

*Journal of Statistical Software*

*MMMMMM YYYY, Volume VV, Issue II.* *http://www.jstatsoft.org/*

# Nonparametric Regression via StatLSSVM

**Kris De Brabanter**
KU Leuven

**Johan A.K. Suykens**
KU Leuven

**Bart De Moor**
KU Leuven

**Abstract**

We present a new Matlab toolbox under Windows and Linux for nonparametric regression estimation based on statistical library for least squares support vector machines (**StatLSSVM**). The **StatLSSVM** toolbox is written so that only a few lines of code are necessary in order to perform standard nonparametric regression, regression with correlated errors and robust regression. In addition, construction of additive models and pointwise or uniform confidence intervals are also supported. A number of tuning criteria such as classical cross-validation, robust cross-validation and cross-validation for correlated errors are available. Also, minimization of the previous criteria is available without any user interaction.

*Keywords*: nonparametric regression, pointwise confidence intervals, uniform confidence intervals, volume-of-tube-formula, asymptotic normality, robustness, reweighting, correlated errors, bimodal kernels.

## 1. Introduction

Nonparametric regression is a very popular tool for data analysis because it imposes few assumptions about the shape of the mean function. Therefore, the nonparametric regression is quite a flexible tool for modeling nonlinear relationships between dependent variable and regressors. The nonparametric and semiparametric regression techniques continue to be an area of active research. In recent decades, methods have been developed for robust regression (Maronna *et al.* 2006; Jurečková and Picek 2006; De Brabanter *et al.* 2009), regression with correlated errors (time series errors) (Chu and Marron 1991; Hart 1991; Hall *et al.* 1995; Opsomer *et al.* 2001; De Brabanter *et al.* 2011b), regression in which the predictor or response variables are curves (Ferraty and Vieu 2006), images, graphs, or other complex data objects, regression methods accommodating various types of missing data (Hastie *et al.* 2009; Marley and Wand 2010), nonparametric regression (Györfi *et al.* 2002), Bayesian methods for regression (Ruppert *et al.* 2003; Brezger *et al.* 2005; Brezger and Lang 2006), regression in which

the predictor variables are measured with error (Carroll *et al.* 2006; Meister 2009; Marley and Wand 2010), inference with regression (Hall 1992; Ruppert *et al.* 2003; Fan and Gijbels 1996; De Brabanter *et al.* 2011a) and nonparametric regression for large scale data sets (De Brabanter *et al.* 2010).

In this article we focus on several areas of nonparametric regression and statistical inference for least squares support vector machines (LS-SVM). Examples include (i) standard nonparametric regression, (ii) robust nonparametric regression, (iii) pointwise and uniform confidence intervals, (iv) additive models and (v) regression in the presence of correlated errors. By means of several examples we demonstrate how effectively the **StatLSSVM** toolbox is able to deal with the regression modeling areas which are stated previously. The Matlab toolbox like **StatLSSVM** has the advantage that an entire analysis can be managed with a single Matlab script or m-file. Because the package is based on Matlab syntax, one can take advantage of Matlab's functionality for data input and pre-processing, as well as summary and graphical display. The current version of **StatLSSVM** is compatible for Matlab R2009b or higher.

To our knowledge there is only one R (R Development Core Team 2009) implementation supporting LS-SVM i.e., the package **kernlab** by Karatzoglou *et al.* (2004). However, **kernlab** does not offer fully automatic data-driven procedures for tuning the parameters of LS-SVM and it is not able to handle several areas implemented in **StatLSSVM**.

The **StatLSSVM** toolbox is a part of the **LSSVMLab** software package (`http://www.esat.kuleuven.be/sista/lssvmlab/`). **StatLSSVM** is published under the GNU general public license and is freely available for research purposes at `http://www.esat.kuleuven.be/sista/statlssvm/`. This part of the **LSSVMLab** software project aims at offering the statistician an easy and fully functional set of nonparametric regression tools based on LS-SVM. A complete user's guide to **StatLSSVM** is available as a supplement to this paper as well as all the Matlab scripts in this paper. All data sets used in the examples are also included in **StatLSSVM**.

Section 2 gives a summary on least squares support vector machines. Section 3 discusses the various model selection criteria available in **StatLSSVM** together with the used optimization routines. Sections 4 through 8 deal with a specific nonparametric regression setting and demonstrates the capabilities of **StatLSSVM** on illustrative and real world examples. Conclusions are summarized in Section 9.

## 2. Some background on least squares support vector machines

In general, one of the key ingredients of support vector machines for regression is the following: Let $\Psi \subseteq \mathbb{R}^{n_f}$ denote a high dimensional (possibly infinite) feature space. Then a random input vector $X \in \mathbb{R}^d$ is mapped into this high dimensional feature space $\Psi$ through some mapping $\varphi : \mathbb{R}^d \to \Psi$. In fact, there is a relation with the existence of a Hilbert space $\mathcal{H}$ (Courant and Hilbert 1953) such that $\varphi : \mathbb{R}^d \to \mathcal{H}$ and $n_f$ is the dimension of $\mathcal{H}$. In this space, one considers a class of linear functions defined as

$$\mathcal{F}_{n,\Psi} = \left\{ f : f(X) = w^T \varphi(X) + b, \varphi : \mathbb{R}^d \to \Psi, w \in \mathbb{R}^{n_f}, b \in \mathbb{R} \right\}. \tag{1}$$

However, even if the linear function in the feature space (1) generalizes well, the problem of how to treat the high-dimensional feature space remains. Notice that for constructing the linear function (1) in the feature space $\Psi$, one does not need to consider the feature space in explicit form i.e., one only needs to replace the inner product in the feature space

$\varphi(X_k)^T\varphi(X_l)$, for all $k, l = 1, \ldots, n$, with the corresponding kernel $K(X_k, X_l)$. This result is known as Mercer's condition (Mercer 1909). As a consequence, to fulfil Mercer's condition one requires a positive (semi-) definite kernel function $K$. Least squares support vector machines (LS-SVM) for regression (Suykens *et al.* 2002b) are related to support vector machines (Vapnik 1999) where the inequality constraints have been replaced by equality constraints and the use of a squared loss is employed. Given a training data set $\mathcal{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ where $X \in \mathbb{R}^d$ and $Y \in \mathbb{R}$ and consider the model class $\mathcal{F}_{n,\Psi}$ defined in (1) and let $\gamma > 0$ be a regularization parameter. Then, LS-SVM for regression is formulated as follows

$$\begin{cases} \min\limits_{w,b,e} \mathcal{J}_P(w, e) = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum\limits_{i=1}^{n} e_i^2 \\ \qquad \text{s.t.} \qquad Y_i = w^T \varphi(X_i) + b + e_i, \quad i = 1, \ldots, n. \end{cases} \tag{2}$$

The square loss in (2) can be replaced by any other empirical loss. By using an $L_2$ loss function (and equality constraints) in LS-SVM, the solution is obtained in a linear system instead of using quadratic programming, see e.g. SVM, which speeds up computations. The problem is that LS-SVM lacks sparseness and robustness. For specialized literature on other loss functions and their properties, consistency and robustness, we refer the reader to Christmann and Steinwart (2007); Steinwart and Christmann (2008) and Steinwart and Christmann (2011). Suykens *et al.* (2002b) provides a benchmarking study on LS-SVM.

In Equation (2), it is clear that this model is linear in the feature space. This principle is illustrated in Figure 1. Consider a nonlinear relationship in the input space (Figure 1 left panel). Then, the inputs ($X$) are mapped into a high dimensional space by means of $\varphi$ (Figure 1 right panel). In this space, a linear model is fitted given the transformed data $\mathcal{D}_n^\star = \{(\varphi(X_1), Y_1), \ldots, (\varphi(X_n), Y_n)\}$ .

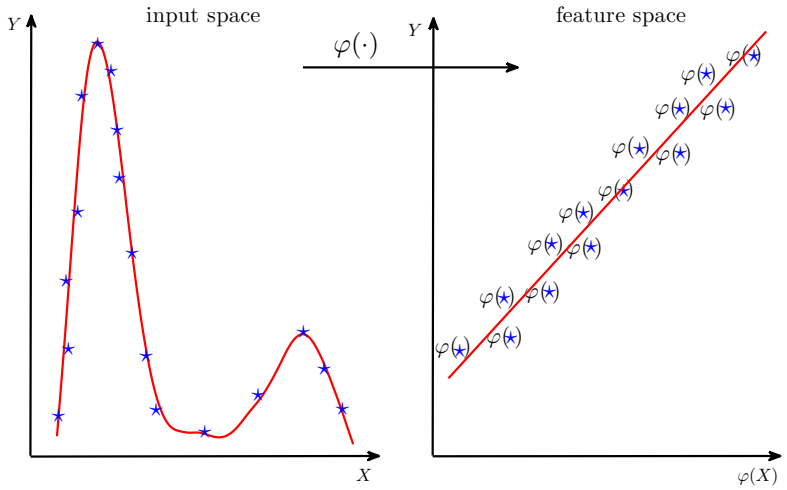



Figure 1: Illustration of the key ingredient of least squares support vector machines: Transformation $\varphi$ of the input data to the feature space.

Since $\varphi$ is in general unknown, problem (2) is solved by using Lagrange multipliers. The

Lagrangian is given by

$$\mathcal{L}(w,b,e;\alpha) = \frac{1}{2}w^T w + \frac{\gamma}{2}\sum_{i=1}^{n} e_i^2 - \sum_{i=1}^{n}\alpha_i\{w^T\varphi(X_i) + b + e_i - Y_i\},$$

where $\alpha_i \in \mathbb{R}$ are Lagrange multipliers. Then, the conditions for optimality are given by

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial w} = 0 & \rightarrow \quad w = \sum_{i=1}^{n}\alpha_i\varphi(X_i) \\ \dfrac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow \quad \sum_{i=1}^{n}\alpha_i = 0 \\ \dfrac{\partial \mathcal{L}}{\partial e_i} = 0 & \rightarrow \quad \alpha_i = \gamma e_i, \qquad\qquad\qquad i = 1,\ldots,n \\ \dfrac{\partial \mathcal{L}}{\partial \alpha_i} = 0 & \rightarrow \quad w^T\varphi(X_i) + b + e_i - Y_i = 0, \quad i = 1,\ldots,n. \end{cases}$$

After elimination of $w$ and $e$, parameters $b$ and $\alpha$ are estimated in the following linear system:

$$\left[\begin{array}{c|c} 0 & 1_n^T \\ \hline 1_n & \Omega + \frac{1}{\gamma}I_n \end{array}\right]\left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \mathbf{y} \end{array}\right], \tag{3}$$

with $\mathbf{y} = (Y_1,\ldots,Y_n)^T$, $1_n = (1,\ldots,1)^T$ and $\alpha = (\alpha_1,\ldots,\alpha_n)^T$. By using Mercer's condition, the $kl$-th element of $\Omega$ is given by

$$\Omega_{kl} = \varphi(X_k)^T\varphi(X_l) = K(X_k, X_l) \qquad k,l = 1,\ldots,n,$$

where $\Omega$ is a positive (semi-) definite matrix and the $kl$-th element of the matrix is $\Omega_{kl} = K(X_k, X_l)$. The kernel function $K$ is a symmetric, continuous positive definite function. Popular choices are the linear, polynomial and RBF kernel. In this paper we take $K(X_i, X_j) = (2\pi)^{-d/2}\exp(-\|X_i - X_j\|_2^2/2h^2)$. The resulting LS-SVM model is given by

$$\hat{m}(x) = \sum_{i=1}^{n}\hat{\alpha}_i K(x, X_i) + \hat{b}.$$

## 3. Model selection

In practical situations it is often preferable to have a data-driven method to estimate learning parameters. For this selection process, many data-driven procedures have been discussed in the literature. Commonly used are those based on the cross-validation criterion (Burman 1989) (leave-one-out and $v$-fold), the generalized cross-validation criterion (Craven and Wahba 1979), Akaike information criterion (Akaike 1973) etc. Several of these criteria are implemented in the toolbox (see the user's manual and the next sections).

Although these model selection criteria assist the user to find suitable tuning parameters or smoothing parameters (bandwidth $h$ of the kernel and the regularization parameter $\gamma$), finding the minimum of these cost functions tends to be tedious. This is due to the fact that the cost functions are often non-smooth and may contain multiple local minima. The latter is theoretically confirmed by Hall and Marron (1991).

A typical method to estimate the smoothing parameters would define a grid over these parameters of interest and apply any type of model selection method for each of these grid values. However, three disadvantages come up with this approach (Bennett *et al.* 2006; Kunapuli *et al.* 2008). A first disadvantage of such a grid-search model selection approach is the limitation of the desirable number of tuning parameters in a model, due to the combinatorial explosion of grid points. A second disadvantage is their practical inefficiency, namely, they are incapable of assuring the overall quality of the produced solution. A third disadvantage in grid-search is that the discretization fails to take into account the fact that the tuning parameters are continuous.

In order to overcome these drawbacks, we have equipped the toolbox with a powerful global optimizer, called coupled simulated annealing (CSA) (Xavier-de-Souza *et al.* 2010) and a derivative-free simplex search (Nelder and Mead 1965; Lagarias *et al.* 1998). The optimization process is twofold: First, determine good initial starting values by means of CSA and second, perform a fine-tuning derivative-free search using the previous end result as starting values. In contrast with other global optimization techniques CSA is not slow and can easily escape from local minima. The CSA algorithm based on coupled multiple starters is more effective than multi-start gradient descent optimization algorithms. Another advantage of CSA is that it uses the acceptance temperature to control the variance of the acceptance probabilities with a control scheme that can be applied to an ensemble of optimizers. This leads to an improved optimization efficiency because it reduces the sensitivity of the algorithm to the initialization parameters while guiding the optimization process to quasi-optimal runs. Because of the effectiveness of the combined methods only a small number of iterations are needed to reach an optimal set of smoothing parameters (bandwidth $h$ of the kernel and the regularization parameter $\gamma$).

## 4. Standard nonparametric regression

In this section we illustrate how to perform a nonparametric regression analysis with **StatLSSVM** in Matlab on the LIDAR data (Holst *et al.* 1996) and a two dimensional toy example. Step-by-step instructions will be given on how to obtain the results. All the data sets used in this paper are included in **StatLSSVM**.

### 4.1. Univariate smoothing

First, load the LIDAR data into the workspace of Matlab. This is done by `load('lidar.mat')`. After loading the data, one should always start by making a model structure using the `initlssvm` command.

```
>> model = initlssvm(x,y,[],[],'gauss_kernel')

model =
          x_dim: 1
          y_dim: 1
        nb_data: 221
         xtrain: [221x1 double]
         ytrain: [221x1 double]
```

```
          gam: []
  kernel_type: 'gauss_kernel'
    bandwidth: []
       status: 'changed'
      weights: []
```

This model structure contains all the necessary information of the given data (`xtrain` and `ytrain`), data size (`nb_data`), dimensionality of the data (`x_dim` and `y_dim`) and the chosen kernel function (`kernel_type`). **StatLSSVM** currently supports five positive (semi-) definite kernels i.e., the Gaussian kernel (`gauss_kernel`), the radial basis function kernel (`RBF_kernel`), the Gaussian additive kernel (`gaussadd_kernel`), a fourth order kernel based on the Gaussian kernel (`gauss4_kernel`) (Jones and Foster 1993) and the linear kernel (`lin_kernel`). Note that we did not specify any value yet for the smoothing parameters i.e., the bandwidth of the kernel (`bandwidth`) and the regularization parameter (`gam`) in the `initlssvm` command. We initialized these two parameters to the empty field in Matlab by `[]` in `initlssvm`. The `status` element of this structure contains information whether the model has been trained with the current set of smoothing parameters (Equation (3) is solved or not). If the model is trained (Equation (3) is solved) then the field `'changed'` will become `'trained'`. The last element `weights` specifies the weights used with robust regression (see Section 5).

Any field in the structure can be accessed by using `model.`*`field_name`*. For example, if one wants to access the regularization parameter in the structure `model`, one simply uses `model.gam`.

The next step is to tune the smoothing parameters. This is done by invoking `tunelssvm`. **StatLSSVM** support several model selection criteria for standard nonparametric regression such as leave-one-out cross-validation (`leaveoneout`), generalized cross-validation (`gcrossval`) and $v$-fold cross-validation (`crossval`). We illustrate the code for the $v$-fold cross-validation. By default, `crossval` uses 10-fold cross-validation and $L_2$ residual loss function. We will not show the complete output of the optimization process but only show the model structure output. The fields `gam` and `bandwidth` are no longer empty but contain their tuned value according the 10-fold cross-validation criterion.

```
>> model = tunelssvm(model,'crossval')

model =
         x_dim: 1
         y_dim: 1
       nb_data: 221
        xtrain: [221x1 double]
        ytrain: [221x1 double]
           gam: 5.4603
  kernel_type: 'gauss_kernel'
    bandwidth: 45.8881
       status: 'changed'
      weights: []
```

Next, to obtain the final model the system of equations (3) has to be solved to acquire the $\alpha$ vector (Lagrange multipliers) and $b$ (bias term). Training the model can be done as follows.

```
>> model = trainlssvm(model)

model =
          x_dim: 1
          y_dim: 1
       nb_data: 221
        xtrain: [221x1 double]
        ytrain: [221x1 double]
           gam: 5.4603
   kernel_type: 'gauss_kernel'
     bandwidth: 45.8881
        status: 'trained'
       weights: []
             b: -0.3089
         alpha: [221x1 double]
      duration: 0.0096
```

Note that the field `status` has been altered from `'changed'` to `'trained'`. Also the Lagrange multipliers `alpha` and bias term `b` have been added to the model structure. The last line in the structure denotes the time needed to solve the system of equations (3) in seconds.

The last step is to visualize the results (if possible). By using

```
>> model = plotlssvm(model);
>> xlabel('range'); ylabel('log ratio')
```

we obtain Figure 2. The full line indicates the resulting LS-SVM solution. The chosen kernel and the tuned smoothing parameters are given above the figure.

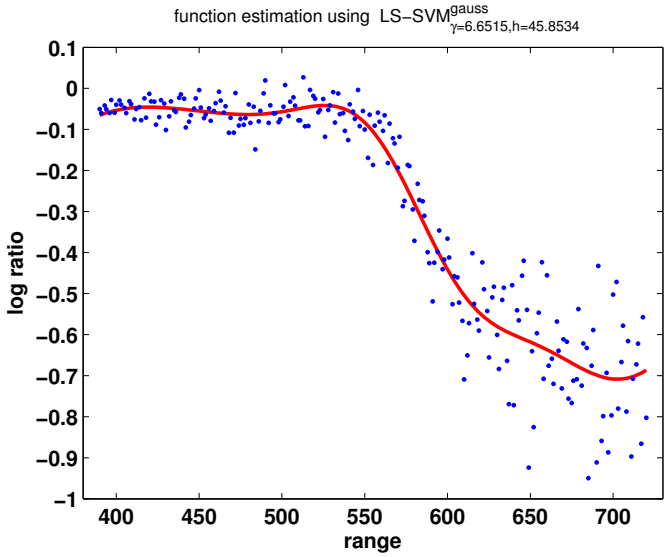



Figure 2: Nonparametric regression estimation with **StatLSSVM** on the LIDAR data set. The title of the figure specifies the chosen kernel and the tuned smoothing parameters.

### 4.2. Bivariate smoothing

In this example of bivariate smoothing, the NBA data set is used (available in **StatLSSVM**) `nba.mat` (Simonoff 1996) into the workspace. Since the workflow is exactly the same as in the previous example we only give the input script and visualize the results. The vector $x \in \mathbb{R}^2$ and $y \in \mathbb{R}$. The fitted regression surface is an estimate of the mean points scored per minute conditional on the number of minutes played per game and height in centimeters for 96 NBA players who played the guard position during the 1992-1993 season. As a model selection method we choose leave-one-out cross-validation. The relevant **Matlab** command is

```
>> load('nba.mat','x','y')
>> model = initlssvm(x,y,[],[],'gauss_kernel');
>> model = tunelssvm(model,'leaveoneout');
>> model = plotlssvm(model);
>> figure(2)
>> model = plotlssvm(model,'contour');
```

Figure 3 shows the three dimensional plot and the contour plot of the LS-SVM estimate. From the figure it is clear that there is a trend towards higher scoring when taller players are longer in the field.

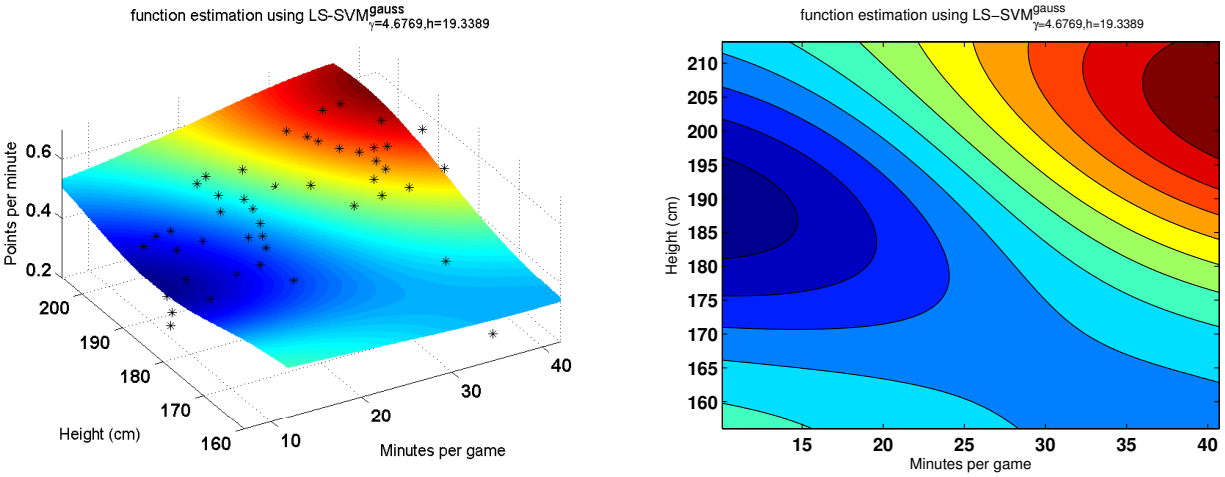



Figure 3: Three dimensional plot and contour plot of the LS-SVM estimate of points scored per minute as a function of minutes played per game and height.

## 5. Robust nonparametric regression

Regression analysis is an important statistical tool routinely applied in most sciences. However, using least squares techniques, there is an awareness of the dangers posed by the occurrence of outliers present in the data. Not only the response variable can be outlying, but also the explanatory part, leading to leverage points. Both types of outliers may totally

spoil an ordinary least squares (LS) analysis. We refer to the books of Hampel *et al.* (1986), Rousseeuw and Leroy (2003), Maronna *et al.* (2006) and Jurečková and Picek (2006) for a thorough survey regarding robustness aspects.

A possible way to robustify (2) is to use an $L_1$ loss function. However, this would lead to a quadratic programming problem and is more difficult to solve than a linear system. Therefore, we opt for a simple but effective method i.e., iterative reweighting (De Brabanter *et al.* 2009; Debruyne *et al.* 2010). This approach solves a weighted least squares problem in each iteration until a certain stopping criterion is satisfied. **StatLSSVM** supports four weight functions: Huber, Hampel, Logistic and Myriad weights. Table 1 illustrates these four weight functions.

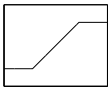
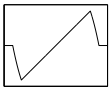
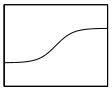
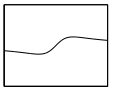
| | Huber | Hampel | Logistic | Myriad |
|---|---|---|---|---|
| $V(r)$ | $\begin{cases} 1, & \text{if } \lvert r\rvert < \beta; \\ \frac{\beta}{\lvert r\rvert}, & \text{if } \lvert r\rvert \geq \beta. \end{cases}$ | $\begin{cases} 1, & \text{if } \lvert r\rvert < b_1; \\ \frac{b_2 - \lvert r\rvert}{b_2 - b_1}, & \text{if } b_1 \leq \lvert r\rvert \leq b_2; \\ 0, & \text{if } \lvert r\rvert > b_2. \end{cases}$ | $\frac{\tanh(r)}{r}$ | $\frac{\delta^2}{\delta^2 + r^2}$ |
| $\psi(r)$ |  |  |  |  |
| $L(r)$ | $\begin{cases} r^2, & \text{if } \lvert r\rvert < \beta; \\ \beta\lvert r\rvert - \frac{1}{2}c^2, & \text{if } \lvert r\rvert \geq \beta. \end{cases}$ | $\begin{cases} r^2, & \text{if } \lvert r\rvert < b_1; \\ \frac{b_2 r^2 - \lvert r^3\rvert}{b_2 - b_1}, & \text{if } b_1 \leq \lvert r\rvert \leq b_2; \\ 0, & \text{if } \lvert r\rvert > b_2. \end{cases}$ | $r\tanh(r)$ | $\log(\delta^2 + r^2)$ |

Table 1: Definitions for the Huber ($\beta \in \mathbb{R}^+$), Hampel, Logistic and Myriad ($\delta \in \mathbb{R}_0^+$) weight functions $V(\cdot)$. The corresponding loss $L(\cdot)$ and score function $\psi(\cdot)$ are also given.

A robust version of (2) is then formulated as follows (Suykens *et al.* 2002a):

$$\begin{cases} \min\limits_{w,b,e} \mathcal{J}_P(w,e) = \frac{1}{2}w^T w + \frac{\gamma}{2}\sum\limits_{i=1}^{n} v_i e_i^2 \\ \qquad \text{s.t.} \qquad Y_i = w^T\varphi(X_i) + b + e_i, \quad i = 1,\ldots,n, \end{cases}$$

where $v_i$ denotes the weight of the $i$th residual. The weights are assigned according to the chosen weight function in Table 1. Again, by using Lagrange multipliers, the solution is given by

$$\left[\begin{array}{c|c} 0 & 1_n^T \\ \hline 1_n & \Omega + D_\gamma \end{array}\right]\left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline Y \end{array}\right],$$

with $Y = (Y_1,\ldots,Y_n)^T$, $1_n = (1,\ldots,1)^T$, $\alpha = (\alpha_1,\ldots,\alpha_n)^T$ and $D_\gamma = \text{diag}\{\frac{1}{\gamma v_1},\ldots,\frac{1}{\gamma v_n}\}$. Suppose we observe the data $\mathcal{D}_n = \{(X_1,Y_1),\ldots,(X_n,Y_n)\}$, but the $Y_i$ are subject to occasional outlying values. An appropriate model is

$$Y_i = m(X_i) + \varepsilon_i,$$

for a smooth function $m$ and the $\varepsilon_i$ come from the gross-error model (Huber 1964) with symmetric contamination. The gross-error model or $\epsilon$-contamination model is defined as

$$\mathcal{U}(F_0,\epsilon) = \{F : F(\varepsilon) = (1-\epsilon)F_0(\varepsilon) + \epsilon G(\varepsilon), 0 \leq \epsilon \leq 1\},$$

where $F_0$ is some given distribution (the ideal nominal model), $G$ is an arbitrary continuous symmetric distribution and $\epsilon$ is the contamination parameter. This contamination model describes the case, where with large probability $(1-\epsilon)$, the data occurs with distribution $F_0$ and with small probability $\epsilon$ outliers occur according to distribution $G$. In our toy example we generate a data set set containing 35% outliers, where the distribution $F_0$ is taken to be the Normal distribution with variance 0.01 and $G$ is the standard Cauchy distribution. In order to obtain a fully robust solution, one must also use a robust model selection method (Leung 2005). Therefore, **StatLSSVM** supports a robust $v$-fold cross-validation procedure (`rcrossval`) based on a robust LS-SVM smoother and robust loss function i.e., the $L_1$ loss (`mae`) or Huber's loss (`huber`) instead of $L_2$ (`mse`). Figure 4 provides an illustration of **StatLSSVM** fitting via the next script for simulated data with $n = 250$, $\epsilon = 0.35$, $m(X) = \text{sinc}(X)$ and $X \sim \mathcal{U}[-5, 5]$. Note that this example requires the Matlab Statistics Toolbox (generation of $t$ distributed random numbers). The relevant **Matlab** command is

```
>> X = -5+10*rand(250,1);
>> epsilon = 0.35;
>> sel = rand(length(X),1)>epsilon;
>> Y = sinc(X)+sel.*normrnd(0,.1,length(X),1)+(1-sel).*trnd(1,length(X),1);
>> model = initlssvm(X,Y,[],[],'gauss_kernel');
>> model = tunelssvm(model,'rcrossval',{10,'mae'},'wmyriad');
>> model = robustlssvm(model);
>> model = plotlssvm(model);
```

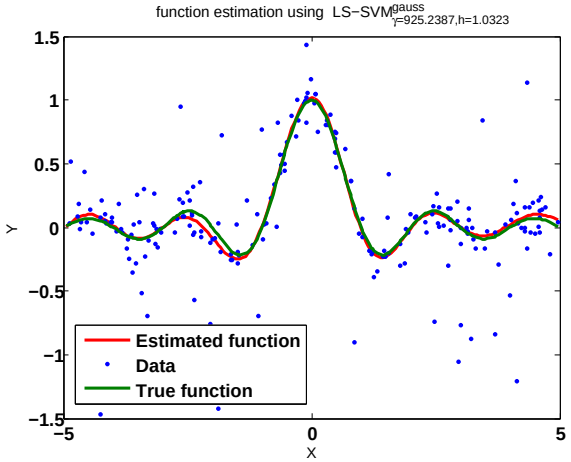



Figure 4: Robust LS-SVM estimate based on iterative reweighting with myriad weights.

**Remark 1.** *The other weights (see Table 1) can be called as* `whuber`, `whampel` *or* `wlogistic` *as last argument of the command* `tunelssvm`. *More information about all functions can be found in the supplements of this paper or via the* Matlab *command window via the* `help` *function. For example* `help robustlssvm`.

*More information and properties regarding the weights in Table 1 and the complete robust tuning procedure can be found in De Brabanter (2011).*

# 6. Confidence intervals

In this section we consider the following nonparametric regression setting

$$Y = m(X) + \sigma(X)\varepsilon,$$

where $m$ is a smooth function, $\mathsf{E}[\varepsilon|X] = 0$, $\mathsf{VAR}[\varepsilon|X] = 1$ and $X$ and $\varepsilon$ are independent. Two possible situations can occur: (i) $\sigma^2(X) = \sigma^2$ (homoscedastic regression model) and (ii) the variance is a function of the explanatory variable $X$ (heteroscedastic regression model). We do not discuss the case when the variance function is a function of the regression function. Our goal is the determine confidence intervals (CI) for $m$.

## 6.1. Pointwise confidence intervals

Under certain regularity conditions, it can be shown that asymptotically (De Brabanter 2011)

$$\frac{\hat{m}(x) - m(x) - b(x)}{\sqrt{V(x)}} \xrightarrow{d} \mathcal{N}(0,1),$$

where $b(x)$ and $V(x)$ are respectively the bias and variance of $\hat{m}(x)$. With the estimated bias and variance given in De Brabanter *et al.* (2011a), an approximate $100(1-\alpha)\%$ pointwise confidence interval for $m$ is

$$\hat{m}(x) - \hat{b}(x) \pm z_{1-\alpha/2}\sqrt{\hat{V}(x)},$$

where $z_{1-\alpha/2}$ denotes the $(1-\alpha/2)^{\text{th}}$ quantile of the standard Gaussian distribution. This approximate CI is valid if

$$\frac{\hat{V}(x)}{V(x)} \xrightarrow{P} 1 \quad \text{and} \quad \frac{\hat{b}(x)}{b(x)} \xrightarrow{P} 1.$$

This in turn requires a different bandwidth used in assessing the bias and variance (Fan and Gijbels 1996) which is automatically done in the **StatLSSVM** toolbox.

The following **Matlab** command generates a $100(1-\alpha)\%$ pointwise confidence interval ($\alpha = 0.05$) for the fossil data set (Ruppert *et al.* 2003). The result is visualized in Figure 5. The relevant **Matlab** command is

```
>> load('fossil.mat','x','y')
>> model = initlssvm(x,y,[],[],'gauss_kernel');
>> model = tunelssvm(model,'crossval');
>> model = trainlssvm(model);
>> model = plotlssvm(model);
>> hold all;
>> ci = cilssvm(model,0.05,'pointwise');
>> fill([x;flipud(x)],[ci(:,1);flipud(ci(:,2))],'g','FaceAlpha',0.5,...
        'EdgeAlpha',1,'EdgeColor','w')
```

## 6.2. Uniform confidence intervals

In order to make simultaneous (or uniform) statements we have to modify the width of the interval to obtain simultaneous confidence intervals (see also multiple comparison theory).

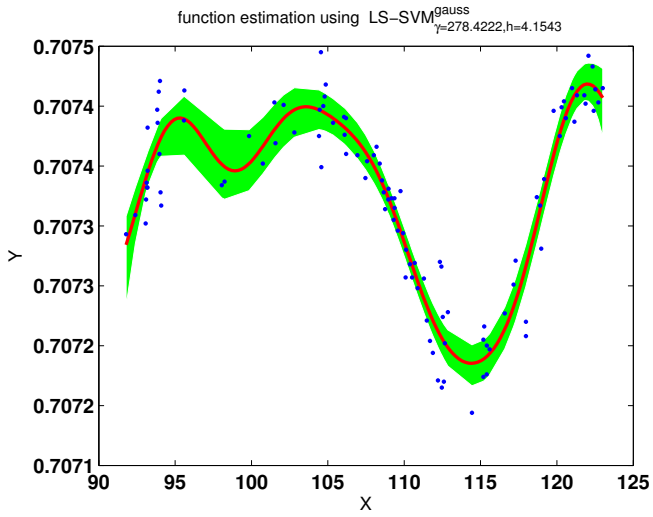



Figure 5: Nonparametric regression estimation with **StatLSSVM** on the fossil data set. The green shaded region corresponds to 95% pointwise biased corrected confidence intervals.

Mathematically speaking, we are searching for the width of the bands $c$, given a confidence level $\alpha \in (0,1)$, such that

$$\inf_{m \in \mathcal{F}_n} \mathsf{P}\left\{\hat{m}(x) - c\sqrt{\hat{V}(x)} \leq m(x) \leq \hat{m}(x) + c\sqrt{\hat{V}(x)}, \forall x \in \mathbb{R}^d\right\} = 1 - \alpha,$$

for some suitable class of smooth functions $\mathcal{F}_n$. In **StatLSSVM** the width of the bands $c$ is determined by the volume-of-tube formula (Sun and Loader 1994). We illustrate the concept of simultaneous confidence intervals with two examples. First, consider the following heteroscedastic regression example. The data generation process is as follows:

$$y_i = \sin(x_i) + \sqrt{0.05x_i^2 + 0.01}\varepsilon \quad i = 1, \ldots, 300,$$

where $x$ are equally spaced over the interval $[-5, 5]$ and $\varepsilon \sim \mathcal{N}(0,1)$. The relevant **Matlab** command is

```
>> x = linspace(-5,5,300)';
>> y = sin(x)+sqrt(0.05*x.^2+0.01).*randn(300,1);
>> model = initlssvm(x,y,[],[],'gauss_kernel');
>> model = tunelssvm(model,'crossval',{10,'mae'});
>> model = trainlssvm(model);
>> model = plotlssvm(model);
>> hold all;
>> ci = cilssvm(model,0.05,'simultaneous','heteroscedastic');
>> fill([x;flipud(x)],[ci(:,1);flipud(ci(:,2))],'g','FaceAlpha',0.5,...
       'EdgeAlpha',1,'EdgeColor','w')
>> plot(x,sin(x),'k','LineWidth',2)
```

As a second example, the LIDAR data set is used. Uniform and simultaneous CI's are shown in Figure 6.

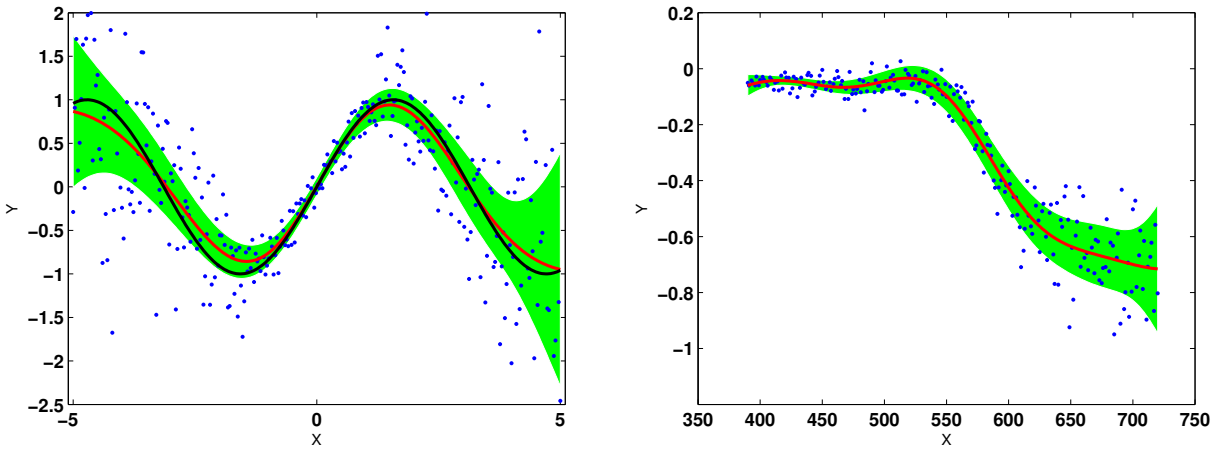



Figure 6: Left panel: Bias-corrected simultaneous CIs for the heteroscedastic toy example (green shaded area) together with the LS-SVM estimate (red line). The black line is the true function. Right panel: Bias-corrected simultaneous CIs for the LIDAR data set together with the LS-SVM estimate (red line).

## 7. Additive LS-SVM models

Suppose a sample of observations $(X_i, Y_i)$ ($X_i \in \mathbb{R}^d$ and $Y_i \in \mathbb{R}$) is generated from an additive model

$$Y_i = b + \sum_{j=1}^{d} m_j(X_i^{(j)}) + \varepsilon_i = m(X_i) + \varepsilon_i,$$

where the error term $\varepsilon_i$ is independent of the $X_i^{(j)}$, $\mathsf{E}[\varepsilon_i|X_i] = 0$, $\mathsf{VAR}[\epsilon_i|X_i] = \sigma^2 < \infty$ and the $m_j$ is a smooth function of a regressor $X_i^{(j)}$. We consider the following model class

$$\mathcal{F}^{\star}_{n,\Psi} = \left\{ f : f(X) = \sum_{j=1}^{d} w_j^T \varphi_j(X^{(j)}) + b, \varphi_j : \mathbb{R} \to \Psi, w \in \mathbb{R}^{n_f}, b \in \mathbb{R} \right\}.$$

The optimization problem (2) can be rewritten w.r.t. the new model class as follows (Pelckmans *et al.* 2005)

$$\begin{cases} \min\limits_{w,b,e} \mathcal{J}_P(w,e) = \frac{1}{2}\sum_{j=1}^{d} w_j^T w_j + \frac{\gamma}{2}\sum\limits_{i=1}^{n} e_i^2 \\ \qquad \text{s.t.} \qquad Y_i = \sum_{j=1}^{d} w_j^T \varphi_j(X_i^{(j)}) + b + e_i, \quad i = 1, \ldots, n, \end{cases}$$

As before, by using Lagrange multipliers, the solution is given by

$$\left[\begin{array}{c|c} 0 & 1_n^T \\ \hline 1_n & \Omega^{\star} + \frac{1}{\gamma} I_n \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline Y \end{array}\right],$$

where $\Omega^\star = \sum_{j=1}^{d} \Omega^{(j)}$ and $\Omega_{kl}^{(j)} = K^{(j)}(X_k^{(j)}, X_l^{(j)})$ for all $k, l = 1, \ldots, n$ (sum of univariate kernels). The resulting additive LS-SVM is given by

$$\hat{m}(x) = \sum_{k=1}^{n} \hat{\alpha}_k \sum_{j=1}^{d} K^{(j)}(x^{(j)}, X_k^{(j)}) + \hat{b}.$$

We illustrate the additive LS-SVMs on two examples. First, we construct a classical example as in Hastie and Tibshirani (1990). The data are generated from the nonlinear regression model with 12 variables:

$$Y_i = 10\sin(\pi X_i^{(1)}) + 20(X_i^{(2)} - 0.5)^2 + 10X_i^{(3)} - 5X_i^{(4)} + 0\sum_{j=5}^{12} X_i^{(j)} + \varepsilon_i,$$

where the 12 variables are uniformly distributed on the interval $[0, 1]$, $\varepsilon \sim \mathcal{N}(0, 1)$ and $n = 300$. By using the option `'multiple'` **StatLSSVM** tunes the bandwidth of the kernel for each estimated function. By setting this option to `'single'` one bandwidth is found for all estimated functions. The relevant **Matlab** command is

```
>> X = rand(300,12);
>> Y = 10*sin(pi*X(:,1))+20*(X(:,2)-0.5).^2+10*X(:,3)-5*X(:,4)+randn(300,1);
>> model = initlssvm(X,Y,[],[],'gaussadd_kernel','multiple');
>> model = tunelssvm(model,'crossval',{10,'mae'});
>> model = trainlssvm(model);
>> model = plotlssvmadd(model);
```

Figure 7 shows the fitted function for the additive LS-SVM model applied to our simulated data data set. In general, the scales on the vertical axes are only meaningful in a relative sense; they have no absolute interpretation. Since we have the freedom to choose the vertical positionings, we should try to make them meaningful in the absolute sense. A reasonable solution, is to plot, for each predictor, the profile of the response surface with each of the other predictors set at their average (see also Ruppert *et al.* (2003)). This is automatically done by the `plotlssvmadd` command.

As a last example we consider the diabetes data set also discussed in Hastie and Tibshirani (1990). The data come from a study (Sockett *et al.* 1987) of the factors affecting patterns of insulin-dependent diabetes mellitus in children. The objective is to investigate the dependence of the level of serum C-peptide on various other factors in order to understand the patterns of residual insulin secretion. The response measurement is the logarithm of C-peptide concentration (pmol/ml) at diagnosis, and the predictor measurements are age and base deficit (a measure of acidity). The **Matlab** command is as follows

```
>> load('diabetes.mat','age','basedef')
>> model = initlssvm([age basedef],Cpeptide,[],[],'gaussadd_kernel','multiple');
>> model = tunelssvm(model,'crossval',{6,'mae'});
>> model = plotlssvmadd(model,{'age','base deficit'});
```

The result is shown in Figure 8 using the vertical alignment procedure discussed above. It can be seen that both effects appear to be nonlinear. The variable `age` has an increasing effect that levels off and the variable `basedef` appears quadratic.

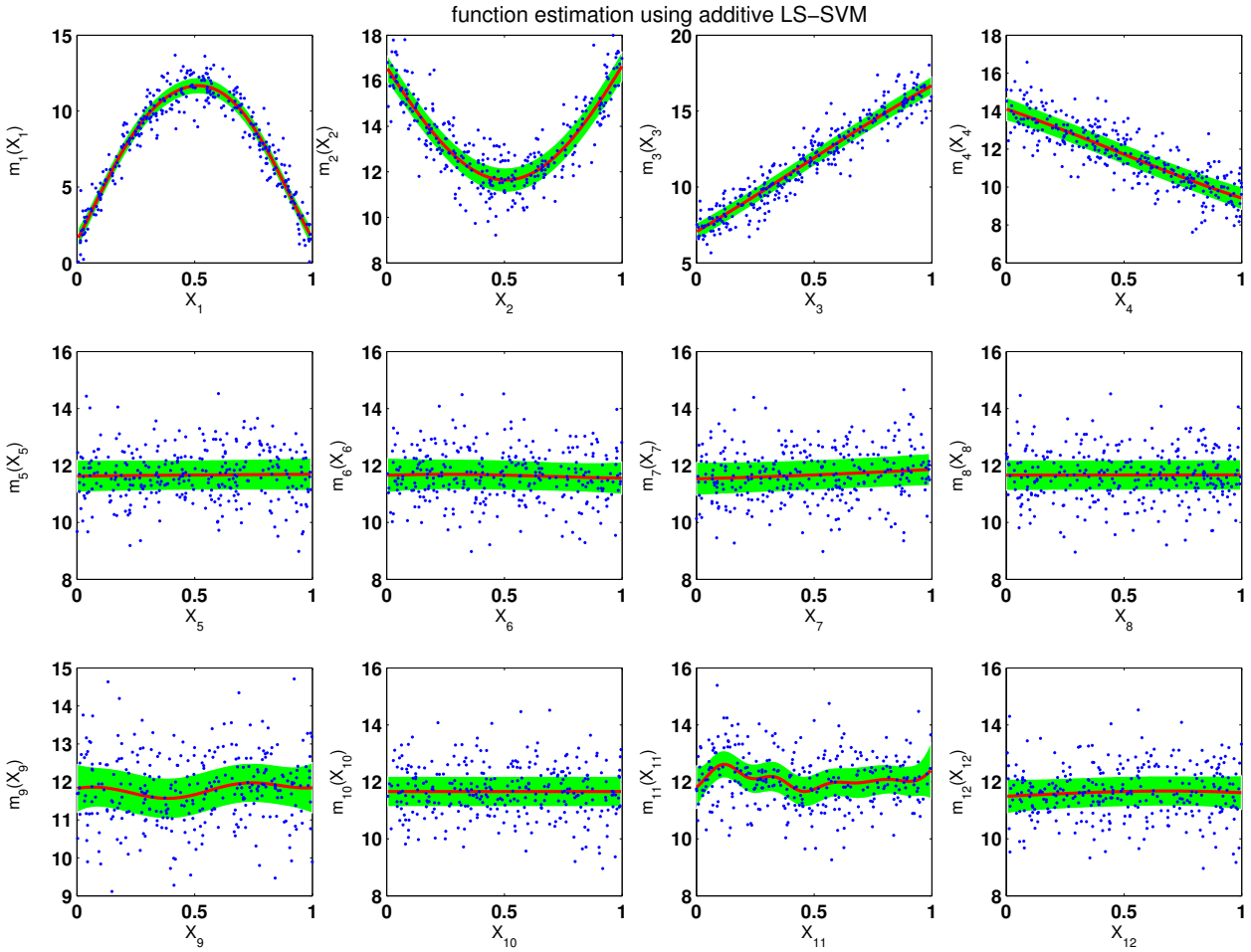



Figure 7: Fitted functions for the additive LS-SVM model for the constructed toy data. The green shaded area represents twice the pointwise standard errors of the estimated curve. The points plotted are the partial residuals: The fitted values for each function plus the overall residuals from the additive model.

## 8. Regression with correlated errors

In this section we consider the nonparametric regression model

$$Y_i = m(x_i) + \varepsilon_i, \quad i = 1, \ldots, n,$$

where $\mathsf{E}[\varepsilon] = 0$, $\mathsf{VAR}[\varepsilon] = \sigma^2 < \infty$, the error term $\varepsilon_i$ is a covariance stationary process and $m$ is a smooth function. However, the presence of correlation between the errors, if ignored, causes breakdown of commonly used automatic tuning parameter selection methods such as cross-validation or plug-in (Opsomer *et al.* 2001; De Brabanter *et al.* 2011b). Data-driven bandwidth selectors tend to be "fooled" by autocorrelation, interpreting it as reflecting the regression relationship and variance function. So, the cyclical pattern in positively correlated errors is viewed as a high frequency regression relationship with small variance, and the bandwidth is set small enough to track the cycles resulting in an undersmoothed fitted regression curve. The

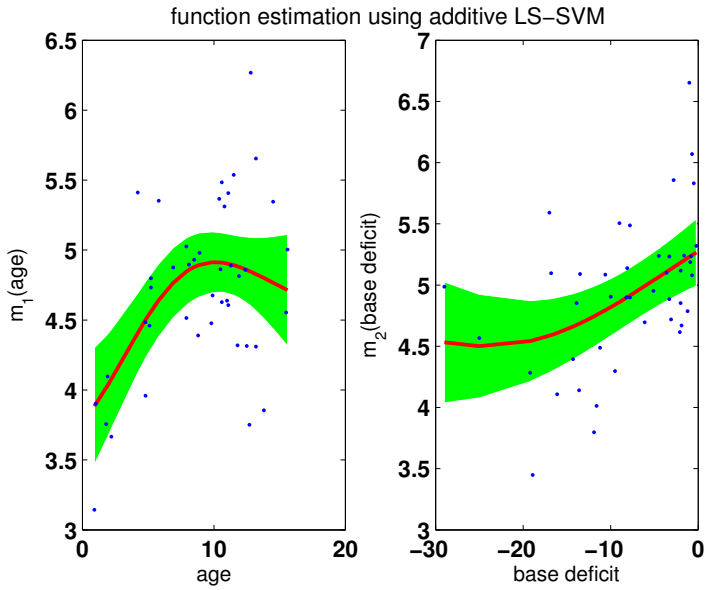



Figure 8: Fitted functions for the additive LS-SVM model for the diabetes data. The green shaded area represent twice the pointwise standard errors of the estimated curve. The points plotted are the partial residuals.

alternating pattern above and below the true underlying function for negatively correlated errors is interpreted as a high variance, and the bandwidth is set high enough to smooth over the variability, producing an oversmoothed fitted regression curve.

The model selection method is based on leave-$(2l+1)$-out cross-validation (Chu and Marron 1991). To tune the parameter $l$, a two-step procedure is used. First, a Nadaraya-Watson smoother with a bimodal kernel is used to fit the data. De Brabanter *et al.* (2011b) have shown that a bimodal kernel satisfying $K(0) = 0$ automatically removes correlation structure without requiring any prior knowledge about its structure. Hence, the obtained residuals are good estimates of the errors. Second, the $k$-th lag sample autocorrelation can be used to find a suitable value for $l$ . More theoretical background about this method can be found in De Brabanter *et al.* (2011b).

Consider the beluga and U.S. birth rate data sets (Simonoff 1996). We will compare the leave-$(2l+1)$-out cross-validation method with classical leave-one-out cross-validation (see Figure 9). It is clear from both results that existence of autocorrelation can seriously affect the regression fit. Ignoring the effects of correlation will cause the nonparametric regression smoother to interpolate the data. This is especially visible in the U.S. birth rate data set. Without removing autocorrelation there is no clear trend visible in the regression fit. By using the above described method for model selection the regression fit shows a clear pattern i.e., the U.S. joined the second world war after the attack on Pearl Harbor (December 1941), decreasing birth rate during the entire course of the war and finally increasing birth rate after the war in Europe and the Pacific was over (mid September 1945).

The relevant **Matlab** command for the beluga data set is

```
>> load('beluga.mat','period','nursingtime')
```

```
>> model = initlssvm(period,nursingtime,[],[],'gauss_kernel');
>> % First, model selection accounting for correlation
>> model = tunelssvm(model,'crossval2lp1',{'mse'});
>> plotlssvm(model); hold on;
>> % second classical leave-one-out CV
>> model = tunelssvm(model,'leaveoneout',{'mse'});
>> model = trainlssvm(model);
>> Yhat = simlssvm(model,period);
>> plot(period,Yhat,'g-')
>> xlabel('Period'); ylabel('Nursing time (s)')
```

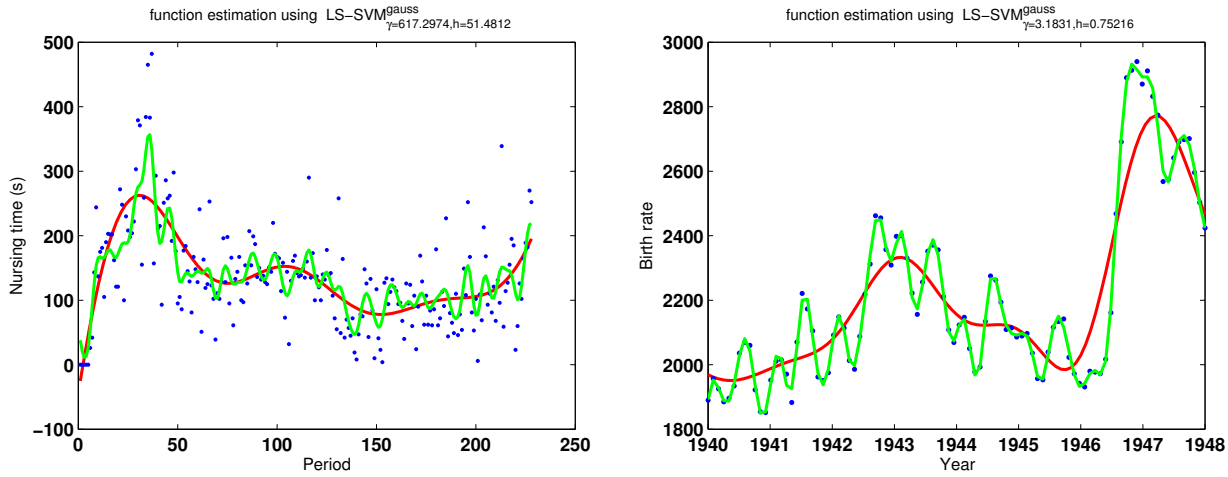



Figure 9: LS-SVM regression estimates for the nursing time of the beluga whale (left panel) and U.S. birth rate data set (right panel). The green line represents the estimate with tuning parameters determined by classical leave-one-out cross-validation and the red line is the estimate based on the above described procedure.

## 9. Conclusions

We have demonstrated that several nonparametric regression problems can be handled with **StatLSSVM**. This Matlab based toolbox can manage standard nonparametric regression, regression with autocorrelated errors, robust regression, pointwise/uniform confidence intervals and additive models with a few simple lines of code. Currently the toolbox is supported for Matlab R2009b or higher.

## Acknowledgements

Kris De Brabanter is a postdoctoral researcher supported by an FWO fellowship grant. JS is a professor at the Katholieke Universiteit Leuven. BDM is a full professor at the Katholieke Universiteit Leuven. Research supported by Research Council KUL: GOA/10/09 MaNet , PFV/10/002 (OPTEC), several PhD/postdoc & fellow grants; Flemish Government:

IOF: IOF/KP/SCORES4CHEM, FWO: PhD/postdoc grants, projects: G.0320.08 (convex MPC), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine), G.0377.09 (Mechatronics MPC); G.0377.12 (Structured systems) research community (WOG: MLDM), IWT: PhD Grants, projects: Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, EUROSTARS SMART, iMinds 2012, Belgian Federal Science Policy Office: IUAP P7/ (DYSCO, Dynamical systems, control and optimization, 2012-2017) EU: ERNSI, FP7-EMBOCON (ICT-248940), FP7-SADCO ( MC ITN-264735), ERC ST HIGHWIND (259 166), ERC AdG A-DATADRIVE-B, COST: Action ICO806: IntelliCIS. The scientific responsibility is assumed by its authors.

## References

Akaike H (1973). "Statistical Predictor Identification." *Annals of The Institute of Statistical Mathematics*, **22**(1), 203–217.

Bennett KP, Hu J, Xiaoyun J, Kunapuli G, Pang JS (2006). "Model Selection via Bilevel Optimization." *International Joint Conference on Neural Networks*, pp. 1922–1929.

Brezger A, Kneib T, Lang S (2005). "**BayesX**: Analyzing Bayesian Structured Additive Regression Models." *Journal of Statistical Software*, **14**(11), 1–22. URL: http://www.jstatsoft.org/v14/i11/.

Brezger A, Lang S (2006). "Generalized Structured Additive Regression Based on Bayesian P-Splines." *Computational Statistics and Data Analysis*, **50**(4), 967–991.

Burman P (1989). "A Comparative Study of Ordinary Cross-Validation, $v$-Fold Crossvalidation and the Repeated Learning-Testing Methods." *Biometrika*, **76**(3), 503–514.

Carroll RJ, Ruppert D, Stefanski LA, Crainiceanu CM (2006). *Measurement Error in Nonlinear Models*. Chapman & Hall/CRC, Boca Raton, USA.

Christmann A, Steinwart I (2007). "Consistency and Robustness of Kernel-Based Regression in Convex Risk Minimization." *Bernoulli*, **13**(3), 799–819.

Chu CK, Marron JS (1991). "Comparison of Two Bandwidth Selectors with Dependent Errors." *The Annals of Statistics*, **19**(4), 1906–1918.

Courant R, Hilbert D (1953). *Methods of Mathematical Physics*. Interscience Publishers.

Craven P, Wahba G (1979). "Smoothing Noisy Data with Spline Functions." *Numerische Mathematik*, **31**(4), 377–403.

De Brabanter K (2011). *Least Squares Support Vector Regression with Applications to Large-Scale Data: a Statistical Approach*. Ph.D. thesis, K.U.Leuven, Leuven, Belgium. URL ftp://ftp.esat.kuleuven.ac.be/pub/SISTA//kdebaban/11-82.pdf.

De Brabanter K, De Brabanter J, Suykens JAK, De Moor B (2010). "Optimized Fixed-Size Kernel Models for Large Data Sets." *Computational Statistics and Data Analysis*, **54**(6), 1484–1504.

De Brabanter K, De Brabanter J, Suykens JAK, De Moor B (2011a). "Approximate Confidence and Prediction Intervals for Least Squares Support Vector Regression." *IEEE Transactions on Neural Networks*, **22**(1), 110–120.

De Brabanter K, De Brabanter J, Suykens JAK, De Moor B (2011b). "Kernel Regression in the Presence of Correlated Errors." *Journal of Machine Learning Research*, **12**, 1955–1976.

De Brabanter K, Pelckmans K, De Brabanter J, Debruyne M, Suykens JAK, Hubert M, De Moor B (2009). "Robustness of Kernel Based Regression: a Comparison of Iterative Weighting Schemes." *in Proc. of the 19th International Conference on Artificial Neural Networks (ICANN)*, pp. 100–110.

Debruyne M, Christmann A, Hubert M, Suykens JAK (2010). "Robustness of Reweighted Least Squares Kernel Based Regression." *Journal of Multivariate Analysis*, **101**(2), 447–463.

Fan J, Gijbels I (1996). *Local Polynomial Modelling and Its Applications*. John Wiley & Sons.

Ferraty F, Vieu P (2006). *Nonparametric Functional Data Analysis: Theory and Practice*. Springer-Verlag.

Györfi L, Kohler M, Krzyżak A, Walk H (2002). *A Distribution-Free Theory of Nonparametric Regression*. Springer-Verlag.

Hall P (1992). "On Bootstrap Confidence Intervals in Nonparametric Regression." *The Annals of Statistics*, **20**(2), 695–711.

Hall P, Lahiri S, Polzehl J (1995). "On Bandwidth Choice in Nonparametric Regression with Both Short- and Long-Range Dependent Errors." *The Annals of Statistics*, **23**(6), 1921–1936.

Hall P, Marron JS (1991). "Local Minima in Cross-Validation Functions." *Journal of the Royal Statistical Society B*, **53**(1), 245–252.

Hampel FR, Ronchetti E, Rousseeuw PJ, Stahel WA (1986). *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons, New York.

Hart JD (1991). "Kernel Regression Estimation with Time Series Errors." *Journal of the Royal Statistical Society B*, **53**(1), 173–187.

Hastie T, Tibshirani R (1990). *Generalized Additive Models*. Chapman & Hall, London.

Hastie T, Tibshirani R, Friedman J (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd edition. Springer-Verlag.

Holst U, Hössjer O, Björklund C, Ragnarson P, Edner H (1996). "Locally Weighted Least Squares Kernel Regression and Statistical Evaluation of Lidar Measurements." *Environmetrics*, **7**(4), 401–416.

Huber PJ (1964). "Robust Estimation of a Location Parameter." *The Annals of Mathematical Statistics*, **35**(1), 73–101.

Jones MC, Foster PJ (1993). "Generalized Jackknifing and Higher Order Kernels." *Journal of Nonparametric Statistics*, **3**(11), 81–94.

Jurečková J, Picek J (2006). *Robust Statistical Methods with R*. Chapman & Hall (Taylor & Francis Group).

Karatzoglou A, Smola A, Hornik K, Zeileis A (2004). "kernlab – An S4 Package for Kernel Methods in R." *Journal of Statistical Software*, **11**(9), 1–20. URL http://www.jstatsoft.org/v11/i09/.

Kunapuli G, Bennett KP, Hu J, Pang JS (2008). *in P.M. Pardalos and P. Hansen (Eds.), Data Mining and Mathematical Programming*, chapter Bilevel Model Selection for Support Vector Machines, pp. 129–158. CRM Proceedings & Lecture Notes Volume 45. American Mathematical Society.

Lagarias JC, Reeds JA, Wright MH, Wright PE (1998). "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions." *SIAM Journal of Optimization*, **9**(1), 112–147.

Leung D (2005). "Cross-Validation in Nonparametric Regression with Outliers." *The Annals of Statistics*, **33**(5), 2291–2310.

Marley JK, Wand MP (2010). "Non-Standard Semiparametric Regression via BRugs." *Journal of Statistical Software*, **37**(5), 1–30. URL: http://www.jstatsoft.org/v37/i05/.

Maronna R, Martin D, Yohai V (2006). *Robust Statistics: Theory and Methods*. John Wiley & Sons.

Meister A (2009). *Deconvolution Problems in Nonparametric Statistics*. Springer-Verlag.

Mercer J (1909). "Functions of Positive and Negative Type and Their Connection With the Theory of Integral Equations." *Philosophical Transactions of the Royal Society A*, **209**, 415–446.

Nelder JA, Mead R (1965). "A Simplex Method for Function Minimization." *The Computer Journal*, **7**(4), 308–313.

Opsomer J, Wang Y, Yang Y (2001). "Nonparametric Regression with Correlated Errors." *Statistical Science*, **16**(2), 134–153.

Pelckmans K, Goethals I, De Brabanter J, Suykens JAK, De Moor B (2005). *Support Vector Machines: Theory and Applications*, chapter Componentwise Least Squares Support Vector Machines, pp. 77–98. Springer-Verlag.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rousseeuw PJ, Leroy AM (2003). *Robust Regression and Outlier Detection*. John Wiley & Sons.

Ruppert D, Wand MP, Carroll RJ (2003). *Semiparametric Regression*. Cambridge University Press.

Simonoff JS (1996). *Smoothing Methods in Statistics*. Springer-Verlag, New York.

Sockett EB, Daneman D, Clarson C, Ehrich RM (1987). "Factors Affecting and Patterns of Residual Insulin Secretion During the First Year of Type I (Insulin Dependent) Diabetes Mellitus in Children." *Diabet*, **30**, 453–459.

Steinwart I, Christmann A (2008). *Support Vector Machines*. Springer.

Steinwart I, Christmann A (2011). "Estimating Conditional Quantiles With the Help of the Pinball Loss." *Bernoulli*, **17**(1), 211–225.

Sun J, Loader C (1994). "Simultaneous Confidence Bands for Linear Regression and Smoothing." *Annals of Statistics*, **22**(3), 1328–1345.

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

Vapnik VN (1999). *Statistical Learning Theory*. John Wiley & Sons.

Xavier-de-Souza S, Suykens JAK, Vandewalle J, Bollé D (2010). "Coupled Simulated Annealing." *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **40**(2), 320–335.

## Affiliation:

Kris De Brabanter
Department of Electrical Engineering (ESAT) - Research Division SCD
Katholieke Universiteit Leuven
Kasteelpark Arenberg 10
B-3001 Leuven, Belgium
Telephone: +32/16/32 86 64
E-mail: kris.debrabanter@esat.kuleuven.be

Johan A.K. Suykens
Department of Electrical Engineering (ESAT) - Research Division SCD
Katholieke Universiteit Leuven
Kasteelpark Arenberg 10
B-3001 Leuven, Belgium
Telephone: +32/16/32 18 02
E-mail: johan.suykens@esat.kuleuven.be

Bart De Moor
Department of Electrical Engineering (ESAT) - Research Division SCD
Katholieke Universiteit Leuven
Kasteelpark Arenberg 10
B-3001 Leuven, Belgium
Telephone: +32/16/32 17 15
E-mail: bart.demoor@esat.kuleuven.be
URL: http://homes.esat.kuleuven.be/~demoor/

*Journal of Statistical Software* http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd