Journal of Machine Learning Research 9 (2008) 2377-2400 Submitted 8/07; Revised 6/08; Published 10/08

Model Selection in Kernel Based Regression using the Influence Function

Michiel Debruyne MICHIEL.DEBRUYNE@UA.AC.BE
Mia Hubert MIA.HUBERT@WIS.KULEUVEN.BE
Department of Mathematics - LStat
K.U.Leuven
Celestijnenlaan 200B, B-3001 Leuven, Belgium

Johan A.K. Suykens JOHAN.SUYKENS@ESAT.KULEUVEN.BE
ESAT-SCD/SISTA
K.U.Leuven
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

Editor: Isabelle Guyon

Abstract

Recent results about the robustness of kernel methods involve the analysis of influence functions. By definition the influence function is closely related to leave-one-out criteria. In statistical learning, the latter is often used to assess the generalization of a method. In statistics, the influence function is used in a similar way to analyze the statistical efficiency of a method. Links between both worlds are explored. The influence function is related to the first term of a Taylor expansion. Higher order influence functions are calculated. A recursive relation between these terms is found characterizing the full Taylor expansion. It is shown how to evaluate influence functions at a specific sample distribution to obtain an approximation of the leave-one-out error. A specific implementation is proposed using a L_1 loss in the selection of the hyperparameters and a Huber loss in the estimation procedure. The parameter in the Huber loss controlling the degree of robustness is optimized as well. The resulting procedure gives good results, even when outliers are present in the data.

Keywords: kernel based regression, robustness, stability, influence function, model selection

1. Introduction

Quantifying the effect of small distributional changes on the resulting estimator is a crucial analysis on many levels. A simple example is leave-one-out which changes the sample distribution slightly by deleting one observation. This leave-one-out error plays a vital role for example in model selection (Wahba, 1990) and in assessing the generalization ability (Poggio et al. 2004 through the concept of stability). Most of these analyses however are restricted to the sample distribution and the addition/deletion of some data points from this sample.

In the field of robust statistics the influence function was introduced in order to analyze the effects of outliers on an estimator. This influence function is defined for continuous distributions that are slightly perturbed by adding a small amount of probability mass at a certain place. In Section 2 some general aspects about the influence function are gathered. Recent results about influence functions in kernel methods include those of Christmann and Steinwart (2004, 2007) for classifica-

tion and regression. In Section 3 these results are stated and their importance is summarized. A new theoretical result concerning higher order influence functions is presented. In Section 4 we show how to evaluate the resulting expressions at sample distributions. Moreover we apply these influence functions in a Taylor expansion approximating the leave-one-out error. In Section 5 we use the approximation with influence functions to select the hyperparameters. A specific implementation is proposed to obtain robustness with a Huber loss function in the estimation step and a L_1 loss in the model selection step. The degree of robustness is controlled by a parameter that can be chosen in a data driven way as well. Everything is illustrated on a toy example and some experiments in Section 6.

2. The Influence Function

In statistics it is often assumed that a sample of data points is observed, all generated independently from the same distribution and some underlying process, but sometimes this is not sufficient. In many applications gathering the observations is quite complex, and many errors or subtle changes can occur when obtaining data. Robust statistics is a branch of statistics that deals with the detection and neutralization of such outlying observations. Roughly speaking a method is called robust if it produces similar results as the majority of observations indicates, no matter how a minority of other observations is placed. A crucial analysis in robust statistics is the behavior of a functional T, not only at the distribution of interest P, but in an entire neighborhood of distributions around P. The influence function measures this behavior. In this section we recall its definition and discuss some links with other concepts.

2.1 Definition

The pioneering work of Hampel et al. (1986) and Huber (1981) considers distributions $P_{\varepsilon,z} = (1-\varepsilon)P + \varepsilon\Delta_z$ where Δ_z denotes the Dirac distribution in the point $z \in \mathcal{X} \times \mathcal{Y}$, representing the contaminated part of the data. For having a robust T, $T(P_{\varepsilon,z})$ should not be too far away from $T(P)$ for any possible z and any small ε. The limiting case of $\varepsilon \downarrow 0$ is comprised in the concept of the influence function.

Definition 1 *Let P be a distribution. Let T be a functional $T : P \to T(P)$. Then the influence function of T at P in the point z is defined as*

$$IF(z;T,P) = \lim_{\varepsilon \to 0} \frac{T(P_{\varepsilon,z}) - T(P)}{\varepsilon}.$$

The influence function measures the effect on the estimator T when adding an infinitesimally small amount of contamination at the point z. Therefore it is a measure of the robustness of T. Of particular importance is the supremum over z. If this is unbounded, then an infinitesimally small amount of contamination can cause arbitrary large changes. For robust estimators, the supremum of its influence function should be bounded. Then small amounts of contamination cannot completely change the estimate and a certain degree of robustness is indeed present. The simplest example is the estimation of the location of a univariate distribution with density f symmetric around 0. The influence function of the mean at $z \in \mathbb{R}$ then equals the function z and is clearly unbounded. If the median of the underlying distribution is uniquely defined, that is if $f(0) > 0$, then the influence function of the median equals $\text{sign}(z)/(2f(0))$ which is bounded. The median is thus more robust than the mean.

2.2 Asymptotic Variance and Stability

From Definition 1 one can see that the influence function is a first order derivative of $T(P_{\varepsilon,z})$ at $\varepsilon = 0$. Higher order influence functions can be defined too:

Definition 2 *Let P be a distribution. Let T be a functional $T : P \to T(P)$. Then the k-th order influence function of T at P in the point z is defined as*

$$IF_k(z;T,P) = \frac{\partial}{\partial^k \varepsilon} T(P_{\varepsilon,z})|_{\varepsilon=0}.$$

If all influence functions exist then the following Taylor expansion holds:

$$T(P_{\varepsilon,z}) = T(P) + \varepsilon IF(z;T,P) + \frac{\varepsilon^2}{2!} IF_2(z;T,P) + \ldots \tag{1}$$

characterizing the estimate at a contaminated distribution in terms of the estimate at the original distribution and the influence functions.

Actually this is a special case of a more general Von Mises expansion (take $Q = P_{\varepsilon,z}$):

$$T(Q) = T(P) + \int IF(x;T,P)d(Q-P)(x) + \ldots$$

Now take Q equal to a sample distribution P_n of a sample $\{z_i\}$ of size n generated i.i.d. from P. Then

$$\begin{aligned} T(P_n) - T(P) &= \int IF(z;T,P)dP_n(z) + \ldots \\ &= \frac{1}{n}\sum_{i=1}^{n} IF(z_i;T,P) + \ldots. \end{aligned}$$

The first term on the right hand side is now a sum of n i.i.d. random variables. If the remaining terms are asymptotically negligible, the central limit theorem thus immediately shows that $\sqrt{n}(T(P_n) - T(P))$ is asymptotically normal with mean 0 and variance

$$ASV(T,P) = \int IF^2(z;T,P)dP(z).$$

Since the asymptotic efficiency of an estimator is proportional to the reciprocal of the asymptotic variance, the integrated squared influence function should be as small as possible to achieve high efficiency. Consider again the estimation of the center of a univariate distribution with density f. At a standard normal distribution the asymptotic variance of the mean equals $\int z^2 dP(z) = 1$, and that of the median equals $\int (\text{sign}(z)/(2f(0)))^2 dP(z) = 1.571$. Thus the mean is more efficient than the median at a normal distribution. However, at a Cauchy distribution for instance, this is completely different: the ASV of the median equals 2.47, but for the mean it is infinite since the second moment of a Cauchy distribution does not exist. Thus to estimate the center of a Cauchy, the median is a much better choice than the mean.

An interesting parallel can be drawn towards the concept of stability in learning theory. Several measures of stability were recently proposed in the literature. The leave-one-out error often plays a vital role, for example in hypothesis stability (Bousquet and Elisseeff, 2001), partial stability (Kutin

and Niyogi, 2002) and CV_{loo}-stability (Poggio et al., 2004). The basic idea is that the result of a learning map T on a full sample should not be very different from the result obtained when removing only one observation. More precisely, let P be a distribution on a set $\mathcal{X} \times \mathcal{Y}$ and $T : P \to T(P)$ with $T(P) : \mathcal{X} \to \mathcal{Y} : x \to T(P)(x)$. Let P_n^{-i} denote the empirical distribution of a sample without the ith observation $z_i = (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$. Poggio et al. (2004) call the map T CV_{loo}-stable for a loss function $L : \mathcal{Y} \to \mathbb{R}^+$ if

$$\lim_{n\to\infty} \sup_{i\in\{1,\ldots,n\}} |L(y_i - T(P_n)(x_i)) - L(y_i - T(P_n^{-i})(x_i))| \to 0 \tag{2}$$

for $n \to \infty$. This means intuitively that the prediction at a point x_i should not be too different whether or not this point is actually used constructing the predictor. If the difference is too large there is no stability, since in that case adding only one point can yield a large change in the result. Under mild conditions it is shown that CV_{loo}-stability is required to achieve good predictions. Let L be the absolute value loss and consider once again the simple case of estimating the location of a univariate distribution. Thus P_n is just a univariate sample of n real numbers $\{y_1, \ldots, y_n\}$. Then the left hand side of (2) equals

$$\lim_{n\to\infty} \sup_{i\in\{1,\ldots,n\}} |T(P_n) - T(P_n^{-i})|.$$

Let $y_{(i)}$ denote the ith order statistic. Consider T the median. Assuming that n is odd and $y_i < y_{(\frac{n+1}{2})}$ (the cases $y_i > y_{(\frac{n+1}{2})}$ and equality can easily be checked as well), we have that

$$|\mathrm{Med}(P_n) - \mathrm{Med}(P_n^{-i})| = \left| y_{(\frac{n+1}{2})} - \frac{1}{2}\left(y_{(\frac{n+1}{2})} + y_{(\frac{n+3}{2})} \right) \right| = \frac{1}{2} |y_{(\frac{n+1}{2})} - y_{(\frac{n+3}{2})}|.$$

If the median of the underlying distribution P is unique, then both $y_{(\frac{n+1}{2})}$ and $y_{(\frac{n+3}{2})}$ converge to this number and CV_{loo} stability is obtained. However, when taking the mean for T, we have that

$$|\mathbb{E}(P_n) - \mathbb{E}(P_n^{-i})| = \left| \frac{1}{n}\sum_{j=1}^{n} y_j - \frac{1}{n-1}\sum_{\substack{j=1 \\ j\neq i}}^{n} y_j \right| = \left| -\frac{1}{n(n-1)}\sum_{\substack{j=1 \\ j\neq i}}^{n} y_j + \frac{y_i}{n} \right|.$$

The first term in this sum equals the sample mean of P_n^{-i} divided by n and thus converges to 0 if the mean of the underlying distribution exists. The second term converges to 0 if

$$\lim_{n\to\infty} \sup_{i\in\{1,\ldots,n\}} \frac{|y_i|}{n} = 0.$$

This means that the largest absolute value of n points sampled from the underlying distribution should not grow too large. For a normal distribution for instance this is satisfied since the largest observation only grows logarithmically: for example the largest of 1000 points generated from a normal distribution only has a very small probability to exceed 5. This is due to the exponentially decreasing density function. For heavy tailed distribution it can be different. A Cauchy density for instance only decreases at the rate of the reciprocal function and $\sup_{i\in\{1,\ldots,n\}} |y_i|$ is of the order $O(n)$. Thus for a normal distribution the mean is CV_{loo} stable, but for a Cauchy distribution it is not.

In summary note that both the concepts of influence function and asymptotic variance on one hand and CV_{loo} stability on the other hand yield the same conclusions: using the sample median as

an estimator is ok as long as the median of the underlying distribution is unique. Then one has CV_{loo} stability and a finite asymptotic variance. Using the sample mean is ok for a normal distribution, but not for a Cauchy distribution (no CV_{loo} stability and an infinite asymptotic variance).

A rigorous treatment of asymptotic variances and regularity conditions can be found in Boos and Serfling (1980) and Fernholz (1983). In any event, it is an interesting link between perturbation analysis through the influence function and variance/efficiency in statistics on one hand, and between leave-one-out and stability/generalization in learning theory on the other hand.

2.3 A Strategy for Fast Approximation of the Leave-one-out Error

In leave-one-out crossvalidation $T(P_n^{-i})$ is computed for every i. This means that the algorithm under consideration has to be executed n times, which can be computationally intensive. If the influence functions of T can be calculated, the following strategy might provide a fast alternative. First note that

$$P_n^{-i} = (1-(\frac{-1}{n-1}))P_n + \frac{-1}{n-1}\Delta_{z_i}.$$

Thus, taking $P_{\varepsilon,z} = P_n^{-i}$, $\varepsilon = -1/(n-1)$ and $P = P_n$, Equation (1) gives

$$T(P_n^{-i}) = T(P_n) + \sum_{j=1}^{\infty}(\frac{-1}{n-1})^j \frac{IF_j(z_i;T,P_n)}{j!}. \tag{3}$$

The right hand side now only depends on the full sample P_n. In practice one can cut off the series after a number of steps ignoring the remainder term, or if possible one can try to estimate the remainder term.

The first goal of this paper is to apply this idea in the context of kernel based regression. Christmann and Steinwart (2007) computed the first order influence function. We will compute higher order terms in (1) and use these results to approximate the leave-one-out estimator applying (3).

3. Kernel Based Regression

In this section we recall some definitions on kernel based regression. We discuss the influence function and provide a theorem on higher order terms.

3.1 Definition

Let $\mathcal{X}, \mathcal{Y}$ be non-empty sets. Denote P a distribution on $\mathcal{X}\times\mathcal{Y} \subseteq \mathbb{R}^d \times \mathbb{R}$. Suppose we have a sample of n observations $(x_i, y_i) \in \mathcal{X}\times\mathcal{Y}$ generated i.i.d. from P. Then P_n denotes the corresponding finite sample distribution. A functional T is a map that maps any distribution P onto $T(P)$. A finite sample approximation is given by $T_n := T(P_n)$.

Definition 3 *A function $K : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is called a kernel on $\mathcal{X}$ if there exists a $\mathbb{R}$-Hilbert space $\mathcal{H}$ and a map $\Phi : \mathcal{X} \to \mathcal{H}$ such that for all $x, x' \in \mathcal{X}$ we have*

$$K(x,x') \;=\; \langle \Phi(x), \Phi(x') \rangle .$$

We call Φ a feature map *and $\mathcal{H}$ a* feature space *of K.*

Frequently used kernels include the linear kernel $K(x_i,x_j) = x_i^t x_j$, polynomial kernel of degree p for which $K(x_i,x_j) = (\tau + x_i^t x_j)^p$ with $\tau > 0$ and RBF kernel $K(x_i,x_j) = \exp(-\|x_i - x_j\|_2^2/\sigma^2)$ with bandwidth $\sigma > 0$. By the reproducing property of $\mathcal{H}$ we can evaluate any $f \in \mathcal{H}$ at the point $x \in \mathcal{X}$ as the inner product of f with the feature map: $f(x) = \langle f, \Phi(x) \rangle$.

Definition 4 *Let K be a kernel function with corresponding feature space $\mathcal{H}$ and let $L : \mathbb{R} \to \mathbb{R}^+$ be a twice differentiable convex loss function. Then the functional $f_{\lambda,K} : P \to f_{\lambda,K}(P) = f_{\lambda,K,P} \in \mathcal{H}$ is defined by*

$$f_{\lambda,K,P} := \underset{f \in \mathcal{H}}{argmin} \mathbb{E}_P L(Y - f(X)) + \lambda \| f \|_{\mathcal{H}}^2$$

where $\lambda > 0$ is a regularization parameter.

The functional $f_{\lambda,K}$ maps a distribution P onto the function $f_{\lambda,K,P}$ that minimizes the regularized risk. When the sample distribution P_n is used, one has that

$$f_{\lambda,K,P_n} := \underset{f \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(y_i - f(x_i)) + \lambda \| f \|_{\mathcal{H}}^2. \tag{4}$$

Such estimators have been studied in detail, see for example Wahba (1990), Tikhonov and Arsenin (1977) or Evgeniou et al. (2000). In a broader framework (including for example classification, PCA, CCA etc.) primal-dual optimization methodology involving least squares kernel estimators were studied by Suykens et al. (2002b). Possible loss functions include

- the least squares loss: $L(r) = r^2$.
- Vapnik's ε-insensitive loss: $L(r) = \max\{|r| - \varepsilon, 0\}$, with special case the L_1 loss if $\varepsilon = 0$.
- the logistic loss: $L(r) = -\log(4\Lambda(r)[1 - \Lambda(r)])$ with $\Lambda(r) = 1/(1 + e^{-r})$. Note that this is not the same loss function as used in logistic regression.
- Huber loss with parameter $b > 0$: $L(r) = r^2$ if $|r| \leq b$ and $L(r) = 2b|r| - b^2$ if $|r| > b$. Note that the least squares loss corresponds to the limit case $b \to \infty$.

3.2 Influence Function

The following proposition was proven in Christmann and Steinwart (2007).

Proposition 5 *Let $\mathcal{H}$ be a RKHS of a bounded continuous kernel K on $\mathcal{X}$ with feature map $\Phi : \mathcal{X} \to \mathcal{H}$. Furthermore, let P be a distribution on $\mathcal{X} \times \mathcal{Y}$ with finite second moment. Then the influence function of $f_{\lambda,K}$ exists for all $z := (z_x, z_y) \in \mathcal{X} \times \mathcal{Y}$ and we have*

$$IF(z; f_{\lambda,K}, P) = -S^{-1}\left(2\lambda f_{\lambda,K,P}\right) + L'(z_y - f_{\lambda,K,P}(z_x)) S^{-1} \Phi(z_x)$$

where $S : \mathcal{H} \to \mathcal{H}$ is defined by $S(f) = 2\lambda f + \mathbb{E}_P\left[L''(Y - f_{\lambda,K,P}(X)) \langle \Phi(X), f \rangle \Phi(X)\right]$.

Thus if the kernel is bounded and the first derivative of the loss function is bounded, then the influence function is bounded as well. Thus L_1 type loss functions for instance lead to robust estimators. The logistic loss as well since the derivative of this loss function equals $L'(r) = 2 -$

$1/(1+e^{-r})$ which is bounded by 2. For the Huber loss $L'(r)$ is bounded by $2b$. This shows that the parameter b controls the amount of robustness: if b is very large than the influence function can become very large too. For a small b the influence function remains small. For a least squares loss function on the other hand, the influence function is unbounded ($L'(r) = 2r$): the effect of the smallest amount of contamination can be arbitrary large. Therefore it is said that the least squares estimator is not robust.

3.3 Higher Order Influence Functions

For the second order influence function as in Definition 2 the following theorem is proven in the Appendix.

Theorem 6 *Let P be a distribution on $\mathcal{X} \times \mathcal{Y}$ with finite second moment. Let L be a convex loss function that is three times differentiable. Then the second order influence function of $f_{\lambda,K}$ exists for all $z := (z_x, z_y) \in \mathcal{X} \times \mathcal{Y}$ and we have*

$$\begin{aligned} IF_2(z; f_{\lambda,K}, P) = & S^{-1}\Big(2\mathbb{E}_P[IF(z; f_{\lambda,K}, P)(X) L''(Y - f_{\lambda,K}(X))\Phi(X)] \\ & + \mathbb{E}_P[(IF(z; f_{\lambda,K}, P)(X))^2 L'''(Y - f_{\lambda,K,P}(X))] \\ & - 2[IF(z; f_{\lambda,K}, P)(z_x) L''(z_y - f_{\lambda,K}(z_x))\Phi(z_x)] \Big) \end{aligned}$$

where $S : \mathcal{H} \to \mathcal{H}$ is defined by $S(f) = 2\lambda f + \mathbb{E}_P\left[L''(Y - f_{\lambda,K,P}(X))\langle \Phi(X), f \rangle \Phi(X)\right]$.

When the loss function is infinitely differentiable, all higher order terms can in theory be calculated, but the number of terms grows rapidly since all derivatives of L come into play. However, in the special case that all derivatives higher than three are 0, a relatively simple recursive relation exists.

Theorem 7 *Let P be a distribution on $\mathcal{X} \times \mathcal{Y}$ with finite second moment. Let L be a convex loss function such that the third derivative is 0. Then the $(k+1)$th order influence function of $f_{\lambda,K}$ exists for all $z := (z_x, z_y) \in \mathcal{X} \times \mathcal{Y}$ and we have*

$$\begin{aligned} IF_{k+1}(z; f_{\lambda,K}, P) = (k+1) S^{-1}\Big(& \mathbb{E}_P[IF_k(z; f_{\lambda,K}, P)(X) L''(Y - f_{\lambda,K}(X))\Phi(X)] \\ & - [IF_k(z; f_{\lambda,K}, P)(z_x) L''(Z_y - f_{\lambda,K}(z_x))\Phi(z_x)] \Big) \end{aligned}$$

where $S : \mathcal{H} \to \mathcal{H}$ is defined by $S(f) = 2\lambda f + \mathbb{E}_P\left[L''(Y - f_{\lambda,K,P}(X))\langle \Phi(X), f \rangle \Phi(X)\right]$.

4. Finite Sample Expressions

Since the Taylor expansion in (1) is now fully characterized for any distribution P and any z, we can use this to assess the influence of individual points in a sample with sample distribution P_n. Applying Equation (3) with the KBR estimator f_{λ,K,P_n} from (4) we have that

$$f_{\lambda,K,P_n^{-i}}(x_i) = f_{\lambda,K,P_n}(x_i) + \sum_{j=1}^{\infty} \left(\frac{-1}{n-1}\right)^j \frac{IF_j(z_i; f_{\lambda,K}, P_n)(x_i)}{j!}. \tag{5}$$

Let us see how the right hand side can be evaluated in practice.

4.1 Least Squares Loss

First consider taking the least squares loss in (4). Denote Ω the $n \times n$ kernel matrix with i,j-th entry equal to $K(x_i,x_j)$. Let I_n be the $n \times n$ identity matrix and denote $S_n = \Omega/n + \lambda I_n$. The value of f_{λ,K,P_n} at a point $x \in \mathcal{X}$ is given by

$$f_{\lambda,K,P_n}(x) = \frac{1}{n}\sum_{i=1}^{n} \alpha_i K(x_i,x) \quad \text{with} \quad \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = S_n^{-1} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{6}$$

which is a classical result going back to Tikhonov and Arsenin (1977). This also means that the vector of predictions in the n sample points simply equals

$$\begin{pmatrix} f_{\lambda,K,P_n}(x_1) \\ \vdots \\ f_{\lambda,K,P_n}(x_n) \end{pmatrix} = H \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{7}$$

with the matrix $H = \frac{1}{n} S_n^{-1}\Omega$, sometimes referred to as the smoother matrix.

To compute the first order influence function at the sample the expression in Proposition 5 should be evaluated at P_n. The operator S at P_n maps by definition any $f \in \mathcal{H}$ onto

$$S_{P_n}(f) = 2\lambda f + \mathbb{E}_{P_n} 2 f(X)\Phi(X) = 2\lambda f + \frac{2}{n}\sum_{j=1}^{n} f(x_j)\Phi(x_j)$$

and thus

$$\begin{aligned} \begin{pmatrix} S_{P_n}(f)(x_1) \\ \vdots \\ S_{P_n}(f)(x_n) \end{pmatrix} &= 2\lambda \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} + \frac{2}{n} \begin{pmatrix} K(x_1,x_1) & \dots & K(x_1,x_n) \\ \vdots & & \\ K(x_n,x_1) & & K(x_n,x_n) \end{pmatrix} \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} \\ &= 2S_n \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} \end{aligned}$$

which means that the matrix $2S_n$ is the finite sample version of the operator S at the sample P_n. From Proposition 5 it is now clear that

$$\begin{pmatrix} IF(z_i; f_{\lambda,K}, P_n)(x_1) \\ \vdots \\ IF(z_i; f_{\lambda,K}, P_n)(x_n) \end{pmatrix} = S_n^{-1} \left((y_i - f_{\lambda,K,P_n}(x_i)) \begin{pmatrix} K(x_i,x_1) \\ \vdots \\ K(x_i,x_n) \end{pmatrix} - \lambda \begin{pmatrix} f_{\lambda,K,P_n}(x_1) \\ \vdots \\ f_{\lambda,K,P_n}(x_n) \end{pmatrix} \right). \tag{8}$$

In order to evaluate the influence function at sample point z_i at a sample distribution P_n, we only need the full sample fit f_{λ,K,P_n} and the matrix S_n^{-1}, which is already obtained when computing f_{λ,K,P_n} (cf. Equation 6). From Theorem 7 one sees similarly that the higher order terms can be computed

recursively as

$$\begin{pmatrix} IF_{k+1}(z_i; f_{\lambda,K}, P_n)(x_1) \\ \vdots \\ IF_{k+1}(z_i; f_{\lambda,K}, P_n)(x_n) \end{pmatrix} = (k+1) S_n^{-1} \frac{\Omega}{n} \begin{pmatrix} IF(z_i; f_{\lambda,K}, P_n)(x_1) \\ \vdots \\ IF_k(z_i; f_{\lambda,K}, P_n)(x_n) \end{pmatrix} - (k+1) IF_k(z_i; f_{\lambda,K}, P_n)(x_i) S_n^{-1} \begin{pmatrix} K(x_i, x_1) \\ \vdots \\ K(x_i, x_n) \end{pmatrix}. \tag{9}$$

Define $[IFM_k]$ the matrix containing $IF_k(z_j; f_{\lambda,K}, P_n)(x_i)$ at entry i, j. Then (9) is equivalent to

$$[IFM_{k+1}] = (k+1)\left(H\,[IFM_k] - nH \bullet [IFM_k]\right)$$

with $\bullet$ denoting the entrywise matrix product (also known as the Hadamard product). Or equivalently

$$[IFM_{k+1}] = (k+1)\left(H([IFM_k] \bullet M(1-n))\right) \tag{10}$$

with M the matrix containing $1/(1-n)$ at the off-diagonal and 1 at the diagonal. A first idea is now to approximate the series in (5) by cutting it off at some step k:

$$f_{\lambda,K,P_n^{-i}}(x_i) \approx f_{\lambda,K,P_n}(x_i) + \sum_{j=1}^{k} \frac{1}{(1-n)^j j!} [IFM_j]_{i,i}. \tag{11}$$

However using (10) we can do a bit better. Expression (5) becomes

$$\begin{aligned} f_{\lambda,K,P_n^{-i}}(x_i) = f_{\lambda,K,P_n}(x_i) &+ \frac{1}{1-n}[IFM_1]_{i,i} + \frac{1}{1-n}[H(IFM_1 \bullet M)]_{i,i} \\ &+ \frac{1}{1-n}[H(H(IFM_1 \bullet M) \bullet M)]_{i,i} + \ldots \end{aligned}$$

In every term there is a multiplication with H and an entrywise multiplication with M. The latter means that all diagonal elements remain unchanged but the non-diagonal elements are divided by $1-n$. So after a few steps the non-diagonal elements will converge to 0 quite fast. It makes sense to set the non-diagonal elements 0 retaining only the diagonal elements:

$$\begin{aligned} f_{\lambda,K,P_n^{-i}}(x_i) &\approx f_{\lambda,K,P_n}(x_i) + \sum_{j=1}^{k-1} \frac{1}{(1-n)^j j!} [IFM_j]_{i,i} + \frac{1}{(1-n)^k k!} \sum_{j=0}^{\infty} H_{i,i}^j [IFM_k]_{i,i} \\ &= f_{\lambda,K,P_n}(x_i) + \sum_{j=1}^{k-1} \frac{1}{(1-n)^j j!} [IFM_j]_{i,i} + \frac{1}{(1-n)^k k!} \frac{[IFM_k]_{i,i}}{1-H_{i,i}} \end{aligned} \tag{12}$$

since $H_{i,i}$ is always smaller than 1.

4.2 Huber Loss

For the Huber loss function with parameter $b > 0$ we have that

$$L(r) = \begin{cases} r^2 & \text{if } |r| < b. \\ 2b|r| - b^2 & \text{if } |r| > b. \end{cases}$$

and thus

$$L'(r) = \begin{cases} 2r & \text{if } |r| < b \\ 2b \text{ sign}(r) & \text{if } |r| > b \end{cases}, \qquad L''(r) = \begin{cases} 2 & \text{if } |r| < b \\ 0 & \text{if } |r| > b \end{cases}.$$

Note that the derivatives in $|r| = b$ do not exist, but in practice the probability that a residual exactly equals b is 0, so we further ignore this possibility. The following equation holds:

$$f_{\lambda,K,P_n}(x) = \frac{1}{n}\sum_{i=1}^{n} \alpha_i K(x_i, x) \quad \text{with} \quad 2\lambda\alpha_j = L'(y_j - \frac{1}{n}\sum_{i=1}^{n} \alpha_i K(x_i, x_j)). \tag{13}$$

Thus a set of possibly non-linear equations has to be solved in α. Once the solution for the full sample is found, an approximation of the leave-one-out error is obtained in a similar way as for least squares. Proposition 5 for P_n gives the first order influence function.

$$\begin{pmatrix} IF(z_i; f_{\lambda,K}, P_n)(x_1) \\ \vdots \\ IF(z_i; f_{\lambda,K}, P_n)(x_n) \end{pmatrix} = S_b^{-1}\left(L'(y_i - f_{\lambda,K,P_n}(x_i)) \begin{pmatrix} K(x_i, x_1) \\ \vdots \\ K(x_i, x_n) \end{pmatrix} - \lambda \begin{pmatrix} f_{\lambda,K,P_n}(x_1) \\ \vdots \\ f_{\lambda,K,P_n}(x_n) \end{pmatrix} \right)$$

with $S_b = 2\lambda I_n + \Omega \bullet B/n$ and B the matrix containing $L''(y_i - f_{\lambda,K,P_n}(x_i)$ at every entry in the ith column. Let $H_b = S_b^{-1}\Omega/n \bullet B$. Starting from Theorem 7 one finds analogously as (10) the following recursion to compute higher order terms.

$$[IFM_{k+1}] = (k+1)\left(H_b([IFM_k] \bullet M(1-n))\right).$$

Finally one can use these matrices to approximate the leave-one-out estimator as

$$f_{\lambda,K,P_n^{-i}}(x_i) \approx f_{\lambda,K,P_n}(x_i) + \sum_{j=1}^{k-1} \frac{1}{(1-n)^j j!}[IFM_j]_{i,i} + \frac{1}{(1-n)^k k!}\frac{[IFM_k]_{i,i}}{1-[H_b]_{i,i}} \tag{14}$$

in the same way as in (12)

4.3 Reweighted KBR

In Equation (14) the full sample estimator f_{λ,K,P_n} is of course needed. For a general loss function L one has to solve Equation (13) to find f_{λ,K,P_n}. A fast way to do so is to use reweighted KBR with a least squares loss. Let

$$W(r) = \frac{L'(r)}{2r}. \tag{15}$$

Then we can rewrite (13) as

$$2\lambda f_{\lambda,K,P_n}(x_k) = \frac{1}{n}\sum_{i=1}^{n} L'(y_i - f_{\lambda,K,P_n}(x_i))K(x_i,x_k) \quad \forall 1 \le k \le n.$$
$$= \frac{1}{n}\sum_{i=1}^{n} 2W(y_i - f_{\lambda,K,P_n}(x_i))(y_i - f_{\lambda,K,P_n}(x_i))K(x_i,x_k).$$

Denoting $w_i = W(y_i - f_{\lambda,K,P_n}(x_i))$ this means that

$$\lambda f_{\lambda,K,P_n}(x_k) = \frac{1}{n}\sum_{i=1}^{n} w_i y_i K(x_i,x_k) - \frac{1}{n}\sum_{i=1}^{n} w_i f_{\lambda,K,P_n}(x_i)K(x_i,x_k) \quad \forall 1 \le k \le n.$$

Let I_w denote the $n \times n$ diagonal matrix with w_i at entry i,i. Then

$$\begin{pmatrix} f_{\lambda,K,P_n}(x_1) \\ \vdots \\ f_{\lambda,K,P_n}(x_n) \end{pmatrix} = \left(\frac{\Omega}{n} + \lambda I_w\right)^{-1} \frac{\Omega}{n} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{16}$$

and thus f_{λ,K,P_n} can be written as a reweighted least squares estimator with additional weights w_i compared to Equations (6) and (7). Of course these weights still depend on the unknown f_{λ,K,P_n}, so (16) only implicitly defines f_{λ,K,P_n}. It does suggest the following iterative reweighting algorithm.

1. Start with simple least squares computing (7). Denote the solution f^0_{λ,K,P_n}.
2. At step $k+1$ compute weights $w_{i,k} = W(y_i - f^k_{\lambda,K,P_n}(x_i))$.
3. Solve (16) using the weights $w_{i,k}$. Let the solution be f^{k+1}_{λ,K,P_n}.

In Suykens et al. (2002a) it is shown that this algorithm usually converges in very few steps. In Debruyne et al. (2006) the robustness of such stepwise reweighting algorithm is analyzed by calculating stepwise influence functions. It is shown that the influence function is stepwise reduced under certain conditions on the weight function.

For the Huber loss with parameter b Equation (15) means that the corresponding weight function equals $W(r) = 1$ if $|r| \le b$ and $W(r) = b/|r|$ if $|r| > b$. This gives a clear interpretation of this loss function: all observations with error smaller than b remain unchanged, but the ones with error larger than b are downweighted compared to the least squares loss. This also explains the gain in robustness. One can expect better robustness as b decreases.

It would be possible to compute higher order terms of such $k-$step estimators as well. Then one could explicitly use these terms to approximate the leave-one-out error of the $k-$step reweighted estimator. In this paper however we use the reweighting only to compute the full sample estimator f_{λ,K,P_n} and we assume that it is fully converged to the solution of (13). For the model selection (14) is then used.

5. Model Selection

Once the approximation of $f_{\lambda,K,P_n^{-i}}$ is obtained, one can proceed with model selection using the leave-one-out principle. In the next paragraphs we propose a specific implementation taking into account performance as well as robustness.

5.1 Definition

The traditional leave-one-out criterion is given by

$$\mathrm{LOO}(\lambda, K) = \frac{1}{n}\sum_{i=1}^{n} V(y_i - f_{\lambda,K,P_n^{-i}}(x_i)) \tag{17}$$

with V an appropriate loss function. The values of λ and of possible kernel parameters for which this criterion is minimal, are then selected to train the model. The idea we investigate is to replace the explicit leave-one-out by the approximation in (12) for least squares and (14) for the Huber loss.

Definition 8 *The k-th order influence function criterion at a regularization parameter $\lambda > 0$ and kernel K for Huber loss KBR with parameter b is defined as*

$$C_{IF}^k(\lambda, K, b) = \frac{1}{n}\sum_{i=1}^{n} V\left(y_i - f_{\lambda,K,P_n}(x_i) - \sum_{j=1}^{k-1}\frac{1}{(1-n)^j j!}[IFM_j]_{i,i} - \frac{1}{(1-n)^k k!}\frac{[IFM_k]_{i,i}}{1-[H_b]_{i,i}}\right).$$

For KBR with a least squares loss we write

$$C_{IF}^k(\lambda, K, \infty) = \frac{1}{n}\sum_{i=1}^{n} V\left(y_i - f_{\lambda,K,P_n}(x_i) - \sum_{j=1}^{k-1}\frac{1}{(1-n)^j j!}[IFM_j]_{i,i} - \frac{1}{(1-n)^k k!}\frac{[IFM_k]_{i,i}}{1-[H]_{i,i}}\right).$$

since a least squares loss is a limit case of the Huber loss as $b \to \infty$.

Several choices need to be made in practice. For k taking five steps seems to work very well in the experiments. If we refer to the criterion with this specific choice $k = 5$ we write C_{IF}^5. For V one typically chooses the squared loss or the absolute value corresponding to the mean squared error and the mean absolute error. Note that V does not need to be the same as the loss function used to compute f_{λ,K,P_n} (the latter is always denoted by L). Recall that a loss function L with bounded first derivative L' is needed to perform robust fitting. It is important to note that this result following from Proposition 5 holds for a fixed choice of λ and the kernel K. However, if these parameters are selected in a data driven way, outliers in the data might have a large effect on the selection of the parameters. Even if a robust estimator is used, the result could be quite bad if wrong choices are made for the parameters due to the outliers. It is thus important to use a robust loss function V as well. Therefore we set V equal to the absolute value loss function unless we explicitly state differently. In Section 6.1 an illustration is given on what can go wrong if a least squares loss is chosen for V instead of the absolute value.

5.2 Optimizing b

With k and V now specified, the criterion C_{IF}^5 can be used to select optimal hyperparameters for a KBR estimator with L the Huber loss with parameter b. Now the final question remains how to choose b. In Section 4.3 it was argued that b controls the robustness of the estimator since all observations with error smaller than b are downweighted compared to the least squares estimator. Thus we want to choose b small enough such that outlying observations receive sufficiently small weight, but also large enough such that the good non outlying observations are not downweighted too much. A priori it is quite difficult to find such a good choice for b, since this will depend on the scale of the errors.

However, one can also treat b as an extra parameter that is part of the optimization, consequently minimizing C^5_{IF} for λ, K and b simultaneously. The practical implementation we propose is as follows:

1. Let Λ be a set of reasonable values for the regularization parameter λ and let $\mathcal{K}$ be a set of possible choices for the kernel K (for instance a grid of reasonable bandwidths if one considers the RBF kernel).

2. Start with L the least squares loss. Find good choices for λ and K by minimizing $C^5_{IF}(\lambda, K, \infty)$ for all $\lambda \in \Lambda$ and $K \in \mathcal{K}$. Compute the residuals r_i with respect to the least squares fit with these optimal λ and K.

3. Compute a robust estimate of the scale of these residuals. We take the Median Absolute Deviation (MAD):

$$\hat{\sigma}_{err} = \text{MAD}(r_1, \ldots, r_n) = \frac{1}{\Phi^{-1}(0.75)} \text{median}(|r_i - \text{median}(r_i)|) \tag{18}$$

   with $\Phi^{-1}(0.75)$ the 0.75 quantile of a standard normal distribution.

4. Once the scale of the errors is estimated in the previous way, reasonable choices of b can be constructed, for example $\{1,2,3\} \times \hat{\sigma}_{err}$. This means that we compare downweighting observations further away than 1, 2, 3 standard deviations. We also want to compare to the least squares fit and thus set

$$\mathcal{B} = \{\hat{\sigma}_{err}, 2\hat{\sigma}_{err}, 3\hat{\sigma}_{err}, \infty\}.$$

5. Minimize $C^5_{IF}(\lambda, K, b)$ over all $\lambda \in \Lambda$, $K \in \mathcal{K}$ and $b \in \mathcal{B}$. The optimal values of b, λ and K can then be used to construct the final fit.

5.3 Generalized Cross Validation

The criterion C^5_{IF} uses influence functions to approximate the leave-one-out error. Other approximations have been proposed in the literature. In this section we very briefly mention some results that are described for example by Wahba (1990) in the context of spline regression. The following result can be proven.

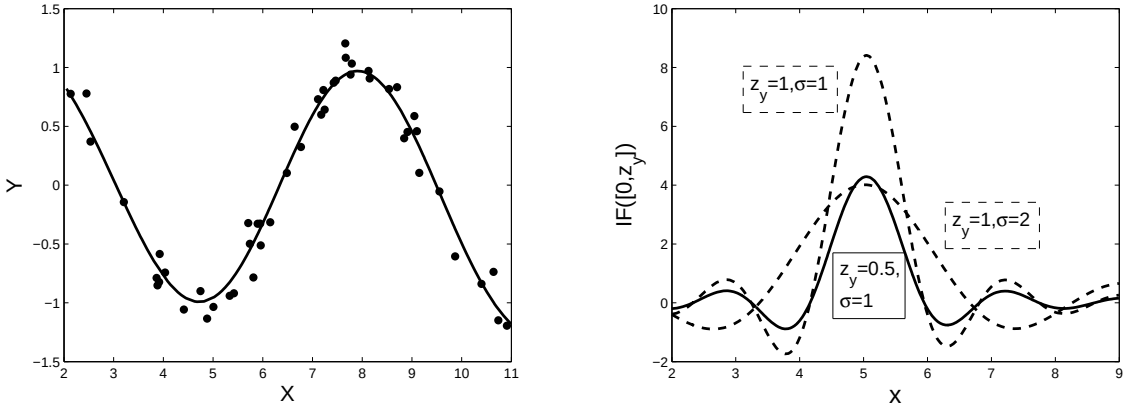
Let $\tilde{P}_n^{-i}$ be the sample P_n with observation (x_i, y_i) replaced by $(x_i, f_{\lambda,K,P_n^{-i}}(x_i))$. Suppose the following conditions are satisfied for any sample P_n:

$$(i) \quad f_{\lambda,K,\tilde{P}_n^{-i}}(x_i) = f_{\lambda,K,P_n^{-i}}(x_i). \tag{19}$$

$$(ii) \quad \text{There exists a matrix } H \text{ such that } \begin{pmatrix} f_{\lambda,K,P_n}(x_1). \\ \vdots \\ f_{\lambda,K,P_n}(x_n) \end{pmatrix} = H \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}. \tag{20}$$

Then

$$f_{\lambda,K,P_n^{-i}}(x_i) = \frac{f_{\lambda,K,P_n}(x_i) - H_{i,i} y_i}{1 - H_{i,i}}. \tag{21}$$



(a) (b)

Figure 1: (a) Data and least squares fit. (b) Influence functions at $[5,0.5]$ with $\sigma = 1$, at $[5,1]$ with $\sigma = 1$ and $\sigma = 2$.

For KBR with the least squares loss condition (22) is indeed satisfied (cf. Equation 7), but condition (19) is not, although it holds approximately. Then (21) can still be used as an approximation of the leave-one-out estimator. The corresponding model selection criterion is given by

$$CV(\lambda,K) = \frac{1}{n}\sum_{i=1}^{n} V\left(\frac{y_i - f_{\lambda,K,P_n}(x_i)}{1-H_{i,i}}\right). \tag{22}$$

We call this approximation CV. Sometimes a further approximation is made replacing every $H_{i,i}$ by $\text{trace}(H)/n$. This is called Generalized Cross Validation (GCV, Wahba, 1990). Note that the diagonal elements of the hatmatrix H play an important role in the approximation with the influence function too (12). Both penalize small values on the diagonal of H.

For KBR with a general loss function one does not have a linear equation of the form of (22), and thus it is more difficult to apply this approximation. We shall thus use CV for comparison in the experiments only in the case of least squares.

6. Empirical Results

We illustrate the results on a toy example and a small simulation study.

6.1 Toy Example

As a toy example 50 data points were generated with x_i uniformly distributed on the interval $[2,11]$ and $y_i = \sin(x_i) + e_i$ with e_i Gaussian distributed noise with standard deviation 0.2. We start with kernel based regression with a least squares loss and a Gaussian kernel. The data are shown in Figure $1(a)$ as well as the resulting fit with $\lambda = 0.001$ and $\sigma = 2$. The first order influence function at $[5,0.5]$ is depicted in Figure $1(b)$ as the solid line. This reflects the asymptotic change in the fit when a point would be added to the data in Figure $1(a)$ at the position $(5,0.5)$. Obviously this influence is the largest at the x-position where we put the outlier, that is, $x = 5$. Furthermore we see that the influence is local, since it decreases as we look further away from $x = 5$. At $x = 8$ for

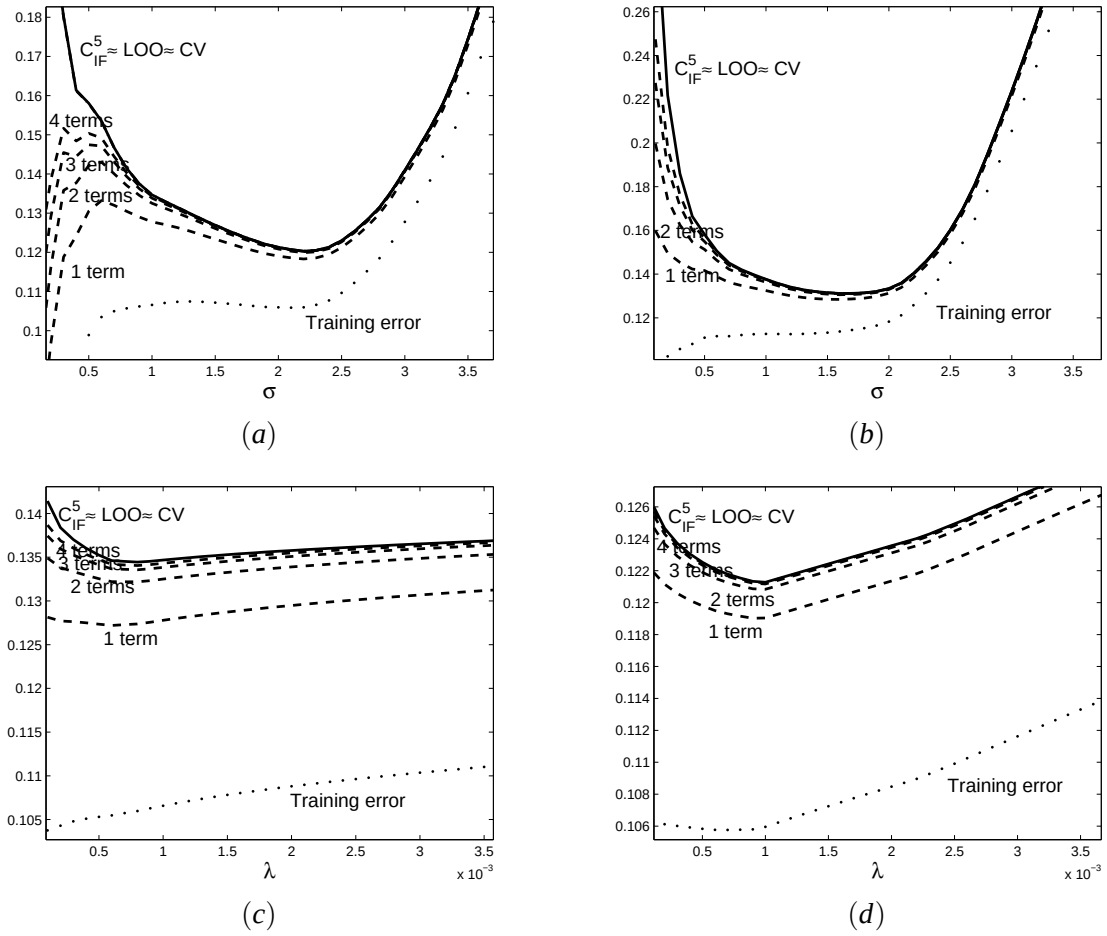


Figure 2: Comparison of training error (dotted line), approximations using (11) (dashed lines), the proposed criterion C_{IF}^k with $k = 5$ (solid line), the exact leave-one-out error and the CV approximation (both collapsing with C_{IF}^k on these plots). Situation (a): as a function of σ at $\lambda = 0.001$, (b) as a function of σ at $\lambda = 0.005$, (c) as a function of λ at $\sigma = 1$, (d) as a function of λ at $\sigma = 2$.

instance the influence function is almost 0. When we change z from $[5, 0.5]$ to $[5, 1]$, the influence function changes too. It still has the same oscillating behavior, but the peaks are now higher. This reflects the non-robustness of the least squares estimator: if we would continue raising the point z, then $IF(z; f_{\lambda,K})$ would become larger and larger, since it is an unbounded function of z. When it comes down to model selection, it is interesting to check the effect of the hyperparameters in play. When we change the bandwidth σ from 1 to 2, the peaks in the resulting influence function in Figure 1 are less sharp and less high. This reflects the loss in stability when small bandwidths are chosen: then the fit is more sensitive to small changes in the data and thus less stable.

Consider now the approximation of the leave-one-out error using the influence functions. We still use the same data as in the previous paragraph. The dashed lines in Figure $2(a)$ show the approximations using (11), that is simply cutting off the expansion after a number of steps, at fixed

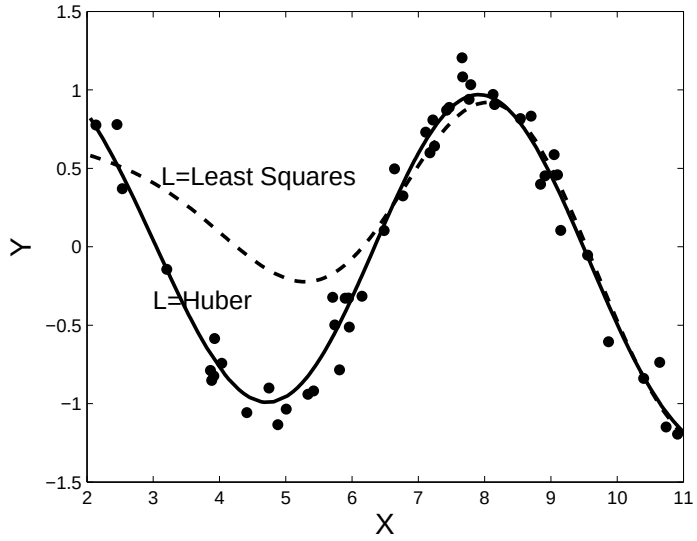
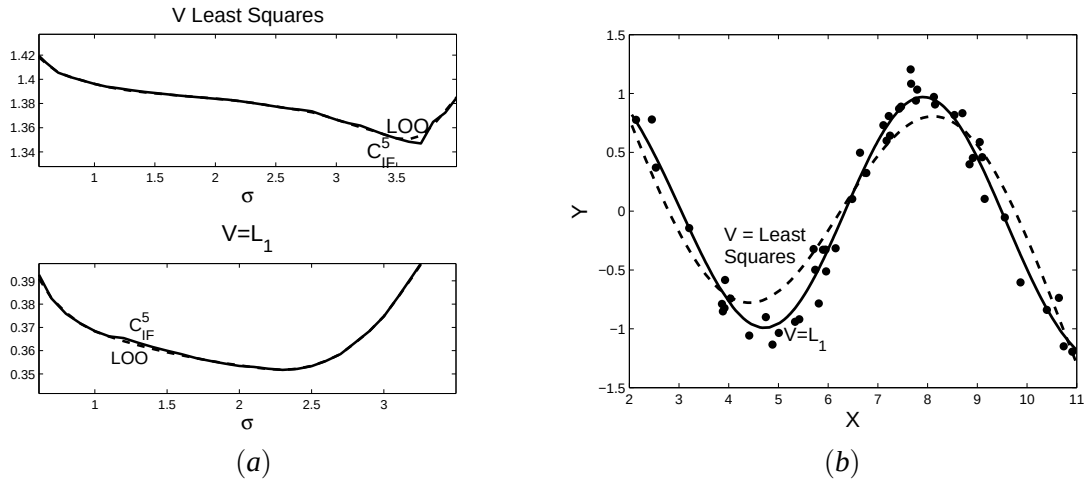


Figure 3: Data with outlier at $(4,5)$. The parameters $\lambda = 0.001$ and $\sigma = 2$ are fixed. Dashed: KBR with least squares loss function. Solid: KBR with Huber loss function ($b = 0.2$).

$\lambda = 0.001$ as a function of the bandwidth σ. We observe convergence from the training error towards the leave-one-out error as the number of terms included is increased. Unfortunately the convergence rate depends on the value of σ: convergence is quite slow at small values of σ. This is no surprise looking at (12). There we approximated the remainder term by a quantity depending on $(1-H_{i,i})^{-1}$. When σ is small, the diagonal elements of H become close to 1. In that case the deleted remainder term can indeed be quite large. Nevertheless, this approach can still be useful if some care is taken not to consider values of λ and σ that are too small. However, the criterion C_{IF}^5 from Definition 8 using the approximation in (12) is clearly superior. We see that the remainder term is now adequately estimated and a good approximation is obtained at any σ. The resulting curve is undistinguishable from the exact leave-one-out error. The mean absolute difference is $3.2\ 10^{-5}$, the maximal difference is $1.8\ 10^{-4}$. The CV approximation also yields a good result being indistinguishable from the exact leave-one-out error on the plot as well. The mean absolute difference is $4.1\ 10^{-4}$ and the maximal difference equals $1.8\ 10^{-3}$. Thus C_{IF}^5 is closer to the true leave-one-out error than CV, although the difference is irrelevant when it comes down to selecting a good σ.

Figure 2 also shows plots for the leave-one-out error and its various approximations at (b) $\lambda = 0.005$ as a function of σ, (c) $\sigma = 1$ as a function of λ, (d) $\sigma = 2$ as a function of λ. In these cases as well it is observed that the cutoff strategy yields decent results if a sufficient number of terms is taken into account and if one does not look at values of λ and σ that are extremely small. The best strategy is to take the remainder term into account using the criterion C_{IF}^k from Definition 8.

In Figure 3 we illustrate robustness. An (extreme) outlier was added at position $(4,5)$ (not visible on the plot). This outlier leads to a bad fit when LS-KBR is used with $\lambda = 0.001$ and $\sigma = 2$ (dashed line). When a Huber loss function is used with $b = 0.2$ a better fit is obtained that still nicely predicts the majority of observations. This behavior can be explained by Proposition 5. The least squares loss has an unbounded first derivative and thus the influence of outliers can be arbitrary large. The Huber loss has a bounded first derivative and thus the influence of outliers is bounded as



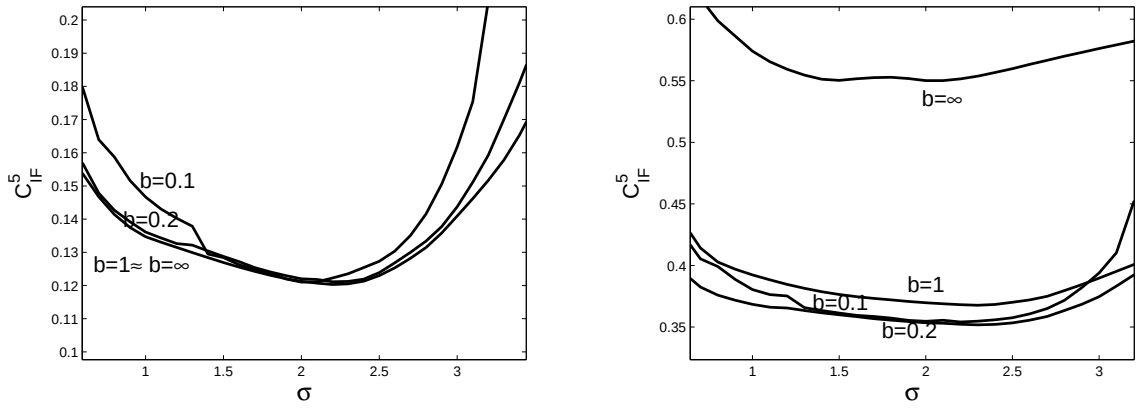
(a) (b)

Figure 4: (a) Optimization of σ at $\lambda = 0.001$. Upper: using least squares loss V in the model selection. Lower: using L_1 loss V in the model selection. For the estimation the loss function L is always the Huber loss with $b = 0.2$. (b) Resulting fits. Dashed line: $\sigma = 3.6$ (optimal choice using V least squares). Solid line: $\sigma = 2.3$ (optimal choice using L_1 loss for V.

well. However, note that in this example as well as in Proposition 5 the hyperparameters λ and σ are assumed to have fixed values. In practice one wants to choose these parameters in a data driven way. Figure $4(a)$ shows the optimization of σ at $\lambda = 0.001$ for KBR with L the Huber loss with $b = 0.2$. In the upper panel the least squares loss is used for V in the model selection criteria. Both exact leave-one-out and C^5_{IF} indicate that a value of $\sigma \approx 3.6$ should be optimal. This results in the dashed fit in Figure $4(b)$. In the lower panel of Figure 4 the L_1 loss is used for V in the model selection criteria. Both exact leave-one-out and C^5_{IF} indicate that a value of $\sigma \approx 2.3$ should be optimal. This results in the solid fit in Figure $4(b)$. We clearly see that, although in both cases a robust estimation procedure is used (Huber loss for L), the outlier can still be quite influential through the model selection. To obtain full protection against outliers, both the estimation and the model selection step require robustness, for example by selecting both L and V in a robust way.

Finally let us investigate the role of the parameter b used in the Huber loss function. We now use C^5_{IF} with V the L_1 loss. When we apply C^5_{IF} to the clean data without the outlier, we observe in Figure $5(a)$ that the choice of b does not play an important role. This is quite expected: since there are no outliers, there is no reason why least squares ($b = \infty$) would not perform well. On the contrary, if we use a small b such as $b = 0.1$ we get a slightly worse result. Again this is not a surprise, since with small b we will downweight a lot of points that are actually perfectly ok.

The same plot for the data containing the outlier yields a different view in Figure $5(b)$. The values of C^5_{IF} are much higher for least squares than for the Huber loss with smaller b. Thus it is automatically detected that a least squares loss is not the appropriate choice, which is a correct assessment since the outlier will have a large effect (cf. the dashed line in Figure 3). The criterion C^5_{IF} indicates a choice $b = 0.2$, which leads to a better result indeed (cf. the solid line in Figure 3)



(a) (b)

Figure 5: C^5_{IF} at $\lambda = 0.001$ as a function of σ for several values of b for (a) the clean data without the outlier, (b) the data with the outlier.

6.2 Other Examples

This part presents the results of a small simulation study. We consider some well known settings.

- Friedman 1 ($d = 10$): $y(x) = 10\sin(\pi x_1 x_2) + 20(x_3 - 1/2)^2 + 10x_4 + 5x_5 + \sum_{i=6}^{10} 0.x_i$. The covariates are generated uniformly in the hypercube in $\mathbb{R}^{10}$.

- Friedman 2 ($d = 4$): $y(x) = \frac{1}{3000}(x_1^2 + (x_2x_3 - (x_2x_4)^{-2}))^{1/2}$, with $0 < x_1 < 100$, $20 < x_2/(2\pi) < 280$, $0 < x_3 < 1$, $1 < x_4 < 11$.

- Friedman 3 ($d = 4$): $y(x) = \tan^{-1}(\frac{x_2x_3 - (x_2x_4)^{-2}}{x_1})$, with the same range for the covariates as in Friedman 2. For each of the Friedman data sets 100 observations were generated with Gaussian noise and 200 noise free test data were generated.

- Boston Housing Data from the UCI machine learning depository with 506 instances and 13 covariates. Each split 450 observations were used for training and the remaining 56 for testing.

- Ozone data from ftp://ftp.stat.berkeley.edu/pub/users/breiman/ with 202 instances and 12 covariates. Each split 150 observations were used for training and the remaining 52 for testing.

- Servo data from the UCI machine learning depository with 167 instances and 4 covariates. Each split 140 observations were used for training and the remaining 27 for testing.

For the real data sets (Boston, Ozone and Servo), new contaminated data set were constructed as well by adding large noise to 10 training points, making these 10 points outliers.

The hyperparameters λ and σ are optimized over the following grid of hyperparametervalues:

- $\lambda \in \{50, 10, 5, 3, 1, 0.8, 0.5, 0.3, 0.1, 0.08, 0.05, 0.01, 0.005\} \times 10^{-3}$.

- For each data set 500 distances were calculated between two randomly chosen observations. Let $d_{(i)}$ be the ith largest distance. Then the following grid of values for σ is considered: $\sigma \in \{\frac{1}{2}d_{(1)}, d_{(1)}, d_{(50)}, d_{(100)}, d_{(150)}, d_{(200)}, d_{(250)}, d_{(300)}, d_{(350)}, d_{(400)}, d_{(450)}, d_{(500)}, 2d_{(500)}\}$.

In each replicate the Mean Squared Error of the test data is computed. For every data set the average MSE over 20 replicates is shown in Table 1 (upper table). A two-sided paired Wilcoxon rank test is used to check statistical significance: values in italic are significantly different from the smallest value at significance level 0.05. If underlined significance holds even at significance level 10^{-4}. Standard errors are shown as well (lower table). First we consider the least squares loss for L with the criterion $C^5_{IF}(\lambda, \sigma, \infty)$ (Definition 8), with exact leave-one-out (17) and with CV (22). These are the first 3 columns in Table 1. We see that the difference between these 3 criteria is very small. This means that both CV and C^5_{IF} provide good approximations of the leave-one-out error.

Secondly, we considered each time the residuals of the least squares fit with optimal λ and σ according to $C^5_{IF}(\lambda, K, \infty)$. An estimate $\hat{\sigma}_{err}$ of the scale of the residuals is computed as the MAD of these residuals (18). Then we applied KBR with a Huber loss and parameter $b = 3\hat{\sigma}_{err}$. The resulting MSE with this loss and λ and σ minimizing $C^5_{IF}(\lambda, \sigma, 3\hat{\sigma}_{err})$ is given in column 4 in Table 1. Similar results are obtained for $b = 2\hat{\sigma}_{err}$ in column 5 and with $b = \hat{\sigma}_{err}$ in column 6. For the data sets without contamination we see that using a Huber loss instead of least squares gives similar results except for the Boston housing data, Friedman 1 and especially Friedman 2. For those data sets a small value of b is inappropriate. This might be explained by the relationship between the loss function and the error distribution. For a Gaussian error distribution least squares is often an optimal choice (cf. maximum likelihood theory). Since the errors in the Friedman data are explicitly generated as Gaussian, this might explain why least squares outperforms the Huber loss. For real data sets, the errors might not be exactly Gaussian, and thus other loss function can perform at least equally well as least squares. For the data sets containing the outliers the situation changes of course. Now least squares is not a good option because of its lack of robustness. Clearly the outliers have a large and bad effect on the quality of the predictions. This is not the case when the Huber loss function is chosen. Then the effect of the outliers is reduced. Choosing $b = 3\hat{\sigma}_{err}$ already leads to a large improvement. Decreasing b leads to even better results (note that the p-values are smaller than 10^{-4} for any significant pairwise comparison).

Finally we also consider optimizing b. We apply the algorithm outlined in Section 5.2. Corresponding MSE's are given in the last column of Table 1. For the Friedman 1 and Friedman 2 data sets for instance this procedure indeed detects that least squares is an appropriate loss function and automatically avoids choosing b too small. For the contaminated data sets the procedure detects that least squares is not appropriate and that changing to a Huber loss with a small b is beneficial, which is indeed a correct choice yielding smaller MSE's. In fact, only for the Friedman 2 data, the automatic choice of b is significantly worse than the optimal choice (p-value=0.03), whereas the benefits at the contaminated data are large (all p-values $< 10^{-4}$).

7. Conclusion

Heuristic links between the concept of the influence function and concepts as leave-one-out cross validation and stability were considered in Section 2, indicating some interesting applications of the influence function and the leave-one-out error in previous literature. New results include the calculation of higher order influence functions and a recursive relation between subsequent terms. It is shown that these theoretical results can be applied in practice to approximate the leave-one-out esti-

	$b=\infty$ (=LS)			$b=3\hat{\sigma}_{err}$	$b=2\hat{\sigma}_{err}$	$b=\hat{\sigma}_{err}$	($b=$ optimized)
	LOO	CV	C^5_{IF}	C^5_{IF}	C^5_{IF}	C^5_{IF}	C^5_{IF}
F1	1.63	1.63	1.63	1.66	*1.70*	*1.82*	1.67
F2	1.30	1.30	1.30	*1.42*	1.71	3.02	*1.39*
F3	2.42	2.42	2.42	2.42	2.42	2.37	2.38
B	10.58	10.58	10.58	10.82	11.30	*12.21*	10.79
O	13.91	13.92	13.91	13.76	13.73	13.91	13.94
S	0.40	0.40	0.40	0.43	0.41	0.41	0.40
B+o	37.54	37.54	37.54	14.60	13.73	12.68	12.78
O+o	78.78	78.78	78.77	21.20	18.85	16.74	16.74
S+o	1.60	1.60	1.60	0.61	0.54	0.46	0.46

	$b=\infty$ (=LS)			$b=3\hat{\sigma}_{err}$	$b=2\hat{\sigma}_{err}$	$b=\hat{\sigma}_{err}$	($b=$ optimized)
	LOO	CV	C^5_{IF}	C^5_{IF}	C^5_{IF}	C^5_{IF}	C^5_{IF}
F1	0.09	0.09	0.09	0.09	0.10	0.08	0.09
F2	0.14	0.14	0.15	0.16	0.20	0.36	0.15
F3	0.03	0.03	0.03	0.03	0.03	0.05	0.05
B	1.39	1.39	1.39	1.40	1.46	1.51	1.39
O	0.86	0.86	0.87	0.78	0.78	0.75	0.81
S	0.05	0.05	0.05	0.09	0.08	0.09	0.09
B+o	2.91	2.91	2.91	1.12	1.09	1.02	1.04
O+o	3.44	3.44	3.44	1.01	0.97	1.03	1.03
S+o	0.16	0.16	0.16	0.07	0.07	0.08	0.08

Table 1: Simulation results. Upper: Mean Squared Errors. Lower: standard errors. Friedman 1 (F1), Friedman 2 (F2), Friedman 3 (F3), Boston Housing (B), Ozone (O), Servo (S), Boston Housing with outliers (B+o), Ozone with outliers (O+o) and Servo with outliers (S+o). Italic values are significantly different from the smallest value in the row with p-value in between 0.05 and 0.001 using a paired Wilcoxon rank test; underlined values are significant with p-value $< 10^{-4}$.

mator. Experiments indicate that the quality of this approximation is quite good. The approximation is used in a model selection criterion to select the regularization and kernel parameters.

We discussed the importance of robustness in the model selection step. A specific procedure is suggested using an L_1 loss in the model selection criterion and a Huber loss in the estimation. Due to an iterative reweighting algorithm to compute such a Huber loss estimator and due to the fast approximation of the leave-one-out error, everything can be computed fast starting from the least squares framework. With an a priori choice of the parameter b in the Huber loss this leads to better robustness if b is chosen small enough. If b is chosen too small on the other hand this might result in worse predictions. However, this parameter can be selected in a data driven way as well. Experiments suggest that this often yields a good trade-off between the robustness of choosing a small b and the sometimes better predictive capacity of least squares.

Acknowledgments

JS acknowledges support from K.U. Leuven, GOA-Ambiorics, CoE EF/05/006, FWO G.0499.04, FWO G.0211.05, FWO G.0302.07, IUAP P5/22.

MH acknowledges support from FWO G.0499.04, the GOA/07/04-project of the Research Fund KULeuven, and the IAP research network nr. P6/03 of the Federal Science Policy, Belgium.

Appendix A.

Proof of Theorem 6

Let P be a distribution, $z \in \mathcal{X} \times \mathcal{Y}$ and $P_{\varepsilon,z} = (1-\varepsilon)P + \varepsilon\Delta_z$ with Δ_z the Dirac distribution in z. We start from the representer theorem of DeVito et al. (2004) (a generalization of (13)):

$$2\lambda f_{\lambda,K,P_{\varepsilon,z}} = \mathbb{E}_{P_{\varepsilon,z}}[L'(Y - f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)].$$

By definition of $P_{\varepsilon,z}$ and since $\mathbb{E}_{\Delta_z} g(X) = g(z)$ for any function g:

$$2\lambda f_{\lambda,K,P_{\varepsilon,z}} = (1-\varepsilon)\mathbb{E}_P[L'(Y - f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)] + \varepsilon L'(z_y - f_{\lambda,K,P_{\varepsilon,z}}(z_x))\Phi(z_x).$$

Taking the first derivative on both sides with respect to ε yields

$$\begin{aligned}
2\lambda\frac{\partial}{\partial\varepsilon} f_{\lambda,K,P_{\varepsilon,z}} =& (1-\varepsilon)\mathbb{E}_P[-\frac{\partial}{\partial\varepsilon} f_{\lambda,K,P_{\varepsilon,z}}(X)L''(Y - f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)] \\
&- \mathbb{E}_P[L'(Y - f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)] + L'(z_y - f_{\lambda,K,P_{\varepsilon,z}}(z_x))\Phi(z_x) \\
&- \varepsilon\frac{\partial}{\partial\varepsilon} f_{\lambda,K,P_{\varepsilon,z}}(z_x))L''(z_y - f_{\lambda,K,P_{\varepsilon,z}}(z_x))\Phi(z_x).
\end{aligned}$$

The second derivative equals

$$\begin{aligned}
2\lambda\frac{\partial}{\partial^2\varepsilon} f_{\lambda,K,P_{\varepsilon,z}} =& -\mathbb{E}_P[-\frac{\partial}{\partial\varepsilon} f_{\lambda,K,P_{\varepsilon,z}}(X)L''(Y - f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)] \\
&+ (1-\varepsilon)\mathbb{E}_P[-\frac{\partial}{\partial^2\varepsilon} f_{\lambda,K,P_{\varepsilon,z}}(X)L''(Y - f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)] \\
&+ (1-\varepsilon)\mathbb{E}_P[-\frac{\partial}{\partial\varepsilon} f_{\lambda,K,P_{\varepsilon,z}}(X)L'''(Y - f_{\lambda,K,P_{\varepsilon,z}}(X))(-\frac{\partial}{\partial\varepsilon} f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)] \\
&- \mathbb{E}_P[L''(Y - f_{\lambda,K,P_{\varepsilon,z}}(X))(-\frac{\partial}{\partial\varepsilon} f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)] \\
&- \frac{\partial}{\partial\varepsilon} f_{\lambda,K,P_{\varepsilon,z}}(z_x)L''(z_y - f_{\lambda,K,P_{\varepsilon,z}}(z_x))\Phi(z_x) \\
&- \varepsilon\frac{\partial}{\partial^2\varepsilon} f_{\lambda,K,P_{\varepsilon,z}}(z_x)L''(z_y - f_{\lambda,K,P_{\varepsilon,z}}(z_x))\Phi(z_x) \\
&- \varepsilon\frac{\partial}{\partial\varepsilon} f_{\lambda,K,P_{\varepsilon,z}}(z_x)L'''(z_y - f_{\lambda,K,P_{\varepsilon,z}}(z_x))(-\frac{\partial}{\partial\varepsilon} f_{\lambda,K,P_{\varepsilon,z}}(z_x))\Phi(z_x) \\
&- L''(z_y - f_{\lambda,K,P_{\varepsilon,z}}(z_x))\frac{\partial}{\partial\varepsilon} f_{\lambda,K,P_{\varepsilon,z}}(z_x)\Phi(z_x).
\end{aligned}$$

Simplifying yields

$$\begin{aligned}2\lambda\frac{\partial}{\partial^2\varepsilon}f_{\lambda,K,P_{\varepsilon,z}} =& 2\mathbb{E}_P[\frac{\partial}{\partial\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(X)L''(Y-f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)] \\ &-(1-\varepsilon)\mathbb{E}_P[\frac{\partial}{\partial^2\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(X)L''(Y-f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)] \\ &+(1-\varepsilon)\mathbb{E}_P[\left(\frac{\partial}{\partial\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(X)\right)^2 L'''(Y-f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)] \\ &-2\frac{\partial}{\partial\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(z_x)L''(z_y-f_{\lambda,K,P_{\varepsilon,z}}(z_x))\Phi(z_x) \\ &-\varepsilon\frac{\partial}{\partial^2\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(z_x)L''(z_y-f_{\lambda,K,P_{\varepsilon,z}}(z_x))\Phi(z_x) \\ &+\varepsilon\left(\frac{\partial}{\partial\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(z_x)\right)^2 L'''(z_y-f_{\lambda,K,P_{\varepsilon,z}}(z_x))\Phi(z_x).\end{aligned} \tag{23}$$

Evaluating at $\varepsilon=0$ and bringing all terms containing $\frac{\partial}{\partial^2\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}$ to the left hand side of the equation yields

$$\begin{aligned}2\lambda\frac{\partial}{\partial^2\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}|_{\varepsilon=0}+\mathbb{E}_P[\frac{\partial}{\partial^2\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(X)|_{\varepsilon=0}L''(Y-f_{\lambda,K,P}(X))\Phi(X)] \\ =2\mathbb{E}_P[\frac{\partial}{\partial\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(X)|_{\varepsilon=0}L''(Y-f_{\lambda,K,P}(X))\Phi(X)] \\ +\mathbb{E}_P[\left(\frac{\partial}{\partial\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}|_{\varepsilon=0}(X)\right)^2 L'''(Y-f_{\lambda,K,P}(X)) \\ -2\frac{\partial}{\partial\varepsilon}f_{\lambda,K,P}(z_x)|_{\varepsilon=0}L''(z_y-f_{\lambda,K,P}(z_x))\Phi(z_x).\end{aligned}$$

Since by definition $\frac{\partial}{\partial\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}|_{\varepsilon=0}$ is $IF(z;f_{\lambda,K},P)$ and $\frac{\partial}{\partial^2\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}|_{\varepsilon=0}$ is $IF_2(z;f_{\lambda,K},P)$ we have that

$$\begin{aligned}S(IF_2(z;f_{\lambda,K},P)) = 2\mathbb{E}_P[&IF(z;f_{\lambda,K},P)(X)L''(Y-f_{\lambda,K,P}(X))\Phi(X)] \\ &+\mathbb{E}_P[\big(IF(z;f_{\lambda,K},P)(X)\big)^2 L'''(Y-f_{\lambda,K,P}(X)) \\ &-2IF(z;f_{\lambda,K},P)(z_x)L''(z_y-f_{\lambda,K,P}(z_x))\Phi(z_x)\end{aligned}$$

with the operator S defined by $S: f\rightarrow\lambda f+\mathbb{E}_P L''(Y-f_{\lambda,K,P}(X))f(X)\Phi(X)$. Christmann and Steinwart (2007) prove that S is an invertible operator and thus Theorem 6 follows.

Proof of Theorem 7

First we proof the following for all $2\leq k\in\mathbb{N}$:

$$\begin{aligned}2\lambda\frac{\partial}{\partial^k\varepsilon}f_{\lambda,K}(P_{\varepsilon,z}) =& (1-\varepsilon)\mathbb{E}_P[-\frac{\partial}{\partial^k\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(X)L''(Y-f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)] \\ &+k\mathbb{E}_P[\frac{\partial}{\partial^{k-1}\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(X)L''(Y-f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)] \\ &-kL''(z_y-f_{\lambda,K,P_{\varepsilon,z}}(z_x))\frac{\partial}{\partial^{k-1}\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(z_x)\Phi(z_x) \\ &-\varepsilon L''(z_y-f_{\lambda,K,P_{\varepsilon,z}}(z_x))\frac{\partial}{\partial^k\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(z_x)\Phi(z_x).\end{aligned} \tag{24}$$

Note that for $k=2$ this immediately follows from (23). For general k we give a proof by induction. We assume that (24) holds for k and we then prove that it automatically holds for $k+1$ as well. Taking the derivatives of both sides in (24) we find

$$\begin{aligned}
\lambda\frac{\partial}{\partial^{k+1}\varepsilon}f_{\lambda,K}(P_{\varepsilon,z}) =&(1-\varepsilon)\mathbb{E}_P[-\frac{\partial}{\partial^{k+1}\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(X)L''(Y-f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)]\\
&-\mathbb{E}_P[-\frac{\partial}{\partial^{k}\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(X)L''(Y-f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)]\\
&+k\mathbb{E}_P[\frac{\partial}{\partial^{k}\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(X)L''(Y-f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)]\\
&-k\frac{\partial}{\partial^{k}\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(z_x)L''(z_y-f_{\lambda,K,P_{\varepsilon,z}}(z_x))\Phi(z_x)\\
&-\varepsilon\frac{\partial}{\partial^{k+1}\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(z_x)L''(z_y-f_{\lambda,K,P_{\varepsilon,z}}(z_x))\Phi(z_x)\\
&-\frac{\partial}{\partial^{k}\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(z_x)L''(z_y-f_{\lambda,K,P_{\varepsilon,z}}(z_x))\Phi(z_x)
\end{aligned}$$

from which it follows that (24) holds for $k+1$ indeed. Evaluating this expression in $\varepsilon=0$ yields:

$$\begin{aligned}
&\lambda\frac{\partial}{\partial^{k+1}\varepsilon}f_{\lambda,K}(P_{\varepsilon,z})|_{\varepsilon=0}+\mathbb{E}_P[\frac{\partial}{\partial^{k+1}\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(X)|_{\varepsilon=0}L''(Y-f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)]\\
&=(k+1)\mathbb{E}_P[\frac{\partial}{\partial^{k}\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}(X)|_{\varepsilon=0}L''(Y-f_{\lambda,K,P_{\varepsilon,z}}(X))\Phi(X)]\\
&-(k+1)\frac{\partial}{\partial^{k}\varepsilon}f_{\lambda,K,P_{\varepsilon,z}}|_{\varepsilon=0}(z_x)L''(z_y-f_{\lambda,K,P_{\varepsilon,z}}(z_x))\Phi(z_x).
\end{aligned}$$

Thus

$$\begin{aligned}
S(IF_{k+1}(z;f_{\lambda,K},P)) = (k+1)\Big(&\mathbb{E}_P[IF_k(z;f_{\lambda,K},P)(X)L''(Y-f_{\lambda,K}(X))\Phi(X)]\\
&-[IF_k(z;f_{\lambda,K},P)(z_x)L''(z_y-f_{\lambda,K}(z_x))\Phi(z_x)]\Big).
\end{aligned}$$

Since S is an invertible operator the result in Theorem 7 follows.

References

D.D. Boos and R.J. Serfling. A note on differentials and the CLT and LIL for statistical functions. *Annals of Statistics*, 8:618–624, 1980.

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2001.

A. Christmann and I. Steinwart. Consistency and robustness of kernel based regression. *Bernoulli*, 13:799–819, 2007.

A. Christmann and I. Steinwart. On robust properties of convex risk minimization methods for pattern recognition. *Journal of Machine Learning Research*, 5:1007–1034, 2004.

M. Debruyne, A. Christmann, M. Hubert, and J.A.K. Suykens. Robustness and stability of reweighted kernel based regression. Technical report TR 06-09, K.U. Leuven, available at http://wis.kuleuven.be/stat/robust, 2006.

E. DeVito, L. Rosasco, A. Caponnetto, M. Piana, and A. Verri. Some properties of regularized kernel methods. *Journal of Machine Learning Research*, 5:1363–1390, 2004.

T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13:1–50, 2000.

L.T. Fernholz. *Von Mises Calculus for Statistical Functionals*. Lecture Notes in statistics 19, Springer, New York, 1983.

F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Wiley, New York, 1986.

P.J. Huber. *Robust Statistics*. Wiley, New York, 1981.

S. Kutin and P. Niyogi. Almost everywhere algorithmic stability and generalization error. In A. Daruich and N. Friedman, editors, *Proceedings of Uncertainty in AI*. Morgan Kaufmann, Edmonton, 2002.

T. Poggio, R. Rifkin, S. Mukherjee, and P. Niyogi. General conditions for predictivity in learning theory. *Nature*, 428:419–422, 2004.

J.A.K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle. Weighted least squares support vector machines : Robustness and sparse approximation. *Neurocomputing*, 48:85–105, 2002a.

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

A.N. Tikhonov and V.Y. Arsenin. *Solutions of Ill Posed Problems*. W.H. Winston, Washington D.C., 1977.

G. Wahba. *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, 1990.