Machine Learning, 62, 217–252, 2006
2006 Springer Science + Business Media, Inc. Manufactured in The Netherlands.
DOI: 10.1007/s10994-005-5315-x

Additive Regularization Trade-Off: Fusion of Training and Validation Levels in Kernel Methods

K. PELCKMANS kristiaan.pelckmans@esat.kuleuven.be
J. A. K. SUYKENS johan.suykens@esat.kuleuven.be
B. DE MOOR
K.U. Leuven, ESAT-SCD-SISTA, Kasteelpark Arenberg 10, B-3001 Leuven, Heverlee, Belgium

Editor: Dale Schuurmans

Published online: 29 January 2006

Abstract. This paper presents a convex optimization perspective towards the task of tuning the regularization trade-off with validation and cross-validation criteria in the context of kernel machines. We focus on the problem of tuning the regularization trade-off in the context of Least Squares Support Vector Machines (LS-SVMs) for function approximation and classification. By adopting an additive regularization trade-off scheme, the task of tuning the regularization trade-off with respect to a validation and cross-validation criterion can be written as a convex optimization problem. The solution of this problem then contains both the optimal regularization constants with respect to the model selection criterion at hand, and the corresponding training solution. We refer to such formulations as the fusion of training with model selection. The major tool to accomplish this task is found in the primal-dual derivations as occuring in convex optimization theory. The paper advances the discussion by relating the additive regularization trade-off scheme with the classical Tikhonov scheme. Motivations are given for the usefulness of the former scheme. Furthermore, it is illustrated how to restrict the additive trade-off scheme towards the solution path corresponding with a Tikhonov scheme while retaining convexity of the overall problem of fusion of model selection and training. We relate such a scheme with an ensemble learning problem and with stability of learning machines. The approach is illustrated on a number of artificial and benchmark datasets relating the proposed method with the classical practice of tuning the Tikhonov scheme with a cross-validation measure.

Keywords: Least Squares Support Vector Machines, regulatization, model selection, optimizatioon

1. Introduction

Regularization has a rich history which dates back to the theory of inverseill-posed and ill-conditioned problems (Ivanov, 1976; Tikhonov & Arsenin, 1977; Morozov, 1984). Hoerl and Kennard (1970) have suggested a method of combatting multicollinearity, called ridge regression. The relation between both is discussed amongst others in Bertero, Poggio & Torre (1988), and Hastie, Tibshirani & Friedman et al., (2001). Non-parametric approaches for observational data relying on similar penalized cost functions include splines (Schumaker, 1981; Wahba, 1990), multilayer perceptrons (Bishop, 1995; MacKay, 1992), regularization networks (Poggio & Girosi, Suykens, & 1990), support vector machines (SVMs) (Vapnik, 1998) and least squares support vector machines (LS-SVMs) (Suykens & Vandewalle et al., 1999; Suykens et al., 2002b, 2003). The latter two are characterized by primal-dual optimization formulations with use of a positive definite kernel and their solution follows from convex programs. Standard SVMs lead to solving convex quadratic programming problems while LS-SVMs for classification and regression lead to solving a set of linear equations. Other primal-dual LS-SVM formulations have been given for kernel principal component analysis, kernel canonical correlation analysis, kernel partial least squares, recurrent networks and optimal control (Suykens et al., 2002b). Advantages of adopting the primal-dual optimization point of view characterizing (LS-)SVM are found amongst others in the flexibility of incorporating additional terms (as the bias term) and additional constraints. The relations between LS-SVMs and related methods as SVMs (Vapnik, 1998), Gaussian Processes (MacKay, 1998) and others are discussed extensively in Suykens et al. (2002b).

The relative importance between the smoothness of the solution and the norm of the residuals in the cost function involves a tuning parameter, usually called the regularization constant. The determination of regularization constants is important in order to achieve good generalization performance with the trained model and is an important problem in statistics and learning theory (Hastie, Tibshirani & Friedman et al., 2001; Vapnik, 1998; Suykens et al., 2003). Several methods have been proposed including validation (Val) and cross-validation (CV) (Stone, 1974; Burman, 1989), generalized cross validation (Golub, Heath & Wahba et al., 1979), Akaike information criteria (Akaike, 1973), Mallows C_p (Mallows, 1973), minimum description length (Rissanen, 1978), bias-variance trade-off (Hoerl & Kennard, 1970), L-curve methods (Hansen, 1992) and many others. For classification problems in pattern recognition, the Receiver Operating Characteristic (ROC) curve has been proposed for model selection (Hanley & McNeil, 1982). In the context of non-Gaussian noise models and outliers, robust counterparts have been presented in De Brabanter et al. (2002, 2003). Translation of a priori knowledge (e.g. norm of the solution, norm of the residuals or the noise variance) into an appropriate regularization constant has been described respectively as the secular equation (Golub & Van Loan, 1989), in Morozov's discrepancy principle (Morozov, 1984) and Pelckmans et al. (2003). In the specific context of kernel machines (Vapnik, 1998) amongst others Chapelle and Vapnik (2000) proposed criteria with bounds on the generalization error based on geometrical concepts (VC bounds, optimal margin and support vector span (Schölkopf & Smola, 2002)) to determine the regularization constant. A bound based on the leave-one-out cross-validation error was introduced in (Kearns, 1997). Bounds on the generalization error with analysis of the approximation and sample error were investigated in Cucker and Smale (2002). Efficient methods for calculating the leave-one-out cross-validation criterion for some kernel algorithms based on the matrix inversion lemma were described in Craven and Whaba (1979) and by Van Gestel (2002); and Cawley and Talbot (2003) for LS-SVMs specifically. In general, the optimization of criteria for determination of unknown regularization constants often leads to non-convex optimization (or even non-smooth) and computationally intensive schemes (depending on the model selection scheme). In Chapelle et al., (2002) the determination of the tuning parameter is done via solving alternating convex problems. Related research can be found in the literature about learning the kernel (see e.g. Herrmann & Bousquet, 2003; Lanckriet et al., 2004).

This paper takes an optimization point of view (Boyd & Vandenberghe, 2004): we tackle the problem of searching the regularization constant resulting in the optimal model selection criterion. Conceptually, the optimization of the regularization constant can be considered in view of different hierarchical levels (as is also the case in classical approaches) (see Figure 1): on the first level of training, the solution to the LS-SVM follows from a set of linear equations. At this level regularization constants are considered to be fixed. On the second level of model selection, one optimizes over the regularization

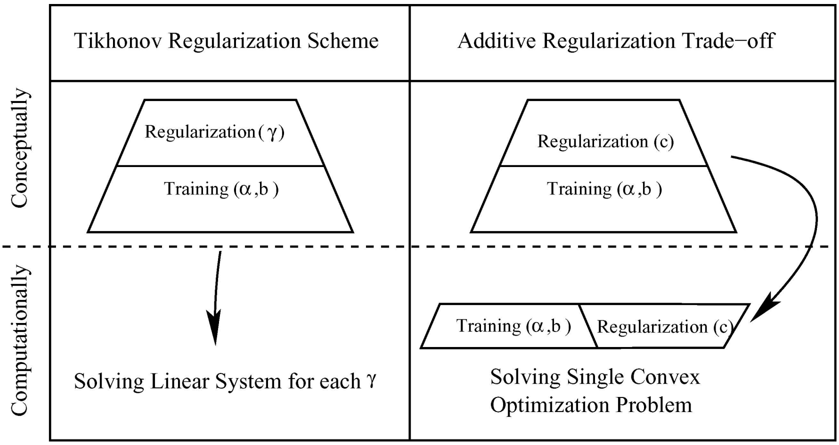


Figure 1. Comparison between the classical Tikhonov regularization scheme and the additive regularization trade-off scheme: Conceptually, training and validation levels are different in both schemes. Computationally, fusion of the training and validation levels results in a single constrained optimization problem. In the case of the additive regularization trade-off this problem becomes a convex problem after fusion.

constants and picks the model with optimal performance. From a computational point of view, the model selection problem can be cast as a constrained optimization problem: model selection amounts to minimizing the validation measure subject to the fact that the training equations hold exactly. This principle is called in this paper the *fusion* of training and validation levels.

An *additive regularization trade-off* scheme is proposed in this paper which penalizes the regularization term in a different way with respect to the loss function. In a classical Tikhonov regularization scheme the regularization constant enters in a multiplicative way with respect to the error variables, while in the proposed additive regularization trade-off scheme the regularization constants enter in an additive way with respect to the error variables. For this additive regularization trade-off scheme the *fusion* of the training and validation level leads to solving a *convex optimization* problem. Both the model parameters and hyperparameters follow as the solution to this problem. Note that also in methods of Bayesian inference one divides the unknowns into parameters and hyperparameters at different hierarchical levels (MacKay, 1992), but this usually leads to non-convex problems and computationally intensive algorithms (e.g. Gibbs sampling and even in the case of approximate solutions). The additive regularization trade-off inherits most properties of the well-known Tikhonov regularization scheme, but differs in the parameterization of the regularization constants. The classical Tikhonov trade-off scheme can be written as a special case. One regularization constant per data point is used which hereby directly influences the distribution of the residuals in the cost function. However, in this case one should carefully prevent *data snooping* (Schölkopf & Smola, 2002, p. 128) because of the fact that one has a regularization constant per data point. Therefor, special attention is paid on limiting the degrees of freedom in the additive regularization scheme. Different restriction schemes are studied (an overview is shown in Table 1).

The described framework is explained for the use of a single validation set as well as in the context of cross-validation. In order to restrict the degrees of freedom in the cross-

Table 1. Overview of the methods used to constrain the regularization constants c in the additive regularization trade-off setting. The basic formulation (8) and (9) has N degrees of freedom, which can cause overfitting on the validation set (data snooping).

	Constraining $\boldsymbol{c}$	Formulation	Reference
1	Minimal norm training and validation errors	$\min_{e,e^v} \|\alpha + c\|_2^2 + \|e^v\|_2^2$	Alg. 3.1
2	Convex solution set	$\mathcal{S}^c_{\gamma(1),\ldots,\gamma(m)}$	Alg. 4.1
3	Constant over cross-validation folds	$c^{(l)} = \mathcal{I}(\mathcal{T}_l, \mathcal{D})c$	Lemma 5.1
4	Minimal training residuals and stability	$\min_{e,e^v} \|\alpha + c\|_2^2 + \xi \|\alpha + c - e^v\|_2^2$	Eq. (52)
5	Tikhonov regularization	$c = (\gamma^{-1} - 1)\alpha \quad \text{s.t. } \gamma \geq 0$	Eq. (70)

validation scheme with additive regularization trade-off, a coupling over the different folds is taken. Straightforward calculation results in a set of linear equations with a number of unknowns (and number of equations) that is proportional to the product of the size of the dataset with the number of folds. It turns out that by exploiting the primal-dual properties, one can further reduce the complexity of the problem to solving a set of linear equations with size proportional to the number of data-points. At this point, it may be interesting to point out the main differences of the proposed method with the literature on learning the kernel (see e.g. Lanckriet et al., 2004) where one proceeds with a model selection criterion based on a statistical bound using training performance and a measure of complexity in terms of the kernel matrix. This method essentially differs from ours by still considering a given complexity control constant and optimizing with respect to training performance.

This paper is organized as follows. Section 2 discusses ridge regression in feature space with primal-dual LS-SVM formulations. The notion of fusion is explained. The additive regularization trade-off is discussed in Section 3. In Section 4, the problem of leakage of information from the validation data to the estimated model parameters is discussed. Ways to prevent this phenomenon are given, including a method with ensemble interpretation. Section 5 extends the framework to a cross-validation setting. Section 6 illustrates the methods on synthetic and real-life data sets. Appendix A discusses the classification case. In Appendix B, it is shown how to recover and to approximate the classical Tikhonov regularization within the additive regularization framework.

2. LS-SVM regressors and fusion of training and validation

2.1. *Ridge regression in feature space*

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times \mathbb{R}$ be the training data with inputs x_i and outputs y_i. Consider the regression model $y_i = f(x_i) + e_i$ where $x_1, \ldots, x_N$ are deterministic points (fixed design), $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is an unknown real-valued smooth function and $e_1, \ldots, e_N$ are uncorrelated random errors with $E[e_i] = 0$, $E[(e_i)^2] = \sigma_e^2 < \infty$. A validation set is assumed to originate from the same underlying function f perturbed with i.i.d. noise with exactly the same properties as the training set. The data points of a validation set are indexed by $j = 1, \ldots, n$ and denoted as $\mathcal{D}^v = \{(x_j^v, y_j^v)\}_{j=1}^n$.

The cost function for the training of the Least Squares Support Vector Machine model $f(x) = w^T\varphi(x) + b$ in the primal space, where $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{n_h}$ denotes the potentially infinite dimensional feature map, is given by Suykens et al., (2002)

$$\min_{w,b,e_i} \mathcal{J}_\gamma(w,e) = \frac{1}{2}w^T w + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 \quad \text{s.t.} \quad w^T\varphi(x_i) + b + e_i = y_i \quad \forall i = 1, \ldots, N. \tag{1}$$

The solution corresponds to a form of ridge regression (Saunders, Gammerman & Vovk 1998), regularization networks (Poggio & Girosi, 1990), Gaussian Processes (MacKay, 1990) and Kriging (Cressie, 1990). The formulation includes a bias term as in most standard SVM formulations, which is usually not the case in the other methods. Note that the regularization constant γ appears here as in the classical Tikhonov regularization trade-off scheme (Tikhonov & Arsenin, 1977). For the corresponding case of classification, see Appendix A. Remark that most ideas of this paper in principle can carry over to other problems as kernel PCA, kernel CCA and kernel PLS, for which primal-dual optimization formulations are available within the context of the class of LS-SVM modelling approaches, see e.g. (Suykens et al., 2002).

The Lagrangian of the constrained optimization problem becomes $\mathcal{L}_\gamma(w, b, e_i; \alpha_i) = 0.5w^T w + 0.5\gamma\sum_{i=1}^N e_i^2 - \sum_{i=1}^N \alpha_i(w^T x_i + b + e_i - y_i)$. By taking the conditions for optimality $\partial\mathcal{L}_\gamma/\partial w = 0$, $\partial\mathcal{L}_\gamma/\partial b = 0$, $\partial\mathcal{L}_\gamma/\partial e_i = 0$, $\partial\mathcal{L}_\gamma/\partial\alpha_i = 0$, and application of the kernel trick $K(x_i, x_j) = \varphi(x_i)^T\varphi(x_j)$ with a positive definite (Mercer) kernel K, one gets $e_i\gamma = \alpha_i$, $w = \sum_{i=1}^N \alpha_i\varphi(x_i)$, $\sum_{i=1}^N \alpha_i = 0$ and $w^T\varphi(x_i) + b + e_i = y_i$. The dual problem is given by the following set of linear equations

$$\left[\begin{array}{c|c} 0 & 1_N^T \\ \hline 1_N & \Omega + I_N/\gamma \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \end{array}\right] \tag{2}$$

where $\Omega \in \mathbb{R}^{N\times N}$ with $\Omega_{ij} = K(x_i, x_j)$. The model can be evaluated at a new point x^* by $\hat{f}(x^*) = \sum_{i=1}^N \alpha_i K(x_i, x^*) + b$.

For the choice of the kernel $K(\cdot,\cdot)$, see e.g. (Genton, 2001; Chapelle & Vapnik, 2000). Typical examples are the use of a linear kernel $K(x_i, x_j) = x_i^T x_j$ or the RBF kernel $K(x_i, x_j) = \exp(-\|x_i - x_j\|_2^2/\sigma^2)$ where σ denotes the bandwidth of the kernel.

2.2. *Fusion of training with validation*

Determination of the optimal value γ with respect to the validation performance of this model can be written as

$$\min_\gamma \sum_{j=1}^{n}\left(y_j^v - \hat{f}_\gamma\left(x_j^v\right)\right)^2 = \sum_{j=1}^{n}\left(y_j^v - \left[\begin{array}{c} 1 \\ \hline \Omega^v \end{array}\right]^T \left[\begin{array}{c|c} 0 & 1_N^T \\ \hline 1_N & \Omega + I_N/\gamma \end{array}\right]^{-1} \left[\begin{array}{c} 0 \\ \hline y \end{array}\right]\right)^2 \tag{3}$$

where $\Omega^v \in \mathbb{R}^{n\times N}$ with $\Omega_{ij}^v = K(x_i, x_j^v)$. The determination of γ becomes a non-convex optimization problem (which is often even non-smooth in the related case of cross-validation methods).

However, one may view this optimization problem also in a somewhat different way, as we will explain now step-by-step. The estimation of the LS-SVM regressor on the training data for a fixed value γ is given as (2)

$$\text{level 1:}\quad (\hat{w}, \hat{b}, \hat{e}) = \arg\min_{w,b,e} \mathcal{J}_\gamma(w, e) \quad \text{s.t.} \quad \text{constraints of (1) hold,} \tag{4}$$

which results in solving a set of linear equations for the dual problem (2) after elimination of w and e. Tuning the regularization parameter by using a validation criterion amounts to minimizing e.g. the following cost on a validation set

$$\text{level 2:}\quad \hat{\gamma} \ = \arg\min_{\gamma} \sum_{j=1}^{n} \left(f(x_j^v; \hat{\alpha}, \hat{b}) - y_j^v \right)^2 \qquad \text{with } (\hat{\alpha}, \hat{b}) \text{ the dual solution to (4).} \tag{5}$$

Using the conditions for optimality (2), one can rewrite (5) as

$$\text{fusion:}\quad (\hat{\gamma}, \hat{\alpha}, \hat{b}, \hat{e}^v) = \arg \min_{\gamma,\alpha,b,e^v} \sum_{j=1}^{n} \left(e_j^v \right)^2 \quad \text{s.t.} \quad \begin{cases} \sum_{i=1}^{N} \alpha_i K\left(x_i, x_j^v\right) + b + e_j^v = y_j^v, & \forall j = 1, \ldots, n \\ \text{(2) holds.} \end{cases} \tag{6}$$

Hence, the training and validation stage are conceptually viewed at different levels, but computationally one can obtain *fusion* of training and validation levels (see Figure 1) by taking the training equations as hard constraints to the objective of the validation cost function.

Remark 2.1. This fusion problem of LS-SVM regression on training data with the validation level is non-convex. Indeed, when eliminating the unknowns α and b, one arrives at the optimization problem (3). This is due to the fact that one works with classical Tikhonov regularization in this scheme. The motivation for the following sections is to remedy this in such a way that the fusion will lead to a convex optimization problem. A key ingredient to achieve this is to consider an alternative parameterization the regularization trade-off for penalizing the loss function with respect to the regularization term.

3. LS-SVMs with additive regularization trade-off

3.1. Training conditions for optimality

A different way of formulating the trade-off between the norm of the residuals and the regularization term is investigated now for the model

$$f(x) = w^T \varphi(x) + b. \tag{7}$$

Instead of taking the regularization constant γ in a multiplicative way, i.e. $\gamma \sum_{i=1}^N e_i^2$, we employ regularization constants c_i which enter the loss in an additive way with respect to the error variables, i.e. $\sum_{i=1}^N (e_i - c_i)^2$. We call this use of regularization constants an *additive regularization trade-off* (AReg). This gives the following primal problem for the regression on training data:

$$\min_{w,b,e_i} \mathcal{J}_c(w,e) = \frac{1}{2} w^T w + \frac{1}{2} \sum_{i=1}^N (e_i - c_i)^2$$
$$\text{s.t.} \quad w^T \varphi(x_i) + b + e_i = y_i \quad \forall i = 1, \ldots, N. \tag{8}$$

Here c denotes the vector of regularization constants for the additive regularization trade-off. Remark that the size of this vector c equals the number of data points N. In the following section we will discuss how one can restrict the degrees of freedom of the regularization trade-off.

Lemma 3.1. (Additive regularization trade-off) *Given a vector of regularization constants $c \in \mathbb{R}^N$, the global solution to the problem (8) is characterized by the dual linear system with dual variables $\alpha \in \mathbb{R}^N$*

$$\left[\begin{array}{c|c} 0 & 1_N^T \\ \hline 1_N & \Omega + I_N \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y - c \end{array}\right]. \tag{9}$$

Proof: The Lagrangian of this constrained optimization problem becomes

$$\mathcal{L}_c(w,b,e_i;\alpha_i) = \frac{1}{2} w^T w + \frac{1}{2} \sum_{i=1}^N (e_i - c_i)^2 - \sum_{i=1}^N \alpha_i (w^T \varphi(x_i) + b + e_i - y_i). \tag{10}$$

The conditions for optimality w.r.t. w, b, e_i, α_i for the training become

$$\begin{cases} \partial \mathcal{L}_c / \partial e_i = 0 \to e_i = c_i + \alpha_i \\ \partial \mathcal{L}_c / \partial w = 0 \to w = \sum_{i=1}^N \alpha_i \varphi(x_i) \\ \partial \mathcal{L}_c / \partial b = 0 \to \sum_{i=1}^N \alpha_i = 0 \\ \partial \mathcal{L}_c / \partial \alpha_i = 0 \to w^T \varphi(x_i) + b + e_i = y_i. \end{cases} \tag{11}$$

This results in the dual linear system (9) after eliminating the variables w and e.

The resulting model is evaluated at a validation point x_j^v by $\hat{f}(x_j^v) = w^T \varphi(x_j^v) + b = \sum_{i=1}^N \alpha_i K(x_i, x_j^v) + b$. The residual $\hat{f}(x_j^v) - y_j^v$ is denoted by e_j^v such that one can write

$$y_j^v = w^T \varphi\left(x_j^v\right) + b + e_j^v = \sum_{i=1}^N \alpha_i K\left(x_i, x_j^v\right) + b + e_j^v. \tag{12}$$

Remark 3.1. The fixed matrix I_N is a consequence of the regularization term as classical and provides (numerical) stability to the linear system. The regularization is then traded by a proper choice of $c \in \mathbb{R}^N$. The classical Tikhonov regularization scheme can be related to this scheme as elaborated in Appendix B . The model training with AReg and the classical scheme (1) correspond when $c = 0_N$ and $\gamma = 1$. Following Eq. (70), one obtains a solution without regularization when $\alpha = c$ or $e \to 0_N$. According to Appendix B.2, the AReg scheme is equivalent to the case of weighted least squares SVMs (Suykens et al., 2002a) when the constraints (76) are satisfied for all $i = 1, \ldots, N$.

Remark 3.2. The following intuitive grasp may be connected to the additive trade-off scheme $\ell(e_i - c_i)$ with loss function $\ell : \mathbb{R} \to \mathbb{R}^+$. Assume a distribution of residuals $\{e_i\}_{i=1}^N$ denoted as $p(e)$. The constants c may be chosen such that the observed datapoints may better satisfy this distribution. Consider e.g. the case of outliers occurring at samples 1, 2 and 10. Then it makes sense to give the constants c_1, c_2 and c_{10} large values such that $(e_i - c_i)$ correspond most likely into the expected nominal distribution $p(e)$ of the other residuals.

3.2. Fusion of training and validation levels

The fusion argument is now applied to the LS-SVM regressor with additive regularization trade-off. The estimation of the LS-SVM regressor on the training data for a fixed value c is given as (9)

$$\text{level 1:} \quad (\hat{w}, \hat{b}, \hat{e}) = \arg\min_{w,b,e} \mathcal{J}_c(w, e) \quad \text{s.t.} \quad \text{constraints (8) hold,} \tag{13}$$

which results in solving a set of linear equations (9) after elimination of the primal variables w and e. Tuning the regularization parameter c by using a validation criterion amounts to minimizing e.g. the following cost on a training and validation set

$$\text{level 2:} \quad \hat{c} = \arg\min_c \frac{1}{2}\sum_{j=1}^{n}\left(f\left(x_j^v; \hat{\alpha}, \hat{b}\right) - y_j^v\right)^2 + \frac{1}{2}\sum_{i=1}^{N}(f(x_i; \hat{\alpha}, \hat{b}) - y_i)^2$$

$$\text{with } (\hat{\alpha}, \hat{b}) \text{ the dual solution to (13).} \tag{14}$$

Using the conditions for optimality (9), one can rewrite (14) as one optimization problem from which both regularization constants and the corresponding training solution follows at once.

Lemma 3.2 (Fusion of training with AReg and validation) *Both the optimal constants c with respect to a validation criterion in (14) as well as the corresponding training solution (α, b) follow from the constrained least squares problem*

$$\text{fusion:} \quad (\hat{c}, \hat{\alpha}, \hat{b}, \hat{e}^v) = \arg\min_{c,\alpha,b,e^v} \frac{1}{2}(e^v)^T e^v + \frac{1}{2}(\alpha + c)^T(\alpha + c)$$

$$\text{s.t.} \quad \begin{cases} (\Omega + I_N)\alpha + 1_N^T b + c = y, \quad 1_N^T \alpha = 0 & \textit{(training equations)} \\ \Omega^v \alpha + 1_N^T b + e^v = y^v & \textit{(validation equations).} \end{cases} \tag{15}$$

Proof: This result follows readily from the necessity and sufficiency of the conditions for optimality (11) (Karush-Kuhn-Tucker conditions). □

Figure 1 gives a schematical representation of the fusion argument. Note again that the conditions for optimality were exploited in order to guide the interaction between training and validation strictly through the regularization constants.

Algorithm 3.1 (Fusion of Areg with validation) *After eliminating the variables e^v and c from (15), one obtains the constrained optimization problem*

$$\min_{\alpha,b} \frac{1}{2} \|\Omega^v \alpha + b - y^v\|_2^2 + \frac{1}{2} \|\Omega \alpha + b - y\|_2^2 \quad s.t. \quad 1_N^T \alpha = 0 \textit{ holds}. \tag{16}$$

with $c = (\Omega \alpha + b - y) - \alpha$ following from the conditions for optimality (11). This optimization problem can be solved analytically as follows. Let

$$M = \begin{bmatrix} \Omega & 1_N \\ \Omega^v & 1_n \end{bmatrix} \in \mathbb{R}^{(N+n)\times(N+1)}, \tag{17}$$

$a = (\alpha; b)^T \in \mathbb{R}^{N+1}$, $g = (y; y^v)^T \in \mathbb{R}^{N+n}$ and $d = (1_N; 0)^T \in \mathbb{R}^{N+1}$. The Lagrangian of (16) becomes then

$$\mathcal{L}_c(a; \rho) = (Ma - g)^T (Ma - g) - \rho(d^T a) \tag{18}$$

with multiplier $\rho \in \mathbb{R}$. One can derive an analytical formula for the optimal Lagrange multiplier ρ by taking the conditions for optimallity with respect to a and ρ:

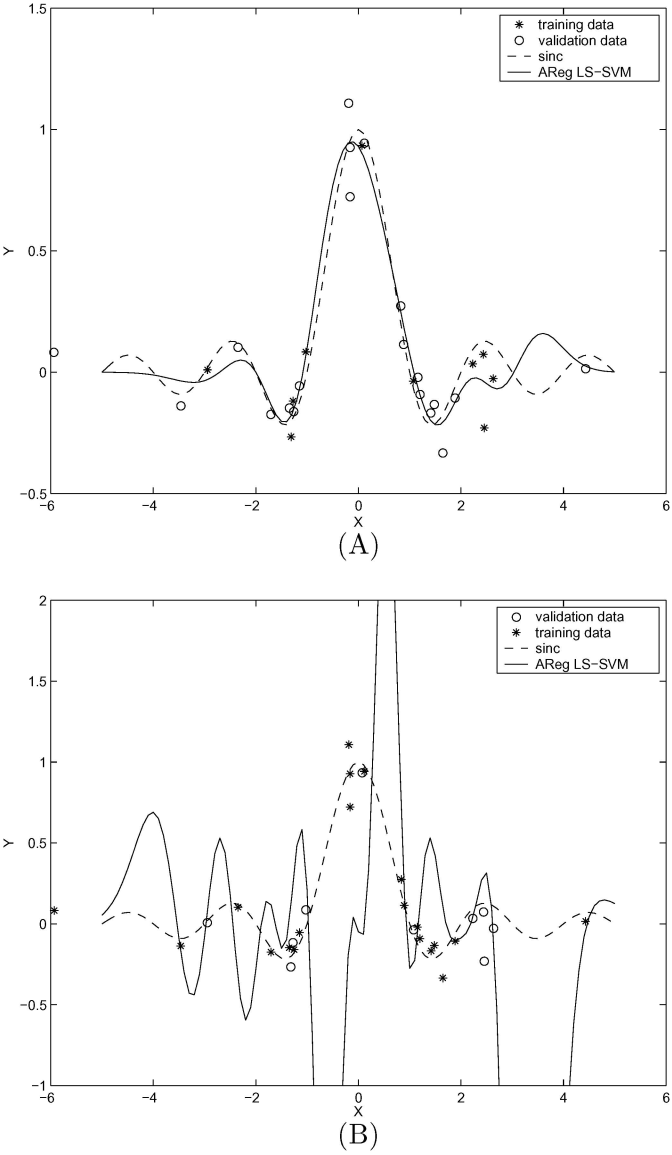
$$\begin{cases} \frac{\partial \mathcal{L}_c}{\partial a} = 0 \to M^T M a - M^T g = \rho d \\ \frac{\partial \mathcal{L}_c}{\partial \rho} = 0 \to \qquad d^T a = 0 \end{cases} \Rightarrow \quad \rho = -\frac{d^T M^\dagger g}{d^T (M^T M)^{-1} d}, \tag{19}$$

where $M^\dagger = (M^T M)^{-1} M^T$ denotes the pseudoinverse of M. Substituting this formulation into the first condition gives an analytical expression for the unknowns $\hat{\alpha}$, $\hat{b}$ as well as $\hat{c}$.

Remark 3.3. The disadvantage of this formulation is that the size of the validation set is required to be significantly larger than the size of the training set ($n \gg N$). Figure 2 illustrates the cases of n smaller and larger than N and its consequences.

4. Link with ensemble methods

In order to avoid the effect of data snooping (Schölkopf & Smola, 2002) or leakage of information from the validation data to the model training, it is crucial to further restrict (either explicitly or implicitly) the degrees of freedom of the regularization constants.



(A)

(B)

Figure 2. Illustration of the AReg LS-SVM minimizing the validation cost when (A) number training data equals 10, number of validation data 20; (B) number of training data equals 20, number of validation points 10.

This section elaborates on a method based on convex hulls of the Tikhonov constraint (see Appendix B). Consider again the model (7)

$$f(x) = w^T \varphi(x) + b = \sum_{i=1}^{N} \alpha_i \, K(x_i, x) + b, \tag{20}$$

but the training criterion (8) will now be restricted such that $c \in \mathcal{S}^c \subset \mathbb{R}^N$.

4.1. Restriction to a convex set

A first approach is to take the set $\mathcal{S}^c$ such that the corresponding solution space of α, b (say $\mathcal{S}^{\alpha,b}$) has nice (convex) properties. To do so, the solutions α, b are required to be a convex combination of $m \geq 2$ solutions $\alpha_{(k)}$, $b_{(k)}$ (referred to as (Tikhonov) *nodes*) corresponding with m prefixed tuning parameters in the Tikhonov regularization scheme, $\gamma_{(k)} \geq 0$ for all $k = 1, \ldots, m$ (see Appendix B).

The m nodes are found as the solutions to the following independent sets of linear equations (2.1) for all $k = 1, \ldots, m$:

$$\begin{cases} (\Omega + I_N \gamma_{(k)}^{-1})\, \alpha_{(k)} + b_{(k)} = y \\ 1_N^T \alpha_{(k)} = 0. \end{cases} \tag{21}$$

referred to as the Tikhonov nodes (see Figure 6.(A)). Formally, the convex solution set $\mathcal{S}^{\alpha,b}_{\gamma_{(1)},\ldots,\gamma_{(m)}}$ is considered (Rockafeller, 1970, Boyd & Vandenberghe, 2004):

Definition 4.1. (Tikhonov Ensemble model). Consider the class of models (20) with unknowns (α, b) restricted by the following convex set spanned by a number of m Tikhonov nodes:

$$\mathcal{S}^{\alpha,b}_{\gamma_{(1)},\ldots,\gamma_{(m)}} = \left\{ (\alpha, b) \mid \alpha = \sum_{k=1}^{m} \lambda_{(k)}\alpha_{(k)}, b = \sum_{k=1}^{m} \lambda_{(k)} b_{(k)}, \sum_{k=1}^{m} \lambda_{(k)} = 1, \lambda_{(k)} \geq 0 \right\} \tag{22}$$

This is called the ensemble model spanned by the given m Tikhonov nodes.

Proposition 4.1 *(Link Tikhonov ensemble model and AReg). Given the solutions to m different Tikhonov nodes (21), every solution $\alpha, b \in \mathcal{S}^{\alpha,b}_{\gamma_{(1)},\ldots,\gamma_{(m)}}$ is a solution to the (more general) LS-SVM regressor with AReg (9) with $c = \sum_{k=1}^{m} \lambda_{(k)}\alpha_{(k)}(\gamma_{(k)}^{-1} - 1)$.*

Proof:

$$\begin{aligned} y &= \sum_{k=1}^{m} \lambda_{(k)} y = \sum_{k=1}^{m} \lambda_{(k)} \left[\left(\Omega + I_N \gamma_{(k)}^{-1}\right) \alpha_{(k)} + b_{(k)} \right] \\ &= (\Omega + I_N) \sum_{k=1}^{m} \lambda_{(k)}\alpha_{(k)} + \sum_{k=1}^{m} \lambda_{(k)} b_{(k)} + \sum_{k=1}^{m} \lambda_{(k)}\alpha_{(k)} \left(\gamma_{(k)}^{-1} - 1\right) \\ &= (\Omega + I_N)\alpha + b + c. \end{aligned} \tag{23}$$

This results into a further intuitive grasp of the additive regularization trade-off scheme (see also Remark 3.1): the additive regularization trade-off scheme enables one to work with convex combinations of solutions corresponding with atomic models (Tikhonov nodes).

Corollary 4.1. *The allowed regularization constant space for c corresponding with $\mathcal{S}^{\alpha,b}_{\gamma_{(1)},\ldots,\gamma_{(m)}}$ can be written explicitly as*

$$\mathcal{S}^{c}_{\gamma_{(1)},\ldots,\gamma_{(m)}} = \left\{ c \mid c = \sum_{k=1}^{m} \lambda_{(k)}\alpha_{(k)} \left(\gamma_{(k)}^{-1} - 1\right), \sum_{k=1}^{m} \lambda_{(m)} = 1, \lambda_{(k)} \geq 0 \right\} \tag{24}$$

which is again a convex set.

Hence, a finite number of Tikhonov node solutions span a convex set, over which one aims at finding the global optimum for the hyperparameters. This is possible thanks to the additive regularization trade-off re-parameterization of the regularization constants. In classical Tikhonov regularization and e.g. using Generalized Cross-Validation (Golub, Heath & Wahba 1979) for model selection one has to explore over a range of regularization constants (either with e.g. grid search or local line search approaches) without guarantees for finding the global optimum and often with non-smooth surfaces to be optimized (though for classical schemes this can be made numerically efficient up to a certain extent as explained in Appendix B).

Remark 4.1. In practice (see Section 6), $m = 2$ can be taken where $\gamma_{(1)}$ and $\gamma_{(2)}$ are two rough guesses of the regularization constant γ in (2). The generalization performance can improve when increasing m at the expense of a higher computational cost, see Figure 6.(B). A good rule of thumb is to take initial guesses as the inverse of the noise variance of the output data (Pelckmans et al., 2003).

4.2. Fusion of the training and the validation

The same argument as in Section 3.2 is followed: the general solution for fusion of LS-SVMs with additive regularization trade-off and validation is described by (22) and (21). However, by application of the previous reasoning, the optimal solution according to

$$\min_{\alpha,b,c,\alpha_{(k)},b_{(k)},\lambda_{(k)}} \|e^v\|_2^2 \quad \text{s.t.} \quad c \in \mathcal{S}^c_{\gamma_{(1)},\ldots,\gamma_{(m)}}$$
$$\text{and (21) and constraints of (15) hold} \qquad (25)$$

is to be found. Equivalently

$$\min_{\alpha,b,c,\alpha^\lambda_{(k)},b^\lambda_{(k)},\lambda_{(k)}} \|e^v\|_2^2$$
$$\text{s.t. constraints of (15) hold \&} \begin{cases} \lambda_{(k)} y = (\Omega + I_N \gamma_{(k)}^{-1})\,\alpha^\lambda_{(k)} + b^\lambda_{(k)} \\ 0 = 1_N^T \alpha^\lambda_{(k)} \\ 0 \leq \lambda_{(m)},\ \sum_{k=1}^m \lambda_{(m)} = 1 \\ c = \sum_{k=1}^m \alpha^\lambda_{(k)} \left(\gamma_{(k)}^{-1} - 1\right) \\ y^v = \Omega^v \sum_{k=1}^m \alpha^\lambda_{(k)} + \sum_{k=1}^m b^\lambda_{(k)} + e^v \end{cases} \qquad (26)$$

where $\alpha^\lambda_{(k)} = \lambda_{(k)}\alpha_{(k)}$ and $b^\lambda_{(k)} = \lambda_{(k)} b_{(k)}$ by definition. This optimization problem is linear in the unknowns $\alpha, b, c, \alpha^\lambda_{(k)}, b^\lambda_{(k)}, \lambda_{(k)}$ for all $k = 1, \ldots, m$ and can be solved as a convex problem. Figure 3(A) shows the solutions of α for varying γ in (2) between two boundary values and for varying λ between the same extrema. The evolution of the corresponding predictors is given in Figure 3(B).

Algorithm 4.1 (Ensemble learning). *As the individual nodes do not depend on the regularization parameters* $\lambda_{(1)}, \ldots, \lambda_{(m)}$, *they can be solved beforehand as in ensemble*

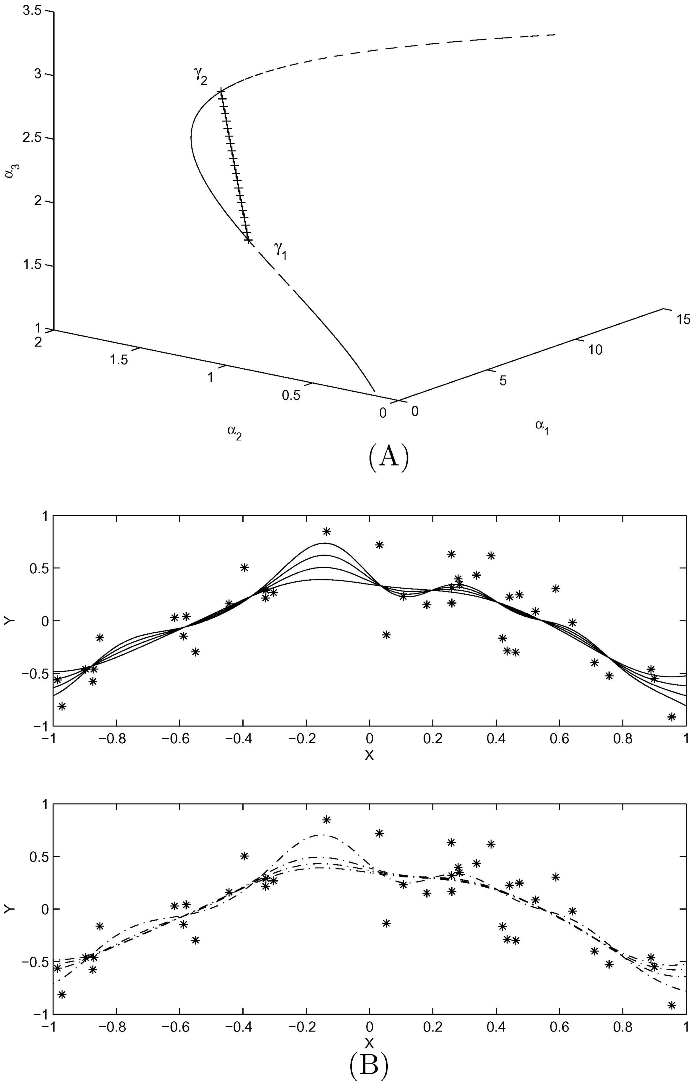


Figure 3. (A) The solid line indicates the solutions α_γ corresponding to a sequence of γ values. The linear interpolation between $\gamma_{(1)}$ and $\gamma_{(2)} = 100$ indicates the convex combination α_λ for $\lambda \in [0, 1]$ between these nodes. The top panel (B) shows the predictions based on α_γ with $\gamma = 1, 1.4, 2.8, 50, 100$. The bottom panel (B) shows the predictions based on α_λ with $\lambda = 0, 0.33, 0.66, 1$.

methods. Given these precomputed nodes, the regularization parameters can be found by solving

$$\min_{\lambda_{(1)},\ldots,\lambda_{(m)}} \|\Omega^v \alpha + b - y^v\|_2^2 \text{ s.t. (22) and (21) holds} \tag{27}$$

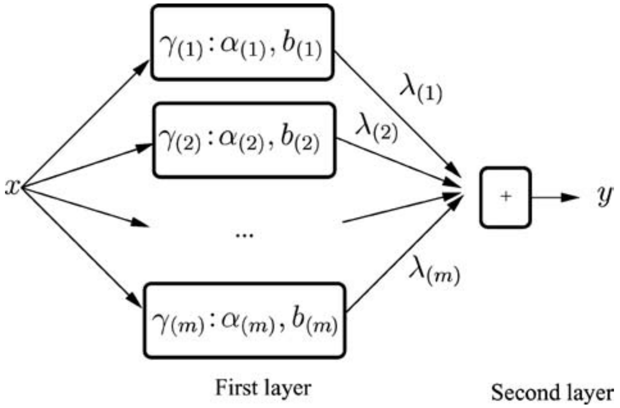

$\gamma_{(1)}:\alpha_{(1)},b_{(1)}$
$\gamma_{(2)}:\alpha_{(2)},b_{(2)}$
$\gamma_{(m)}:\alpha_{(m)},b_{(m)}$
x
...
$\lambda_{(1)}$
$\lambda_{(2)}$
$\lambda_{(m)}$
+
y
First layer
Second layer

Figure 4. Schematical illustration of the ensemble interpretation explained in Section 4 and A.4. In the first layer one computes the solutions of a finite number of Tikhonov nodes, in the second layer, the optimal (convex) combination of the nodes is determined by optimizing the validation performance of the nodes. A major difference with classical ensemble methods is that the additive regularization trade-off provides a global optimality principle and one can recover the additive regularization constants c_i from $\alpha_{(m)}$, $b_{(m)}$ and $\lambda_{(m)}$.

which is equivalent to

$$\min_{\lambda_{(1)},\ldots,\lambda_{(m)}} \left\| \sum_{k=1}^{m} \left(\Omega^v \alpha_{(k)} + b_{(k)}\right) \lambda_{(k)} - y^v \right\|_2^2$$
$$\text{s.t.} \quad \sum_{k=1}^{m} \lambda_{(k)} = 1, \quad \lambda_{(k)} \geq 0 \quad \forall k = 1,\ldots,m. \tag{28}$$

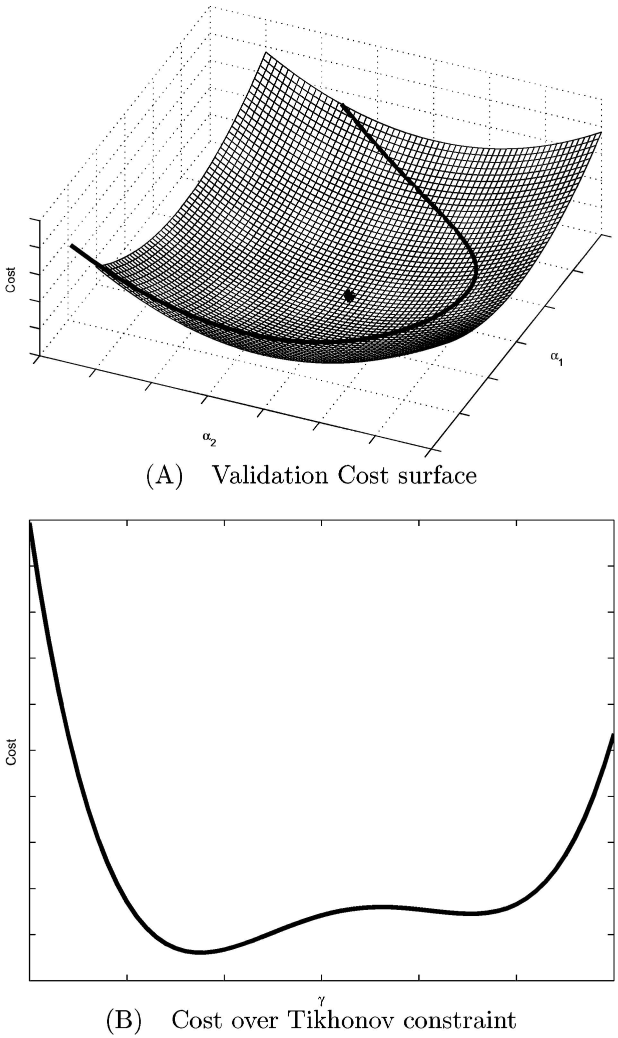
This can be solved efficiently as a QP problem or constrained linear least squares problem (when omitting the inequality constraint). Note that this implementation is closely related to the point of view of ensemble methods (Perrone & Cooper, 1993; Bishop, 1995; Breiman, 1996):

1. *(first layer): compute the m Tikhonov node solutions,*
2. *(second layer): combine them using (28) to obtain the final estimator.*

This layered interpretation is shown in Figure 4. A main difference of this interpretation and the classical literature on ensemble methods is that a global optimality principle holds for the final model of the ensemble: once $\lambda_{(l)}$ for all $l = 1,\ldots,m$ are known, one can recover the corresponding regularization constants c_i for which the ensemble model is globally optimal according to that criterion.

Remark 4.2. Other validation criteria can be considered. Examples in the case of classification are found in Appendix A.5. However, convexity is not preserved in general when other model selection criteria are considered.

Remark 4.3. The goal of this section was to propose a way of restricting the degrees of freedom of the additive regularization constants. Appendix B considers the problem of recovering the classical Tikhonov scheme within the context of the additive regularization trade-off framework (see Figures 5 and 6(A)).



(A) Validation Cost surface

(B) Cost over Tikhonov constraint

Figure 5. (A) Convex cost surface over the training solutions α (12) w.r.t. to the validation set performance. The optimal validation performance is achieved at the dot. When the solutions α are obtained from ridge regression with Tikhonov regularization, one considers α_γ satisfying the quadratic Tikhonov constraint (70) (solid line). (B) Optimization over a quadratic constraint as in (A) results in an optimization problem with possibly multiple local minima.

Remark 4.4. The computation of the solutions of the Tikhonov nodes, has a complexity of $O(mlN^2)$ where $0 < l \ll N$ is the number of iterations of the conjugate gradient algorithm. The second step, the determination of the optimal hyper-parameters $\lambda_{(k)}$ for all $k = 1, \dots, m$ is typically of complexity $O(mN^2)$ for the evaluation of the nodes, and $O(m^3)$ for the solution of the constrained least squares problem. The total complexity of the fast implementation becomes then $O(mlN^2 + m^3)$. As in most applications m will be chosen relatively small, the computational cost will be approximately $O(mlN^2)$.

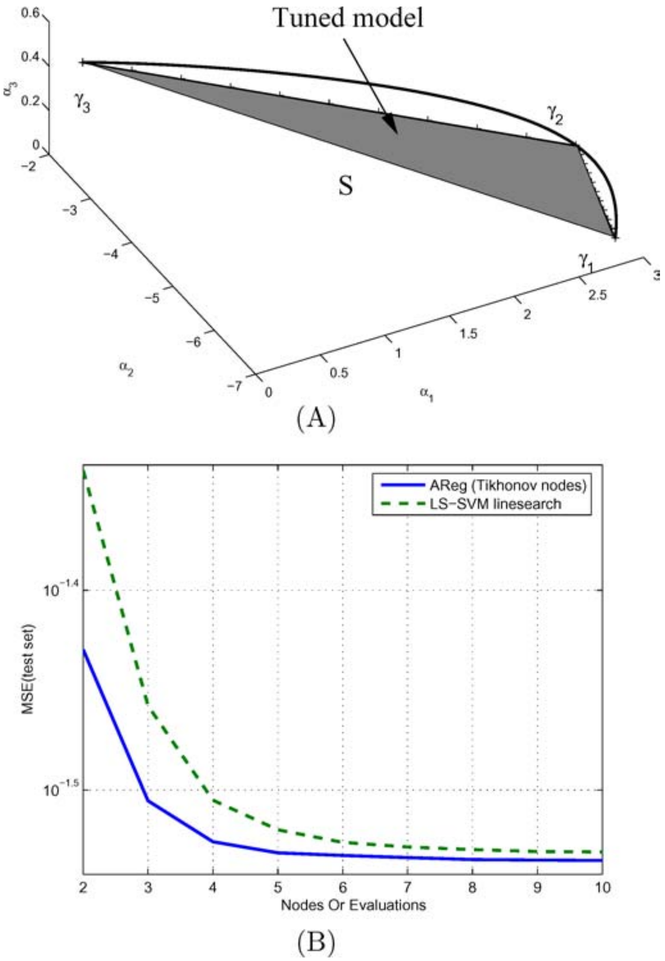


Figure 6. (A) For the fusion scheme with the restriction as discussed in Section 4 one looks for the model inside the set $\mathcal{S}$ with optimal validation performance. Approximation of the quadratic Tikhonov constraint can be obtained by a convex combination of the Tikhonov nodes describing the convex set $\mathcal{S}$. (B) Comparison of (solid line) the generalization performance (MSE of the test set) of the model resulting from the convex scheme having additive regularization trade-off (as explained in 4) with $m = 2, \ldots, 10$ nodes and (dashed line) an LS-SVM with regularization constant γ (classical Tikhonov regularization) tuned by a line-search using $m = 2, \ldots, 10$ evaluations. While the computational cost of both methods are the same along the number of nodes (or evaluations) axis, the generalization performance of the convex AReg scheme (solid line) outperforms the latter method (dashed line), especially for smaller values of m. These experiments are done on the sinc data set.

This exploration shows that the argument of fusion of the training equations with a simple validation criterion may be approached efficiently with techniques of convex optimization and the adoption of the additive regularization trade-off scheme. It is illustrated how one may restrict the degrees of freedom of the regularization scheme itself in order to increase the generalization performance. We proceed by considering the more powerful model selection scheme of cross-validation.

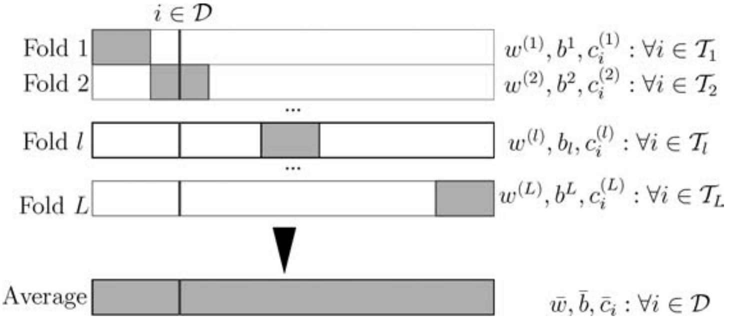

$i \in \mathcal{D}$
Fold 1
Fold 2
Fold l
Fold L
Average
$w^{(1)}, b^1, c_i^{(1)} : \forall i \in \mathcal{T}_1$
$w^{(2)}, b^2, c_i^{(2)} : \forall i \in \mathcal{T}_2$
$w^{(l)}, b_l, c_i^{(l)} : \forall i \in \mathcal{T}_l$
$w^{(L)}, b^L, c_i^{(L)} : \forall i \in \mathcal{T}_L$
$\bar{w}, \bar{b}, \bar{c}_i : \forall i \in \mathcal{D}$

Figure 7. Schematical illustration of the L-fold cross-validation procedure.

5. Fusion of training and cross-validation levels within the additive regularization framework

In order to avoid the non-trivial process of dividing valuable data into a separate training and validation set, Cross-Validation (CV) (Stone, 1974) has been introduced. The following is based on the L-fold CV (where Leave-One-Out CV is a special case with $L = N$). The data $\mathcal{D}$ are repeatedly divided into a training set $\mathcal{T}_l$ and a corresponding disjoint validation set $\mathcal{V}_l, \forall l = 1, \ldots, L$ such that $\mathcal{D} = \mathcal{T}_l \cup \mathcal{V}_l = \cup_{l=1}^{L} \mathcal{V}_l$ and $\mathcal{V}_l \cap \mathcal{V}_k = \oslash, \forall l \neq k = 1, \ldots, L$. In the following, $N_{(l)}$ denotes the number of training points and $n_{(l)}$ the number of validation points of the lth fold. Figure 7 illustrates this repeated training and validation process.

5.1. *Training and validation set evaluation per fold*

Straightforward application of the fusion of training and validation of the regression model $f^{(l)}(x) = w^{(l)^T} \varphi(x) + b^{(l)}$ in the lth fold results in the following constrained optimization problem problem (compare with (8)):

$$\min_{w^{(l)}, b^{(l)}, e_i^{(l)}} \mathcal{J}_c^{(l)} = \frac{1}{2} w^{(l)^T} w^{(l)} + \frac{1}{2} \sum_{i \in \mathcal{T}_l} \left(e_i^{(l)} - c_i^{(l)} \right)^2$$
$$\text{s.t.} \quad w^{(l)^T} \varphi(x_i) + b^{(l)} + e_i^{(l)} = y_i, \quad \forall i \in \mathcal{T}_l \tag{29}$$

where the shorthand notation $i \in \mathcal{T}_l$ is employed to denote that the ith data point belongs to the lth fold for the training and only the index of the data point is used in the notation.

After construction of the Lagrangian and taking the conditions for optimality, one obtains

$$\begin{cases} e_i^{(l)} = c_i^{(l)} + \alpha_i^{(l)} & \forall i \in \mathcal{T}_l, \quad \text{(a)} \\ w^{(l)} = \sum_{j \in \mathcal{T}_l} \alpha_j^{(l)} \varphi(x_j) & \quad \text{(b)} \\ \sum_{j \in \mathcal{T}_l} \alpha_j^{(l)} = 0 & \quad \text{(c)} \\ w^{(l)} \varphi(x_i) + b^{(l)} + e_i^{(l)} = y_i & \forall i \in \mathcal{T}_l. \quad \text{(d)} \end{cases} \tag{30}$$

The validation errors are computed as follows:

$$e_j^{(l)v} = w^{(l)^T} \varphi\left(x_j^v\right) + b^{(l)} - y_j^v, \quad \forall j \in \mathcal{V}_l \tag{31}$$

where the shorthand notation $j \in \mathcal{V}_l$ is employed to denote that the jth data point belongs to the validation set in the lth fold.

Similar as in Section 3.2, one obtains

$$\left[\begin{array}{c|c|cc} 0_{N_l}^T & 0_{n_l}^T & 0 & 1_{N_l}^T \\ \hline 0_{n_l \times N_l} & I_{n_l} & 1_{n_l} & \Omega_l^v \\ \hline I_{N_l} & 0_{N_l \times n_l} & 1_{N_l} & \Omega_l + I_{N_l} \end{array}\right] \left[\begin{array}{c} c^{(l)} \\ \hline e^{(l)v} \\ \hline b^{(l)} \\ \alpha^{(l)} \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y^{(l)v} \\ \hline y^{(l)} \end{array}\right]. \tag{32}$$

Note that each point is used for validation only once, hence we can write $e_j^{(l)v} = e_j^v$. The matrices Ω_l, Ω_l^v are defined as before where l denotes the lth fold.

5.2. *Simultaneous training and validation of all folds*

All L training and validation steps can be solved simultaneously but independently by stacking them into a block diagonal linear system. For notational convenience, the indicator matrix $I[\mathcal{S}_1, \mathcal{S}_2]$ is introduced denoting a sparse matrix with (i, j)th entry 1 if $\mathcal{S}_1(i) = \mathcal{S}_2(j)$ and 0 otherwise for sets $\mathcal{S}_1$ and $\mathcal{S}_2$, e.g.:

$$I[\mathcal{S}_1, \mathcal{S}_2] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{where} \mathcal{S}_1 = \{a, b, d\} \text{ and } \mathcal{S}_2 = \{a, b, c, d\}. \tag{33}$$

As argued in Section 3.2 and Figure 2, in each fold the number of validation data may not be smaller than the number of training data when using the cost function $\mathcal{J}^{(l)}(e^{(l)}, e^{(l)v}) = e^{(l)^T} e^{(l)} + e^{(l)v^T} e^{v(l)}$. To avoid this difficulty in the cross-validation setting, there is an opportunity to restrict in a natural way the degrees of freedom of the additive regularization constants $c^{(l)}$ for all $l = 1, \ldots, N_{(l)}$. As in classical cross-validation practice, the additive regularization constants should be held constant over the different folds, i.e.

$$c^{(l)} = \mathcal{I}(\mathcal{T}_l, \mathcal{D})\, c, \quad \forall l = 1, \ldots, L. \tag{34}$$

This reduces the freedom of the regularization constants from $(L-1)N$ to N. Embedding this in a single linear system results in the following set of $(N+1)L$ linear equations for the $(N+1)L + N$ variables.

$$\left[\begin{array}{cc|cc|c|cc}
I[\mathcal{T}_1, \mathcal{D}] & & 1_{N_{(1)}} & (\Omega_1 + I_{N_{(1)}}) & & & \\
& I[\mathcal{V}_1, \mathcal{D}] & 1_{n_{(1)}} & \Omega_1^v & & & \\
& & 0 & 1^T_{N_{(l)}} & & & \\ \hline
\vdots & & & & \ddots & & \\
& \vdots & & & \ddots & & \\
& & & & \ddots & & \\ \hline
I[\mathcal{T}_L, \mathcal{D}] & & & & & 1_{N_{(L)}} & (\Omega_l + I_{N_{(l)}}) \\
& I[\mathcal{V}_L, \mathcal{D}] & & & & 1_{n_{(L)}} & \Omega_L^v \\
& & & & & 0 & 1^T_{N_{(L)}}
\end{array}\right]
\left[\begin{array}{c}
c \\ e^v \\ \hline b^{(1)} \\ \alpha^{(1)} \\ \hline \\ \vdots \\ \\ \hline b^{(1)} \\ \alpha^{(1)}
\end{array}\right]
=
\left[\begin{array}{c}
y^{(1)} \\ y^{(1)v} \\ 0 \\ \hline \\ \vdots \\ \\ \hline y^{(L)} \\ y^{(L)v} \\ 0
\end{array}\right]
\tag{35}$$

where empty entries indicate zeros. Different optimality criteria can be considered to choose a 'best' solution to this under-determined linear system.

For notational convenience, the bias term b is left out below. A simple criterion is to minimize the sum of squared training and validation residuals as motivated in previous section

$$\min_{c, \alpha^{(l)}, e^{(l)v}} \frac{1}{2} \sum_{l=1}^{L} e^{(l)v^T} e^{(l)v} + \frac{1}{2} \sum_{l=1}^{L} e^{(l)^T} e^{(l)} \quad \text{s.t. (35) holds.} \tag{36}$$

The dual solution is characterized as follows

Lemma 5.1 (Dual solution of cross-validation based tuning) *The dual solution to (36) is given by*

$$A_{CV} \left[\begin{array}{c}
c \\ e^v \\ \hline \alpha^{(1)} \\ \psi^{(1)} \\ \hline \\ \vdots \\ \\ \hline \alpha^{(L)} \\ \psi^{(L)}
\end{array}\right]
=
\left[\begin{array}{c}
y^{(1)} \\ y^{(1)v} \\ 0_{N_{(l)}} \\ \hline \\ \vdots \\ \\ \hline y^{(L)} \\ y^{(L)v} \\ 0_{N_{(l)}} \\ \hline 0_N
\end{array}\right], \tag{37}$$

with Lagrange multipliers $\alpha^{(l)} \in \mathbb{R}^{N_l}$ for all $l = 1, \ldots, L$: From this formulation, one can obtain the global model by averaging the individual models from the different folds:

$$\hat{y}^v = \frac{1}{L} \sum_{l=1}^{L} \hat{y}^{(l)v} = \frac{1}{L-1} \Omega^v \sum_{l=1}^{L} I[\mathcal{D}, \mathcal{T}_l] \alpha^{(l)}. \tag{38}$$

Proof: The Lagrangian corresponding to (36) using (34) and the equality $e^{(l)} = \alpha^{(l)} + c^{(l)}$ from (30) becomes

$$
\begin{aligned}
\mathcal{L}\left(c, \alpha^{(l)}, e^{(l)v}; \nu^{(l)}, \psi^{(l)}\right) = {} & \frac{1}{2}\sum_{l=1}^{L} e^{(l)v^T} e^{(l)v} + \frac{1}{2}\sum_{l=1}^{L}\left(\alpha^{(l)} + c^{(l)}\right)^T\left(\alpha^{(l)} + c^{(l)}\right) \\
& + \sum_{l=1}^{L} \nu^{(l)T}\left(\Omega^{(l)v}\alpha^{(l)} + e^{(l)v} - y^{(l)v}\right) \\
& + \sum_{l=1}^{L} \psi^{(l)T}\left(\left(\Omega^{(l)} + I_N\right)\alpha^{(l)} + c^{(l)} - y^{(l)}\right). \qquad (39)
\end{aligned}
$$

Taking the conditions for optimality of (39) w.r.t. $\alpha^{(l)}$, $e^{(l)v}$, $\nu^{(l)}$, $\psi^{(l)}$ and c

$$
\forall l = 1, \ldots, L \quad
\begin{cases}
\partial\mathcal{L}/\partial\alpha^{(l)} = 0 \rightarrow \left(\alpha^{(l)} + c^{(l)}\right) + \Omega^{(l)vT}\nu^{(l)} & \\
\qquad\qquad + \left(\Omega^{(l)} + I_{N_{(l)}}\right)^T \psi^{(l)} = 0 & (a) \\
\partial\mathcal{L}/\partial e^{(l)v} = 0 \rightarrow e^{(l)v} + \nu^{(l)} = 0 & (b) \\
\partial\mathcal{L}/\partial\nu^{(l)} = 0 \rightarrow \Omega^{(l)v}\alpha^{(l)} + e^{(l)v} = y^{(l)v} & (c) \\
\partial\mathcal{L}/\partial\psi^{(l)} = 0 \rightarrow \left(\Omega^{(l)} + I_{N_L}\right)\alpha^{(l)} + c^{(l)} = y^{(l)} & (d)
\end{cases}
$$

$$
\textit{and } \partial\mathcal{L}/\partial c = 0 \rightarrow (L-1)c + \sum_{l=1}^{L}\left(I[\mathcal{D}, \mathcal{T}_l]\alpha^{(l)} + I[\mathcal{D}, \mathcal{T}_l]\psi^{(l)}\right) = 0. \quad (e)(40)
$$

By elimination of $\nu^{(l)}$, one obtains the dual linear system (37) with A_{CV} defined as

$$
\left[\begin{array}{cc|cc|c|cc}
I[\mathcal{T}^1, \mathcal{D}] & & \Omega^{(1)} + I_{N_{(1)}} & & & & \\
& I[\mathcal{V}_1, \mathcal{D}] & \Omega^{(1)v} & & & & \\
-I[\mathcal{T}_1, \mathcal{D}] & -\Omega^{(1)vT} I[\mathcal{V}_1, \mathcal{D}] & I_{N_{(1)}} & \Omega^{(l)} + I_{N_{(1)}} & & & \\
\hline
\vdots & & & & \ddots & & \\
& \vdots & & & \ddots & & \\
\hline
I[\mathcal{T}_L, \mathcal{D}] & & & & & \Omega^{(L)} + I_{N_{(L)}} & \\
& I[\mathcal{V}_L, \mathcal{D}] & & & & \Omega^{(L)v} & \\
-I[\mathcal{T}^{(1)}, \mathcal{D}] & -\Omega^{(L)vT} I[\mathcal{V}_L, \mathcal{D}] & & & & I_{N_{(L)}} & \Omega^{(L)} + I_{N_{(L)}} \\
\hline
(L-1)I_N & & I[\mathcal{D}, \mathcal{T}_1] & I[\mathcal{D}, \mathcal{T}_1] & \ldots & I[\mathcal{D}, \mathcal{T}_L] & I[\mathcal{D}, \mathcal{T}_L]
\end{array}\right].
$$

We refer to this model as to AReg LS-SVM (CV), which works with additive regularization trade-off in a cross-validation setting (see Figure 8).

Remark 5.1. The complexity of this implementation becomes $O((LN)^2 l)$ where $0 < l \ll LN$ is the number of iteration steps in the conjugate gradient algorithm for solving the set of linear equations.

5.3. Alternative formulation with fast algorithm

It turns out that a similar result can be obtained without computing explicitly the solutions $\alpha^{(l)}$ and $b^{(l)}$ of the individual folds. To show this, one starts again from the

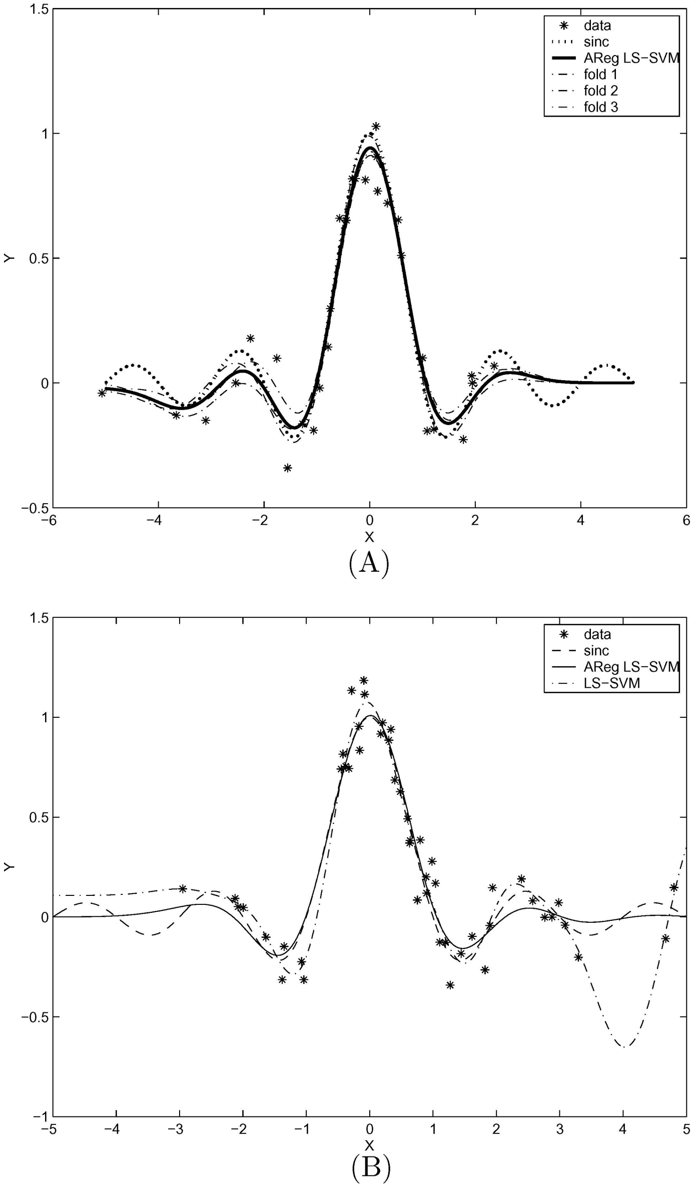


Figure 8. Illustration of AReg LS-SVM (CV) (37) on a *sinc* function (25 data): (A) AReg based on 3-fold cross-validation and the resulting estimate; (B) comparison of AReg LS-SVM and the LS-SVM withTikhonov regularization showing less boundary effects for AReg LS-SVM.

primal formulation:

$$\hat{f}_{CV}(x) = \bar{w}^T \varphi(x) + b \quad \text{with } \bar{w} = \frac{1}{L}\sum_{l=1}^{L} w^{(l)}. \tag{41}$$

One can view the cross-validation based learning strategy explained in the previous subsection as a multi-criterion optimization problem

$$00\big(w^{(l)}, e_k^{(l)}, b\big) = \underset{w^{(l)}, e_k^{(l)}, b}{\arg\min} \mathcal{J}_{(cv)} \begin{bmatrix} \frac{1}{2} w^{(1)^T} w^{(1)} + \frac{1}{2}\sum_{k\in\mathcal{T}^{(1)}} (e_k^{(1)} - c_k)^2 \\ \dots \\ \frac{1}{2} w^{(L)^T} w^{(L)} + \frac{1}{2}\sum_{k\in\mathcal{T}^{(L)}} (e_k^{(L)} - c_k)^2 \end{bmatrix}$$
$$\text{s.t.} \quad \begin{cases} w^{(1)^T}\varphi(x_k) + b + e_k^{(1)} = y_k, & \forall k \in \mathcal{T}^{(1)} \\ \dots \\ w^{(L)^T}\varphi(x_k) + b + e_k^{(L)} = y_k, & \forall k \in \mathcal{T}^{(L)} . \end{cases} \tag{42}$$

Note that the b-term is not regularized, as common e.g. in ridge regression (Hoerl & Kennard, 1970), SVMs (Vapnik, 1998) and LS-SVMs (Suykens et al., 2002). A Pareto optimal solution can be found to this (Boyd & Vandenberghe, 2004). The scalarization technique with weights $1_N = (1, \dots, 1)^T \in \mathbb{R}^L$ in the objective function is used, as no fold can be favored a priori in general. The following cost criterion is considered:

$$J_{(cv)}^P\big(w^{(l)}, e_k^{(l)}\big) = \frac{1}{2(L-1)}\sum_{l=1}^{L} w^{(l)^T} w^{(l)} + \frac{1}{2(L-1)}\sum_{l=1}^{L}\sum_{k\in\mathcal{T}_l}\big(e_k^{(l)} - c_k\big)^2$$
$$\text{s.t.} \quad \begin{cases} w^{(1)^T}\varphi(x_k) + b + e_k^{(1)} = y_k, & \forall k \in \mathcal{T}^{(1)} \\ \dots \\ w^{(L)^T}\varphi(x_k) + b + e_k^{(L)} = y_k, & \forall k \in \mathcal{T}^{(L)} \end{cases} \tag{43}$$

where the normalization terms $\frac{1}{2(L-1)}$ appear for notational convenience in the following. Furthermore, the criterion (43) is relaxed by replacing the fitting term $\sum_{l|k\in\mathcal{T}_l}(e_k^{(l)} - c_k)^2$ by the average fitting term $(\sum_{l|k\in\mathcal{T}_l}\frac{e_k^{(l)}}{L-1} - c_k)^2$. This states that at least the average of different folds should perform optimal. Lateron the individual folds will be optimized with respect to their respective validation cost. This motivates the following cost function, obtained after elimination of the individual error terms $e_k^{(l)}$ for all $k = 1, \dots, N$ and $l = 1, \dots, L$ by $\bar{e}_k = \sum_{l|k\in\mathcal{T}_l}\frac{e_k^{(l)}}{L-1}$:

$$J_{(cv)}^{P'}\left(w^{(l)}, \bar{e}_k\right) = \frac{1}{2(L-1)}\sum_{l=1}^{L} w^{(l)^T} w^{(l)} + \frac{1}{2(L-1)}\sum_{k=1}^{N}(\bar{e}_k - c_k)^2$$
$$\text{s.t.} \quad \frac{1}{L-1}\sum_{l|i\in\mathcal{T}_l} w^{(l)^T}\varphi(x_i) + b + \bar{e}_k = y_i, \quad \forall i = 1, \dots, N. \tag{44}$$

The dual characterization of the solution is given by the following Lemma.

Lemma 5.2 (Dual characterization of CV based LS-SVMs). *Let the matrix $\Omega_{CV} \in \mathbb{R}^{N\times N}$ be defined as*

$$\Omega_{CV} = \begin{bmatrix} \Omega_{\mathcal{V}_1,\mathcal{V}_1} & & & \\ & \Omega_{\mathcal{V}_2,\mathcal{V}_2} & & \\ & & \ddots & \\ & & & \Omega_{\mathcal{V}_L,\mathcal{V}_L} \end{bmatrix} \tag{45}$$

where $\Omega_{\mathcal{V}_l,\mathcal{V}_l}$ is the kernel matrix between elements of the validation set of the lth fold $[\Omega_{\mathcal{V}_l,\mathcal{V}_l}]_{i,j} = K(x_i, x_j), \forall i, j \in \mathcal{V}_l$. The dual solution to (43) is then given by

$$\left[\begin{array}{c|c} 0 & 1_N^T \\ \hline 1_N & \frac{L-2}{L-1}\Omega + \frac{1}{L-1}\Omega_{CV} + I_N \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y - c \end{array}\right] \tag{46}$$

with Lagrange multipliers $\alpha \in \mathbb{R}^N$. One can obtain an expression for the individual models of the different folds such that the lth model can be evaluated in point x_j^v for $j \in \mathcal{V}_l$ as

$$y_j^v = w^{(l)^T}\varphi\left(x_j^v\right) + b + e_j^v = \sum_{i\in\mathcal{T}_l} \alpha_i K\left(x_i, x_j^v\right) + b + e_j^v \tag{47}$$

with residual $\hat{f}^{(l)}(x_j^v) - y_j^v$ denoted as e_j^v. The primal model (41) can also be recovered

$$\bar{w} = \sum_{i=1}^{N} \alpha_i \varphi(x_i) \tag{48}$$

and can be evaluated in a new point x^* by $\hat{f}_{CV}(x^*) = \bar{w}^T\varphi(x^*) + b = \sum_{i=1}^N \alpha_i K(x_i, x^*) + b$.

Proof: The Lagrangian of the constrained optimization problem (44) becomes

$$\mathcal{L}_{CV}^{P'}\left(w^{(l)}, b, \bar{e}_i; \alpha_i\right) = \sum_{l=1}^{L} \frac{w^{(l)^T} w^{(l)}}{2(L-1)} + \sum_{i=1}^{N} \frac{(\bar{e}_i - c_i)^2}{2(L-1)} - \sum_{i=1}^{N} \alpha_i \left(\frac{1}{L-1} \sum_{l|i\in\mathcal{T}_l} w^{(l)^T}\varphi(x_i) + b + \bar{e}_i - y_i \right). \tag{49}$$

The conditions for optimality w.r.t. $w^{(l)}, b, \bar{e}_i, \alpha_i$ for all $i = 1, \ldots, N$ and $l = 1, \ldots, L$ for the training become:

$$\begin{cases} \partial \mathcal{L}_{CV}^{P'}/\partial \bar{e}_i = 0 & \rightarrow \bar{e}_i = c_i + \alpha_i & \text{(a)} \\ \partial \mathcal{L}_{CV}^{P'}/\partial w^{(l)} = 0 & \rightarrow w^{(l)} = \sum_{i \in \mathcal{T}_l} \alpha_i \varphi(x_i) & \text{(b)} \\ \partial \mathcal{L}_{CV}^{P'}/\partial b = 0 & \rightarrow \sum_{i=1}^{N} \alpha_i = 0 & \text{(c)} \\ \partial \mathcal{L}_{CV}^{P'}/\partial \alpha_i = 0 & \rightarrow \frac{1}{L-1} \sum_{l|i \in \mathcal{T}_l} w^{(l)^T} \varphi(x_i) + b + \bar{e}_i = y_i. & \text{(d)} \end{cases} \tag{50}$$

From (50(b)) one can recover the model used in the training equations (44). Suppose one wants to evaluate the kth data-point during training, then

$$\begin{aligned} \sum_{l|k \in \mathcal{T}_l} w^{(l)^T} \varphi(x_k) &= \sum_{l|k \in \mathcal{T}_l} \sum_{i \in \mathcal{T}_l} \alpha_i \varphi(x_i)^T \varphi(x_k) \\ &= (L-2) \sum_{i=1}^{N} \alpha_i \varphi(x_i)^T \varphi(x_k) + \sum_{j \in \mathcal{V}_l} \alpha_j \varphi(x_j)^T \varphi(x_k). \end{aligned} \tag{51}$$

This leads to the dual linear system (46). □

Algorithm 5.1. (*Fast cross-validation based learning*). *The fusion of the training equations (46) and the validation set of equations (47) results in the following constrained optimization problem in the unknowns α, b, $\bar{e}$ and e^v:*

$$\min_{\alpha, b, \bar{e}, e^v} \mathcal{J}_2^{CV}(\bar{e}, e^v) = \|\bar{e}\|_2^2 + \xi \|\bar{e} - e^v\|_2^2$$

$$\textit{such that} \left[\begin{array}{c|c} 0 & 1_N^T \\ \hline 1_N & \frac{L-2}{L-1}\Omega + \frac{1}{L}\Omega_{CV} \\ \hline 0_N & \frac{-1}{L-1}\Omega_{CV} + \frac{L}{L-1}\Omega \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] + \left[\begin{array}{c} 0 \\ \hline \bar{e} \\ \hline \bar{e} - e^v \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \\ \hline 0_N \end{array}\right]. \tag{52}$$

Note that the drawback of using this fast method is that an extra tuning parameter ξ appears.

Remark 5.2. A motivation for the cost function (52) is given. Consider the more intuitive choice of the objective function $e^T \bar{e} + e^{v^T} e^v$. One can see that the residuals $\bar{e}_i$ and $e_i^v, \forall i = 1, \ldots, N$ are strongly correlated with each other as they estimate the same quantity (the true error of the ith observation). In order to take into account this cross-correlation, a generalized least squares argument can be used (Sen & Srivastave, 1990). To avoid building the correlation matrix of size $2N \times 2N$ explicitly, a different route is taken. The correlation structure of $\bar{e}$ and e^v is described by

$$\begin{bmatrix} \bar{e}^T \\ e^{v^T} \end{bmatrix} \begin{bmatrix} \bar{e} & e^v \end{bmatrix} = \begin{bmatrix} \bar{e}^T \bar{e} & \bar{e}^T e^v \\ e^{v^T} \bar{e} & e^{v^T} e^v \end{bmatrix}. \tag{53}$$

Consider the correlation matrix associated with the 2-norm of $\bar{e}$ and $\bar{e} - e^v$

$$\begin{bmatrix} \bar{e}^T \\ (\bar{e} - e^v)^T \end{bmatrix} \begin{bmatrix} \bar{e} & (\bar{e} - e^v) \end{bmatrix} = \begin{bmatrix} \bar{e}^T \bar{e} & \bar{e}^T (\bar{e} - e^v) \\ (\bar{e} - e^v)^T \bar{e} & (\bar{e} - e^v)^T (\bar{e} - e^v) \end{bmatrix}. \tag{54}$$

As $\bar{e}$ and e^v are strongly correlated ($\bar{e}^T e^v \approx \bar{e}^T \bar{e}$), one can see that $\bar{e}^T(\bar{e} - e^v) = \bar{e}^T \bar{e} - \bar{e}^T e^v \ll \bar{e}^T e^v$. As such, (54) is more diagonally dominant than (53) and (52) will result in more efficient estimates when using ordinary least squares.

Remark 5.3. An alternative for (52) is related to the α–stability of the learning machine f as defined amongst others in (Bousquet & Elisseeff, 2002; Schölkopf & Smola, 2002). Here, one works in a leave-one-out setting ($L = N$). The criterion bounds the following difference:

$$\max_{i \in \mathcal{D}} \left| \bar{e}_i - e_i^v \right| \leq \alpha_{\mathcal{D}}. \tag{55}$$

Note that the notation $\alpha_{\mathcal{D}}$ is used here instead of the common notation α, in order to avoid confusion with support values α_i in the model. As such, optimizing the following constrained optimization leads to an optimal α-stable machine:

$$\mathcal{J}^{CV}_{\infty,2}(\bar{e}, e^v) = \|\bar{e}\|_2 + \xi \|\bar{e} - e^v\|_\infty \quad \textit{s.t. constraints of (52) holds.} \tag{56}$$

Remark 5.4. Via ordinary least squares (52) can be solved in $O(lN^2)$ where $0 < l \ll N$ is the number of iteration steps in the conjugate gradient algorithm.

6. Experimental results

In this section, applications of the proposed method of additive regularization with fusion of training and validation levels for regression as well as classification problems are reported. The RBF kernel was used, except for the first linear experiment. As the focus of this paper is not on optimal kernel design, the bandwidth was held constant over the experiments of a task once it was tuned.

6.1. Regression benchmark studies

In order to test the performance of the proposed methods, a few classical benchmark studies were examined. The benchmarked algorithms are:

- LS-SVM according to (2) with Tikhonov regularization parameter γ determined by minimizing the validation cost. One third of the available dataset is reserved for validation purposes;
- LS-SVM according to (2) with Tikhonov regularization parameter γ set by minimizing the 10-fold cross-validation cost. The total dataset was used for computing the cross-validation cost;
- AReg LS-SVM with the tuning parameter λ found by minimizing the convex validation cost as in (28) according with the ensemble interpretation with two Tikhonov nodes ($m = 2$, $\gamma_{(1)} = 1$, $\gamma_{(2)} = 400$). Again one third of the dataset was reserved for validation purposes;
- AReg LS-SVM with the additive regularization constants c found by minimizing the 10-fold cross-validation cost function as in (37). As the size of the resulting set of

linear equations grows with NL, this full implementation was only tractable when the number of training and validation points is relatively small. The total dataset was used for computing the cross-validation cost;

- Fast algorithm of AReg LS-SVM based on the 10-fold cross-validation procedure as described in (52). The total dataset was used for computing the cross-validation cost.

The final model performance criterion was computed on an a priori fixed test-set using a model computed on the complete dataset with the regularization trade-off tuned as described above.

At first, two toy examples have been studied. The first dataset is a linear regression problem. 30 training points were generated in a two dimensional input space with Gaussian noise ($y = 2x^1 + 5x^2 + e$ where $e \sim \mathcal{N}(0, 3)$). According to the same distribution, 20 points were generated as validation data. 500 independent noise free data points were generated from the known underlying function in order to have a good criterion of the final performance. For the cross-validation based tuning algorithms, all 50 data-points were used. The second dataset (*sinc*) consists of data generated from $y = \mathrm{sinc}(x) + e$ where $e \sim \mathcal{N}(0, 0.3)$. The number of generated data of the training, validation and the noise free test set are 30, 20 and 500, respectively. The experiments were repeated 1000 times. Numerical results (Table 2) are expressed in terms of the average mean square error of the estimated regressor on the test set (Mean(MSE)) and the standard deviation of the MSE (Std(MSE)). This latter dataset was used to conduct a related experiment: the generalization performance is plotted with respect to the number of Tikhonov nodes as explained in Section 4 (solid line) of and a line search using m evaluations (dashed line). Both methods have the same computational cost for a fixed number of nodes respectively evaluations, while the AReg case outperforms the classical line-search. Figure 6(B) shows the average performances after repeating the experiment 1000 times.

In addition, two regression benchmark studies from the UCI Machine Learning Repository were studied, respectively the Abalone data (100 times a different division in training, validation and test set with $N = 700, n = 500, n_{\text{test}} = 2977$) and the Boston housing dataset (1000 times a different division in training, validation and test set with $N = 220, n = 120, n_{\text{test}} = 166$). In the first case, the full implementation of AReg LS-SVM (CV) was based on 4-fold CV. The average computation times on a PIV, 2 GHz linux pc in cpu seconds (Matlab's `cputime`) are given. Note that the computational cost of the classical tuning procedure is mostly due to the computation of the singular value decomposition. The mean performance and the standard deviation were expressed in Mean Square Error (MSE) on the test set.

Although the setup of the experiments was rather to illustrate the speedup of the methods with respect to speed and variance of the solution, the results show also an increased performance in the case of the first two experiments using the full implementation of AReg LS-SVM based on 10-fold cross-validation. According to the Wilcoxon Rank Sum test, the test set performance is even significantly better using the AReg (CV) LS-SVM for the first two toy examples (with confidence level 99%). While efficient numerical computations of the classical criteria can be done based on the singular value decomposition of the kernel matrix ($O(N^3)$), (see Appendix B), fusion of LS-SVMs with additive regularization and (cross-)validation results at much lower computational costs (see Remarks 3.3.1, 4.3.3, 5.2.1 and 5.3.3). Furthermore, the fusion-argument results

Table 2. Results of numerical experiments on regression benchmark datasets with the Tikhonov regularization based LS-SVMs (tuned for γ using validation (Val) and cross-validation (CV)) and the LS-SVMs with additiveregularization trade-off (AReg) (tuned for λ with validation and cross-validation). For the latter, results are given based on the full implementation (37) and the fast implementation (52). Results of two artificial datasets (a two-dimensional linear function and the *sinc* function) are given. The size of the training, validation and noise free test set were 30, 20, 500, respectively. Cross-validation based tuning procedures were provided with the joint training-validation dataset. Data generation, training and testing were repeated 1000 times. Performance is measured in average mean squared error (Mean(MSE)) and standard deviation (Std(MSE)) of the predictions on the test set which is fixed a priori in the different randomizations. Additionally, the techniques were compared on two benchmark data sets from the UCI Machine Learning Repository, the Abalone data ($N = 700$, $n = 500$, $n_{test} = 2977$ and $d = 7$) and the Boston housing dataset ($N = 220$, $n = 120$, $n_{test} = 166$, $d = 11$). Data division in training and validation set, tuning, training and testing were repeated respectively 100 and 1000 times. The results show also an increased performance in the case of the first two experiments using the full implementation of AReg LS-SVM based on 10-fold cross-validation. According to the Wilcoxon Rank Sum test, the test set performance is even significantly better using the AReg (CV) LS-SVM for the first two toy examples.

	Tuned		AReg		
LS-SVM	Val	CV	Val	CV	Fast CV
Linear regression (30, 20, 500)					
Mean(MSE):	0.5887	0.5931	0.5887	0.3796	0.5858
Std(MSE):	0.5108	0.5125	0.5108	0.4069	0.5074
Sinc (30, 20, 500)					
Mean(MSE):	0.0289	0.0269	0.0286	0.0174	0.0240
Std(MSE):	0.0217	0.0185	0.0210	0.0086	0.0145
Abalone (700, 500, 2977)					
Mean(MSE):	4.6609	4.8502	4.6622	5.0258	4.6216
Std(MSE):	0.1188	0.2311	0.1164	0.1808	0.0952
Computation time (s):	67.81	126.39	10.672	1401.6	19.28
Boston Housing (220, 120, 166)					
Mean(MSE):	0.1815	0.1883	0.1814	0.1874	0.1260
Std(MSE):	0.0491	0.0523	0.0500	0.0446	0.0262
Computation time (s):	0.1199	9.1834	0.0732	10.0728	1.0195

here in a global optimum (at least with the respect to the parametrization of the regularization constants according to the additive regularization trade-off), while the classical tuning methods do not guarantee global optimality.

6.2. *Classification benchmark studies*

Although the methods are explained in the framework of regression, they can be applied to classification problems in a similar fashion as explained in Appendix A. Table 3 reports results of benchmark studies on different datasets using the following methods:

Table 3. Results of numerical experiments on different classification benchmark datasets comparing the standard Tikhonov regularization based LS-SVMs and the LS-SVM based on the additive regularization trade-off scheme (AReg) (both tuned using the 2-norm on the validation residuals, the number of misclassifications (*mis*) of the validation data and the area under the curve of the ROC curve based on the validation data (*ROC*)). The results using the fast implementation of AReg LS-SVM based on 10-fold cross-validation (see Appendix A.6) are also reported. The performance is expressed as the percentage of correctly classified (PCC) test data. The first data set are the Ripley data. Other classification benchmark sets are the UCI Machine Learning Repository Pima Indians Diabetes Database (Pima) with $N = 768$, $d = 7$ and Liver-disorders Database (BUPA) with $N = 345$, $d = 6$. The division in 2/3 training and 1/3 validation set is 1000 times randomized. Finally, the UCI Adult Database ($N = 6000$, $n = 26561$, $n_{test} = 16281$) is used to illustrate the computational benefit in the case of large scale data. The fast implementation of the cross-validation based AReg LS-SVM used only the 6000 data-points. This experiment was only repeated once.

	Tuned			AReg (Val)			AReg (CV)
LS-SVM	2-norm	mis	ROC	2-norm	mis	ROC	2-norm
Ripley (150, 100, 1000)							
Mean(PCC):	90.42	90.43	90.46	90.41	90.43	90.46	90.45
Std(PCC):	0.07	0.17	0.12	0.10	0.14	0.12	0.11
Computation time (s):	0.33	0.33	1.38	0.013	0.018	1.06	0.015
Pima (325, 187, 256)							
Mean(PCC):	76.26	75.80	76.19	71.80	75.68	76.16	76.23
Std(PCC):	2.22	2.30	2.23	2.36	2.35	2.24	2.16
Computation time (s):	7.47	7.47	10.44	0.30	0.35	3.33	1.40
BUPA (155, 76, 115)							
Mean(PCC):	70.82	69.77	70.08	66.78	69.56	70.20	70.49
Std(PCC):	3.84	4.02	4.13	4.09	4.20	3.94	3.91
Computation time (s):	0.92	0.92	2.43	0.036	0.046	1.55	1.75
Adult (6000, 26561, 6281)							
PCC:	83.94	83.94	84.07	83.99	83.63	83.88	84.05
Computation time(h):	16.12	16.98	18.65	0.63	0.56	2.57	3.54

- LS-SVM classifier based on (57) with minimizing the 2-norm of the validation residuals, the number of misclassifications (*mis*) of the validation data and the area under the curve of the ROC curve based on the validation data (*ROC*);
- AReg LS-SVM classifier according to (63)(64) with $m = 2, \gamma_{(1)} = 1, \gamma_{(2)} = 20$. The same criteria as in the previous item were taken: minimizing the 2-norm on the validation residuals (leading to one single set of linear equations), the number of misclassifications (*mis*) (66) of the validation data and the area under the curve of the ROC curve based on the validation data (*ROC*) (Section A.5);
- Fast implementation of AReg LS-SVM based on 10-fold cross-validation (see Appendix A.6) using an 2-norm on the validation data.

In the case of a validation performance, one third of the dataset was reserved for validation purposes, while cross-validation used the complete dataset. The final model performance criterion was computed on an a priori fixed test-set using a model computed on the complete dataset with the regularization trade-off tuned as described above.

At first, the performance on the classical synthetic dataset (Ripley, 1996) is studied. The training data (250 points) are repeatedly divided into a disjoint training set of size 150 and a validation set of size 100. The misclassification rate and area under the curve (ROC) criterion lead to non-convex optimization problems. The fast AReg LS-SVM based on 10-fold CV uses all 250 data points. The synthetic test set of 1000 points is used to compare the different classifiers trained on all 250 data points with γ (λ) tuned as described above. The test set performance is expressed in the average percentage correctly classified data (Mean(PCC)) and standard deviation (Std(PCC)) of the 1000 randomizations.

Except for this toy example, the classifiers are also benchmarked using data sets of the UCI Machine Learning Repository:

- Pima Indians Diabetes Database (Pima) with $N = 325$, $n = 187$ and $n_{\text{test}} = 256$;
- Liver-disorders Database (BUPA) with $N = 155$, $n = 76$ and $n_{\text{test}} = 115$.

The experiment was repeated 1000 times. For aims of comparisons, the (average) computation time on a PIV, 2GHz linux pc in cpu seconds (Matlab's `cputime`) is given for the cases where the algorithmical complexity exceeds the computational overhead.

Special attention was paid to the larger scale Adult Database benchmark set for which a separate test set (16281 instances) was available. From the training set of 32561 instances, 6000 were used for training and the others for validation purpose. The fast implementation of the cross-validation based AReg LS-SVM used only this 6000 points. Experiments suggest that (the ratio of) performances do not change significantly in this case using a larger chunk of data during training.

The main conclusion one can draw from these examples is that the methods speed up considerably using AReg while the good performance remains preserved. Although no preference can be concluded from the experiments, it is clear that the choice of the norm used in validation has some importance.

7. Conclusions

While literature on model selection often focuses on the derivation, implementation or approximation of a suitable model selection criterion for a fixed model with given hyper-parameters, this paper considered the optimization of the regularization constants in LS-SVM regressors and classifiers. By introducing a different parameterization of the regularization trade-off, called the additive regularization trade-off, fusion of training and validation (or cross-validation) levels can be formulated as convex problems or even linear systems which result in the hyper-parameters and the corresponding training parameters at once. Computationally efficient implementations have been given together with methods to restrict the degrees of freedom. The latter has also led to a relation with ensemble methods. Simulation results indicate comparable results over the classical schemes that are based on classical Tikhonov regularization schemes, while global optimality is guaranteed and efficient state-of-the-art convex optimization algorithms can be employed. Within the use of an additive regularization trade-off scheme the global optimality of the model parameters and hyperparameters is also guaranteed as it follows from a convex problem. Further extensions and applications of this framework can be studied including e.g. the construction of stable kernel machines, the design of hierarchical kernel machines, extensions to input selection and the study of fusion in the case of unsupervised learning problems and links with robust statistics.

Appendix A: Classification

A.1. LS-SVM for classification

The framework of this paper applies equally well to pattern recognition tasks. In the case of a classification problem, the output data $y_i, y_j^v \in \{-1, 1\}$ $\forall i, j$. The cost function of the classification LS-SVM model $f(x) = \text{sgn}(w^T \varphi(x) + b)$ becomes (Suykens et al., 1999, 2002) $\min_{w,b,e_i} \mathcal{J}_\gamma(w, e) = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^N e_i^2$ such that $y_i(w^T \varphi(x_i) + b) = 1 - e_i$ $\forall i = 1, \ldots, N$. This formulation corresponds to a form of kernel Fisher discriminant analysis. (The well-known formulation of the Support Vector Machine classifier (Vapnik, 1998) has the following primal problem: $\min_{w,b,e_i} \mathcal{J}_\gamma(w, e) = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^N e_i$ such that $y_i(w^T \varphi(x_i) + b) \geq 1 - e_i$ & $e_i \geq 0, \forall i = 1, \ldots, N$).

By constructing the Lagrangian and taking the conditions for optimality w.r.t. w, b, e_i, α_i and applying the kernel trick, one obtains $w = \sum_{i=1}^N \alpha_i y_i \varphi(x_i)$, $e_i \gamma = \alpha_i$, $\sum_{i=1}^N \alpha_i y_i = 0$ and $y_i[w^T x_i + b] - 1 + e_i = 0$. The solution is given by the dual linear equations

$$\left[\begin{array}{c|c} 0 & y^T \\ \hline y & \Omega_y + I_N/\gamma \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 1_N \end{array}\right] \tag{57}$$

where $\Omega_{y(i,j)} = y_i y_j K(x_i, x_j)$. As such, the model is evaluated in a new point x^* as $\hat{f}(x^*) = \text{sgn}(\sum_{i=1}^N \alpha_i y_i K(x_i, x^*) + b)$. Again, the optimization of the regularization constant γ may become a non-convex optimization problem w.r.t. the validation errors.

A.2. Additive regularization trade-off

Additive regularization for LS-SVM classifiers (corresponding to a form of kernel Fisher discriminant analysis) for the model

$$f(x) = \text{sign}(w^T \varphi(x) + b) \tag{58}$$

is given by

$$\min_{w,b,e_i} \mathcal{J}_c(w, e_i) = \frac{1}{2} w^T w + \frac{1}{2} \sum_{i=1}^N (e_i - c_i)^2 \text{ s.t. } y_i(w^T \varphi(x_i) + b) = 1 - e_i \ \forall i = 1, \ldots, N. \tag{59}$$

Remark that the number of regularization constants c_i equals the number of data points N. After taking the Lagrangian, the conditions for optimality w.r.t. w, b, e_i, α_i for the training can be written in a set of dual linear equations:

$$\left[\begin{array}{c|c} 0 & y_N^T \\ \hline y_N & \Omega_y + I_N \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 1_N - (c.y) \end{array}\right]. \tag{60}$$

The residual $y_j^v \hat{f}(x_j^v) - 1$ (denoted as e_j^v) can be written as

$$e_j^v = 1 - y_j^v \left(w^T \varphi(x_j^v) + b \right) = 1 - \left(\Omega_y^v \alpha + y_j^v b \right) \tag{61}$$

where Ω_y^v denotes the matrix with as (i, j)th element $y_i y_j^v K(x_i, x_j^v)$.

A.3. *Fusion of training and validation*

After combination of the training (60) and the validation constraints (61), one obtains

$$\left[\begin{array}{cc|cc} 0_N^T & 1_n^T & 0 & y_N^T \\ \hline I_N & 0_{N\times n} & y & \Omega_y + I_N \\ \hline 0_{n\times N} & I_n & y^v & \Omega_y^v \end{array}\right] \left[\begin{array}{c} c \\ e^v \\ \hline b \\ \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 1_N \\ \hline 1_n \end{array}\right]. \tag{62}$$

Depending on the appropriate definition of optimality, different schemes to find the 'best' solution to this underdetermined set of equations can be considered. Similar to Section 3.2, one can solve this underdetermined set of equation using a least squares loss on the training errors $e = \alpha + c$ and on the validation errors e^v.

A.4. *Ensemble interpretation, convex combinations of Tikhonov node solutions*

Similar to Section 4, the degrees of freedom of the regularization constants can be restricted while preserving the convexity property. Application of the arguments for multiple nodes m can be applied as in Section 4.1 using the 2-norm classification criterion. Similar to Algorithm 4.1, this can be solved from an ensemble point of view:

1. (*First layer*): compute the Tikhonov node solutions corresponding to $\gamma_{(k)}$ for all $k = 1, \ldots, m$

$$\begin{cases} (\Omega_y + I_N \gamma_{(k)}^{-1})\,\alpha_{(k)} + y b_{(k)} = 1_N \\ \qquad\qquad\qquad\quad y_N^T \alpha_{(k)} = 0. \end{cases} \tag{63}$$

2. (*Second layer*): optimize the regularization constants $\lambda_{(k)}$ for all $k = 1, \ldots, m$

$$\min_{\lambda_{(1)},\ldots,\lambda_{(m)}} \left\| \sum_{k=1}^{m} \left[\Omega_y^v \alpha_{(k)} + y^v b_{(k)} \right] \lambda_{(k)} - 1_n \right\|_2^2 \text{ s.t. } \sum_{k=1}^{m} \lambda_{(k)} = 1,\, \lambda_{(k)} \geq 0 \forall k. \tag{64}$$

A.5. *Other validation criteria in view of the ensemble interpretation*

Validation of a classifier typically involves measuring the performance in terms of the number of misclassifications on a validation set. One can translate this using an indicator function $\mathcal{I}(\cdot)$ on the residuals defined as

$$\mathcal{I}(e) = \begin{cases} 1 & \text{if } \leq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{65}$$

As such, a good alternative for the cost criterion in (64) is

$$\min_{\lambda_{(1)},\ldots,\lambda_{(m)}} \mathcal{J}(\alpha, b) = \sum_{j=1}^{n} \mathcal{I}\left(\Omega_y^v \alpha + b y_j^v\right) \text{ s.t. } \sum_{k=1}^{m} \lambda_{(k)} = 1,\, \lambda_{(k)} \geq 0. \tag{66}$$

The ensemble point of view as explained in Section 4 is appropriate to solve this optimization problem: only the second step needs to be adapted. The drawback is that the minimization becomes non-convex (multiple local minima) and one should use a more costly line search algorithm. This algorithm is efficient compared to classical practice as the number of training steps (m) is small compared to the number of evaluations of the validation set (or the evaluations of the cost function).

In a similar way the area under the Receiver Operating Characteristic (ROC) curve (Hanley & McNeil, 1982) can be used as performance measure for binary classifiers.

A.6. Fusion of training and cross-validation levels for the classifier

A similar methodology as outlined in Section 5 applies. We report here only the fast implementation related to Subsection 5.3. Corresponding to (66), fusion of training and additive regularization with the cross-validation paradigm leads to

$$\left[\begin{array}{c|c} 0 & y_N^T \\ \hline y & \frac{L-2}{L-1}\Omega_y + \frac{1}{L}\Omega_{CV,y} \\ \hline y & \frac{-1}{L-1}\Omega_{CV,y} - \frac{L}{L-1}\Omega_y \end{array}\right] \left[\begin{array}{c} \bar{b} \\ \hline \bar{\alpha} \end{array}\right] + \left[\begin{array}{c} 0 \\ \hline e \\ \hline e - e^v \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 1_N \\ \hline 0_N \end{array}\right] \tag{67}$$

where

$$\Omega_{CV,y} = \begin{bmatrix} \Omega_{y\mathcal{V}_1,\mathcal{V}_1} & & & \\ & \Omega_{y\mathcal{V}_2,\mathcal{V}_2} & & \\ & & \ddots & \\ & & & \Omega_{y\mathcal{V}_L,\mathcal{V}_L} \end{bmatrix}. \tag{68}$$

This can be solved e.g. in least squares sense as follows

$$\mathcal{J}_2^{CV}(e, e^v) = \|e\|_2^2 + \xi\,\|e - e^v\|_2^2 \text{ s.t. (67) holds} \tag{69}$$

(see also the remarks in Subsection 5.3).

Appendix B: Relation with Tikhonov regularization trade-off scheme

B.1. Relation to standard LS-SVMs

The relation between the classical ridge regression with Tikhonov regularization and the additive regularization trade-off formulation (9) is investigated here. The solution α of the AReg LS-SVM (9) is equal to a solution α of the LS-SVM (1) provided that

$$c = (\gamma^{-1} - 1)\alpha \quad \text{s.t.} \quad \gamma \geq 0 \tag{70}$$

meaning that the vectors α and c need to be collinear. We refer to this (non-convex) constraint as the *Tikhonov constraint* within the context of the additive regularization framework. The training conditions (9) together with this quadratic constraint in α, c, λ would lead in fact to a non-convex optimization problem (see Figure 5).

Then, one may be interested in trying to recover the Tikhonov regularization constant γ in (1) from c (or $\lambda, \alpha_{(1)}$ and $\alpha_{(2)}$) obtained from (21). As both solution sets of α do not coincide over the entire range of possible regularization constants, a good thing to do is to obtain a value γ that leads to a solution (f_γ) which optimally approximates the solution related to c (f_c). For notational convenience, the bias term b is omitted here from the following criterion:

$$\gamma_c = \arg\min_{\gamma} \|f_\gamma(x) - f_c(x)\|_2^2 = \|\Omega(\Omega + I_N\gamma^{-1})^{-1}y - \Omega(\Omega + I_N)^{-1}(y - c)\|_2^2 \tag{71}$$

which results in a non-convex optimization problem.

The singular value decomposition of Ω can be used to make this optimization more efficient. The regression case of the LS-SVM of Eq. (1) without bias term b is considered without loss of generality. Consider the Singular Value Decomposition (SVD) (Golub & Van Loan, 1989) of the kernel matrix $\Omega = U^T SU$ with $U, S \in \mathbb{R}^{N\times N}$, U orthonormal ($U^T U = I$) and S diagonal matrices (note here that Ω is a symmetric, square and positive definite matrix). The elements S_{ii} for all $i = 1, \ldots, N$ of the diagonal matrix S are called the singular values. Consider the training equations (2) for given γ:

$$y = (\Omega + I_N\gamma^{-1})\alpha = U^T(S + I_N\gamma^{-1})U\,\alpha. \tag{72}$$

A well-known result is that $(\Omega^{-1} + I_N\gamma^{-1}) = U^T S_\gamma^{-1} U$ and S_γ^{-1} is a diagonal matrix with diagonal elements $(S_\gamma^{-1})_{ii} = (S_\gamma)_{ii}^{-1}$. Hence $S_\gamma^{-1} y_U = \alpha_U$ where $S_\gamma^{-1} = (S + I_N\gamma^{-1})^{-1}$, $y_U = Uy$ and $\alpha_U = U\,\alpha$. Note that this formula shows that the regularization constant γ affects only the singular values. Equation (71) can be rewritten as

$$\arg\min_{\gamma} \|U(SS_\gamma^{-1})y_U - U(SS_1^{-1})y_U^c\|_2^2 = \|(SS_\gamma^{-1})y_U - (SS_1^{-1})y_U^c\|_2^2 \tag{73}$$

as U preserves the norm (orthonormal matrix). This calculations results in an efficient line search once the SVD of the kernel matrix is computed as (SS_γ^{-1}) and (SS_1^{-1}) is diagonal. Note that S_1 denotes here $S_1 = S_\gamma|_{\gamma=1}$.

B.2 Relation to weighted LS-SVMs

Consider the weighted LS-SVM as described in (Suykens et al., 2002a) where a weighting term $\gamma_i \geq 0$ is considered per datapoint. Given model (7), observations $\{(x_i, y_i)\}_{i=1}^N$ and fixed regularization constants $\Gamma = (\gamma_1, \ldots, \gamma_N)^T \in \mathbb{R}^N$, the weighted cost function becomes

$$\min_{w,b,e_i} \mathcal{J}_\Gamma(w, e) = \frac{1}{2}w^T w + \frac{1}{2}\sum_{i=1}^N \gamma_i e_i^2 \quad \text{s.t.} \quad w^T\varphi(x_i) + b + e_i = y_i \;\forall i = 1, \ldots, N. \tag{74}$$

The dual solution is given analoguous to the derivation in Section 2.1 as follows

$$\left[\begin{array}{c|c} 0 & 1_N^T \\ \hline 1_N & \Omega + I_\Gamma \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \end{array}\right], \tag{7.5}$$

where $\Omega \in \mathbb{R}^{N\times N}$ with $\Omega_{ij} = K(x_i, x_j)$ and $I_\Gamma = \operatorname{diag}(\gamma_1^{-1}, \ldots, \gamma_N^{-1}) \in \mathbb{R}^{N\times N}$. The model can be evaluated at a new point x^* by $\hat{f}(x^*) = \sum_{i=1}^N \alpha_i K(x_i, x^*) + b$. Note that in Suykens et al. 2003 this scheme is employed in order to achieve robustness via an iterative reweighting procedure based on (74).

By comparison of the dual solution of the weighted LS-SVM and the LS-SVM with the AReg scheme (9), one can derive the conditions on Γ and c when the dual solutions will correspond:

$$c_i = (\gamma_i^{-1} - 1)\,\alpha_i \quad \text{s.t.} \quad \gamma_i \geq 0, \forall i = 1, \ldots, N. \tag{76}$$

This prooves the following corollary:

Corollary B.1 (Relation weighted LS-SVM and AReg) *One can rewrite the AReg scheme (8) as the solution to a weighted LS-SVM (74) with weighting constants* Γ *when the following non-convex conditions on c are satisfied*

$$\operatorname{sign}(\alpha_i)(c_i + \alpha_i) > 0. \tag{77}$$

Acknowledgments

We thank the anonymous reviewers for their constructive remarks. This research is supported by Research Council KUL: GOA-Mefisto 666, GOA AMBioRICS, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects, G.0240.99 (multilinear algebra), G.0197.02 (power islands), G.0141.03 (identification and cryptography), G.0491.03 (control for intensive care glycemia), G.0120.03 (QIT), G.0452.04 (new quantum algorithms), G.0407.02 (support vector machines), G.0499.04 (robust statistics), G.0211.05 (nonlinear identification), G.0080.01 (collective behaviour) AWI: Bil. Int. Collaboration Hungary/Poland; research communities (ICCoS, ANMMM, MLDM); IWT: PhD Grants,GBOU (McKnow); Belgian Federal Science Policy Office: IUAP P5/22 ('Dynamical Systems and Control: Computation, Identification and Modelling', 2002-2006), IUAP V; PODO-II (CP/40: TMS and Sustainability); EU: FP5-Quprodis; ERNSI; Eureka 2063-IMPACT; Eureka 2419-FliTE; Contract Research/agreements: ISMC/IPCOS, Data4s, TML, Elia, LMS, Mastercard.

References

Akaike, H. (1973). Statistical predictor identification. *Ann. Inst. Statist. Math,* 22, 203–217.

Bertero, M., Poggio, T. A., & Torre, T. (1988). Ill-posed problems in early vision. *Proceedings of the IEEE,* 76(8), 869–889.

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

Boyd, S., Vandenberghe, L. (2004). *Convex Optimization.* Cambridge University Press.

Bousquet, O., & Elisseeff, A. (2002). Stability and generalisation. *Journal of Machine Learning Research,* 2, 499–526.

Herrmann, D. J. L. & Bousquet, O. (2003). On the complexity of learning the kernel matrix. In Becker, S., Thrun, S. & Obermayer, K. (Eds.), *Advances in Neural Information Processing Systems* MIT Press, Cambridge, MA, (pp. 399–406).

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140.

Burman, P. (1989). A comparative study of ordinary cross-validation, *v*-fold cross-validation and the repeated learning-testing methods. *Biometrika, 76*(3), 503–514.

Cawley, G. C., & Talbot, N.L.C (2003). Efficient leave-one-out cross-validation of kernel Fisher discriminant classifiers. *Pattern Recognition, 36*(11), 2585–2592.

Chapelle, O., & Vapnik, V. (2000). Model selection for support vector machines. *Advances in Neural Information Processing Systems,* S. A. Solla, T. K. Leen & K.-R. Muller, (Eds) MIT Press, 12.

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning, 46*(1–3), 131–159.

Cherkassky, V., & Mulier, F. (1998). *Learning from Data.* Wiley, New York.

Craven, P., & Wahba, G. (1979). Smoothing noisy data with spline functions. *Numer. Math.* 31, 377–390.

Cucker, F., & Smale, S. (2002). Best choices for regularization parameters in learning theory: On the bias-variance problem. *Foundations of Computational Mathematics, 2*(4), 413–428.

De Brabanter, J., Pelckmans, K., Suykens, J. A. K., & Vandewalle, J. (2000). Robust cross-validation score function for non-linear function estimation. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN 2002),* Madrid, Spain, 713–719.

De Brabanter, J., Pelckmans, K., Suykens, J. A. K., De Moor, B., & Vandewalle, J. (2003). Robust complexity criteria for nonlinear regression in NARX models. In *Proceedings of the 13th System Identification Symposium (SYSID2003),* Rotterdam, the Netherlands (pp. 79–84).

Golub, G. H., Heath, M., & Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics, 21,* 215–223.

Golub, G. H., & Van Loan, C. F. (1989). *Matrix Computations.* The John Hopkins University Press.

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristics. *Radiology, 143,* 29–36.

Hansen, P. C., (1992). Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Review, 34*(4), 561–580.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning.* Springer-Verlag, Heidelberg.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics, 12*(1), 5582.

Ivanov, V. V. (1976). *The Theory of Approximate Methods and Their Application to the Numerical Solution of Singular Integral Equations.* Nordhoff International.

Kearns, M. (1997). A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split. *Neural Computation,* 9(5), 1143–1161.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., El Ghaoui, L., Jordan, M. I. (2004). Learning the Kernel Matrix with semidefinite programming, *Journal of Machine Learning Research, 5*(Jan), 27–72.

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

MacKay, D. J. C. (1998). Introduction to Gaussian processes. In *Neural networks and machine learning* (Ed. C.M. Bishop), Springer NATO-ASI Series F: Computer and Systems Sciences, 168, 133–165.

Mallows, C. L. (1973). Some comments on C_p. *Technometrics,* 15, 661–675.

Morozov, V. A. (1984). *Methods for Solving Incorrectly Posed Problems.* Springer-Verlag.

Pelckmans, K., De Brabanter, J., Suykens, J. A. K., & De Moor, B. (2003). Variogram based noise variance estimation and its use in Kernel Based Regression. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing,* 8, Toulouse, France (pp. 199–208).

Perrone, M. P. & Cooper, L. N. (1993). When networks disagree: Ensemble method for neural networks. In R. J. Mammone (Ed.), *Neural Networks for Speech and Image Processing,* Chapman-Hall.

Poggio, T., & Girosi, F. (1990). Networks for approximation and learning. *Proceedings of the IEEE, 78,* 1481–1497.

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

Rissanen, J. (1978). Modelling by shortest Data Description. *Automatica, 14,* 465–471.

Rockafeller, T. R. (1970). *Convex Analysis.* Princeton University Press.

Saunders, C., Gammerman, A., & Vovk, V. (1998). Ridge regression learning algorithm in dual variables. *Proceedings of the 15th Int. Conf. on Machine learning (ICML'98),* Morgan Kaufmann, 515–521.

Sen, A., & Srivastava, M. (1990). *Regression Analysis, Theory, Methods, and Applications.* Springer.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *J. Royal Statist. Soc. Ser. B, 36,* 111–147.

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

Suykens, J. A. K., De Brabanter, J., Lukas, L., & Vandewalle, J. (2002a). Weighted least squares support vector machines : robustness and sparse approximation. *Neurocomputing*, Special issue on fundamental and information processing aspects of neurocomputing, 48(1–4), 85–105.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., & Vandewalle, J. (2002b). *Least Squares Support Vector Machines.* World Scientific, Singapore.

Suykens, J. A. K., Horvath, G., Basu, S., Micchelli, C., Vandewalle, J. (Eds.) (2003) *Advances in Learning Theory: Methods, Models and Applications*, NATO Science Series III: Computer & Systems Sciences, 190, IOS Press Amsterdam.

Schölkopf, B., & Smola, A. (2002). *Learning with Kernels.* MIT Press, Cambridge, MA.

Schumaker, L. L. (1981). *Spline Functions: Basic Theory*. New York: Wiley.

Tikhonov, A., & Arsenin, V. Y. (1977). *Solutions of Ill-posed problems.* V.H.Winston & Sons, Washington D.C.

Van Gestel T. (2002). *From Linear to Kernel Based Methods in Classification, Modelling and Prediction.* Ph.D. Thesis, Faculty of Engineering, K.U.Leuven (Leuven, Belgium).

Vapnik, V. N. (1998). *Statistical Learning Theory*. John Wiley & Sons.

Wahba, G. (1990). *Splines Models for Observational Data*. Series in Applied Mathematics, 59, SIAM, Philadelphia.

Received January 3, 2005
Final Revision October 2, 2005
Accepted October 2, 2005