Automatic Relevance Determination for Least Squares Support Vector Machine Regression

T. Van Gestel, J.A.K. Suykens, B. De Moor & J. Vandewalle

K.U.Leuven, Dept. of Electrical Engineering, ESAT-SISTA,
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
E-mail: {tony.vangestel,johan.suykens}@esat.kuleuven.ac.be.

Abstract

In the past Automatic Relevance Determination (ARD) has been applied to multilayer perceptrons in order to select the relevant inputs of the regression problem at hand. This is obtained by inferring different regularization parameters for the input interconnection layer within the evidence framework. In this paper, the idea is extended towards Least Squares Support Vector Machines (LS-SVMs) for regression by weighting the inputs so as to obtain the most probable model for the given finite number of data points. On the first level of Bayesian inference, a probabilistic framework is related to the LS-SVM regression formulation. The hyperparameters that control the trade-off between regularization and error minimization are inferred on the second level of inference. Model comparison is performed on the third level in order to select the parameters of the kernel function. ARD is performed by introducing a diagonal weighting matrix in the kernel function. These diagonal elements are obtained by evidence maximization on the third level of inference. Inputs with a low scaling in the Radial Basis Function kernel are less relevant and can be removed.

1 Introduction

The Bayesian evidence framework has been successfully applied to the design of multilayer perceptrons (MLPs) in [1, 4]. The model parameters are inferred from the data by applying Bayes' rule on the first level of inference, with the prior and likelihood corresponding to the regularization and error term, respectively. The hyperparameters that control the trade-off between error minimization and regularization are inferred on the second level. Model comparison can be performed on the third level. Automatic Relevance Determination [4, 6] (ARD) involves the automatic determination of relevant inputs. Within the evidence framework, ARD is applied to MLPs by introducing additional regularization hyperparameters for the interconnections that are associated with each input. Evidence maximization is used to infer the regularization parameters and input selection can be performed by removing inputs with relatively large regularization constants.

In Support Vector Machines (SVMs) [3, 13] and Least Squares SVMs (LS-SVMs) [10], the inputs $x \in \mathbb{R}^n$ are preprocessed in a nonlinear way by the mapping $\varphi(\cdot): \mathbb{R}^n \to \mathbb{R}^{n_f}$ that maps the input $x \to \varphi(x)$ in a nonlinear way to a high n_f-dimensional feature space. A linear decision line is then constructed in the feature space space. The mapping $\varphi(x)$ is never explicitly calculated and Mercer's condition $\varphi(x_1)^T\varphi(x_2) = K(x_1, x_2)$ is applied instead. The weights and bias term of the SVM and LS-SVM can be obtained by applying Bayes' rule on the first level of inference [3, 12]. Hyperparameters are inferred on the second level, while the kernel parameters are obtained from model comparison on the third level of inference. In this paper, an ARD algorithm is proposed for LS-SVM classifiers [10] within the Bayesian evidence framework [12]. Since the mapping $\varphi(\cdot)$ is not explicitly known, also the weights of the input layer are unknown and in this sense ARD by optimal hyperparameter selection on level 2 cannot be applied. Instead, ARD for SVMs is obtained by assigning a weight [7] to each input of the kernel function K. These weights are inferred by applying model comparison on the third level of Bayesian inference. We illustrate the use of ARD for an Radial Basis Function (RBF) kernel on an artificial example with 1 relevant and 2 irrelevant inputs.

This paper is organized as follows. The inference of the model- and hyperparameters on level 1 and 2 is reviewed in Sections 2 and 3, respectively. Automatic Relevance Determination by model comparison on level 3 is discussed in Section 4. An example is given in Section 5.

2 Inference of the Model Parameters (Level 1)

The LS-SVM regressor $y = w^T\varphi(x) + b$ is inferred from the data $D = \{(x_i, y_i)\}_{i=1}^N$ by minimizing the cost function [10, 11]

$$\min_{w,b} \mathcal{J}_1(w, b) = \mu E_W + \zeta E_D = \frac{\mu}{2} w^T w + \frac{\zeta}{2} \sum_{i=1}^N e_i^2 \quad (1)$$

subject to the constraints

$$e_i = y_i - (w^T\varphi(x_i) + b), \qquad i = 1, \ldots, N. \quad (2)$$

The regularization and error term are defined as $E_W = \frac{1}{2}w^T w$ and $E_D = \frac{1}{2}\sum_{i=1}^N e_i^2$, respectively. The trade-off between regularization and training error is determined by the ratio $\gamma = \zeta/\mu$.

Vapnik's SVM formulation for regression [2, 13] involves the use of the so-called ϵ-insensitive loss function which leads to a convex Quadratic Programming problem in the dual space. The use of a least squares cost function [8, 10, 11, 12] results into a linear Karush-Kuhn-Tucker system in the dual space and allows to obtain analytic expressions on the three levels of Bayesian inference. Constructing the Lagrangian $\mathcal{L}_1(w, b, e; \alpha) = \mathcal{J}_1(w, e) - \sum_{i=1}^N \alpha_i(w^T\varphi(x_i) + b + e_i - y_i)$, by introducing the Lagrange multipliers $\alpha_i \in \mathbb{R}$ for the equality constraints (2) and taking the conditions for optimality, a linear Karush-Kuhn-Tucker system is obtained in the dual space

$$\left[\begin{array}{c|c} 0 & 1_v^T \\ \hline 1_v & \Omega + \gamma^{-1} I_N \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline Y \end{array}\right] \quad (3)$$

with $Y = [y_1; \ldots; y_N]$, $1_v = [1; \ldots; 1]$, $e = [e_1; \ldots; e_N]$, $\alpha = [\alpha_1; \ldots; \alpha_N]$, and where Mercer's condition is applied within the Ω matrix $\Omega_{ij} = \varphi(x_i)^T\varphi(x_j) = K(x_i, x_j)$. Possible kernel functions are, e.g.:

- Linear kernel $K(x_1, x_2) = x_1^T x_2$
- Radial Basis Function (RBF)-kernel $K(x_1, x_2) = \exp(-||x_1 - x_2||_2^2/\sigma^2)$,

where Mercer's condition holds for all possible choices of the kernel parameter $\sigma \in \mathbb{R}$. The LS-SVM regressor is then constructed as follows:

$$y(x) = \sum_{i=1}^N \alpha_i K(x, x_i) + b, \quad (4)$$

with latent variable $z = \sum_{i=1}^N \alpha_i K(x, x_i) + b$, by definition.

A probabilistic interpretation for (1)-(2) is obtained by applying Bayes' rule [12]

$$P(w, b|D, \log\mu, \log\zeta, \mathcal{H}) = \frac{P(D|w, b, \log\mu, \log\zeta, \mathcal{H})P(w, b|\log\mu, \log\zeta, \mathcal{H})}{P(D|\log\mu, \log\zeta, \mathcal{H})}, \quad (5)$$

where the model $\mathcal{H}$ corresponds to the kernel function K, possibly with kernel parameters. The evidence $P(D|\log\mu, \log\zeta, \mathcal{H})$ is a normalizing constant. The prior is assumed to be of the form $P(w, b|\log\mu, \log\zeta, \mathcal{H}) = P(w|\log\mu, \mathcal{H})P(b|\mathcal{H})$, with $P(b|\mathcal{H})$ a non-informative uniform distribution. A Gaussian prior

$$P(w|\log\mu, \mathcal{H}) = (\frac{\mu}{2\pi})^{n_f/2} \exp(-\frac{\mu}{2} w^T w) \quad (6)$$

is assumed. The likelihood is equal to

$$P(D|w, b, \log\zeta, \mathcal{H}) = \prod_{i=1}^N P(y_i|x_i, w, b, \log\zeta, \mathcal{H})P(x_i|w, b, \log\zeta, \mathcal{H}), \quad (7)$$

with constant $P(x_i|w, b, \log\zeta, \mathcal{H}) = P(x_i)$ and where the following conditional probability is assumed:

$$P(y_i|x_i, w, b, \log\zeta, \mathcal{H}) = (\frac{\zeta}{2\pi})^{1/2} \exp[-\frac{\zeta}{2}(y_i - (w^T\varphi(x_i) + b))^2]. \quad (8)$$

Substituting (6), (7) and (8) into Bayes' rule (5), we obtain the posterior probability

$$P(w, b|D, \log\mu, \log\zeta, \mathcal{H}) \propto \exp(-\frac{\mu}{2} w^T w) \exp(-\frac{\zeta}{2} \sum_{i=1}^N e_i^2). \quad (9)$$

The maximum a posteriori estimates w_{MP} and b_{MP} are obtained by minimizing the corresponding negative logarithm (1). This is equivalent to solving the linear system (3) in the dual space.

3 Inference of Hyperparameters (Level 2)

Applying Bayes' rule on the second level of inference [4, 12], we obtain:

$$P(\log\mu, \log\zeta|D, \mathcal{H}) \propto \frac{\sqrt{\mu^{n_f}\zeta^N}}{\sqrt{\det H}} \exp(-\mathcal{J}_1(w_{MP}, b_{MP})), \quad (10)$$

with the Hessian $H = \partial^2\mathcal{J}_1(w, b)/\partial[w; b]^2$. In a similar reasoning as in kernel PCA [9] the expression for the determinant of the Hessian H is given by $\det H = N\mu^{n_f - N_{eff}}\zeta\prod_{i=1}^{N_{eff}}(\mu+\zeta\lambda_{G,i})$, where the N_{eff} non-zero eigenvalues $\lambda_{G,i}$ corresponds to the N_{eff} non-zero eigenvalues of the centered Gram matrix in the feature space and are the solutions to the symmetric eigenvalue problem [12] (or Singular Value Decomposition)

$$(M\Omega M)v_{G,i} = \lambda_{G,i}v_{G,i}, \quad i = 1, ..., N_{eff} \leq N-1, \quad (11)$$

with $M = (I_N - \frac{1}{N}1_v 1_v^T)$ where $V_G = [v_{G,1}, ..., v_{G,N_{eff}}]$ and $D_G = \mathrm{diag}([\lambda_{G,1}, ..., \lambda_{G,N_{eff}}])$. The optimal parameters μ_{MP} and ζ_{MP} are found as the solution to the minimization problem:

$$\min_{\mu,\zeta} \mathcal{J}_2(\mu,\zeta) = \mu E_W(w_{MP}) + \zeta E_D(w_{MP}, b_{MP}) + \frac{1}{2}\sum_{i=1}^{N_{eff}}\log(\mu+\zeta\lambda_{G,i}) - \frac{N_{eff}}{2}\log\mu - \frac{N-1}{2}\log\zeta. \quad (12)$$

In the optimum, the following relations hold [3, 4, 12]:

$$2\mu_{MP}E_W(w_{MP}) = \gamma_{eff} - 1 \quad (13)$$

$$2\zeta_{MP}E_D(w_{MP}, b_{MP}) = N - \gamma_{eff}, \quad (14)$$

which is the Bayesian estimate estimate of the variance $\zeta^{-1} = \sum_{i=1}^N e_i^2/(N-\gamma_{eff})$ of the noise e_i. Combining both relations, we obtain a relation between μ_{MP} and the ratio $\gamma_{MP} = \zeta_{MP}/\mu_{MP}$:

$$2\mu_{MP}[E_W(w_{MP}) + \gamma_{MP}E_D(w_{MP}, b_{MP})] = N-1. \quad (15)$$

For the LS-SVMs, the effective number of parameters [1, 4, 12] is equal to:

$$\gamma_{eff} = 1 + \sum_{i=1}^{N_{eff}} \frac{\zeta_{MP}\lambda_{G,i}}{\mu_{MP} + \zeta_{MP}\lambda_{G,i}} = 1 + \sum_{i=1}^{N_{eff}} \frac{\gamma_{MP}\lambda_{G,i}}{1+\gamma_{MP}\lambda_{G,i}}, \quad (16)$$

where the first term is obtained because no regularization on the bias term b is used.

A practical way to find the maximum a posteriori estimates μ_{MP}, ζ_{MP} is to reformulate (12) in terms of $\gamma = \zeta/\mu$ using (15). Then the following scalar minimization problem in γ is obtained [12]:

$$\min_{\gamma} \mathcal{J}_3(\gamma) = \sum_{i=1}^{N-1}\log[\lambda_{G,i} + \frac{1}{\gamma}] + (N-1)\log[E_W(w_{MP}) + \gamma E_D(w_{MP}, b_{MP})] \quad (17)$$

with $\lambda_{G,i} = 0$ for $i > N_{eff}$. In this optimization problem, expressions for $E_{D,MP}$ and $E_{W,MP}$ can be obtained from the conditions for optimality of the Lagrangian on level 1 [10, 12].

As for Gaussian Processes, alternative expressions for $E_W + \gamma E_D$ can be derived:

$$E_W + \gamma E_D = \frac{1}{2}(Y - \hat{m}_Y 1_v)^T V_G (D_G + \gamma^{-1}I_{N_{eff}})^{-1} V_G^T (Y - \hat{m}_Y 1_v), \quad (18)$$

with the empirical mean $\hat{m}_Y = \frac{1}{N}\sum_{i=1}^N y_i$ Separate expressions for E_W and E_D can be obtained:

$$E_W = \frac{1}{2}(Y - \hat{m}_Y 1_v)^T V_G D_G (D_G + \gamma^{-1}I_{N_{eff}})^{-2} V_G^T (Y - \hat{m}_Y 1_v) \quad (19)$$

$$E_D = \frac{1}{2\gamma^2}(Y - \hat{m}_Y 1_v)^T V_G (D_G + \gamma^{-1}I_{N_{eff}})^{-2} V_G^T (Y - \hat{m}_Y 1_v). \quad (20)$$

From the optimal γ_{MP}, one easily obtains μ_{MP} and ζ_{MP} using the relations (13), (14) and (15) for μ, ζ, γ, $E_W(w_{MP})$ and $E_D(w_{MP}, b_{MP})$ that hold in the optimum of the cost function (12) of the second level of inference.

4 Automatic Relevance Determination by Inference of Kernel Parameters (Level 3)

By applying Bayes' rule on the third level, the posterior for the model $\mathcal{H}_j$ is obtained: $P(\mathcal{H}_j|D) \propto P(D|\mathcal{H}_j)P(\mathcal{H}_j)$. At this level, no evidence or normalizing constant is used since it is impossible to compare all possible models $\mathcal{H}_j$. The prior $P(\mathcal{H}_j)$ over all possible models is assumed to be uniform here. Hence, we obtain $P(\mathcal{H}_j|D) \propto P(D|\mathcal{H}_j)$. The likelihood $P(D|\mathcal{H}_j)$ corresponds to the evidence (10) of the previous level and can be approximated by [3, 4, 12]

$$P(D|\mathcal{H}_j) \propto P(D|\log\mu_{MP}, \log\zeta_{MP}, \mathcal{H}_j)\frac{\sigma_{\log\mu|D}\sigma_{\log\zeta|D}}{\sigma_{\log\mu}\sigma_{\log\zeta}}, \quad (21)$$

with $\sigma_{\log\mu}, \sigma_{\log\zeta}$ the standard deviations of the Gaussian priors (level 2) on $\log\mu$, $\log\zeta$, respectively. The first factor $P(D|\log\mu_{MP}, \log\zeta_{MP}, \mathcal{H}_j)$ corresponds to the goodness of fit, while the second factor corresponds to the Occam factor preferring simple models [4, 12].

The error bars $\sigma_{\log\mu|D}$ and $\sigma_{\log\zeta|D}$ can be approximated [4] as follows: $\sigma^2_{\log\mu|D} \simeq \frac{2}{\gamma_{eff}-1}$ and $\sigma^2_{\log\zeta|D} \simeq \frac{2}{N-\gamma_{eff}}$. The posterior (21) becomes [12]:

$$P(D|\mathcal{H}_j) \propto \sqrt{\frac{\mu_{MP}^{N_{eff}}\zeta_{MP}^{N-1}}{(\gamma_{eff}-1)(N-\gamma_{eff})\prod_{i=1}^{N_{eff}}(\mu_{MP}+\zeta_{MP}\lambda_{G,i})}}. \quad (22)$$

One selects the kernel parameters, e.g, σ_j for an RBF-kernel, with maximal posterior $P(D|\mathcal{H}_j)$.

For Automatic Relevance Determination, we now introduce a diagonal[1] weighting matrix [7] $U = \mathrm{diag}([u(1);...;u(n)])$. Each $u(l) \in \mathbb{R}^+$ weights the corresponding input $x(l)$, $l = 1, ..., n$ in the kernel function K. For an RBF-kernel, the kernel function becomes

$$\begin{aligned} K(x_1, x_2) &= \exp(-(x_1 - x_2)^T U (x_1 - x_2)/\sigma^2) \\ &= \exp(-(x_1 - x_2)^T \bar{U} (x_1 - x_2)), \end{aligned}$$

where the positive scale parameter σ is taken into account by defining $\bar{U} = U/\sigma$ and $\bar{U} = \mathrm{diag}(\bar{u}) = U/\sigma$. The weights $\bar{u}$ are inferred by maximizing the model evidence (22). The most relevant inputs will have larger weights, while the less important inputs will have relatively small weights.

In order to find a good starting value for optimizing $\bar{u}$ with respect to (22), we will first infer the optimal σ from (22) for $u = [1;...;1]$. This value then serves as the starting point $\bar{u} = [1;...;1]/\sigma$ for the more complex optimization of the weights $\bar{u}$. A practical algorithm consists of the following steps:

1. Normalize the inputs to zero mean and unit variance [1].
2. Optimize σ_j with respect to $P(D|\mathcal{H}_j)$ from (22). For each σ_j, the optimal μ_{MP}, ζ_{MP} and γ_{MP} are inferred on level 2 as follows:
3. Solve the eigenvalue problem (11).
4. Minimize $\mathcal{J}_3(\gamma)$ from (17).
5. Given γ_{MP}, calculate μ_{MP}, ζ_{MP} and γ_{eff}.
6. Calculate $P(D|\mathcal{H}_j)$ from (22).
7. Select an initial choice for $\bar{u}$, e.g, $\bar{u} = [1;...;1]/\sigma$ (when Step 2 was the previous step) or $\bar{u} = \bar{u}_{prev}(...;l-1;l+1;...)$ otherwise (lth input has been removed).
8. Optimize $\bar{u}_j$ with respect to $P(D|\mathcal{H}_j)$ from (22). See Step 2 for the different steps on level 1 (and level 2.
9. Remove inputs l with low $\bar{u}(l)$ values, go back to Step 3.

The main differences of this approach with Gaussian Processes [6] is that GP typically infer the kernel parameters on level 2, together with the hyperparameters.

[1] Instead of using a diagonal weighting matrix, the approach may be generalized towards any positive definite weighting matrix $\bar{U} \in \mathbb{R}^{n\times n}$. However, a physical interpretation of the importance of the weights is less obvious when there are significantly non-zero off-diagonal elements.

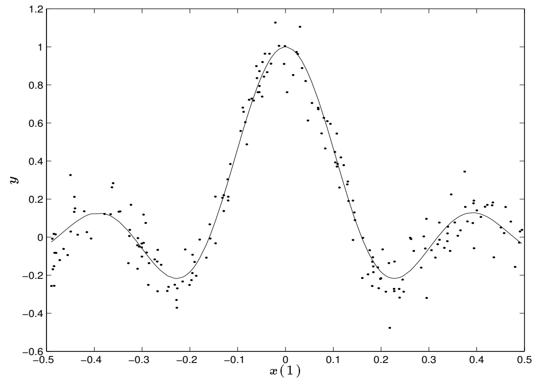

Figure 1: Training set (250 points) and true function for the ARD example.

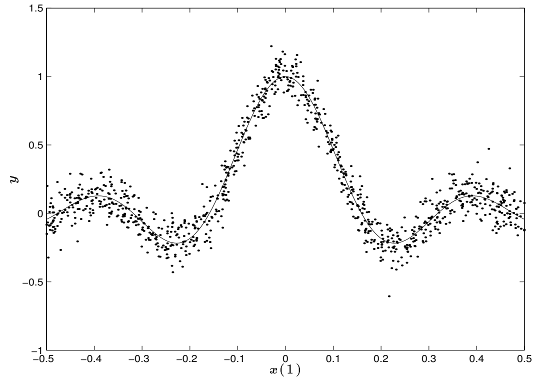

Figure 2: Test set (1000 points) and true function for the ARD example.

In this paper, we explicitly use the insight of the feature space and infer the model likelihood of LS-SVMs in a similar way as for MLPs. While in MLPs the weights are known, model comparison is now performed on the third level of inference in the global optimum of the level 1 cost function. In MLPs the analysis is done for a local optimum at the first level of inference.

5 Example: ARD with an RBF-kernel

We illustrate the ARD algorithm for an RBF-kernel on a synthetic dataset constructed in a similar way as in [5]. The data set consists of a training set and test set of $N = 200$ and $N_{test} = 1000$ data points, respectively. The 3 inputs were generated as follows. The first input $x(1)$ is generated by sampling uniformly in the interval $[-0.5, 0.5]$. The second input $x(2)$ is constructed from the first input by adding zero mean Gaussian distributed noise with variance 0.05^2. The third

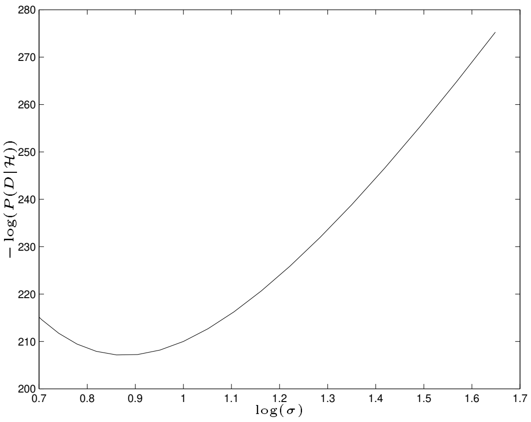

Figure 3: Evolution of the negative level 3 model probability $-\log(P(D|\mathcal{H}))$ as a function of $\log(\sigma)$. The corresponding test set MSE is depicted in Figure 4.

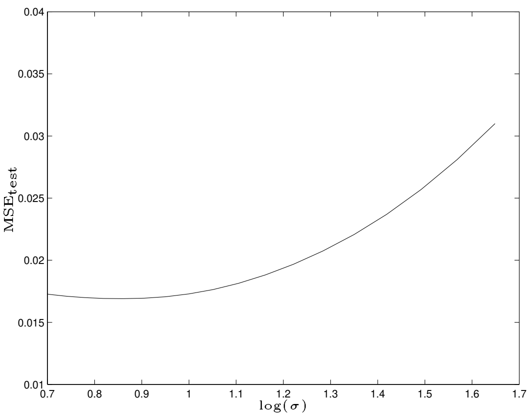

Figure 4: Evolution of the test set MSE as a function of $\log(\sigma)$.

input $x(3)$ is zero mean random noise with variance 0.5^2. The output data y are generated from the first input by adding Gaussian noise as follows:

$$y_i = \mathrm{sinc}(2\pi x_i(1)) + e_i, \tag{23}$$

with $\mathrm{sinc}(x) = \sin(\pi x)/(\pi x)$ and e_i zero mean Gaussian noise with variance $\mathrm{var}(e_i) = 0.1^2$. Hence, the first input is the most relevant for the regression problem, the second input is somewhat relevant and the third input is not relevant. The training and test data points are depicted in Figures 1 and 2, respectively.

From Step 2, we obtained $\sigma = 0.88$, $\mu_{MP} = 5.24$ and $\zeta_{MP} = 106.87$ (with $u = [1;1;1]$). The negative model probability $-\log(P(D|\mathcal{H}))$ and corresponding test set mean squared error (MSE) are depicted in Figures 5

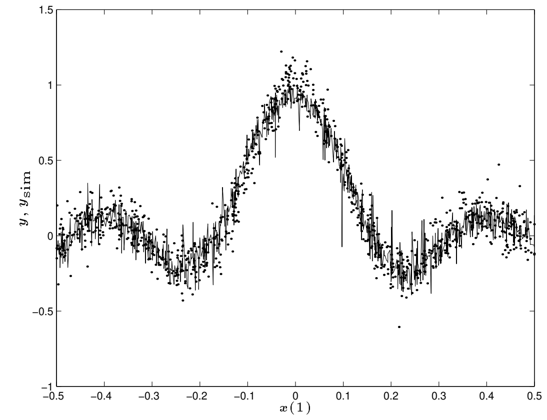

Figure 5: Test set data points $\{(x_i(1), y_i)\}_{i=1}^{1000}$ and predicted outputs $y_i = f(x_i([1:3]))$ for a RBF-kernel function with constant σ and 3 inputs. The simulated output is a non-smooth function of the first (relevant) input because there are two irrelevant noise inputs with equal importance in the RBF-kernel/

and 5, respectively. For the optimal $\sigma = 0.88$ the training set and test set MSE are 0.0058 and 0.0169, respectively. The simulated or predicted output y_{sim} from (4) are plotted in Figure 5 together with the test set data points. Observe that the simulated output is rather noise due to the two noisy inputs. In Step 3, we optimized $\bar{u}$ with respect to (22) for all inputs $x(1:3)$. This yielded $\bar{u} = [2.3908, 0.0020, 0.0007]$, $\mu_{MP} = 4.26$, $\zeta_{MP} = 113.77$, with training and test set MSE of 0.0083 and 0.0102, respectively. The test set performance is much smaller than the previous model by reducing the relevance of the noise inputs and is now almost equal to the variance of the additive noise. The simulated output is now a smooth function of the first input $x(1)$ as we can see from Figure 6. The evolution of $\bar{u}$ during the optimization is depicted in Figure 1. Removing input $x(3)$ with very low relevance, we restarted the optimization with $\bar{u}_{old} = [2.3908, 0.0020]$ for inputs $x(1:2)$. We obtained $\bar{u} = [1.9855, 0.0000]$, $\mu_{MP} = 2.99$, $\zeta_{MP} = 109.05$, while the training and test performances were 0.0085 and 0.0102, respectively. Input $x(1)$ is now far more important than input $x(2)$. The corresponding simulated output is depicted in Figure 7.

6 Conclusions

An Automatic Relevance Determination (ARD) algorithm is proposed for LS-SVM regression within the evidence framework. A diagonal weighting matrix is

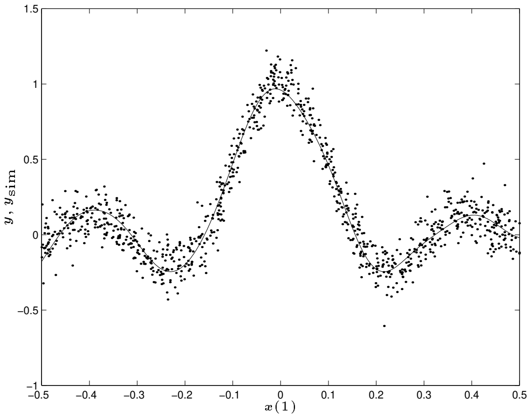

Figure 6: Test set data points $\{(x_i(1), y_i)\}_{i=1}^{1000}$ and predicted outputs $y_i = f(x_i([1 : 3]))$ for a RBF-kernel function with $\bar{u} = [2.3908, 0.0020, 0.0007]$ and 3 inputs.

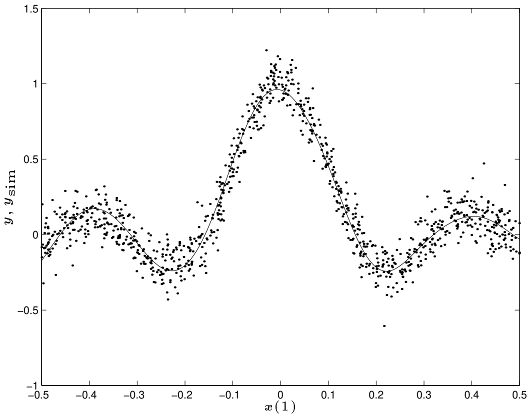

Figure 7: Test set data points $\{(x_i(1), y_i)\}_{i=1}^{1000}$ and predicted outputs $y_i = f(x_i([1 : 2]))$ for a RBF-kernel function with $\bar{u} = [1.9855, 0.0000]$ and 2 inputs.

taken for the inputs of the RBF-kernel. The weights are inferred on the third level of Bayesian inference. Inputs corresponding to small weights have low relevance in the kernel function and can be removed. Although the RBF kernel is known to be quite insensitive to irrelevant inputs, the generalization behavior in our experiment is improved by using a weighting matrix.

Acknowledgments

T. Van Gestel and J. Suykens are a Research Assistant and a Postdoctoral Researcher with the Fund for Scientific Research-Flanders (FWO-Vlaanderen), respectively. This work was partially supported by grants and projects from the Flemish Gov.: (Research council KULeuven: Grants, GOA-Mefisto 666; FWO-Vlaanderen: Grants, res. proj. G.0240.99, G.0256.97, and comm. (ICCoS and ANMMM); AWI: Bil. Int. Coll.; IWT: STWW Eureka IMPACT); from the Belgian Fed. Gov. (IUAP-IV/02, IV/24; Program Dur. Dev.); from the Eur. Comm.: (TMR Netw. (Alapedes, Niconet); Science: ERNSI). The scientific responsibility is assumed by its authors.

References

[1] Bishop, C.M. *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

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

[3] Kwok, J.T. "The evidence framework applied to support vector machines," *IEEE Transactions on Neural Networks, 11(5)*, 1162-1173, 2000.

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

[5] Nabney, I. *Netlab: Algorithms for Pattern Recognition*, 2001, to appear.

[6] Neal, R.M. *Bayesian Learning for Neural Networks.* Lecture Notes in Statistics 118, Springer, New York, 1996.

[7] Poggio, T., & Girosi, F. "Networks for Approximation and Learning," *Proceedings of the IEEE*, 78(9), 1481-1497, 1990.

[8] Saunders, C., Gammerman, A., & Vovk, V. "Ridge regression learning algorithm in dual variables", in *Proc. 15th Int. Conf. on Machine Learning ICML-98*, pp. 515-521, Madison-Wisconsin, 1998.

[9] Schölkopf, B., Smola A., & Müller, K.-M., "Nonlinear component analysis as a kernel eigenvalue problem", *Neural Computation*, vol. 10, pp. 1299-1319, 1998.

[10] Suykens, J.A.K. & Vandewalle, J. "Least squares support vector machine classifiers," *Neural Processing Letters, 9*, 293-300, 1999.

[11] Suykens, J.A.K., "Least Squares Support Vector Machines for Classification and Nonlinear Modelling," *Neural Network World*, vol. 10, pp. 29-48, 2000.

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

[13] Vapnik, V. *Statistical learning theory,* John Wiley, New-York, 1998.

[14] Vapnik, V. *Statistical Learning Theory,* John Wiley, NY, 1998.

[15] Williams, C.K.I., "Prediction with Gaussian Processes: from Linear Regression to Linear Prediction and Beyond," in *Learning and Inference in Graphical Models*, M.I. Jordan, ed., Kluwer Academic Press, 1998.