so as to obtain a small overlap between the classes. The feature space expression for the regularized kernel Fisher discriminant is then found by maximizing

$$\max_w \mathcal{J}_{FDA}(w) = \frac{(w^T(\hat{m}_+-\hat{m}_-))^2}{s} = \frac{w^T(\hat{m}_+-\hat{m}_-)(\hat{m}_+-\hat{m}_-)^Tw}{w^TS_{\mathcal{WC}}w}, \quad (28)$$

with $S_{\mathcal{WC}} = \sum_{i\in\mathcal{I}_+}(\varphi(x_i)-\hat{m}_+)(\varphi(x_i)-\hat{m}_+)^T+\sum_{i\in\mathcal{I}_-}(\varphi(x_i)-\hat{m}_-)(\varphi(x_i)-\hat{m}_-)^T+\gamma^{-1}I_{n_f}$. The solution to the generalized Rayleigh quotient (28) follows from a generalized eigenvalue problem in the feature space $(\hat{m}_+-\hat{m}_-)(\hat{m}_+-\hat{m}_-)^Tw = \lambda S_{\mathcal{WC}}w$ from which one obtains

$$w = S_{\mathcal{WC}}^{-1}(\hat{m}_+-\hat{m}_-). \quad (29)$$

As the mapping φ is typically unknown, practical expressions need to be derived in the dual space, e.g., by solving a generalized eigenvalue problem (Baudat & Anouar, 2000; Mika *et al.*, 2001). Also the SVM formulation has been related to Fisher Discriminant Analysis (Shashua, 1999). The bias term b is not determined by Fisher discriminant analysis. Fisher Discriminant Analysis is typically used as a first step which yields the optimal linear discriminant between the two classes. The bias term b has to be determined in the second step so as to obtain an optimal classifier.

It can be easily shown in the feature space that the LS-SVM regression formulation (14) yields the same discriminant vector w. Defining $\Upsilon = [\varphi(x_1),\ldots,\varphi(x_N)] \in \mathbb{R}^{n_f\times N}$, the conditions for optimality in the primal space are

$$\begin{bmatrix} \Upsilon\Upsilon^T+\gamma^{-1}I_{n_f} & \Upsilon 1_v \\ 1_v^T\Upsilon^T & N \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} \Upsilon Y \\ \Upsilon 1_v \end{bmatrix}. \quad (30)$$

From the second condition, we obtain $b = w^T(N_+\hat{m}_+ + N_-\hat{m}_-)/N + (N_+ - N_-)/N$. Substituting this into the first condition, one obtains

$$(S_{\mathcal{WC}} + \tfrac{N_+N_-}{N^2}(\hat{m}_+ - \hat{m}_-)(\hat{m}_+ - \hat{m}_-)^T)w = 2\tfrac{N_+N_-}{N^2}(\hat{m}_+ - \hat{m}_-),$$

which yields, up to a scaling constant, the same discriminant vector w as (29) since $(\hat{m}_+ - \hat{m}_-)(\hat{m}_+ - \hat{m}_-)^T)w \propto (\hat{m}_+ - \hat{m}_-)$. In the regression formulation the bias b is determined so as to obtain an optimal least squares approximation (17) for the discriminant function (2).

3 Probabilistic Interpretation of the LS-SVM Classifier (Level 1)

A probabilistic framework is related to the LS-SVM classifier. The outline of our approach is similar to the work of (Kwok, 1999a; 1999b) for SVMs, but there are significant differences concerning the Bayesian interpretation of the cost function and the algebra involved for the computations in the feature space. First, Bayes' rule is applied in order to obtain the LS-SVM cost function. The moderated output is obtained by marginalizing over w and b.

3.1 Inference of the Model Parameters w and b

Given the data points $D = \{(x_i, y_i)\}_{i=1}^N$ and the hyperparameters μ and ζ of the model $\mathcal{H}$, the model parameters w and b are estimated by maximizing the posterior $p(w, b|D, \log\mu, \log\zeta, \mathcal{H})$. Applying Bayes' rule at the first level

(Bishop, 1995; MacKay, 1995), we obtain[1]

$$p(w,b|D,\log\mu,\log\zeta,\mathcal{H}) = \frac{p(D|w,b,\log\mu,\log\zeta,\mathcal{H})p(w,b|\log\mu,\log\zeta,\mathcal{H})}{p(D|\log\mu,\log\zeta,\mathcal{H})}, \quad (31)$$

where the evidence $p(D|\log\mu,\log\zeta,\mathcal{H})$ is a normalizing constant such the integral over all possible w and b values is equal to 1.

We assume a separable Gaussian prior, which is independent of the hyperparameter ζ, i.e., $p(w,b|\log\mu,\log\zeta,\mathcal{H}) = p(w|\log\mu,\mathcal{H})p(b|\log\sigma_b,\mathcal{H})$, where $\sigma_b \to \infty$ to approximate a uniform distribution. By the choice of the regularization term E_W in (15), we obtain for the prior with $\sigma_b \to \infty$:

$$\begin{aligned} p(w,b|\log\mu,\mathcal{H}) &= \left(\tfrac{\mu}{2\pi}\right)^{\frac{n_f}{2}} \exp(-\tfrac{\mu}{2}w^Tw)\tfrac{1}{\sqrt{2\pi\sigma_b^2}}\exp(-\tfrac{b^2}{2\sigma_b^2}) \\ &\propto \left(\tfrac{\mu}{2\pi}\right)^{\frac{n_f}{2}} \exp(-\tfrac{\mu}{2}w^Tw). \end{aligned} \quad (32)$$

To simplify the notation, the step of taking the limit of $\sigma_b \to \infty$ is already made in the remainder of this paper.

The probability $p(D|w,b,\log\mu,\log\zeta,\mathcal{H})$ is assumed to depend only on w, b, ζ and $\mathcal{H}$. We assume that the data points are independent $p(D|w,b,\log\zeta,\mathcal{H}) = \prod_{i=1}^N p(x_i,y_i|w,b,\log\,\zeta,\mathcal{H})$. In order to obtain the least squares cost function (16) from the LS-SVM formulation, it is assumed that the probability of a data point is proportional to

$$p(x_i,y_i|w,b,\log\zeta,\mathcal{H}) \propto p(e_i|w,b,\log\zeta,\mathcal{H}), \quad (33)$$

where the normalizing constant is independent of w and b. A Gaussian distribution is taken for the errors $e_i = y_i - (w^T\varphi(x_i)+b)$ from (13)

$$p(e_i|w,b,\log\zeta,\mathcal{H}) = \sqrt{\tfrac{\zeta}{2\pi}}\exp(-\tfrac{\zeta e_i^2}{2}). \quad (34)$$

[1]The notation $p(\cdot|\cdot,\log\mu,\log\zeta,\cdot)$ used here is somewhat different form the notation $p(\cdot|\cdot,\mu,\zeta,\cdot)$ used in (MacKay, 1995). We prefer this notation since μ and ζ are (positive) scale parameters (Gull, 1988). By doing so, a uniform notation over the three levels of inference is obtained. The change in notation does not affect the results.

An appealing way to interpret this probability is depicted in Figure 1. It is assumed that the w and b are determined in such a way that the class centers $\hat{m}_-$ and $\hat{m}_+$ are mapped onto the targets -1 and $+1$, respectively. The projections $w^T\varphi(x) + b$ of the class elements $\varphi(x)$ of the multivariate Gaussian distributions are then normally disturbed around the corresponding targets with variance $1/\zeta$. One can then write $p(x_i, y_i|w, b, \zeta, \mathcal{H}) = p(x_i|y_i, w, b, \zeta, \mathcal{H})P(y_i) = p(e_i|w, b, \zeta, \mathcal{H})P(y_i)$, where the errors $e_i = y_i - (w^T\varphi(x_i) + b)$ are obtained by projecting the feature vector $\varphi(x_i)$ onto the discriminant function $w^T\varphi(x_i)+b$ and comparing them with the target y_i. Given the binary targets $y_i \in \{-1, +1\}$ the error e_i is a function of the input x_i in the classifier interpretation. Assuming a multivariate Gaussian distribution of feature vector $\varphi(x_i)$ in the feature space, the errors e_i are also Gaussian distributed as is depicted in Figure 1 (Bishop, 1995; Duda & Hart, 1973). However, the assumptions that $w^T\hat{m}_- + b = -1$ and $w^T\hat{m}_+ + b = +1$ may not always hold and will be relaxed in the next Subsection.

By combining (32) and (34) and neglecting all constants, Bayes' rule (31) for the first level of inference becomes

$$p(w, b|D, \log\mu, \log\zeta, \mathcal{H}) \propto \exp(-\tfrac{\mu}{2}w^Tw - \tfrac{\zeta}{2}\textstyle\sum_{i=1}^N e_i^2) = \exp(-\mathcal{J}_2(w, b)). \quad (35)$$

The maximum a posteriori estimates w_{MP} and b_{MP} are then obtained by minimizing the negative logarithm of (35). In the dual space, this corresponds to solving the linear set of equations (19). The quadratic cost function (14) is linear in w and b and can be related to the posterior

$$p(w, b|D, \mu, \zeta, \mathcal{H}) = \frac{1}{\sqrt{(2\pi)^{n_f+1}\det Q}}\exp(-\tfrac{1}{2}g^TQ^{-1}g), \quad (36)$$

with[2] $g = [w - w_{\text{MP}};\, b - b_{\text{MP}}]$ and $Q = \text{covar}(w, b) = \mathcal{E}(g^Tg)$, taking the expectation over w and b. The covariance matrix Q is related to the Hessian

[2]The matlab notation $[X;\, Y]$ is used, where $[X;\, Y] = [X^T\, Y^T]^T$.

H of (14):

$$Q = H^{-1} = \left[\begin{array}{cc} H_{11} & H_{12} \\ H_{12}^T & H_{22} \end{array} \right]^{-1} = \left[\begin{array}{cc} \frac{\partial^2 \mathcal{J}_2}{\partial w^2} & \frac{\partial^2 \mathcal{J}_2}{\partial w \partial b} \\ \frac{\partial^2 \mathcal{J}_2}{\partial b \partial w} & \frac{\partial^2 \mathcal{J}_2}{\partial b^2} \end{array} \right]^{-1}. \tag{37}$$

When using MLPs, the cost function is typically non-convex and the covariance matrix is estimated using a quadratic approximation in the local optimum (MacKay, 1995).

3.2 Class Probabilities for the LS-SVM Classifier

Given the posterior probability (36) of the model parameters w and b, we will now integrate over all w and b values in order to obtain the posterior class probability $P(y|x, D, \mu, \zeta, \mathcal{H})$. First, it should be remarked that the assumption $w^T \hat{m}_+ + b = +1$ and $w^T \hat{m}_- + b = -1$ may not always be satisfied. This typically occurs when the training set is unbalanced ($N_+ \neq N_-$): in this case the discriminant shifts towards the a priori most likely class so as to yield the optimal least squares approximation (17). Therefore, we will use

$$\begin{aligned} p(x|y = \bullet 1, w, b, \log \zeta_\bullet, \mathcal{H}) &= \sqrt{\frac{\zeta_\bullet}{2\pi}} \exp\left(-\frac{\zeta_\bullet (w^T(\varphi(x) - \hat{m}_\bullet))^2}{2}\right) \\ &= \sqrt{\frac{\zeta_\bullet}{2\pi}} \exp\left(-\frac{\zeta_\bullet e_\bullet^2}{2}\right) \end{aligned} \tag{38}$$

with $e_\bullet = w^T(\varphi(x) - \hat{m}_\bullet)$ by definition, where $\zeta_\bullet^{-1}$ is the variance of $e_\bullet$. The $\bullet$ notation is used to denote either $+$ or $-$, since analogous expressions are obtained for classes $\mathcal{C}_+$ and $\mathcal{C}_-$, respectively. In this paper, we assume $\zeta_+ = \zeta_- \triangleq \zeta_*$.

Since $e_\bullet$ is a linear combination of the Gaussian distributed w, marginalizing over w will yield a Gaussian distributed $e_\bullet$ with mean $m_{e_\bullet}$ and variance $\sigma_{e_\bullet}^2$. The expression for the mean is

$$m_{e_\bullet} = w_{\mathrm{MP}}^T(\varphi(x) - \hat{m}_\bullet) = \textstyle\sum_{i=1}^N \alpha_i K(x, x_i) - \hat{m}_{\mathrm{d}\bullet}, \tag{39}$$

with $\hat{m}_{\mathrm{d}\bullet} = \frac{1}{N_\bullet}\sum_{i=1}^{N}\alpha_i \sum_{j\in\mathcal{I}_\bullet} K(x_i, x_j)$, while the corresponding expression for the variance is

$$\sigma_{e_\bullet}^2 = [\varphi(x) - \hat{m}_\bullet]^T Q_{11}[\varphi(x) - \hat{m}_\bullet] \tag{40}$$

The expression for the upper left $n_f \times n_f$ block Q_{11} of the covariance matrix Q is derived in Appendix A. By using matrix algebra and applying the Mercer condition, we obtain:

$$\begin{aligned}\sigma_{e_\bullet}^2 &= \mu^{-1}K(x,x) - 2\mu^{-1}N_\bullet^{-1}\textstyle\sum_{i\in\mathcal{I}_\bullet} K(x,x_i) + \mu^{-1}N_\bullet^{-2}1_v^T\Omega(\mathcal{I}_\bullet,\mathcal{I}_\bullet)1_v - [\theta(x)^T \\ &- \tfrac{1}{N_\bullet}1_v^T\Omega(\mathcal{I}_\bullet,\mathcal{I})]MU_G[\mu^{-1}I_{n_{eff}} - (\mu I_{n_{eff}} + \zeta D_G)^{-1}]U_G^T M[\theta(x) - \tfrac{1}{N_\bullet}\Omega(\mathcal{I},\mathcal{I}_\bullet)1_v],\end{aligned} \tag{41}$$

where 1_v is a vector of appropriate dimensions with all elements equal to one and where we used the Matlab index notation $X(\mathcal{I}_a,\mathcal{I}_b)$ which selects the corresponding rows $\mathcal{I}_a$ and columns $\mathcal{I}_b$ of the matrix X. The vector $\theta(x) \in \mathbb{R}^N$ and the matrices $U_G \in \mathbb{R}^{N\times N_{eff}}$ and $D_G \in \mathbb{R}^{N_{eff}\times N_{eff}}$ are defined as follows:

$$\theta_i(x) = K(x, x_i), \qquad i = 1,\ldots,N \tag{42}$$

$$U_G(:,i) = \lambda_{G,i}^{-\frac{1}{2}} v_{G,i}, \qquad i = 1,\ldots,N_{eff} \leq N-1 \tag{43}$$

$$D_G = \mathrm{diag}([\lambda_{G,1},\ldots,\lambda_{G,N_{eff}}]), \tag{44}$$

where $v_{G,i}$ and $\lambda_{G,i}$ are the solutions to the eigenvalue problem (68)

$$M\Omega M\nu_{G,i} \;=\; \lambda_{G,i}v_{G,i}, \quad i = 1,\ldots,N_{eff} \leq N-1, \tag{45}$$

with $V_G = [v_{G,1},\ldots,v_{G,N_{eff}}] \in \mathbb{R}^{N\times N_{eff}}$. The vector Y and the matrix Ω are defined in (20) and (21), respectively, while $M \in \mathbb{R}^{N\times N}$ is the idempotent centering matrix $M = I_N - 1/N 1_v 1_v^T$ with rank $N-1$. The number of non-zero eigenvalues is denoted by $N_{eff} < N$. For rather large datasets, one may choose to reduce to computational requirements and to approximate the variance σ_z^2 by using only the most significant eigenvalues ($\lambda_{G,i} \gg \frac{\mu}{\zeta}$) in the above expressions. In this case, N_{eff} denotes the number of most significant eigenvalues (see Appendix A for details).

The conditional probabilities $p(x|y=+1, D, \log\mu, \log\zeta, \log\zeta_*, \mathcal{H})$ and $p(x|y=-1, D, \log\mu, \log\zeta, \log\zeta_*, \mathcal{H})$ are then equal to

$$p(x|y{=}\bullet 1, D, \log\mu, \log\zeta, \log\zeta_\bullet, \mathcal{H}) = (2\pi(\zeta_\bullet^{-1} + \sigma_{e_\bullet}^2))^{-\frac{1}{2}} \exp(-\tfrac{m_{e_\bullet}^2}{2(\zeta_\bullet^{-1}+\sigma_{e_\bullet}^2)}) \quad (46)$$

with $\bullet$ either $+$ or $-$, respectively. By applying Bayes' rule (1) the following class probabilities of the LS-SVM classifier are obtained:

$$P(y|x, D, \log\mu, \log\zeta, \log\zeta_*, \mathcal{H}) = \frac{P(y)p(x|y, D, \log\mu, \log\zeta, \log\zeta_*, \mathcal{H})}{p(x|D, \log\mu, \log\zeta, \log\zeta_*, \mathcal{H})}, \quad (47)$$

where the denominator $p(x|D, \log\mu, \log\zeta, \log\zeta_*, \mathcal{H}) = P(y{=}{+}1)p(x|y{=}{+}1, D, \log\mu, \log\zeta, \log\zeta_*, \mathcal{H}) + P(y{=}{-}1)p(x|y{=}{-}1, D, \log\mu, \log\zeta, \log\zeta_*, \mathcal{H})$ follows from normalization. Substituting the expression (46) for $\bullet = +$ and $\bullet = -$ into (47) a quadratic expression is obtained since $\zeta_*^{-1} + \sigma_{e_-}^2 \neq \zeta_*^{-1} + \sigma_{e_+}^2$. When $\sigma_{e_-}^2 \simeq \sigma_{e_+}^2$, one can define $\sigma_e^2 = \sqrt{\sigma_{e_+}^2 \sigma_{e_-}^2}$ and one obtains the linear discriminant function

$$y(x) = \mathrm{sign}[\textstyle\sum_{i=1}^N \alpha_i K(x, x_i) - \frac{\hat{m}_{\mathrm{d}+}+\hat{m}_{\mathrm{d}-}}{2} + \frac{\zeta^{-1}+\sigma_e^2}{\hat{m}_{\mathrm{d}+}-\hat{m}_{\mathrm{d}-}} \log \frac{P(y=+1)}{P(y=-1)}]. \quad (48)$$

The second and third term in (48) correspond to the bias term b in the LS-SVM classifier formulation (22), where the bias term was determined to obtain an optimal least squares approximation to the Bayes discriminant. The decision rules (47) and (48) allow to take into account prior class probabilities in a more elegant way. This also allows to adjust the bias term for classification problems with different prior class probabilities in the training and test set. Due to the marginalization over w the bias term correction is also a function of the input x since σ_e^2 is a function of x. The idea of a (static) bias term correction has also been applied in (Evgeniou *et al.*, 2000) in order to improve the validation set performance. In (Mukherjee *et al.*, 1999) the probabilities $p(e|y, w_{\mathrm{MP}}, b_{\mathrm{MP}}, \mu, \zeta, \mathcal{H})$ were estimated using leave-one-out cross validation given the obtained SVM classifier and the corresponding classifier decision was made in a similar way as in (48). A simple density estimation algorithm was used and no Gaussian assumptions were made, while no marginalization over the model parameters was performed. A bias term correction was also applied in the softmax interpretation for the SVM output (Platt, 1999) using a validation set. Given the asymptotically optimal least squares approximation one can approximate the class probabilities $P(y = +1|x, w, b, D, \log\mu, \log\zeta, \mathcal{H}) = (1 + g_0(x))/2$ replacing $g_0(x)$ by $w_{\mathrm{MP}}^T\varphi(x) + b_{\mathrm{MP}}$ for the LS-SVM formulation. However, such an approach does not yield true probabilities that are between 0 and 1 and sum up to 1. Using a softmax function (Bishop, 1995; MacKay, 1992; 1995) one obtains $P(y = +1|x, w, b, D, \log\mu, \log\zeta, \mathcal{H}) = (1 + \exp(-(w^T\varphi(x) + b)))^{-1}$ and $P(y = -1|x, w, b, D, \log\mu, \log\zeta, \mathcal{H}) = (1 + \exp(+(w^T\varphi(x) + b)))^{-1}$. In order to marginalize over the model parameters in the logistic functions, one can use the approximate expressions of (MacKay, 1992; 1995) in combination with the expression for the moderated output of the LS-SVM regressor derived in (Van Gestel *et al.*, 2001). In the softmax interpretation for SVMs (Platt, 1999) no marginalization over the model parameters is applied and the bias term is determined on a validation set. Finally it is remarked that, because of the equivalence between classification costs and prior probabilities (Duda & Hart, 1973), the results for the moderated output of the LS-SVM classifier can be extended in a straightforward way in order to take different classification costs into account.

For large scale data sets, the computation of the eigenvalue decomposition (45) may require long computations and one may choose to compute only the largest eigenvalues and corresponding eigenvectors using an expectation maximization approach (Rosipal & Girolami, 2001). This will result into an increased variance as explained in Appendix A. An alternative approach is to use the cheap and chearful approach described in (MacKay, 1995).

4 Inference of the Hyperparameters μ and ζ (Level 2)

Bayes' rule is applied on the second level of inference to infer the most likely $\mu_{\rm MP}$ and $\zeta_{\rm MP}$ values from the given data D. The differences with the expressions obtained in (MacKay, 1995) are due to the fact that no regularization is applied on the bias term b and that all practical expressions are obtained in the dual space by applying the Mercer condition. Up to a centering, these expressions are similar to the expressions obtained with GP for regression. By combination of the conditions for optimality, the minimization problem in μ and ζ is reformulated into a scalar minimization problem in $\gamma = \zeta/\mu$.

4.1 Inference of μ and ζ

In the second level of inference, Bayes' rule is applied to infer the most likely μ and ζ values from the data:

$$\begin{aligned} p(\log\mu, \log\zeta|D, \mathcal{H}) &= \frac{p(D|\log\mu, \log\zeta, \mathcal{H})p(\log\mu, \log\zeta|\mathcal{H})}{p(D|\mathcal{H})} \\ &\propto p(D|\log\mu, \log\zeta, \mathcal{H}). \end{aligned} \tag{49}$$

Because the hyperparameters μ and ζ are scale parameters (Gull, 1988), we take a uniform distribution in $\log\mu$ and $\log\zeta$ for the prior $p(\log\mu, \log\zeta|\mathcal{H}) = p(\log\mu|\mathcal{H})p(\log\zeta|\mathcal{H})$ in (49). The evidence $p(D|\mathcal{H})$ is again a normalizing constant, which will be needed in level 3. The probability $p(D|\log\mu, \log\zeta, \mathcal{H})$ is equal to the evidence in (31) of the previous level. Substituting (32), (34) and (36) into (49), we obtain:

$$p(\log\mu, \log\zeta|D, \mathcal{H}) \propto \frac{\sqrt{\mu^{n_f}}\sqrt{\zeta^N}}{\sqrt{\det H}} \frac{\exp(-\mathcal{J}_2(w,b))}{\exp(-\frac{1}{2}g^T H g)} \propto \frac{\sqrt{\mu^{n_f}\zeta^N}}{\sqrt{\det H}} \exp(-\mathcal{J}_2(w_{\rm MP}, b_{\rm MP})),$$

where $\mathcal{J}_2(w,b) = \mathcal{J}_2(w_{\rm MP}, b_{\rm MP}) + \frac{1}{2}g^T H g$ with $g = [w - w_{\rm MP};\ b - b_{\rm MP}]$. The expression for $\det H$ is given in Appendix B and is equal to $\det H = N\mu^{n_f - N_{eff}}\zeta$

$\prod_{i=1}^{N_{eff}}(\mu+\zeta\lambda_{G,i})$, where the N_{eff} eigenvalues $\lambda_{G,i}$ are the non-zero eigenvalues of $M\Omega M$. Taking the negative logarithm of $p(\log\mu, \log\zeta|D, \mathcal{H})$, the optimal parameters $\mu_{\rm MP}$ and $\zeta_{\rm MP}$ are found as the solution to the minimization problem:

$$\begin{aligned}\min_{\mu,\zeta} \mathcal{J}_3(\mu,\zeta) =& \mu E_W(w_{\rm MP}) + \zeta E_D(w_{\rm MP}, b_{\rm MP}) \\ & + \tfrac{1}{2}\textstyle\sum_{i=1}^{N_{eff}} \log(\mu+\zeta\lambda_{G,i}) - \tfrac{N_{eff}}{2}\log\mu - \tfrac{N-1}{2}\log\zeta. \end{aligned} \quad (50)$$

In Appendix B it is also shown that the level 1 cost function evaluated in $w_{\rm MP}$ and $b_{\rm MP}$ can be written as $\mu E_W(w_{\rm MP}) + \zeta E_D(w_{\rm MP}, b_{\rm MP}) = \frac{1}{2}Y^T M(\mu^{-1}M\Omega M + \zeta^{-1}I_N)^{-1}MY$. The cost function $\mathcal{J}_3$ from (50) can be written as

$$\begin{aligned}\min_{\mu,\zeta} \mathcal{J}_3(\mu,\zeta) = & \tfrac{1}{2}Y^T M(\tfrac{1}{\mu}M\Omega M + \tfrac{1}{\zeta}I_N)^{-1}MY \\ & + \tfrac{1}{2}\log\det(\tfrac{1}{\mu}M\Omega M + \tfrac{1}{\zeta}I_N) - \tfrac{1}{2}\log\tfrac{1}{\zeta}, \end{aligned} \quad (51)$$

where the last term is due to the extra bias term b in the LS-SVM formulation. Neglecting the centering matrix M, the first two terms in (51) correspond to the level 2 cost function used in GP (Gibbs, 1997; Rasmussen, 1996; Williams, 1998). Hence, the use of the unregularized bias term b in the SVM and LS-SVM formulation results into a centering matrix M in the obtained expressions compared to GP. The eigenvalues $\lambda_{G,i}$ of the centered Gram matrix are also used in kernel PCA (Schölkopf *et al.*, 1998) and can also be used to infer improved error bounds for SVM classifiers (Schölkopf *et al.*, 1999). In the Bayesian framework, the capacity is controlled by the prior.

The effective number of parameters (Bishop, 1995; MacKay, 1995) is equal to $\gamma_{eff} = \sum \lambda_{i,u}/\lambda_{i,r}$, where $\lambda_{i,u}$ and $\lambda_{i,r}$ are the eigenvalues of Hessians of the unregularized cost function ($\mathcal{J}_u = \zeta E_D$) and regularized cost function ($\mathcal{J}_r = \mu E_W + \zeta E_D$), respectively. For the LS-SVM, the effective number of parameters is equal to:

$$\gamma_{eff} = 1 + \textstyle\sum_{i=1}^{N_{eff}} \frac{\zeta_{\rm MP}\lambda_{G,i}}{\mu_{\rm MP}+\zeta_{\rm MP}\lambda_{G,i}} = 1 + \sum_{i=1}^{N_{eff}} \frac{\gamma_{\rm MP}\lambda_{G,i}}{1+\gamma_{\rm MP}\lambda_{G,i}}, \quad (52)$$

with $\gamma = \zeta/\mu$. The term $+1$ is obtained because no regularization on the bias term b is applied. Notice that since $N_{eff} \leq N-1$, the effective number of parameters γ_{eff} can never exceed the number of given training data points: $\gamma_{eff} \leq N$, although we may choose a kernel function K with possibly $n_f \rightarrow \infty$ degrees of freedom in the feature space.

The gradient of the cost function $\mathcal{J}_3(\mu, \zeta)$ is (MacKay, 1992):

$$\begin{aligned} \frac{\partial \mathcal{J}_3}{\partial \mu} &= E_W(w_{\mathrm{MP}}) + \frac{1}{2}\sum_{i=1}^{N_{eff}} \frac{1}{\mu+\zeta\lambda_{G,i}} - \frac{N_{eff}}{2\mu} && (53) \\ \frac{\partial \mathcal{J}_3}{\partial \zeta} &= E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}}) + \frac{1}{2}\sum_{i=1}^{N_{eff}} \frac{\lambda_{G,i}}{\mu+\zeta\lambda_{G,i}} - \frac{N-1}{2\zeta}. && (54) \end{aligned}$$

Putting the partial derivatives (53) and (54) equal to zero, we obtain the following relations in the optimum of the level 2 cost function: $2\mu_{\mathrm{MP}}E_W(w_{\mathrm{MP}}) = \gamma_{eff}-1$ and $2\zeta_{\mathrm{MP}}E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}}) = N-\gamma_{eff}$. The last equality can be viewed as the Bayesian estimate of the variance $\zeta_{\mathrm{MP}}^{-1} = \sum_{i=1}^{N} e_i^2/(N-\gamma_{eff})$ of the noise e_i. While this yields an implicit expression for the optimal ζ_{MP} for the regression formulation, this may not be equal to the variance ζ_* since the targets ± 1 does not necessarily correspond to the projected class centers. Therefore, we will use the estimate $\zeta_*^{-1} = (N-\gamma_{eff})^{-1}(\sum_{i\in\mathcal{I}_+} e_{i,+}^2 + \sum_{i\in\mathcal{I}_-} e_{i,-}^2)$ in the remainder of this paper. Combining both relations, we obtain that for the optimal μ_{MP}, ζ_{MP} and $\gamma_{\mathrm{MP}} = \zeta_{\mathrm{MP}}/\mu_{\mathrm{MP}}$:

$$2\mu_{\mathrm{MP}}[E_W(w_{\mathrm{MP}}) + \gamma_{\mathrm{MP}}E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}})] = N-1. \tag{55}$$

4.2 A Scalar Optimization Problem in $\gamma = \zeta/\mu$

We reformulate the optimization problem (50) in μ and ζ into a scalar optimization problem in $\gamma = \zeta/\mu$. Therefore, we first replace the optimization problem (50) by an optimization problem in μ and γ. We can use that $E_W(w_{\mathrm{MP}})$ and $E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}})$ in the optimum of (14) only depend on γ. Since in the optimum also (55) holds, we only have the search for the optimum along this curve in the (μ, γ) space.

By elimination of μ from (55), the following minimization problem is obtained in a straightforward way:

$$\min_{\gamma} \mathcal{J}_4(\gamma) = \textstyle\sum_{i=1}^{N-1} \log[\lambda_{G,i} + \frac{1}{\gamma}] + (N-1)\log[E_W(w_{\mathrm{MP}}) + \gamma E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}})] \tag{56}$$

with $\lambda_{G,i} = 0$ for $i > N_{eff}$. The derivative $\frac{\partial \mathcal{J}_4}{\partial \gamma}$ is obtained in a similar way as $\frac{\partial \mathcal{J}_3}{\partial \mu}$:

$$\tfrac{\partial \mathcal{J}_4}{\partial \gamma} = -\textstyle\sum_{i=1}^{N-1} \frac{1}{\gamma + \lambda_{G,i}\gamma^2} + (N-1)\frac{E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}})}{E_W(w_{\mathrm{MP}}) + \gamma E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}})}. \tag{57}$$

Due to the second logarithmic term, this cost function is not convex and it is useful to start from different initial values for γ. The condition for optimality $(\partial \mathcal{J}_4 / \partial \gamma = 0)$ is:

$$\gamma_{\mathrm{MP}} = \tfrac{N - \gamma_{eff}}{\gamma_{eff} - 1} \, \tfrac{E_W(w_{\mathrm{MP}})}{E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}})}. \tag{58}$$

We also need the expressions for E_D and E_W in (56) and (57). It is explained in Appendix B that these terms can be expressed in terms of the output vector Y and the eigenvalue decomposition of the centered kernel matrix $M\Omega M$:

$$E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}}) = \tfrac{1}{2\gamma^2} Y^T M V_G (D_G + \gamma^{-1} I_{n_{eff}})^{-2} V_G^T M Y \tag{59}$$

$$E_W(w_{\mathrm{MP}}) = \tfrac{1}{2} Y^T M V_G D_G (D_G + \gamma^{-1} I_{n_{eff}})^{-2} V_G^T M Y \tag{60}$$

$$E_W(w_{\mathrm{MP}}) + \gamma E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}}) = \tfrac{1}{2} Y^T M V_G (D_G + \gamma^{-1} I_{n_{eff}})^{-1} V_G^T M Y. \tag{61}$$

When the eigenvalue decomposition (45) is calculated, the optimization (56) involves only vector products which can be evaluated very fast.

Although the eigenvalues $\lambda_{G,i}$ only have to be calculated once, their calculation in the eigenvalue problem (45) becomes computationally expensive for large datasets. In this case, one can choose to calculate only the largest eigenvalues in (45) using an expectation maximization approach (Rosipal & Girolami, 2001), while the linear system (19) can be solved using the Hestenes-Stiefel conjugate gradient algorithm (Suykens, 2000). The obtained α and b can also be used to derive an alternative expression for $E_D = \frac{1}{2\gamma^2} \sum_{i=1}^{N} \alpha_i^2$ and $E_W = \frac{1}{2} \sum_{i=1}^{N} \alpha_i (y_i - \frac{\alpha_i}{\gamma} - b_{\mathrm{MP}})$, instead of using (59) and (60).

5 Bayesian Model Comparison (Level 3)

After determination of the hyperparameters μ_{MP} and ζ_{MP} on the second level of inference, we still have to select a suitable model $\mathcal{H}$. For SVMs, different models correspond to different kernel functions K, e.g., a linear kernel or an RBF kernel with tuning parameter σ. We describe how to rank different models $\mathcal{H}_j$ ($j = 1, 2, \ldots$, corresponding to, e.g., RBF kernels with different tuning parameters σ_j) in the Bayesian evidence framework (MacKay, 1999). By applying Bayes' rule on the third level, we obtain the posterior for the model $\mathcal{H}_j$:

$$p(\mathcal{H}_j|D) \propto p(D|\mathcal{H}_j)p(\mathcal{H}_j). \tag{62}$$

At this level, no evidence or normalizing constant is used since it is computationally infeasible to compare all possible models $\mathcal{H}_j$. The prior $p(\mathcal{H}_j)$ over all possible models is assumed to be uniform here. Hence, (62) becomes $p(\mathcal{H}_j|D) \propto p(D|\mathcal{H}_j)$. The likelihood $p(D|\mathcal{H}_j)$ corresponds to the evidence (49) of the previous level.

A separable Gaussian prior $p(\log \mu_{\mathrm{MP}}, \log \zeta_{\mathrm{MP}}|\mathcal{H}_j)$ with error bars $\sigma_{\log \mu}$ and $\sigma_{\log \zeta}$ is assumed for all models $\mathcal{H}_j$. To estimate the posterior analytically, it is assumed (MacKay, 1999) that the evidence $p(\log \mu, \log \zeta|D, \mathcal{H}_j)$ can be very well approximated by using a separable Gaussian with error bars $\sigma_{\log \mu|D}$ and $\sigma_{\log \zeta|D}$. As in Section 4, the posterior $p(D|\mathcal{H}_j)$ then becomes (MacKay, 1995; 1999)

$$p(D|\mathcal{H}_j) \quad \propto \quad p(D|\log \mu_{\mathrm{MP}}, \log \zeta_{\mathrm{MP}}, \mathcal{H}_j)\frac{\sigma_{\log \mu|D}\sigma_{\log \zeta|D}}{\sigma_{\log \mu}\sigma_{\log \zeta}}. \tag{63}$$

Ranking of models according to model quality $p(D|\mathcal{H}_j)$ is thus based on the goodness of fit $p(D|\log \mu_{\mathrm{MP}}, \log \zeta_{\mathrm{MP}}, \mathcal{H}_j)$ from the previous level and the Occam factor $\frac{\sigma_{\log \mu|D}\sigma_{\log \zeta|D}}{\sigma_{\log \mu}\sigma_{\log \zeta}}$ (Gull, 1988; Mackay, 1995; 1999).

Following a similar reasoning as in (MacKay, 1999), the error bars $\sigma_{\log \mu|D}$ and $\sigma_{\log \zeta|D}$ can be approximated as follows: $\sigma^2_{\log \mu|D} \simeq \frac{2}{\gamma_{eff}-1}$ and $\sigma^2_{\log \zeta|D} \simeq$

$\frac{2}{N-\gamma_{eff}}$. Using (49) and (55) in (63) and neglecting all constants yields

$$p(D|\mathcal{H}_j) \propto \sqrt{\frac{\mu_{\mathrm{MP}}^{N_{eff}} \zeta_{\mathrm{MP}}^{N-1}}{(\gamma_{eff}-1)(N-\gamma_{eff}) \prod_{i=1}^{N_{eff}} (\mu_{\mathrm{MP}}+\zeta_{\mathrm{MP}}\lambda_{G,i})}}. \tag{64}$$

6 Design and Application of the LS-SVM Classifier

In this Section, the theory of the previous Sections is used to design the LS-SVM classifier in the Bayesian evidence framework. The obtained classifier is then used to assign class labels and class probabilities to new inputs x by using the probabilistic interpretation of the LS-SVM classifier.

6.1 Design of the LS-SVM Classifier in the Evidence Framework

The design of the LS-SVM classifier consists of the following steps:

1. The inputs are normalized to zero mean and unit variance (Bishop, 1995). The normalized training data are denoted by $D = \{(x_i, y_i)\}_{i=1}^N$, with x_i the normalized inputs and $y_i \in \{-1, 1\}$ the corresponding class label.

2. Select the model $\mathcal{H}_j$ by choosing a kernel type K_j (possibly with a kernel parameter, e.g., σ_j for an RBF-kernel). For this model $\mathcal{H}_j$, the optimal hyperparameters μ_{MP} and ζ_{MP} are estimated on the second level of inference. This is done as follows. (a) Estimate the N_{eff} important eigenvalues (and eigenvectors) of the eigenvalue problem (45) to obtain D_G (and V_G). (b) Solve the scalar optimization problem (56) in $\gamma = \zeta/\mu$ with cost function (56) and gradient (57). (c) Use the optimal γ_{MP} to calculate μ_{MP} from (55), while $\zeta_{\mathrm{MP}} = \mu_{\mathrm{MP}}\gamma_{\mathrm{MP}}$. Calculate the effective number of parameters γ_{eff} from (52).

3. Calculate the model evidence $p(D|\mathcal{H}_j)$ from (64).

4. For a kernel K_j with tuning parameters, refine the tuning parameters. E.g., for the RBF-kernel with tuning parameter σ_j, refine σ_j such that a higher model evidence $p(D|\mathcal{H}_j)$ is obtained. This can be done by maximizing the model evidence with respect to σ_j by evaluating the model evidence for the refined kernel parameter starting from step 2(a).

5. Select the model $\mathcal{H}_j$ with maximal model evidence $p(D|\mathcal{H}_j)$. Go to step 2, unless the best model has been selected.

For a kernel function without tuning parameter, like the linear kernel and polynomial kernel with (already) fixed degree d, steps 2 and 4 are trivial, since no tuning parameter of the kernel has to be chosen in step 2 and since no refining of the tuning parameter is needed in step 4. The model evidence obtained at step 4 can then be used to rank the different kernel types and to select the most appropriate kernel function.

6.2 Decision making with the LS-SVM classifier

The designed LS-SVM classifier $\mathcal{H}_j$ is now used to calculate class probabilities. By combination of these class probabilities with Bayesian decision theory (Duda & Hart, 1973), class labels are assigned in an optimal way. The classification is done in the following steps:

1. First, normalize the input in the same way as the training data D. The normalized input vector is denoted by x.

2. Assuming that the parameters α, b_{MP}, μ_{MP}, ζ_{MP}, γ_{MP}, γ_{eff}, D_G, U_G, N_{eff} are available from the design of the model $\mathcal{H}_j$, one can calculate m_{e_+}, m_{e_-}, $\sigma^2_{e_+}$ and $\sigma^2_{e_-}$ from (39) and (41), respectively. Compute ζ_* from $\zeta_*^{-1} = (N - \gamma_{eff})^{-1}(\sum_{i\in\mathcal{I}_+} e_{i,+}^2 + \sum_{i\in\mathcal{I}_-} e_{i,-}^2)$.

3. Calculate $p(x|y = +1, D, \log \mu_{\text{MP}}, \log \zeta_{\text{MP}}, \log \zeta_*, \mathcal{H})$ and $p(x|y = +1, D, \log \mu, \log \zeta, \log \zeta_*, \mathcal{H})$ from (46).

4. Calculate $P(y|x, D, \mathcal{H}_j)$ from (47) using the prior class probabilities $P(y{=}{+}1)$ and $P(y{=}{-}1)$. When these prior class probabilities are not available, compute $P(y{=}{+}1) = N_+/N$ and $P(y{=}{-}1) = N_-/N$.

5. Assign the class label to class with maximal posterior $P(y|x, D, \mathcal{H}_j)$.

7 Examples

The synthetic binary classification benchmark dataset from (Ripley, 1996) is used to illustrate the theory of this paper. Randomized test set performances of the 'Bayesian LS-SVM' are reported on 10 binary classification datasets.

7.1 Design of the Bayesian LS-SVM: a Case Study

We illustrate the design of the LS-SVM classifier within the evidence framework on the synthetic dataset (`rsy`) from (Ripley, 1996). The dataset consists of a training and test set of $N = 250$ and $N_{test} = 1000$ data points, respectively. There are two inputs ($n = 2$) and each class is an equal mixture of two normal distributions with the same covariance matrices. Both classes have the same prior probability $P(y{=}{+}1) = P(y{=}{-}1) = 1/2$. The training data are visualized in Figure 3, with class $+1$ and class -1 depicted by $+$ and $\times$, respectively.

In a first step, both inputs $x^{(1)}$ and $x^{(2)}$ where normalized to zero mean and unit variance (Bishop, 1995). For the kernel function K of the model $\mathcal{H}$, an RBF-kernel with parameter σ was chosen. Assuming a flat prior on the value of $\log \sigma$, the optimal σ_{MP} was selected by maximizing $p(D|\mathcal{H}_j) = p(D|\log \sigma_j)$, given by (64). The maximum likelihood is obtained for $\sigma_{\text{MP}} = 1.3110$. This yields a test set performance of 90.6% for both LS-SVM classifiers. We also trained a Gaussian Process for classification with the Flexible Bayesian Modeling toolbox (Neal, 1997). A logistic model with constant term and RBF-kernel in the covariance function yielded an average test set performance of 89.9%, which is not significantly different from the LS-SVM result given the standard deviation from Table 2. This Table is further discussed in the next Subsection. In the logistic model, the parameters are directly optimized with respect to the output probability using sampling techniques. The LS-SVM classifier formulation assumes a Gaussian distribution on the errors between the projected feature vectors and the targets (or class centers), which allows to derive analytic expressions on the three levels of inference.

The evolution of the posterior class probabilities $P(y{=}{+}1|x, D, \mathcal{H})$ is plotted in Figure 2 for $x^{(1)} \in [-1.5,\ 1.5]$ and $x^{(2)} \in [-0.5,\ 1.5]$. The corresponding contour plot is given in Figure 3, together with the location of the training points. Notice how the uncertainty on the class labels increases as the new input x is further away from the training data. The value z_{MP} decreases while the variance $\sigma_{z,t}^2$ increases when moving away from the training data.

We also intentionally unbalanced the test set by defining a new test set from the original set: the negatively and positively labeled instances of the original set are repeated three times and once in the new set, respectively. This corresponds to prior class probabilities $P(y = -1) = 0.75$ and $P(y = +1) = 0.25$. Taking these class probabilities not into account, a test set accuracy of 90.9% is obtained, while one achieves a classification performance of 92.5% when the prior class probabilities are taken into account.

7.2 Empirical Results on Binary Classification Datasets

The test set classification performance of the Bayesian (Bay) LS-SVM classifier with RBF-kernel was assessed on 10 publicly available binary classification datasets. We compare the results with LS-SVM and SVM classification and GP regression (LS-SVM without bias term) where the hyperparameter and kernel parameter are tuned by 10-fold cross-validation (CV10). The BUPA Liver Disorders (`bld`), the Statlog German Credit (`gcr`), the Statlog Heart Disease (`hea`), the John Hopkins University Ionosphere (`ion`), the Pima Indians Diabetes (`pid`), the Sonar (`snr`) and the Wisconsin Breast Cancer (`wbc`) datasets were retrieved from the UCI benchmark repository (Blake & Merz, 1998). The synthetic dataset (`rsy`) and Leptograpsus crabs (`cra`) are described in (Ripley, 1996). The Titanic data (`tit`) was obtained from Delve. Each dataset was split up into a training (2/3) and test set (1/3), except for the `rsy` dataset, where we used $N = 250$ and $N_{test} = 1000$. Each data set was randomized ten times in order to reduce possible sensitivities in the test set performances to the choice of training and test set.

For each randomization, the design procedure from Subsection 6.1 was used to estimate μ and ζ from the training data for the Bayesian LS-SVM, while selecting σ from a candidate set $\Sigma = [\sigma_1, \sigma_2, ..., \sigma_j, ..., \sigma_{N_s}]$ using model comparison. The classification decision (LS-SVM BayM) is made by the Bayesian decision rule (47) using the moderated output and is compared with the classifier (22) which is denoted by (LS-SVM Bay). A 10-fold cross-validation (LS-SVM, SVM and GP CV10) procedure was used to select the parameters[3] γ or C and σ yielding best CV10 performance from the set Σ and an additional set $\Gamma = [\gamma_1, \gamma_2, ..., \gamma_{N_g}]$. The same sets were used for each algorithm. In a second step, more refined sets were defined for each algorithm[4] in order to select the optimal parameters. The classification decisions were obtained from (4) with the corresponding w and b determined in the dual space for each algorithm. We also designed the GP regressor within the evidence framework for a GP with RBF-kernel (GP Bay) and for a GP with RBF-kernel and an

[3]Notice that the parameter C of the SVM plays a similar role as the parameter γ of the LS-SVM.

[4]We used the matlab SVM toolbox (Cawley, 2000), while the GP CV10 was obtained from the linear system (23).

additional bias term b in the kernel function ($\mathrm{GP_b}$ Bay).

In Table 1, we report the average test set performance and sample standard deviation on ten randomizations for each dataset (De Groot, 1986). The best average test set performance was underlined and denoted in bold face for each data set. Bold face script is used to tabulate performances that are not significantly different at the 5% level from the top performance using a two-tailed paired t-test. Statistically significant underperformances at the 1% level are emphasized. Other performances are tabulated using normal script. Since the observations are not independent, we remark that the t-test is used here only as a heuristic approach to show that the average accuracies on the ten randomizations can be considered to different. Ranks are assigned to each algorithm starting from 1 for the best average performance. Averaging over all datasets, a Wilcoxon signed rank test of equality of medians is carried out on both Average Performance (AP) and Average Ranks (AR). Finally, the significance Probability of a Sign Test ($\mathrm{P_{ST}}$) is reported comparing each algorithm to the algorithm with best performance (LS-SVM CV10). These results are denoted in the same way as the performances on each individual data set.

No significant differences are obtained between the different algorithms. Comparing SVM CV10 with GP and LS-SVM CV10 it is observed that similar results are obtained with all three algorithms, which means that the loss of sparseness does not result into a degradation of the generalization performance on these data sets. It is also observed that the LS-SVM and GP designed within the evidence framework yield consistently comparable results when compared with CV10, which indicates that the Gaussian assumptions of the evidence framework hold well for the natural domains at hand. Estimating the bias term b in the GP kernel function by Bayesian inference on level 3 yields comparable, but different results than the LS-SVM formulation where the bias term b is obtained on the first level. Finally, it is observed that assigning the class label from the moderated output (47) also yields comparable results with respect to the classifier (22), but the latter formulation does yield an analytic expression to adjust the bias term for different prior class probabilities, which is useful, e.g., in the case of unbalanced training and test set or in the case of different classification costs.

8 Conclusions

In this paper, a Bayesian framework has been related to the Least Squares Support Vector Machine (LS-SVM) classifier formulation. This least squares formulation was obtained by modifying the Support Vector Machine (SVM) formulation and implicitly corresponds to a regression problem with binary targets. The LS-SVM formulation is also related kernel Fisher Discriminant Analysis. Without the bias term in the LS-SVM formulation, the dual space formulation is equivalent to Gaussian Processes (GPs) for regression. The least squares regression approach to classification allows to derive analytic expressions on the different levels of inference. On the first level, the model parameters are obtained from solving a linear Karush-Kuhn-Tucker system in the dual space. The regularization hyperparameters are obtained from a scalar optimization problem on the second level, while the kernel parameter is determined on the third level by model comparison. Starting from the LS-SVM feature space formulation, the analytic expressions obtained in the dual space are similar to the expressions obtained for GPs. The use of an unregularized bias term in the LS-SVM formulation results into a zero mean training error and an implicit centering of the Gram matrix in the feature space, also used in kernel PCA. The corresponding eigenvalues can be used to obtain improved bounds for SVMs. Within the evidence framework, these eigenvalues are used to control the capacity by the regularization term. Class probabilities are obtained within the defined probabilistic framework by marginalizing over the model- and hyperparameters. By combination of the posterior class probabilities with an appropriate decision rule, class labels can be assigned in an optimal way. Comparing LS-SVM, SVM classification and GP regression with binary targets on ten normalized public domain datasets, no significant differences in performance are observed. The Gaussian assumptions made in the LS-SVM formulation and the related Bayesian framework allow to obtain analytical expressions on all levels of inference using reliable numerical techniques and algorithms.

Acknowledgments

The authors would like to thank the two anonymous reviewers for constructive comments and would also like to thank David MacKay for helpful comments related to the second and third level of inference. T. Van Gestel and J.A.K. Suykens are a Research Assistant and a Postdoctoral Researcher with the Fund for Scientific Research-Flanders (FWO-Vlaanderen), respectively. This work was partially supported by grants and projects from the Flemish Gov.: (Research council KULeuven: Grants, GOA-Mefisto 666; FWO-Vlaanderen: Grants, res. proj. G.0240.99, G.0256.97, and comm. (ICCoS and ANMMM); AWI: Bil. Int. Coll.; IWT: STWW Eureka SINOPSYS, IMPACT); from the Belgian Fed. Gov. (Interuniv. Attr. Poles: IUAP-IV/02, IV/24; Program Dur. Dev.); from the Eur. Comm.: (TMR Netw. (Alapedes, Niconet); Science: ERNSI). The scientific responsibility is assumed by its authors.

References

Baudat, G., & Anouar, F. (2000). Generalized Discriminant Analysis Using a Kernel Approach. *Neural Computation,* 12, 2385-2404.

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press.

Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. Irvine, CA: University of California, Dept. of Information and Computer Science.

Brown, P.J. (1977). Centering and scaling in ridge regression. *Technometrics,* 19, 35-36.

Cawley, G.C. (2000). MATLAB Support Vector Machine Toolbox (v0.54β). [http://theoval.sys.uea.ac.uk/∼gcc/svm/toolbox]. University of East Anglia, School of Information Systems, Norwich, Norfolk, U.K.

Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines.* Cambridge University Press.

De Groot, M.H. (1986). *Probability and Statistics* (Second Edition). Reading, MA: Addison-Wesley.

Duda, R.O., & Hart, P.E. (1973). *Pattern Classification and Scene Analysis.* John Wiley, New York.

Evgeniou, T., Pontil, M., Papageorgiou, C., & Poggio, T. (2000) Image representations for object detection using kernel classifiers. In *Proc. Fourth Asian Conference on Computer Vision (ACCV 2000)*, (pp. 687-692).

Gibbs, M.N. (1997). *Bayesian Gaussian Processes for Regression and Classification.* PhD thesis, University of Cambridge.

Golub, G.H., & Van Loan, C.F. (1989). *Matrix Computations*. Baltimore MD: Johns Hopkins University Press.

Gull, S.F. (1988). Bayesian Inductive Inference and Maximum Entropy. In G.J. Erickson & R. Smith (Eds.), *Maximum-Entropy and Bayesian Methods in Science and Engineering, 1*, 73-74. Kluwer Academic Publishers.

Horn, R.A., & Johnson, C.R. (1991). *Topics in Matrix Analysis*. Cambridge University Press.

Kwok, J.T. (1999). Moderating the outputs of support vector machine classifiers. *IEEE Trans. on Neural Networks, 10*, 1018-1031.

Kwok, J.T. (2000). The evidence framework applied to support vector machines. *IEEE Trans. on Neural Networks, 11*, 1162-1173.

MacKay, D.J.C. (1992). The evidence framework applied to classification networks. *Neural Computation, 4(5)*, 698-714.

MacKay, D.J.C. (1995). Probable Networks and Plausible Predictions - A Review of Practical Bayesian Methods for Supervised Neural Networks. *Network: Computation in Neural Systems, 6*, 469-505.

MacKay, D.J.C. (1999). Comparison of Approximate Methods for Handling Hyperparameters. *Neural Computation, 11(5)*, 1035-1068.

Mika, S., Rätsch, G., & Müller, K.-R. (2001). A mathematical programming approach to the Kernel Fisher algorithm. In *Proc. Neural Information Processing Systems (NIPS 2000)*, to appear.

Mukherjee, S., Tamayo, P., Mesirov, J.P., Slonim, D., Verri, A., & Poggio, T. (1999). Support vector machine classification of microarray data. *CBCL Paper #182/AI Memo #1676.* Massachusetts Institute of Technology, Cambridge, MA.

Neal, R.M. (1997). Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification. *Technical Report CRG-TR-97-2*, Dept. of Computer Science, University of Toronto.

Platt, J. (1999). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In A. Smola, P. Bartlett, B. Schölkopf, & D. Schuurmans (Eds.), *Advances in Large Margin Classifiers.* MIT Press.

Rasmussen, C. (1996). *Evaluation of Gaussian Processes and other Methods for Non-Linear Regression.* PhD. Thesis, Dept. of Computer Science, University of Toronto, Canada.

Ripley, B.D. (1996). *Pattern Recognition and Neural Networks.* Cambridge University Press.

Rosipal, R., Girolami, M. (2001) An Expectation-Maximization Approach to Nonlinear Component Analysis. *Neural Computation 13(3)*, 505-510.

Saunders, C., Gammerman, A., Vovk, K. (1998). Ridge regression learning algorithm in dual variables. In *Proc. of the 15th Int. Conf. on Machine Learing (ICML-98)*, Madison-Wisconsin, (pp. 515-521).

Schölkopf, B., Smola, A., & Müller, K.-M. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation, 10*, 1299-1319.

Schölkopf, B., Shawe-Taylor, J., Smola, A., & Williamson, R.C. (1999). Kernel-dependent Support Vector Error Bounds. In *Proc. of the 9th Int. Conf. on Artificial Neural Networks (ICANN-99)*, Edinburgh, UK, (pp. 304-309).

Shashua, A. (1999). On the Equivalence Between the Support Vector Machine for Classification and Sparsified Fisher's Linear Discriminant. *Neural Processing Letters, 9*, 129-139.

Smola, A., Schölkopf, B., & Müller, K.-R. (1998). The connection between regularization operators and support vector kernels. *Neural Networks, 11*, 637-649.

Sollich, P. (2000). Probabilistic methods for support vector machines. In S.A. Solla, T.K. Leen & K.-R. Müller (Eds.), *Advances in Neural Information Processing Systems, 12, (NIPS'99)*. MIT Press.

Suykens, J.A.K., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters, 9*, 293-300.

Suykens, J.A.K. (2000). Least Squares Support Vector Machines for Classification and Nonlinear Modelling. *Neural Network World, 10*, 29-48.

Suykens, J.A.K., & Vandewalle, J. (2000). Recurrent least squares support vector machines. *IEEE Transactions on Circuits and Systems-I, 47*, 1109-1114.

Suykens, J.A.K., Vandewalle, J., & De Moor, B. (2001). Optimal control by least squares support vector machines. *Neural Networks, 14*, 23-35.

Van Gestel, T., Suykens, J.A.K., Baestaens, D.-E., Lambrechts, A., Lanckriet, G., Vandaele, B., De Moor, B., & Vandewalle, J. (2001). Predicting Financial Time Series using Least Squares Support Vector Machines within the Evidence Framework. *IEEE Transactions on Neural Networks*, to appear.

Vapnik, V. (1995). *The nature of statistical learning theory.* Springer-Verlag, New-York.

Vapnik, V. (1998). *Statistical learning theory.* John Wiley, New-York.

Williams, C.K.I. (1998) Prediction with Gaussian Processes: from Linear Regression to Linear Prediction and Beyond. In M.I. Jordan (Ed.), *Learning and Inference in Graphical Models*. Kluwer Academic Press.

Williams. C.K.I., & Barber, D. (1998). Bayesian Classification with Gaussian Processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*, 1342-1351.

A Derivations Level 1 Inference

In the expression for the variances $\sigma^2_{e_+}$ and $\sigma^2_{e_-}$, the upper left $n_f \times n_f$ block of the covariance matrix $Q = H^{-1}$ is needed. Therefore, we first calculate the inverse of the block Hessian H. Using $\Upsilon = [\varphi(x_1), \ldots, \varphi(x_N)]$, with $\Omega = \Upsilon^T \Upsilon$, the expressions for the block matrices in the Hessian (37) are $H_{11} = \mu I_{n_f} + \zeta \Upsilon \Upsilon^T$, $H_{12} = \zeta \Upsilon 1_v$ and $H_{22} = N\zeta$. Calculating the inverse of the block matrix, the inverse Hessian is obtained as follows

$$H^{-1} = \left(\begin{bmatrix} I_{n_f} & X \\ 0 & 1 \end{bmatrix} \begin{bmatrix} H_{11} - H_{12} H_{22}^{-1} H_{12}^T & 0 \\ 0 & H_{22} \end{bmatrix} \begin{bmatrix} I_{n_f} & 0 \\ X^T & 1 \end{bmatrix} \right)^{-1} \tag{65}$$

$$= \begin{bmatrix} (\mu I_{n_f} + \zeta G)^{-1} & -(\mu I_{n_f} + \zeta G)^{-1} H_{12} H_{22}^{-1} \\ -H_{22}^{-1} H_{12}^T (\mu I_{n_f} + \zeta G)^{-1} & H_{22}^{-1} + H_{22}^{-1} H_{12}^T (\mu I_{n_f} + \zeta G)^{-1} H_{12} H_{22}^{-1} \end{bmatrix} \tag{66}$$

with $G = \Upsilon M \Upsilon^T$, $X = H_{12} H_{22}^{-1}$ and where $M = I_N - \frac{1}{N} 1_v 1_v^T$ is the idempotent centering matrix with rank $N-1$. Observe that the above derivation results into a centered Gram matrix G, as is also done in kernel PCA (Schölkopf *et al.*, 1998b). The eigenvalues of the centered Gram matrix can be used to derive improved bounds for SVM classifiers (Schölkopf *et al.*, 1999). In the Bayesian framework of this paper, the eigenvalues of the centered Gram matrix are also used on level 2 and 3 of Bayesian inference to determine the amount of weight decay and to select the kernel parameter, respectively. The inverse $(\mu I_{n_f} + \zeta G)^{-1}$ will be calculated using the eigenvalue decomposition of the symmetric matrix $G = G^T = P^T D_{G,f} P = P_1^T D_G P_1$, with $P = [P_1 \, P_2]$ a unitary matrix and with the diagonal matrix $D_G = \mathrm{diag}([\lambda_{G,1}, ..., \lambda_{G,N_{eff}}])$ containing the N_{eff} non-zero diagonal elements of full size diagonal matrix $D_{G,f} \in \mathbb{R}^{n_f}$. The matrix P_1 corresponds to the eigenspace corresponding to the non-zero eigenvalues and the null space is denoted by P_2. There are maximally $N-1$ eigenvalues $\lambda_{G,i} > 0$ and their corresponding eigenvectors $\nu_{G,i}$ are a linear combination of ΥM: $\nu_{G,i} = c_{G,i} \Upsilon M v_{G,i}$, with $c_{G,i}$ a normalization constant such that $\nu_{G,i}^T \nu_{G,i} = 1$. The eigenvalue problem we need to solve is the following:

$$\Upsilon M \Upsilon^T (\Upsilon M v_{G,i}) \;=\; \lambda_{G,i} (\Upsilon M v_{G,i}). \tag{67}$$

Multiplication of (67) to the left with $M\Upsilon^T$ and applying the Mercer condition yields: $M\Omega M\Omega M v_{G,i} = \lambda_{G,i} M\Omega M v_{G,i}$,which is a generalized eigenvalue problem of dimension N. An eigenvector $v_{G,i}$ corresponding to a nonzero eigenvalue $\lambda_{G,i}$ is also a solution of

$$M\Omega M v_{G,i} \;=\; \lambda_{G,i} v_{G,i}. \tag{68}$$

since $M\Omega M\Omega M v_{G,i} = \lambda_{G,i} M\Omega M v_{G,i} \neq 0$. By applying the normality condition of $\nu_{G,i}$, which corresponds to $c_{G,i} = 1/\sqrt{v_{G,i}^T M\Omega M v_{G,i}}$, one finally obtains $P_1 = [\nu_{G,1} \ldots \nu_{G,N_{eff}}]$ where $\nu_{G,i} = \frac{1}{\sqrt{v_{G,i}^T M\Omega M v_{G,i}}} \Upsilon M v_{G,i}$, and $P_1 = \Upsilon M U_G$, with $U_G(:,i) = \frac{1}{\sqrt{v_{G,i}^T M\Omega M v_{G,i}}} v_{G,i} = \lambda_{G,i}^{-1/2} v_{G,i}$, $i = 1, \ldots, N_{eff}$. The remaining $n_f - N_{eff}$ dimensional orthonormal null space P_2 of G cannot be explicitly calculated, but using that $[P_1 \, P_2]$ is a unitary matrix, we have $P_2 P_2^T = I_{n_f} - P_1 P_1^T$. Observe that this is different from (Kwok, 1999; 2000), where the space P_2 is neglected. This yields

$$(\mu I_{n_f} + \zeta G)^{-1} = P_1((\mu I_{N_{eff}} + \zeta D_G)^{-1} - \mu^{-1} I_{N_{eff}}) P_1^T + \mu^{-1} I_{n_f}.$$

By defining $\theta(x) = \Upsilon^T \varphi(x)$ and applying the Mercer condition in (40) one finally obtains the expression (41).

For large N, the calculation of all eigenvalues $\lambda_{G,i}$ and corresponding eigenvectors ν_i, $i = 1, \ldots, N$ may require long computations. One may expect that little error is introduced by putting the $N - r_G$ smallest eigenvalues, $\mu \gg \zeta\lambda_{G,i}$, of $G = G^T \geq 0$. This corresponds to an optimal rank r_G approximation of $(\mu I_{n_f} + \zeta G)^{-1}$. Indeed, calling G_R the rank r_G approximation of G, we obtain $\min_{G_R} ||(\mu I_{n_f} + \zeta G)^{-1} - (\mu I_{n_f} + \zeta G_R)^{-1}||_F$. Using the Sherman-Morrison-Woodbury formula (Golub & Van Loan, 1989) this becomes: $||(\mu I_{n_f} + \zeta G)^{-1} - (\mu I_{n_f} + \zeta G_R)^{-1}||_F = ||\frac{\zeta}{\mu}(\mu I_{n_f} + \zeta G)^{-1} G - \frac{\zeta}{\mu}(\mu I_{n_f} + \zeta G_R)^{-1} G_R||_F$. The optimal rank r_G approximation for $(\mu I_{n_f} + \zeta G)^{-1}G$ is obtained by putting its smallest eigenvalues to zero. Using the eigenvalue decomposition of G, these eigenvalues are $\frac{\lambda_{G,i}}{\mu+\zeta\lambda_{G,i}}$. The smallest eigenvalues of $(\mu I_{n_f} + \zeta G)^{-1}G$ correspond to the smallest eigenvalues of G. Hence, the optimal rank r_G approximation is obtained by putting the smallest $N - r_G$ eigenvalues to zero. Also notice that σ_z^2 is increased by putting $\lambda_{G,i}$ equal to zero. A decrease of the variance would introduce a false amount of certainty on the output.

B Derivations Level 2 and 3 Inference

First, an expression for $\det(H)$ is given using the eigenvalues of G. By block diagonalizing (37), $\det(H)$ is not changed, see, e.g., (Horn & Johnson, 1991). From (65), we obtain $\det H = N\zeta \det(\mu I_{n_f} + \zeta G)$. The determinant is the product of the eigenvalues, this yields $\det H = N\zeta\mu^{n_f - N_{eff}} \prod_{i=1}^{N_{eff}} (\mu + \zeta\lambda_{G,i})$. Due to the regularization term μE_W, the Hessian is regular. The inverse exists and we can write $\det H^{-1} = 1/\det H$.

Using (30) the simulated error $e_i = y_i - (w_{\mathrm{MP}}^T \varphi(x_i) + b_{\mathrm{MP}})$ can also be written as $e_i = y_i - \hat{m}_Y - w_{\mathrm{MP}}^T(\varphi(x_i) - \hat{m}_\Upsilon)$, with $\hat{m}_Y = \sum_{i=1}^N y_i/N$ and $\hat{m}_\Upsilon = \sum_{i=1}^N \varphi(x_i)/N$. Since $w_{\mathrm{MP}} = (\Upsilon M \Upsilon^T + \gamma^{-1} I_{n_f})^{-1}\Upsilon M Y$, the error term $E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}})$ is equal to

$$\begin{aligned} E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}}) = \frac{1}{2}(Y - \hat{m}_Y 1_v)^T (I_N - M\Upsilon(\Upsilon M \Upsilon^T + \gamma^{-1} I_{n_f})^{-1}\Upsilon M)^2 \\ (Y - \hat{m}_Y 1_v) = \frac{1}{2\gamma^2} Y^T M V_G (D_G + \gamma^{-1} I_{n_{eff}})^{-2} V_G^T M Y, \end{aligned} \tag{69}$$

where we used the eigenvalue decomposition of $G = \Upsilon M \Upsilon$. In a similar way one can obtain the expression for E_W in the dual space starting from $w_{\mathrm{MP}} = (\Upsilon M \Upsilon^T + \gamma^{-1} I_{n_f})^{-1}\Upsilon M Y$:

$$E_W(w_{\mathrm{MP}}) = \frac{1}{2} Y^T M V_G D_G (D_G + \gamma^{-1} I_{n_{eff}})^{-2} V_G^T M Y. \tag{70}$$

The sum $E_W(w_{\mathrm{MP}}) + \gamma E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}})$ is then equal to

$$\begin{aligned} E_W(w_{\mathrm{MP}}) + \gamma E_D(w_{\mathrm{MP}}, b_{\mathrm{MP}}) &= \frac{1}{2} Y^T M V_G (D_G + \gamma^{-1} I_{n_{eff}})^{-1} V_G^T M Y \\ &= \frac{1}{2} Y^T M (M \Omega M + \gamma^{-1} I_n)^{-1} M Y, \end{aligned} \tag{71}$$

which is the same expression as obtained with GP when no centering M is applied on the outputs Y and the feature vectors Υ.

Captions of figures

Figure 1: Two Gaussian distributed classes with the same covariance matrix are separated by the hyperplane $w^T\varphi(x)+b=0$ in the feature space. The class center of classes -1 and $+1$ are located on the hyperplanes $w^T\varphi(x)+b=-1$ and $w^T\varphi(x)+b=1$, respectively. The projections of the features onto the linear discriminant result in Gaussian distributed errors with variance ζ^{-1} around the targets -1 and $+1$.

Figure 2: The posterior class probability $P(y{=}{+}1|x,D,\mathcal{H})$ as a function of the inputs $x^{(1)}$ and $x^{(2)}$ for the `rsy` dataset.

Figure 3: Contour plot of the posterior class probability $P(y=+1|x,D,\mathcal{H})$ for the `rsy` dataset. The training data are marked with $+$ and $\times$ for class $y=+1$ and $y=-1$, respectively.

Captions of tables

Table 1: Comparison of the 10 times randomized test set performances of LS-SVMs, GPs and SVM. Both CV10 and the Bayesian (Bay) framework were used to design the LS-SVMs. For the Bayesian LS-SVM the class label was assigned using the moderated output (BayM). A Radial Basis Function (RBF) kernel was used for all models. The model $\mathrm{GP_b}$ has an extra bias term in the kernel function. The Average Performance (A.P.), Average Rank (A.R.) and Probability of equal medians using the Sign Test ($\mathrm{P_{ST}}$) taken over all domains are reported in the last three rows. Best performances are underlined and denoted in bold face, performances not significantly different at the 5% level are denoted in bold face, performances significantly different at the 1% level are emphasized. No significant differences are observed between the different algorithms.

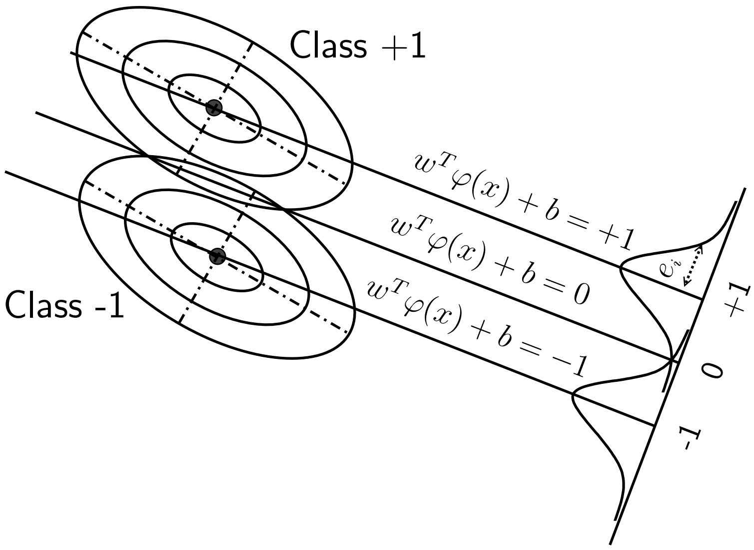

Figure 1: Van Gestel *et al*, Manuscript no. 2281

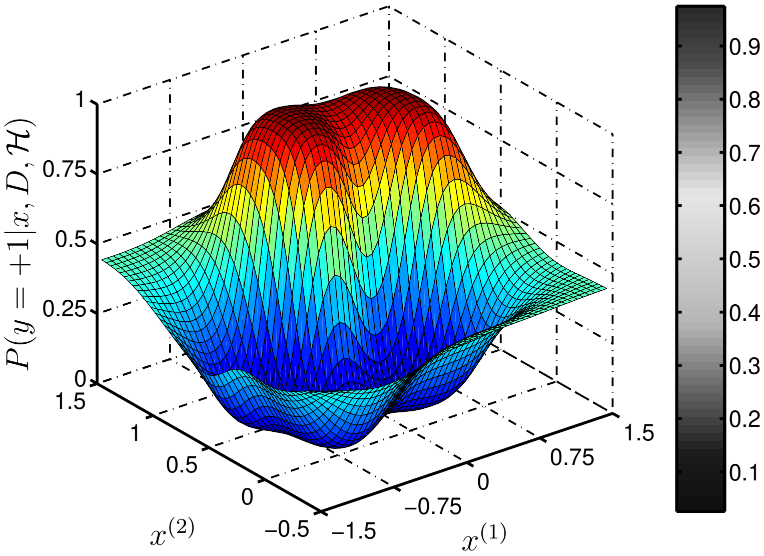

Figure 2: Van Gestel *et al*, Manuscript no. 2281

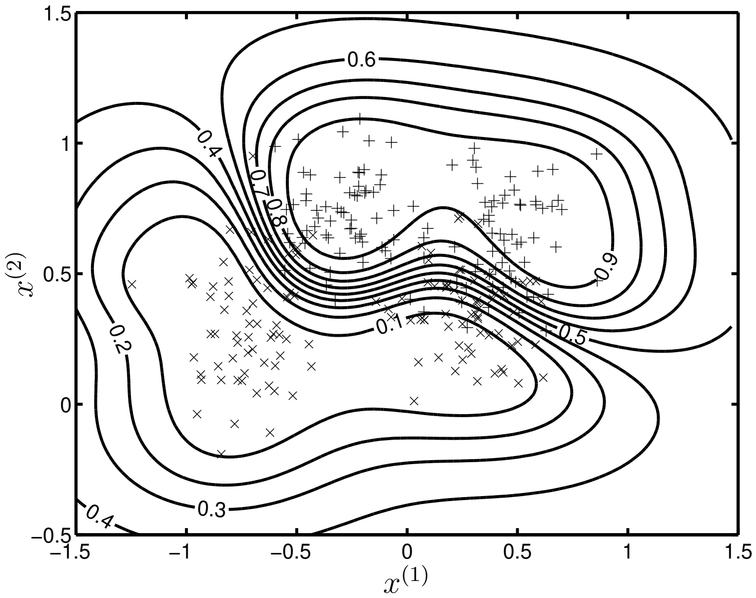

Figure 3: Van Gestel *et al*, Manuscript no. 2281

	n	N	N_{test}	N_{tot}	LS-SVM (BayM)	LS-SVM (Bay)	LS-SVM (CV10)	SVM (CV10)	GP (Bay)	GP_{b} (Bay)	GP (CV10)
bld	6	230	115	345	**69.4**(2.9)	**69.4**(3.1)	**69.4**(3.4)	**69.2**(3.5)	**69.2**(2.7)	**68.9**(3.3)	**69.7**(4.0)
cra	6	133	67	200	**96.7**(1.5)	**96.7**(1.5)	**96.9**(1.6)	**95.1**(3.2)	**96.4**(2.5)	94.8(3.2)	**96.9**(2.4)
gcr	20	666	334	1000	*73.1*(3.8)	73.5(3.9)	**75.6**(1.8)	*74.9*(1.7)	**76.2**(1.4)	**75.9**(1.7)	**75.4**(2.0)
hea	13	180	90	270	**83.6**(5.1)	**83.2**(5.2)	**84.3**(5.3)	**83.4**(4.4)	**83.1**(5.5)	**83.7**(4.9)	**84.1**(5.2)
ion	33	234	117	351	95.6(0.9)	**96.2**(1.0)	**95.6**(2.0)	95.4(1.7)	*91.0*(2.3)	*94.4*(1.9)	*92.4*(2.4)
pid	8	512	256	768	**77.3**(3.1)	**77.5**(2.8)	**77.3**(3.0)	**76.9**(2.9)	**77.6**(2.9)	**77.5**(2.7)	**77.2**(3.0)
rsy	2	250	1000	1250	**90.2**(0.7)	**90.2**(0.6)	89.6(1.1)	**89.7**(0.8)	**90.2**(0.7)	**90.1**(0.8)	**89.9**(0.8)
snr	60	138	70	208	76.7(5.6)	**78.0**(5.2)	**77.9**(4.2)	76.3(5.3)	**78.6**(4.9)	**75.7**(6.1)	**76.6**(7.2)
tit	3	1467	734	2201	**78.8**(1.1)	**78.7**(1.1)	**78.7**(1.1)	**78.7**(1.1)	**78.5**(1.0)	77.2(1.9)	**78.7**(1.2)
wbc	9	455	228	683	95.9(0.6)	*95.7*(0.5)	**96.2**(0.7)	**96.2**(0.8)	95.8(0.7)	*93.7*(2.0)	**96.5**(0.7)
A.P.					**83.7**	**83.9**	**84.1**	83.6	**83.7**	83.2	**83.7**
A.R.					**2.3**	**2.5**	**2.5**	**3.8**	**3.2**	**4.2**	**2.6**
P_{ST}					**1.000**	**0.754**	**1.000**	**0.344**	**0.754**	**0.344**	**0.508**

Table 1: Van Gestel *et al*, Manuscript no. 2281