KATHOLIEKE UNIVERSITEIT LEUVEN

Faculteit Wetenschappen

Departement Wiskunde

Robustness of censored depth quantiles, PCA and kernel based regression, with new tools for model selection

Michiel Debruyne

Promotor :
Prof. dr. Mia Hubert

Proefschrift ingediend tot het
behalen van de graad van Doctor
in de Wetenschappen

juli 2007

Kasteelpark Arenberg 11, B-3001 Heverlee (Belgium)

Wettelijk depot D/2007/10.705/36
ISBN 978-90-8649-114-8

Voorwoord

Graag bedank ik een aantal mensen die cruciaal hebben bijgedragen aan het welslagen van dit werk.

In de eerste plaats natuurlijk promotor Mia Hubert. Zij gaf me de kans dit onderzoek aan te vatten. De vrijheid en ruimte die ik hierbij kreeg, heb ik bijzonder geapprecieerd. Haar expertise omtrent robuuste statistiek is onmiskenbaar en haar enthousiasme aanstekelijk: ook ik kan geen data meer bestuderen zonder onmiddellijk naar uitschieters te zoeken.

Johan Suykens introduceerde me in de wereld der kernel methodes. De combinatie met robuustheid verliep bijwijlen moeizaam, maar de discussies waren altijd interessant. Deze thesis toont dan ook niet alleen inhoudelijk aan dat kleinste kwadraten en robuustheid wèl kunnen samengaan.

Andreas Christmann is een van de weinigen met een uitgebreide kennis over zowel kernel methodes als robuuste statistiek. Zijn nauwkeurige analyses van mijn werk waren dan ook van goudwaarde. Ook de andere leden van de jury wil ik bedanken voor de nuttige suggesties en correcties: Jan Beirlant, Pavel Čížek, Bart De Moor en Irène Gijbels.

Johan Van Horebeek nodigde me uit naar het CIMAT in Mexico, een onvergetelijke ervaring. Niet alleen was dit een unieke kans om een andere (onderzoeks)cultuur te leren kennen; ook op het gebied van *robustes* bleek mijn verblijf een zeer leerrijke maand.

Peter Rousseeuw en Steve Portnoy wil ik bedanken voor hun bijdrage aan de respectievelijke publicaties die samen verwezenlijkt werden; Jurgen Vercauteren voor de interessante samenwerking rond zijn AIDS-onderzoek.

Uiteraard is ook een goede werksfeer onontbeerlijk om een doctoraat succesvol af te ronden. Daaraan was op het UCS gelukkig geen gebrek. Een dikke merci aan alle (ex-)collega's. Het wordt wennen om 's middags niet meer met een leuke bende in de alma te eten.

Voor de nodige ontspanning kon ik altijd bij mijn vrienden terecht. Squash, tennis, terrasje, voetbal met bijhorende derde helft: ideaal om het hoofd even leeg te maken en er daarna weer fris tegenaan te gaan.

Mijn ouders zorgden voor een warme thuis en onvoorwaardelijke steun in alles wat ik deed. Katrien, als geen ander heb jij met me meegeleefd tijdens dit doctoraat. Altijd stond je klaar met steun en een luisterend oor. Bedankt voor alle mooie momenten tot nu, en op naar de vele die nog zullen volgen.

Table of contents

Introduction

In statistics, classical methods often heavily rely on assumptions which can not always be met in practice. For instance, it is often assumed that the data are generated from a specific underlying distribution. And even if the model assumptions are distribution-free, most methods assume that the sample contains independent and identically distributed observations. However, when outliers are present such methods can perform very poorly. Robust statistics seeks to provide methods that are not unlimitedly affected by outliers. The goal is to learn the structure of the majority of the data, even if a minority of observations disturbs the pattern.

In this work robustness is studied in two settings: regression and Principal Component Analysis (PCA). Regression analysis models the relationship between a response variable and a set of explanatory variables (also called covariates). Interest lies in the conditional distribution of the response, conditional on values of the explanatory variables. One can concentrate on estimating certain aspects of this conditional distribution, e.g. the mean, leading to least squares regression.

However, in some applications a more detailed description beyond the mean might be useful. Quantile regression [Koenker, 2005] aims at estimating all conditional quantiles, thus fully characterizing the conditional distribution. Assuming a linear relationship between response and covariates, linear quantile regression can be performed using an L_1 loss function [Koenker and Bassett, 1978]. Although this is less sensitive to outlying observations than a linear least squares method, robustness problems still appear. A more robust approach was proposed by Rousseeuw and Hubert [1999], named deepest regression. In Chapter 1 we shortly review both methods and their properties. Next we consider the difficulties appearing when right-censoring is present. This means that the response value is not always exactly measured for each observation, but only a lower limit can be obtained. This frequently occurs in medicine, for instance when patients are taken into a study but leave before the final results are measured, and in economics, for instance in auction type sales when the sales price is not yet known but a current bid is running. A quantile regression algorithm dealing with such censoring was proposed by Portnoy [2003] for the L_1 estimator. In Chapter 1 we apply similar ideas for the deepest regression estimator. We derive the new optimization criterion and propose a grid algorithm to perform the computations. Robustness is shown in a small simulation study and on two data examples.

The second major framework of this dissertation concerns Principal Component Analysis (PCA). This is a technique designed to reduce the dimension of multivariate data. These days data sets sometimes contain hundreds or thousands of variables, for instance in chemometrics, where measurements for several samples are taken at a very large amount of different wavelenghts. Also in genetics such high-dimensional data often appears, when information about many genes is gathered for each patient. In such cases it is sometimes preferable to reduce the huge number of variables. In traditional linear PCA this reduction is obtained by a projection onto a linear lower dimensional subspace, spanned by the eigenvectors of the classical covariance matrix. However, these eigenvectors are again very sensitive to outlying observations in the data. A more robust procedure called ROBPCA was proposed by Hubert et al. [2005]. In Chapter 2 we give a short description of this method. Next we analyze some theoretical properties of the underlying robust covariance estimator. We obtain its asymptotic efficiency and make a comparison to some other robust covariance estimators. We also provide some insight in the robustness of ROBPCA by calculating its influence function. The concept of influence function was introduced by Hampel et al. [1986] and plays an important role in Chapter 2 but also in the chapters thereafter.

Definition 1 *Given a statistical functional T mapping a distribution P onto $T(P)$. Consider the contaminated distribution*

$$P_{\epsilon,z} = (1-\epsilon)P + \epsilon\Delta_z$$

for small enough ϵ. The distribution Δ_z is the Dirac distribution which puts all probability mass at the point z. Then the influence function of T at the distribution P is defined as

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

The influence function measures the effect on an estimator when making an infinitesimally small change in the distribution. Of specific interest is the supremum of this function. If the influence function is unbounded, then the influence of outliers can be arbitrary large. For a robust method this function should thus be bounded and preferably as small as possible. In Chapter 2 we prove that the influence function of ROBPCA is indeed bounded. We also consider the extension towards RSIMPLS [Hubert and Vanden Branden, 2003], a method combining ideas from robust PCA and regression. A similar analysis again shows a bounded influence function and gives some insight in the robustness of this method.

Both Chapters 1 and 2 assume a linear underlying structure. In practice more complicated structures can occur as well. Chapters 3, 4 and 5 fit in the framework of kernel methods [Schölkopf and Smola, 2002, Shawe-Taylor and Cristianini, 2004]. This is a broadly applicable methodology to transfer ideas from linear multivariate applications to more complex situations. We will use the following notations.

Definition 2 *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 $u, v \in \mathcal{X}$ we have*

$$K(u,v) \;=\; \langle \Phi(u), \Phi(v) \rangle \,.$$

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

Some well known kernels in case $\mathcal{X} \subseteq \mathbb{R}^d$ are the linear kernel

$$K(u,v) = u^t v,$$

the polynomial kernel of degree $p > 0$ with offset $\tau > 0$

$$K(u,v) = (u^t v + \tau)^p,$$

and the RBF kernel with bandwidth $\sigma > 0$

$$K(u,v) = e^{||u-v||^2/\sigma^2},$$

but many more types of kernels exist.

The general idea behind kernel methods is to apply a linear method in the feature space $\mathcal{H}$ rather than in the original input space $\mathcal{X}$. If this linear method in $\mathcal{H}$ can be formulated in terms of inner products $\langle \Phi(u), \Phi(v) \rangle$ only, one can use the kernel function and evaluate $K(u,v)$ instead. This allows data analysis in a potentially very high dimensional space $\mathcal{H}$ without having to compute or even knowing the explicit feature maps. A typical application of kernel methods is a setting where the data consists of objects rather than numerical vectors. In Chapter 5 the case of string analysis is considered. If we want to analyze text strings, we have to convert them into numerical vectors first. A way to achieve this can be to count all possible substrings. However, the size of such vectors can easily be several billions. Defining the corresponding kernel and applying kernel methodology on the other hand avoids the explicit computation of these high-dimensional vectors, making such an analysis computationally possible. Secondly kernels can be very useful in classical multivariate analysis fitting non linear models. Using a polynomial kernel for instance, an explicit power expansion of all variables is avoided, whereas these powers are still taken into the analysis. Using an RBF kernel, one can even work with implicit (even unknown!) feature vectors, modelling data in a semi-parametric way.

This kernel framework applied in a regression setting together with ideas from convex optimization and regularization, leads to Kernel Based Regression (KBR). Christmann and Steinwart [2006] proved that the influence function of Least Squares KBR (LS-KBR) is unbounded, in contrast to KBR using a loss function with a bounded first derivative. Suykens et al. [2002a] proposed a reweighted LS-KBR method in order to improve the robustness of LS-KBR, at the same time retaining the least squares methodology and its benefits. In Chapter 3 we investigate some theoretical properties of this reweighted LS-KBR method. We derive the influence function of $k-$step reweighted LS-KBR. Under some restrictions we analyze the behavior of this series of influence functions as we keep on reweighting in an iterative way ($k \to \infty$). An important result states that the influence function of iteratively reweighted LS-KBR is bounded if the kernel is bounded, and if the weight function $w(r)$, with $r \in \mathbb{R}$ the residual, can be written as $w(r) = \psi(r)/r$ with ψ bounded but increasing. This condition is not trivial, since it is not satisfied by some popular weight functions, i.e. Hampel's suggestion. We propose logistic weights as it is a smooth weight function fitting our conditions perfectly.

Under some specific model assumptions we are also able to analyze the convergence of this iterative reweighting showing quite fast results. We conclude Chapter 3 by linking the influence function to concepts of stability [Poggio et al., 2004]. This way we motivate that reweighting is not only useful to reduce effects due to outliers, but also to deal with heavy-tailed noise situations.

In Chapter 4 we continue linking the influence function of KBR and its reweighted version to some other concepts. We estimate the influence function based on a sample and use these results to construct pointwise confidence intervals. Secondly, we consider the influence function as an asymptotic leave-one-out criterion. We construct a fast and robust model selection criterion to select some of the hyperparameters in play, i.e. the regularization parameter and possible kernel parameters such as the bandwidth σ in case of a RBF kernel.

In Chapter 5 we return to the PCA setting. Here as well kernels can be incorporated in order to detect more complex structures. The influence function of Kernel PCA (KPCA, [Schölkopf et al., 1998]) is obtained. Just like in the regression case, bounded kernels lead to a bounded influence function, whereas KPCA with an unbounded kernel possibly leads to arbitrary large effects from outliers. We propose a new method, Spherical KPCA, to perform robust KPCA with any type of kernel. It is an extension of Spherical PCA [Locantore et al., 1999] to a feature space $\mathcal{H}$ only using the kernel. Finally we construct a diagnostic tool based on the influence function and Spherical KPCA to detect influential observations in a sample.

List of publications

Debruyne, M and Hubert, M. (2004)
Robust regression quantiles with censored data.
Proceedings in Computational Statistics, editor J. Antoch, p. 887-893. Springer-Verlag, Heidelberg.

Rousseeuw, P.J., Debruyne, M., Engelen, S. and Hubert, M. (2006)
Robustness and outlier detection in chemometrics.
Critical Reviews in Analytical Chemistry, 36, p. 221-242.

Debruyne, M., Hubert, M., Portnoy, S. and Vanden Branden, K. (2006)
Censored depth quantiles.
Computational Statistics and Data Analysis, accepted for publication.

Debruyne, M. and Hubert, M. (2006)
The influence function of Stahel-Donoho type methods for robust covariance estimation and PCA.
Technical Report K.U.Leuven, Section of Statistics, TR-06-01,
available at http://wis.kuleuven.be/stat/publications.

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

Debruyne, M., Hubert, M. and Suykens J.A.K. (2007)
Model selection for kernel regression using the influence function,
submitted.

Van Horebeek J., Debruyne M. and Hubert, M. (2007)
Influential observations in Kernel PCA,
in preparation.

Vercauteren, J., Deforche, K., Theys, K., Debruyne, M., Duque, J.L., Peres, S., Carvalho, A.P., Mansinho, K., Vandamme, A.-M. and Camacho, R. (2007)
The incidence of Multidrug and Class Resistance in HIV-1 infected patients is decreasing over time (2001-2006),
submitted.

Chapter 1

Censored depth quantiles

1.1 Introduction

Since its introduction by Koenker and Bassett [1978], quantile regression has become more and more popular. The possibility to estimate the entire conditional distribution, instead of only the conditional mean as in e.g. ordinary least squares regression, has proven to be advantageous in many applications. In recent years, quantile regression has been extended to many possible settings, such as non-linear and non-parametric regression, time series, etc. [Koenker, 2005]. In this chapter we want to focus on linear quantile regression with right censored observations. These are observations for which the true value of the response variable is not measured, but only a lower limit is given. This kind of data is frequently encountered in many domains. In medicine for example, when time until healing is measured, patients might not yet be healed when finishing the study. In that case, the exact healing time is not observed, but we do know it is at least the time the patient spent in the study. Also in economics, censoring can be an issue. The first example we will give in Section 1.6, considers sales prices in auction type sales. When a bid is running on an object, but the deadline is not reached yet, this is a right censored observation. The true sales price is not known, but we do know it will be at least the current bid.

Let us first describe the model under consideration. We want to estimate the conditional quantiles of a real random variable Y given $x \in \mathbb{R}^{d+1}$ where we take the first component $x_1 = 1$. However, also models through the origin can be considered. We suppose throughout that these conditional quantiles denoted by $Q_\tau(Y|x)$ are linear in x. So for $\tau \in (0,1)$

$$Q_\tau(Y|x) = \inf\{y : P(Y \leq y|X = x) = \tau\} = x^t\beta(\tau), \tag{1.1}$$

for $\beta(\tau) = (\beta_0(\tau), \beta_1(\tau), \ldots, \beta_{d+1}(\tau))^t$ the τth regression quantile. These assumptions correspond to the problem of estimating the regression quantiles in a linear, possible heterogeneous, regression setting with response variable Y and covariates x. Especially in the case of heterogeneous data these quantiles offer an overall view on the data as they catch much more the variability present in the sample when τ varies over the interval $(0,1)$.

In Koenker and Bassett [1978], a consistent estimator $\hat{\beta}(\tau)$ for $\beta(\tau)$ has been defined. We will however assume that the observations can be right censored. This implies that instead of observing the true response y_i, we observe $\tilde{y}_i = \min(y_i, c_i)$ for a set of n covariates $x_i \in \mathbb{R}^{d+1}$, where c_i is the censoring time. A censoring indicator $\Delta_i = I(y_i \leq c_i)$, with I the indicator function, denotes whether observation i is censored ($\Delta_i = 0$) or observed ($\Delta_i = 1$). We assume independence between the response variable and the censoring times, conditionally on the covariates x. The censoring times are however allowed to depend on the covariates, contrary to most other censored regression methods, e.g. Honoré et al. [2002], assuming censoring at random.

In Portnoy [2003] a reweighting scheme based on the Kaplan-Meier estimator has been developed for adapting Koenker and Bassetts L_1-quantiles to the censored case. In Section 1.2, we will shortly review this L_1-methodology and its extension towards censoring. A serious drawback of these estimators is the lack of robustness. Although they are resistant to vertical outliers, i.e. observations that are outlying in y given x, L_1-quantiles can be heavily influenced by leverage points, i.e. observations outlying in x-space.

A more robust quantile estimator has been proposed in Rousseeuw and Hubert [1999], based on the concept of regression depth. The main goal of this chapter is to extend these depth quantiles to the framework of censored observations, using the same reweighting scheme as Portnoy [2003]. The most important ideas are outlined in Section 1.3. A detailed description of the algorithm can be found in Section 1.4. We illustrate our method with a simulation study in Section 1.5 and with two real data examples in Section 1.6.

1.2 L_1-quantiles

A consistent estimator of the vector $\beta(\tau)$ was proposed in Koenker and Bassett [1978] as the solution of

$$\underset{\beta \in \mathbb{R}^{d+1}}{\operatorname{argmin}} \sum_{i=1}^{n} \rho_\tau(y_i - x_i^t \beta), \tag{1.2}$$

for a sample $(x_i^t, y_i)^t \in \mathbb{R}^{d+2}$ where $i = 1, \ldots, n$ and $\rho_\tau(u) = u(\tau - I(u < 0))$. When τ equals $\frac{1}{2}$, the function $\rho_{\frac{1}{2}}(u)$ reduces to the half of the absolute value. Thus for the special case of the median, this estimator corresponds to the L_1-estimator. Therefore the solution for general τ will be denoted by L_1-quantiles further on. The asymptotic distribution of $\sqrt{n}(\hat{\beta}(\tau) - \beta(\tau))$ has also been derived in Koenker and Bassett [1978]. In the same article it was shown that L_1-quantiles only depend on the sign of the residuals and not on the exact value of the response variable.

This is a very important observation when it comes to censored data. First of all, it allows an easy start for the lowest quantiles. Remember, the exact value y_i of the response variable is unknown for a censored observation, but we do know a lower limit c_i. Thus, as long as c_i lies above the τth regression quantile, y_i certainly will. Hence the residual $y_i - x_i^t \beta(\tau)$ will be positive no matter the true value of y_i. Therefore we can just use ordinary quantile regression for the smallest quantiles.

This changes of course as τ increases. Sooner or later a censored observation will have a negative residual $c_i - x_i^t \beta(\tau)$. Then the true residual $y_i - x_i^t \beta(\tau)$ might be either negative or positive, and there is no way of knowing the sign for sure. We will call such observations *crossed* from now on. The quantile at which the ith censored observation is crossed, will be denoted $\hat{\tau}_i$, thus

$$c_i - x_i^t \beta(\hat{\tau}_i) \geq 0 \qquad \text{and} \qquad c_i - x_i^t \beta(\tau) \leq 0 \;\text{ for all }\; \tau > \hat{\tau}_i.$$

The crucial idea explained in Portnoy [2003] is to estimate the probabilities of crossed censored observations having a positive respectively negative residual, using these estimates as weights further on. More precisely, such a crossed censored observation is split into two new pseudo-observations, one at (x_i, c_i) with weight $w_i(\tau) \approx P(y_i - x_i^t \beta(\tau) \leq 0)$ and one at (x_i, ∞) with weight $1 - w_i(\tau)$. Finally it is noted that the weights $w_i(\tau)$ can easily be found in quantile regression, since the number $1 - \hat{\tau}_i$ is an estimate of the censoring probability

$P(y_i > c_i)$. Thus we can define

$$w_i(\tau) = \frac{\tau - \hat{\tau}_i}{1 - \hat{\tau}_i} \qquad \tau > \hat{\tau}_i. \tag{1.3}$$

This leads to the following method to deal with censored observations in linear L_1-quantile regression:

- As long as no censored observations are crossed, use ordinary quantile regression as in (1.2).

- When the ith censored observation is crossed at the τth regression quantile, store this value as $\hat{\tau}_i = \tau$.

- When estimating the τth regression quantile and censored observations have been crossed, optimize a weighted version of (1.2):

$$\begin{aligned} \underset{\beta \in \mathbb{R}^{d+1}}{\operatorname{argmin}} \Big\{ & \sum_{i \in K_\tau^c} \rho_\tau(\tilde{y}_i - x_i^t \beta) \\ & + \sum_{i \in K_\tau} [w_i(\tau)\rho_\tau(\tilde{y}_i - x_i^t\beta) + (1 - w_i(\tau))\rho_\tau(y^* - x_i^t\beta)] \Big\}, \end{aligned} \tag{1.4}$$

  where the set K_τ represents the crossed and censored observations at τ and K_τ^c is the complement of K_τ. The weights $w_i(\tau)$ are as defined in (1.3). The number y^* is any value sufficiently large to exceed all $\{x_i^t\beta\}$.

To compute the regression quantile function in practice, a sequence of breakpoints $\{\tau_1^*, \ldots, \tau_L^*\}$ is defined such that $\hat{\beta}(\tau)$ is piecewise constant between these breakpoints. By simplex pivoting we can move from one breakpoint to another, using the subgradients of (1.4). Luckily, the resulting gradient conditions are linear in τ, making this linear programming approach possible. A detailed description of the algorithm can be found in Portnoy [2003], together with some consistency results (see also Neocleous et al. [2006]).

1.3 Depth quantiles

As already mentioned before, L_1-quantiles only depend on the sign of the residual, not on the exact value of the response variable. Therefore one can immediately see that observations with outlying y-value will not have a large impact on the estimates. Contrary to for example linear least squares regression, L_1-quantiles can resist vertical outliers. However, L_1-quantiles are sensitive to data points outlying in x-space. This is among others reflected in its breakdown value, which equals 0. Since the break down value represents the smallest percentage of contamination needed to completely destroy the estimator, this means that the slightest amount of contamination can have a disastrous effect on the resulting estimates. A more robust method was proposed in Rousseeuw and Hubert [1999], based on the concept of regression depth.

1.3.1 Definition

The regression depth of a hyperplane β with respect to a sample
$Z_n = \{(x_i^t, y_i)^t \in \mathbb{R}^{d+2}\}$ is defined as

$$rdepth(\beta, Z_n) = \min_{\lambda \in \mathbb{R}^{d+1}} \left(\#\{x_i : \text{sign}(y_i - x_i^t\beta)) \neq \text{sign}(x_i^t\lambda)\}\right),$$

with $\text{sign}(u) = -1$ if $u < 0$, $\text{sign}(u) = 0$ if $u = 0$ and $\text{sign}(u) = 1$ if $u > 0$. It has a nice geometrical interpretation as it represents the smallest number of observations one has to pass in order to turn the hyperplane β into vertical position. As such, regression depth gives an indication of how well the data surround the hyperplane.

The maximal depth (or deepest regression) estimator $\hat{\beta}(\frac{1}{2})$ is defined as

$$\hat{\beta}(\frac{1}{2}) = \underset{\beta \in \mathbb{R}^{d+1}}{\text{argmax}}\{rdepth(\beta, Z_n)\},$$

or equivalently

$$\hat{\beta}(\frac{1}{2}) = \underset{\beta \in \mathbb{R}^{d+1}}{\text{argmax}} \min_{\lambda \in \mathbb{R}^{d+1}} \left(\#\{x_i : \text{sign}(y_i - x_i^t\beta)) \neq \text{sign}(x_i^t\lambda)\}\right), \tag{1.5}$$

Properties of this estimator have been studied in Rousseeuw and Hubert [1999], Van Aelst et al. [2002], Van Aelst and Rousseeuw [2000] and Bai and He [2000]. In the latter paper it is proven that deepest regression is a consistent estimator of the median regression quantile $\beta(\frac{1}{2})$. The first papers show that the breakdown value of the deepest regression is around 33%. This means that theoretically smaller percentages of outliers cannot completely destroy the fit. Practical results show that the method can indeed easily resist at least up to 20% of outliers (vertical outliers as well as leverage points). Note that this is a big difference compared to the L_1-estimator, which has breakdown value 0%.

The deepest regression estimator can be extended to the regression quantile setting by introducing the idea behind the function ρ_τ in definition (1.2) [Rousseeuw and Hubert, 1999]. This ρ_τ function can be seen as a weight function where a weight τ is given to positive residuals, and a weight $1-\tau$ to negative residuals. Let $\Psi_\tau(u) = \tau - I(u < 0)$. Then the τth regression depth quantile $\hat{\beta}(\tau)$ is defined as that value $\beta \in \mathbb{R}^{d+1}$ for which

$$\inf_{\lambda \in \mathbb{R}^{d+1}} \sum_{i=1}^{n} \Psi_\tau(y_i - x_i'\beta)\,\mathrm{sign}(x_i'\lambda)$$

is maximized. The case $\tau = 0.5$ coincides with the conditional median as defined in (1.5). In Bai and He [2000] the $\sqrt{n}$-consistency of $\hat{\beta}(\tau)$ has been shown and the limiting distribution of $\sqrt{n}(\hat{\beta}(\tau) - \beta(\tau))$ has been characterized.

An extension of regression depth to the case of censored data is proposed in Park and Hwang [2003]. Their approach however only covers the special case $\tau = 0.5$. We introduce a general extension to censored data for all quantiles in the next section.

1.3.2 Censored depth quantiles

Similarly to the method defined above for the L$_1$-regression quantiles, we introduce the reweighting scheme for the depth quantiles. Since depth quantiles also depend on the sign of the residuals only, the same idea can be used as in the previous section. The only thing changing is the objective function (1.4). We replace this expression by the corresponding depth quantile objective function, defined as

$$\begin{aligned}\operatorname*{argmax}_{\beta \in \mathbb{R}^{d+1}} \inf_{\lambda \in R^{d+1}} \Big\{ &\sum_{i \in K_\tau^c} \Psi_\tau(\tilde{y}_i - x_i^t\beta)\,\mathrm{sign}(x_i^t\lambda) \\ &+ \sum_{i \in K_\tau} [w_i(\tau)\Psi_\tau(\tilde{y}_i - x_i^t\beta)\,\mathrm{sign}(x_i^t\lambda) + \tau(1 - w_i(\tau))\,\mathrm{sign}(x_i^t\lambda)] \Big\},\end{aligned} \tag{1.6}$$

where the set K_τ and the weights $w_i(\tau)$ are as defined in (1.3) and (1.4).

1.4 Computation

Although the idea for censored depth quantiles is the same as for L_1-quantiles, the computation has to be done differently. Working with breakpoints τ_j^* is impossible, since gradient conditions cannot be obtained for depth quantiles.

As such, the linear programming algorithm from Section 1.2 cannot be used. We now introduce another algorithm, using a grid $\{t_j : 0 < t_1 < t_2 < \ldots < t_M < 1\}$ where M is the total number of grid points. Each censored observation receives a weight $w_i(\tau) = 1$ as long as it is not crossed by $\beta(\tau)$, i.e. $c_i - x_i^t\beta(\tau) > 0$. Once the censored observation c_i is crossed, say at grid point t_j, a weight $w_i(\tau)$ that varies along the grid is assigned to that observation c_i and a weight $1 - w_i(\tau)$ is placed at infinity. This weight $w_i(\tau)$ is defined as

$$w_i(\tau) = \frac{\tau - t_j}{1 - t_j},$$

for grid points $\tau > t_j$, corresponding to (1.3).

1.4.1 Algorithm

We will now in detail list all the steps of the algorithm for obtaining the censored depth quantiles. The MATLAB routine (cdq.m) is part of LIBRA, Matlab Library for Robust Analysis [Verboven and Hubert, 2005], freely available at http://wis.kuleuven.be/stat/robust.html.

STEP 1 Choose a set of grid points $\{0 < t_1 < \ldots < t_M < 1\}$.
Estimate the t_1th regression quantile using the regression depth quantile for uncensored data. Crossed censored observations can be ignored since they almost do not contain any information, if t_1 is small enough.

STEP 2 Suppose we have estimated the t_lth regression quantile $\hat{\beta}(t_l)$. Then we also know the set of crossed censored observations $K_{t_l} = \{(x_i, c_i) : c_i - x_i^t\hat{\beta}(t_l) \leq 0\}$. For each of these crossed censored observations a number $\hat{\tau}_i$ has been given following equation (1.8) that will be explained in step 3 of the algorithm. The according weight is

$$w_i(\tau) = \frac{\tau - \hat{\tau}_i}{1 - \hat{\tau}_i}.$$

STEP 3 Estimate the t_{l+1}th regression quantile using expression (1.6), which in practice can be implemented as follows:

$$\begin{aligned}\hat{\beta}(t_{l+1}) = \underset{\beta \in \mathbb{R}^{d+1}}{\operatorname{argmax}} \inf_{\lambda \in \mathbb{R}^{d+1}} \Big(& t_{l+1} \, \#\{\tilde{y}_i \notin K_{t_l} : (r_i(\beta) > 0, x_i^t \lambda < 0)\} \\ & + (1 - t_{l+1}) \, \#\{\tilde{y}_i \notin K_{t_l} : (r_i(\beta) < 0, x_i^t \lambda > 0)\} \\ & + t_{l+1} \, w_i(t_{l+1}) \ \ \#\{\tilde{y}_i \in K_{t_l} : (r_i(\beta) > 0, x_i^t \lambda < 0)\} \\ & + (1 - t_{l+1}) w_i(t_{l+1}) \, \#\{\tilde{y}_i \in K_{t_l} : (r_i(\beta) < 0, x_i^t \lambda > 0)\} \\ & + t_{l+1} \, (1 - w_i(t_{l+1})) \ \ \#\{\tilde{y}_i \in K_{t_l} : (x_i^t \lambda < 0)\} \Big). \quad (1.7)\end{aligned}$$

The maximization is performed on a random grid of β and λ vectors and will be further explained in Section 1.4.2.

Consider the set $K_{\tau_{l+1}} = \{(x_i, c_i) : c_i - x_i^t \hat{\beta}(t_{l+1}) \leq 0\}$.

IF $K_{t_{l+1}} = K_{t_l}$,

the current estimate $\hat{\beta}(t_{l+1})$ was found using the correct weights and is therefore a correct solution.

IF $K_{t_{l+1}} \neq K_{t_l}$,

then the weights should be changed. Observations in $K_{t_l} \backslash K_{t_{l+1}}$ are censored observations that were crossed but are not anymore. These receive weight 1 again. Observations from $K_{t_{l+1}} \backslash K_{t_l}$ are censored observations that are crossed just now, during the transition from t_l to t_{l+1}. We define the number

$$\hat{\tau}_i = t_l, \quad (1.8)$$

for each of these observations. Their weight is then

$$w_i(\tau) = \frac{\tau - \hat{\tau}_i}{1 - \hat{\tau}_i} = \frac{\tau - t_l}{1 - t_l}.$$

The remaining weight $1 - w_i(\tau)$ is assigned to a pseudo-observation arbitrarily far away. Thus we find a new set of crossed censored observations. The regression quantile $\hat{\beta}_{\tau_{l+1}}$ is then recomputed with this new set of weights.

We repeat this step until we find an estimate for which the weights remain the same, or until a predefined number of iterations is exceeded.

STEP 4 The algorithm stops when we have dealt with the last grid point τ_M, or when all observations with a positive residual are censored.

□

Note that at any grid point t_{l+1}, step 3 of the algorithm is repeated until a stable solution is found with $K_{t_l} = K_{t_{l+1}}$. The existence of such a stable solution is not guaranteed. However, this situation rarely occurred in our examples and simulations. Moreover, when this happened it was always for a very restricted area of τ values. Thus, by changing the problematic t_{l+1} value by a few thousandths, a stable solution can usually be found.

1.4.2 Optimization

One part of the algorithm still needs some explanation, i.e. the optimization of (1.7). We propose two methods.

A straightforward approach is to define a set of N hyperplanes at the start of the algorithm. We take $B = \{\beta_j,\ j = 1, \dots, N\}$ with each β_j a hyperplane through $d+1$ randomly chosen data points. This way affine equivariance of the depth quantiles is retained. We can then maximize the objective function in (1.7) over this finite set B:

$$\hat{\beta}(t_{l+1}) = \operatorname*{argmax}_{\beta_j \in B} \inf_{\beta_k \in B} \Big(\text{objective function}\Big).$$

This approach was also used in Adrover et al. [2004] to compute regression quantiles in the uncensored case. Note that it is usually not necessary to scan all possibilities. Take for example $\beta_j \in B$, and suppose we kept track of the current maximum over $\beta_i \in B$, $i = 1, \dots, j-1$. Then we do not really need to compute the infimum over all $\beta_k \in B$. As soon as we find a β_k such that the objective function is smaller than the current maximum, the infimum will certainly be smaller. Thus we can immediately discard β_j and proceed with β_{j+1}. As noted in Adrover et al. [2004], this leads to roughly $O(N \log(N))$ calculations to find $\hat{\beta}(t_l)$. Since we have M grid points, we roughly need $O(MN \log(N))$ calculations.

We propose a faster approach, explicitly making use of the iterative character of our algorithm. Suppose we want to compute the regression quantile at a grid point t_{l+1}. Then we already have an estimate $\hat{\beta}(t_l)$ of the regression quantile at t_l. Since t_l and t_{l+1} will not differ a lot, we can expect $\hat{\beta}(t_{l+1})$ to be close to $\hat{\beta}(t_l)$. We therefore suggest to perform the maximization in (1.7) over a set $B(\hat{\beta}(t_l))$ of N^* hyperplanes close to $\hat{\beta}(t_l)$:

$$\hat{\beta}(t_{l+1}) = \underset{\beta_j \in B(\hat{\beta}(t_l))}{\text{argmax}} \inf_{\beta_k \in B} \Big(\text{objective function}\Big).$$

The complexity of this algorithm is very roughly $O(MN^* \log(N))$. The gain in speed comes from the fact that N^* can usually be chosen much smaller than N. The set $B(\hat{\beta}(t_l))$ of hyperplanes close to $\hat{\beta}(t_l)$ can be obtained in several ways. We take hyperplanes that have d observations in common with $\hat{\beta}(t_l)$. Such a set can be constructed very fast using updating techniques.

1.5 Simulation study

We compare three algorithms: the L_1-estimator using the package *crq* in R [Portnoy, 2003], the depth estimator using the basic algorithm and the depth estimator using the faster updating algorithm. The setting for our simulations is as follows: let ϵ be the percentage contamination and n the sample size with $m = \text{round}(n\epsilon)$, then we generated $n - m$ datapoints $(x_i, \tilde{y}_i)$ as follows:

- $x_i = (1, x_{i2}, \ldots, x_{ip})^t$ with each $x_{ij} \sim N(0, 1)$.
- $y_i = x_{i2} + e_i$ with $e_i \sim N(0, 1)$.
- $c_i = 0.8x_{i2} + b + f_i$ with $f_i \sim N(0, 1)$. We considered different values for b, controlling the amount of censored observations in the data. All values reported are for $b = 1$, corresponding to roughly 20% of censored data points. Other percentages yielded similar results, at least up to 40%.
- $\tilde{y}_i = \min(c_i, y_i)$.

Note that the censoring c_i depends on the covariates and on the response variable. In other methods this is often not allowed, but the regression quantile approach only assumes conditional independence of c_i and y_i, given x_i.

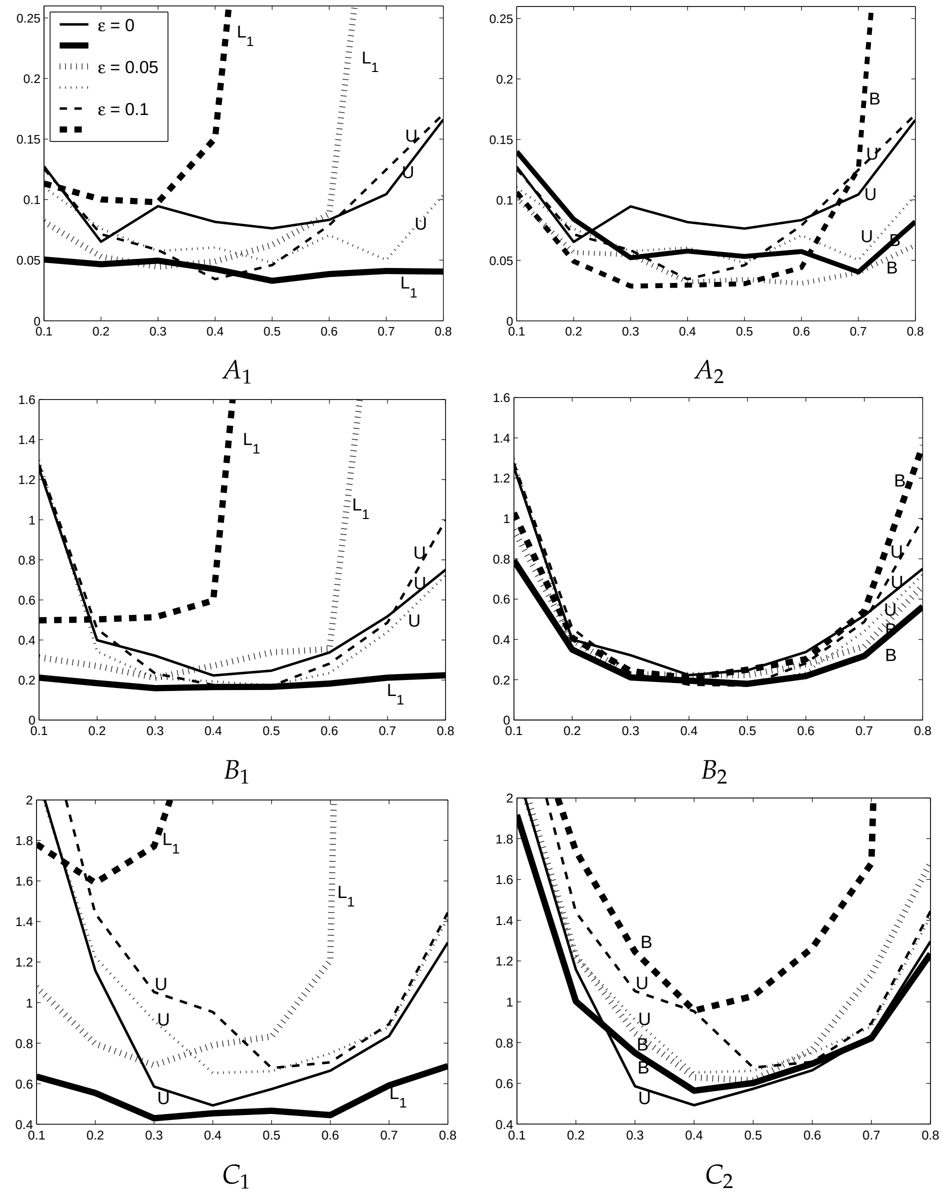
- We considered 2 cases: we took the m outliers all coinciding in (x_0, y_0) (point contamination), but we also distributed the m outliers around (x_0, y_0). Since there was no big difference in results, we only report the case of point contamination. The outlier location was taken equal to $((1, -5, 0, \ldots, 0)^t, 10)^t$. This is motivated by Adrover et al. [2002], where this setup appeared as the worst case scenario in a very similar simulation study for robust (but uncensored) quantile regression.

Each simulation consists of 50 replications. We report the median of the squared errors ($||\hat{\beta}(\tau) - \beta(\tau)||^2$) in the grid points $0.1, 0.2, \ldots, 0.8$. We considered sample sizes $n = 50, 100$, dimensions $d = 2, 5, 10$ and percentages of contamination $\epsilon = 0, 0.05, 0.1, 0.2$. We took $M = 20$ equally spaced grid points, although even $M = \sqrt{n}$ is probably sufficient, as already proposed in Portnoy [2003]. In the basic algorithm, we took $N = 500$. In the updating algorithm we chose $N = 500$ and $N^* = 100$.

Results for $n = 100$ are summarized in Figure 1.1. At the left side of the figure, we compare L_1-quantiles (thick lines) to depth quantiles (thin lines), for $\tau = 0.1, \ldots, 0.8$. Solid lines correspond to the case $\epsilon = 0$, dotted lines to $\epsilon = 0.05$ and dashed lines to $\epsilon = 0.1$. Plot A_1 shows results in 2 dimensions, B_1 in 5 and C_1 in 10 dimensions. It is clear that L_1-quantiles are superior when there is no contamination. The medians of squared errors are uniformly smaller. Especially for lower and higher quantiles the difference can be quite significant.

When contamination is added, the situation changes. With 5% of contamination, depth quantiles are more efficient from the 0.4-quantile on. At the 0.6-quantile, L_1 even breaks down, as it is completely attracted towards the outliers. In case of 10% of outliers, breakdown occurs already at the 0.4 quantile. Depth quantiles on the other hand, suffer very little from outliers. Their efficiencies remain about the same, no matter the value of $\epsilon \leq 0.1$, showing the robustness of our method.

At the right side of Figure 1.1, we compare the basic algorithm (thick lines) with the updating algorithm (thin lines). Otherwise the setting is the same as previously, with solid/dotted/dashed lines corresponding to $\epsilon = 0/0.05/0.1$ and A_2, B_2, C_2 plots for $d = 2$, $d = 5$ and $d = 10$ respectively. The difference between both algorithms is not too big. In lower dimensions, the naive approach is slightly better. In higher dimensions, the updating algorithm sometimes even improves on the basic algorithm. In any event, the updating algorithm is certainly not much worse than the basic approach. Note however that it only took about half as much time. In the updating algorithm we constructed sets of 100 hyperplanes having d point in common. We also tried this with hyperplanes having $d - 1$ points in common. The results were however almost the same, whereas the computation time slightly increased. Therefore we propose to stick to the algorithm replacing only one point at a time.



A_1 A_2

B_1 B_2

C_1 C_2

Figure 1.1 Summary of simulation results: median of squared errors for different quantiles and amounts of contamination. Left side plots: L_1 (thick) versus depth (thin lines, updating algorithm U). Right side: updating (U) versus basic (B) algorithm. Upper: $d = 2$, middle: $d = 5$, lower: $d = 10$. Solid lines: $\epsilon = 0$, dotted: $\epsilon = 0.05$, dashed: $\epsilon = 0.1$.

The effective computation time of the algorithm of course highly depends on the choice of parameters. Setting the parameters as in our simulations and examples, i.e. $N = 500$, $N^* = 100$ and 20 grid points, the matlab routine takes about 15 seconds on average, running Matlab 6.1 on a 2.4Ghz pc. For moderate-sized data sets decent results can thus be obtained in feasible time. However, the L_1-quantiles are obviously much faster: a similar analysis took us 0.02 seconds with the *crq* implementation for R 2.4.1.

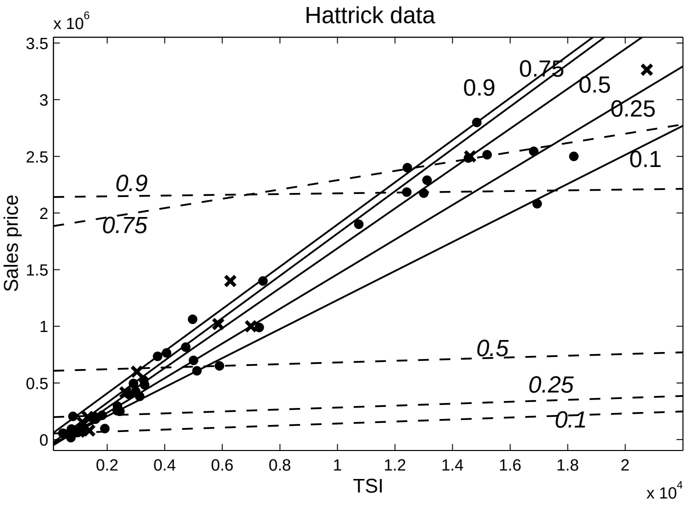


Figure 1.2 Hattrick data. Censored observations are shown as stars, uncensored ones as dots. One data point at $(37.5 * 10^4, 3, 38 * 10^6)$ (not visible on plot) destroys L_1-quantiles (dashed lines), but is resisted by the depth quantiles (solid lines).

1.6 Examples

1.6.1 Hattrick data

Hattrick is a free online soccer game at www.hattrick.org, played by over half a million of people worldwide. Each participant owns one team, consisting of virtual soccer players, fans, money etc. Just like in real soccer, all teams are put into divisions and play weekly competition games. The winner can promote to a higher division. The ultimate goal is to become one of the top teams of your country.

An important and interesting aspect of Hattrick is its economy. Players can be sold and bought on the transfer market. The transfer system works as follows. A team can put a player on the transfer list with a starting price and a certain deadline. Other teams can make a bid; the highest bid at the deadline buys the player. All bids are publicly available to all users.

In our study, we followed 90 players that were on the transfer list on May 15, 2005. On May 17, 2005, we stopped our study. At that point, 55 players were sold, so we know their true sales price. For 14 players, at least one bid was already made, but the deadline was not yet reached. These are censored observations: we do not know the true sales price, but we do know it will be at least the current highest bid. The remaining 21 players did not recieve a bid higher than their starting price, and thus were of no use in our study. This way, we obtained a data set of 69 observations, of which 14 are censored. As a covariate, we took the Total Skill Index (TSI) of each player, an in-game statistic measuring the quality of a player.

Our data contains one outlier: a player with a TSI of 374 600. Note that its sales price is relatively low: 3 381 000 euros. This is explained by the players' wage, which is closely related to their TSI. Moderate players with TSI around 5000 earn 4000 euro a week, better players with TSI around 20000 make about 15000. Since a team in the game typically has a budget of a few million euros, this difference is negligible. Our outlying player on the other hand has a wage of 299 496 euro a week. Although this player makes your team perform much better on the (virtual) pitch, his high wage becomes disadvantageous from an economical viewpoint. Therefore, the linear structure between TSI and market value is violated for these extremely good players.

The data is plotted in Figure 1.2. The outlier is not visible for aesthetic purposes, but was taken into account in our analysis. The dashed lines are the 0.1, 0.25, 0.5, 0.75 and 0.9 regression quantiles, estimated by *crq*, the method from Portnoy [2003] using the L_1-quantiles. As one can see, the outlier has a huge effect. The estimates clearly make no sense.

The depth quantiles are plotted as solid lines. They provide quite a nice view of the linear and heteroscedastic nature of the data. The effect of the outlier is minimal, showing the robustness of our approach. Also note that the outlier is not outlying in y direction. Furthermore, its L_1-residual is not outlying compared to the other residuals, since the L_1-quantiles are completely tilted towards the outlier. Thus our extremely good player can only be detected in x-space. In simple regression as in this example, this is of course very easy.

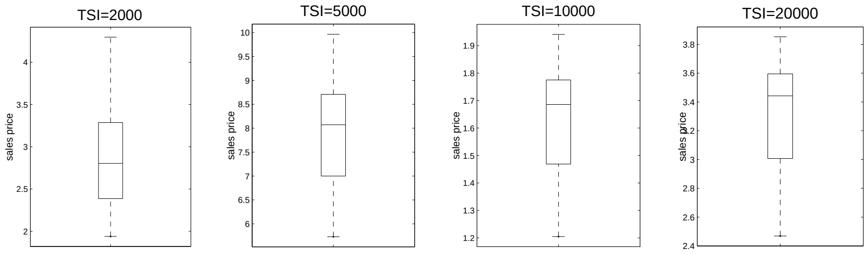


Figure 1.3 Boxplots of conditional distribution of sales price given TSI= 2000, 5000, 10000 and 20000.

However, in a higher dimensional x-space, outlier detection might be far from trivial. In that case, depth quantiles provide a way out.

An interesting feature of quantile regression is that one can visualize the entire conditional distribution at a certain point in $x-$space. This is shown in Figure 1.3 by means of boxplots at four different values of TSI: 2000, 5000, 10000 and 20000. Note the evolution from right skewed to left skewed as the value of TSI increases. This might reflect that people are more careful when dealing with more money. When a team with a budget of a few million euros wants to buy a player worth around 30 000, he might easily pay 50 000. When buying one worth around 3 000 000, he will probably make more effort to search for a relative good buy.

As a final note we should mention that the proposed model is probably not very optimal. Due to the rightskewness of both variables, a log transform would be a logical option. In that case the effect of the outlier is reduced, making the difference between L_1-quantiles and depth quantiles rather small.

1.6.2 Granule data

Our second example concerns a process in pharmacy called fluidized bed granulation. The data was taken from Rambali et al. [2003]. In an experimental design, they studied the granulation process on a fluidized bed in a semi-full scale (30 kg batch). There are 30 observations of which 8 are censored, because the process conditions were too bad to determine the granule size correctly.

In an empirical model they consider four variables: airflow rate, inlet air temperature, scaled spray rate and inlet air humidity. Their final proposal yields a 9 dimensional model including these four standardized variables and some interaction terms as independent variables. The response variable of interest is the observed granule size. Rambali et al. [2003] estimated the conditional median in two steps. First they obtained estimates for the value of the response variable for the censored observations. Then they used ordinary deepest regression on this completed data set.

We will now compare these results to the ones obtained by our algorithm. The right column of Table 1.1 shows the original results from Rambali et al. [2003]. The middle column shows the results from our censored depth quantiles. Note that the results are pretty similar. We also performed ordinary deepest regression on the data set without the censored observations (see the first column in Table 1.1). Some estimates are completely different, eg. the coefficients of A^2, S and AS. This shows that deleting censored observations can lead to severely biased estimates. It is absolutely necessary to use specific methods dealing with censored observations.

Also note that it would be nice to have some inference tools about the estimates, especially when comparing different methods like here. Portnoy [2003] uses a fast bootstrap scheme for the L_1-quantiles based on He and Hu [2002]. Unfortunately the discrete nature of the deepest regression objective function hampers extending this approach. Moreover bootstrap procedures themselves can suffer robustness problems, even when the underlying method is robust, see for instance Salibian-Barrera and Zamar [2002].

	Censoring ignored	CDQ	Rambali et al.
intercept	537.00	520.75	536.2
airflow rate (A)	-221.37	-309.32	-326.1
inlet air temperature (T)	-231.34	-166.50	-184.6
spray rate (S)	134.28	215.25	226.5
inlet air humidity (H)	35.90	28.40	30.60
A^2	52.63	197.60	164.4
T^2	172.34	123.75	145.4
AT	135.60	118.50	123.3
AS	4.47	-118.39	-110.7

Table 1.1 Granule data: parameter estimates of the conditional median.

1.7 Conclusion

We extended the idea of regression depth quantiles to data sets where censored observations are present. A grid algorithm was used and we introduced a relatively fast way of optimizing the objective function over this grid. A simulation study showed that this updating algorithm is particularly useful in higher dimensions, when similar efficiency as the naive approach is reached twice as fast. The simulation study revealed a loss in efficiency compared to the L_1-quantiles for normally distributed data, especially at lower and higher values of τ. When contamination was added on the other hand, depth quantiles performed better. They appear to have excellent robustness properties whereas L_1-quantiles break down.

Chapter 2

The influence function of Stahel-Donoho type estimators of covariance and PCA

2.1 Introduction

A very popular technique for analyzing multivariate data is principal component analysis (PCA). It consists of finding orthogonal directions which maximize the variance captured in the data. These directions can be computed as the eigenvectors of an estimate of the covariance matrix. Classical PCA uses the classical sample covariance matrix to do so. However, when outliers are present in the data, they can heavily influence the resulting eigenvectors. Thus, there is a need for robust estimators of the covariance matrix.

The Stahel-Donoho [Stahel, 1981, Donoho, 1982] estimator was the first introduced high-breakdown and affine equivariant estimator of multivariate location and scatter. It is based on the outlyingness of data points, which is obtained by projecting the observation on univariate directions. The original Stahel-Donoho estimator then computes a weighted mean and covariance matrix, with weights inverse proportional to the outlyingness. In this chapter we denote this approach as the ‘weighted outlyingness’ Stahel-Donoho estimator

(SD$_{wo}$). Alternatively, we can consider the mean and covariance matrix of the $(1-\alpha)n$ observations (with $0 < \alpha < 1/2$) with smallest outlyingness, denoted as SD$_{so}$. Both estimators are introduced and discussed in Section 2.2. In Section 2.3 we describe how SD$_{so}$ leads to a robust method for PCA, called ROBPCA [Hubert et al., 2005]. This PCA method yields fast, accurate and robust results, even in high dimensions, as illustrated in [Hubert and Engelen, 2004, Engelen et al., 2005]. A short sketch of this method is provided, as well as a real data example.

Next we investigate the robustness of SD$_{so}$ and ROBPCA by means of their influence function. For SD$_{wo}$ this has been established in Gervini [2002]. In Section 2.4 we first derive the influence function of SD$_{so}$. The result leads to several important applications. First of all the influence function gives us additional insight in the robustness of these methods, for example by examining the gross error sensitivities. Secondly, asymptotic efficiencies are computed and compared to those of SD$_{wo}$ and the MCD estimator [Rousseeuw, 1984]. In Section 2.5 we derive the influence function of the ROBPCA eigenvectors and eigenvalues. Finally we consider an application of ROBPCA, namely robust Partial Least Squares Regression [Hubert and Vanden Branden, 2003] of which the influence function is obtained in Section 2.6. All proofs are collected in Section 2.8.

2.2 Stahel-Donoho type estimators of covariance

Consider a d-dimensional sample $\mathbf{X} = (x_1, \ldots, x_n)^t$ of size n. Stahel [Stahel, 1981] and Donoho [Donoho, 1982] introduced the outlyingness $r(x_i, \mathbf{X})$, defined as follows:

$$r(x_i, \mathbf{X}) = \sup_{a \in \mathbb{R}^d} \left| \frac{a^t x_i - m(a^t \mathbf{X})}{s(a^t \mathbf{X})} \right| \tag{2.1}$$

where $m(.)$ and $s(.)$ are affine equivariant univariate robust estimators of location and scale. Popular options for $m(.)$ and $s(.)$ are the median and the median absolute deviation (denoted mad), univariate M-estimators [Gervini, 2002] and univariate MCD-estimators [Hubert et al., 2005].

Equation (2.1) can be interpreted as follows: for every univariate direction $a \in \mathbb{R}^d$ we consider the standardized distance of the projection $a^t x_i$ of observation x_i to the robust center of all the projected data points. Thus suppose $r(x_i, \mathbf{X})$ is large, then there exists a direction in which the projection of x_i lies far away from the bulk of the other projections. As such, one might suspect x_i of being an outlier. Thus, in order to obtain a robust estimator, we want to concentrate on the observations x_i with small outlyingness $r(x_i; \mathbf{X})$. We consider two possibilities to do so.

A first approach consists of downweighting all observations according to their outlyingness, as in the original proposal of Stahel and Donoho. In this case, we need a non-negative weighting function $w(.)$, which gives a smaller weight to observations with a large outlyingness. Several functions have been proposed, for example Huber type weights in Maronna and Yohai [1995], Gaussian weights in Gervini [2002], and exponential weights in Zuo et al. [2004]. The corresponding estimators of location and covariance, $SD_{wo} = (T_{wo}(\mathbf{X}), V_{wo}(\mathbf{X}))$, are then weighted versions of the classical sample mean and sample covariance matrix:

$$\begin{aligned} T_{wo}(\mathbf{X}) &= \frac{\sum_{i=1}^{n} w(r^2(x_i; \mathbf{X}))\, x_i}{\sum_{i=1}^{n} w(r^2(x_i; \mathbf{X}))} \\ V_{wo}(\mathbf{X}) &= c_w \frac{\sum_{i=1}^{n} w(r^2(x_i; \mathbf{X}))(x_i - T_{wo}(\mathbf{X}))(x_i - T_{wo}(\mathbf{X}))^t}{\sum_{i=1}^{n} w(r^2(x_i; \mathbf{X}))}. \end{aligned} \tag{2.2}$$

The constant c_w ensures consistency at the specified model. This estimator belongs to the class of depth weighted scatter estimators, is asymptotically normal distributed (under mild conditions on the weight function, the location $m(.)$ and the scale estimator $s(.)$) and achieves high asymptotic efficiencies if an appropriate weight function is chosen [Zuo and Cui, 2005]. As the underlying distribution function typically is unknown, it can be difficult to define a good weight function and more specifically to determine in advance from which cutoff value on the weight function should decrease.

To circumvent this problem, a second approach was proposed in Hubert et al. [2005]. A proportion $0 < \alpha < 1/2$ is chosen and only the $(1 - \alpha)n$ observations with smallest outlyingness are used in the estimation. We will call this the Stahel-Donoho estimator with smallest outlyingness (SD$_{so}$) from now on. The corresponding estimators of location and covariance, $T_{so}(\mathbf{X})$ and V$_{so}(\mathbf{X})$, are defined as

$$\begin{aligned} T_{so}(\mathbf{X}) &= \frac{\sum_{i \in I_\alpha} x_i}{n_I} \\ V_{so}(\mathbf{X}) &= c_\alpha \frac{\sum_{i \in I_\alpha} (x_i - T_{so}(\mathbf{X}))(x_i - T_{so}(\mathbf{X}))^t}{n_I - 1} \end{aligned} \tag{2.3}$$

where I_α denotes the set containing the $n_I = [(1 - \alpha)n]$ data points with smallest outlyingness. The constant c_α ensures consistency at the specified model.

2.3 ROBPCA

2.3.1 Algorithm

Classical Principal Component Analysis (PCA) performs a dimension reduction by replacing the d variables in the sample $\mathbf{X}$ with $k \ll d$ principal components. These are computed as the eigenvectors of the classical sample covariance matrix. However, if the sample $\mathbf{X}$ contains some outliers, the effect on the resulting principal components can be devastating. In order to resist these outliers, robust PCA techniques have been developed in recent years. In this paper we focus on the ROBPCA algorithm [Hubert et al., 2005], which consists of the following four major steps:

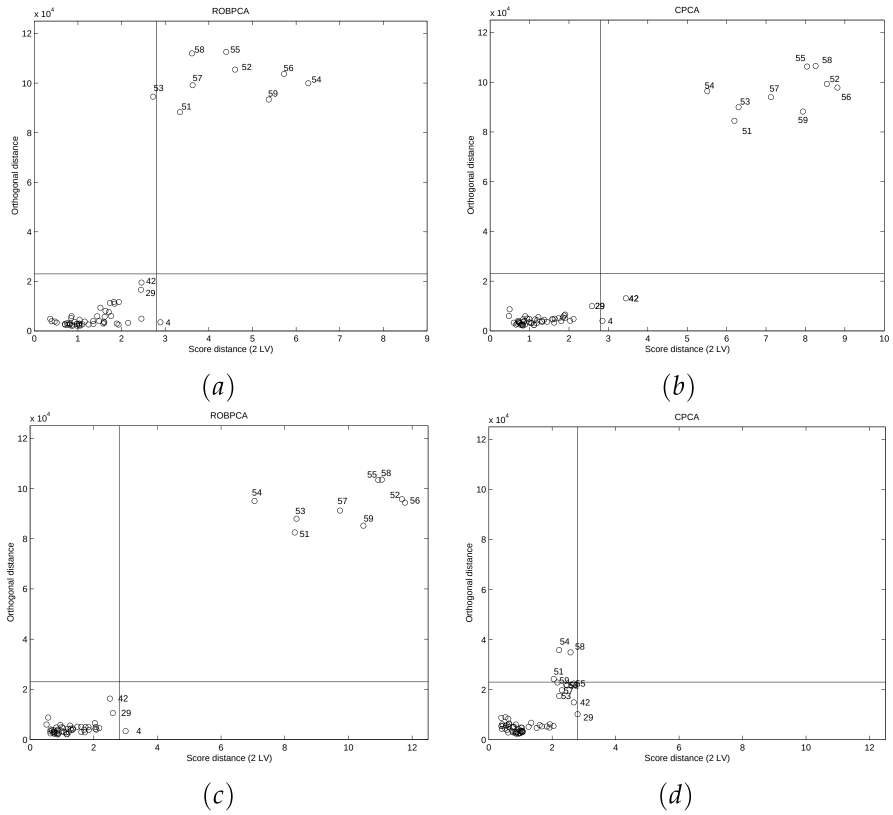
1. Choose $0 < \alpha < 1/2$.

2. Calculate the outlyingness of every data point x_i as in (2.1), using univariate MCD estimators in [Rousseeuw, 1984] for $m(.)$ and $s(.)$ with breakdown value α. Compute the $\mathrm{SD}_{so} = (T_{so}, V_{so})$ estimator (2.3) as the mean and covariance matrix of the $[(1-\alpha)n]$ observations with smallest outlyingness.

3. Reduce the dimension by projecting all data on the k-dimensional subspace spanned by the first k eigenvectors of the robust covariance estimator V_{so}, obtained in step 2. The choice of k can for example be made using a scree plot of the eigenvalues, or by a robust PRESS algorithm [Hubert and Engelen, 2007].

4. In this k-dimensional subspace, a robust center and covariance matrix are recomputed by applying the reweighted MCD estimator [Rousseeuw and Van Driessen, 1999] to the projected data. The final principal components are the eigenvectors of this robust covariance matrix.

For a detailed explanation including computational and practical aspects, we refer to Hubert et al. [2005]. We conclude this section by giving an example of ROBPCA on a real data set, showing its robustness and its usefulness in outlier detection. Applications of ROBPCA in bioinformatics, multivariate calibration and classification can be found in Hubert and Engelen [2004], Hubert and Verboven [2003], Hubert and Vanden Branden [2003], Vanden Branden and Hubert [2005].



(*a*) (*b*)

(*c*) (*d*)

Figure 2.1 Outlier maps for the Singh data, obtained with (a) ROBPCA on the training data; (b) classical PCA on the training data; (c) ROBPCA on the training and the test data; (d) classical PCA on the training and the test data.

2.3.2 Prostate cancer data set

We consider the prostate cancer data from Singh et al. [2002]. The full data set consists of gene expression profiles that were obtained from 52 prostate tumors and 50 non tumor prostate samples from patients undergoing surgery. Moreover a test set of 25 tumor and 9 normal samples was obtained from a different experiment. The number of gene expression levels is 12600. Here, we will only consider the normal samples, which means that our data consists of 59 observations (50 from the training and 9 from the test set) and 12600 variables.

First, we estimated the PCA subspace only using the training data and decided to retain two principal components, based on a scree plot of the eigenvalues. Applying ROBPCA with $\alpha = 0.25$ or classical PCA did not really make a difference as can be seen from Figures 2.1(a) and (b) which look very similar. These so called outlier maps plot the orthogonal distances versus the score distances. For each observation, the y-axis shows the orthogonal distance between the original data point in the 12600-dimensional space and the projected data point in the two-dimensional PCA subspace. The score distance on the x-axis is a robust distance of the projected observations (or scores). When using ROBPCA these distances are based on the MCD estimates derived in step 4 of the algorithm, whereas for classical PCA they coincide with the Mahalanobis distances (thus based on the sample mean and covariance) of the scores. The vertical and horizontal lines define cutoff lines which can be used to separate the regular observations from the outliers (see Hubert et al. [2005] for all details).

The outlier map in Figures 2.1(a) and (b) show the orthogonal and score distances of the training data in the lower left corner, and also those of the test data (which are not used to produce the estimates!) in the upper right corner. We immediately see that the test set is very different from the training set, and should not be used to validate the results.

Assume now that we would ignore the difference between the training and test data and that we would use the complete data set to find an appropriate PCA subspace. Figure 2.1(c) shows the resulting diagnostic plot for ROBPCA (with $\alpha = 0.25$), whereas Figure 2.1(d) is the plot for classical PCA. As ROBPCA looks for the $[(1-\alpha)n]$ least outlying data points, most of the 50 training observations determine the robust principal components. The influence of the 9 test cases on the other hand is small. As a result, we get about the same diagnostic plot as in Figure 2.1(a), where the test data was not used at all. In classical PCA however, the 9 observations from the test set have a huge effect. The principal components are tilted towards them and the diagnostic plot is totally different from what we observed in Figure 2.1(b). Only observations 54 and 58 are now clearly above the horizontal cut-off line, indicating that they lie far from the PCA subspace. Furthermore, observations 29 and 42 are closer to the test set than to the training set which they really belong to.

This illustrates that small subgroups in the data do not have a large effect on ROBPCA, and often even can be detected using this robust PCA method.

2.4 Influence functions

2.4.1 Definitions and setup

In this section we will derive the influence function (Definition 1) of the SD_{so} estimator. This function gives useful information about the behavior of a statistical functional when outliers are present. For the classical covariance matrix we denote the corresponding functional C as

$$C(F) = \int (x - \mathbb{E}_F(X))(x - \mathbb{E}_F(X))^t dF(x).$$

The notation $\mathbb{E}_F$ denotes the expectation with respect to the distribution F, so $\mathbb{E}_F(X) = \int x dF(x)$. Straightforward computations show that

$$IF(z; C, F) = (z - \mathbb{E}_F(X))(z - \mathbb{E}_F(X))^t - C(F). \tag{2.4}$$

We see that the influence function of the classical covariance matrix enlarges quickly as z is chosen away from the mean of the distribution F, and is unbounded. This means that an infinitesimally small amount of outliers can have an arbitrary large effect on the result. So the classical covariance matrix is anything but robust. For any robust estimator of covariance, a minimal requirement should be to have a bounded influence function.

Let us now consider the Stahel-Donoho type estimators. The weighted Stahel-Donoho estimator SD_{wo} has bounded influence function, see Gervini [2002] and Zuo and Cui [2005]. Here, we concentrate on SD_{so}. We define the outlyingness $r(x; F)$ of a point $x \in \mathbb{R}^d$ as

$$r(x; F) = \sup_{a \in \mathbb{R}^d} \left| \frac{a^t x - m(F^a)}{s(F^a)} \right|$$

with $m(.)$ and $s(.)$ affine equivariant univariate estimators of location and scale and $F^a = \mathcal{L}(a^t X)$, $X \sim F$. Due to the affine equivariance of m and s, we can assume from now on that $\|a\| = 1$. Note that this definition is equivalent to (2.1) when replacing the theoretical distribution F by the empirical distribution F_n.

In a similar way, definitions (2.2) and (2.3) can be extended from the sample case to the functional case leading to:

$$\begin{aligned} T_{wo}(F) &= \frac{\mathbb{E}_F(w(r^2(X; F))X)}{\mathbb{E}_F(w(r^2(X; F)))} \\ V_{wo}(F) &= c_w \frac{\mathbb{E}_F(w(r^2(X; F))(X - T_{wo}(F))(X - T_{wo}(F))^t)}{\mathbb{E}_F(w(r^2(X; F)))} \end{aligned} \tag{2.5}$$

with w a non-negative weighting function and

$$\begin{aligned} T_{so}(F) &= \frac{1}{1-\alpha}\int_{A(F)} x\, dF(x) \\ V_{so}(F) &= \frac{c_\alpha}{1-\alpha}\int_{A(F)} (x - T_{so}(F))(x - T_{so}(F))^t dF(x) \end{aligned} \tag{2.6}$$

with $A(F) = \{y \in \mathbb{R}^d : r^2(y, F) \leqslant qr_\alpha(F)\}$ and $qr_\alpha(F)$ the $(1-\alpha)$ quantile of the r^2, i.e. the smallest value such that $D(r^2(y, F) \leqslant qr_\alpha(F)) = 1 - \alpha$. The factor c_α is chosen in order to ensure Fisher consistency at the specified model.

We will assume that F belongs to the class of d-dimensional elliptical symmetric distributions around μ. This means that its density exists and that it is of the form

$$f(x) = |\Sigma|^{-1/2} g((x-\mu)^t \Sigma^{-1}(x-\mu))$$

with g a monotone decreasing function and Σ a symmetric positive-definite matrix. Further we assume that $m(.)$ is Fisher consistent, which means that $m(F^a) = a^t\mu$. To obtain Fisher consistency for V_{so}, we then take c_α in (2.6) such that $V_{so}(F) = \Sigma$. Due to the affine equivariance of SD_{so}, we can set $\mu = 0$ and $\Sigma = I$. The distribution F then becomes spherical, $F^a := F^1$ is identical for every a on the unit sphere in $\mathbb{R}^d$ and satisfies $m(F^a) = 0$ and $s(F^a) = s_0 \in \mathbb{R}$. We will denote q_α as the $1-\alpha$ quantile of the distribution of X^tX with $X \sim F$.

2.4.2 Main result

First we need to know how the outlyingness $r(x; F)$ behaves when contamination is added to F.

Lemma 2.1 *Denote $F_{\epsilon,z} = (1-\epsilon)F + \epsilon\Delta_z$. Assume that the influence function of m exists, and that the function $(\epsilon, z) \to s((1-\epsilon)F + \epsilon\Delta_z)$ is twice differentiable at $(0, z)$. Then, for each $x, z \in \mathbb{R}^d$, $x \neq 0$*

$$\frac{\partial r^2(x, F_{\epsilon,z})}{\partial\epsilon}|_{\epsilon=0} = -2\frac{\|x\|}{s_0^2}\; IF(\tilde{x}^t z; m, F^1) - 2\frac{\|x\|^2}{s_0^3}\; IF(\tilde{x}^t z; s, F^1)$$

where $\tilde{x} = x/\|x\|$ and F^1 is the one-dimensional marginal distribution of F.

Note that this is a slight extension of Theorem 2 in Gervini [2002], where the case of $m(.)$ and $s(.)$ being M-estimators was covered. Our main condition basically states that the influence function of the scale estimator is continuous and differentiable. This is the case for a suitable M-estimator of scale and for the Q_n-estimator [Rousseeuw and Croux, 1993] for instance. Whether this condition is really necessary is still an open problem. Gervini [2002] mentions that his expressions are still valid for the mad, although the influence function of the mad is not differentiable everywhere. For a univariate MCD estimator of scale, the differentiability condition is not satisfied either, and we were unable to proof the existence of the influence function in that case. To keep close to the original implementation in Hubert et al. [2005], we still use univariate MCD-estimators in the next results.

We can now derive the influence function of V_{so} at a spherical distribution F.

Theorem 2.2 *The influence function of the V_{so} estimator of scatter at a spherical distribution F is given by*

$$IF(z; V_{so}, F) = w_1(\|z\|)zz^t + w_2(\|z\|)I$$

where

$$w_1(\|z\|) = \frac{c_\alpha}{1-\alpha}\mathbb{1}(\|z\|^2 \leq q_\alpha) - \frac{c_\alpha}{1-\alpha}g(q_\alpha)\frac{d_3}{2\|z\|^2}$$
$$w_2(\|z\|) = \frac{c_\alpha}{1-\alpha}\left(\frac{q_\alpha}{d}\left(1-\alpha-\mathbb{1}(\|z\|^2 \leq q_\alpha)\right) + g(q_\alpha)\left(\frac{q_\alpha d_1}{2d} - \frac{d_2}{2(d-1)}\right)\right) - 1$$

with

$$d_1 = \frac{2\pi^{\frac{d-1}{2}}}{\Gamma(\frac{d-1}{2})}\int_{-\sqrt{q_\alpha}}^{\sqrt{q_\alpha}} h(x_1, z)(q_\alpha - x_1^2)^{\frac{d-3}{2}}dx_1$$
$$d_2 = \frac{2\pi^{\frac{d-1}{2}}}{\Gamma(\frac{d-1}{2})}\int_{-\sqrt{q_\alpha}}^{\sqrt{q_\alpha}} h(x_1, z)(q_\alpha - x_1^2)^{\frac{d-1}{2}}dx_1$$
$$d_3 = \frac{2\pi^{\frac{d-1}{2}}}{\Gamma(\frac{d-1}{2})}\int_{-\sqrt{q_\alpha}}^{\sqrt{q_\alpha}} h(x_1, z)\frac{px_1^2 - q_\alpha}{d-1}(q_\alpha - x_1^2)^{\frac{d-3}{2}}dx_1$$

and

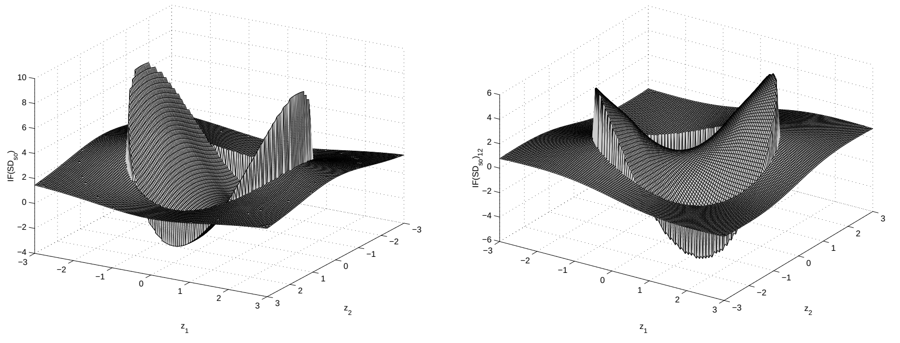
$$h(x_1, z) = -\frac{2\sqrt{q_\alpha}}{s_0^2}IF(\frac{x_1\|z\|}{\sqrt{q_\alpha}}, m, F^1) - \frac{2q_\alpha}{s_0^3}IF(\frac{x_1\|z\|}{\sqrt{q_\alpha}}, s, F^1).$$

In Figure 2.2 the influence functions of (a) the first diagonal element and (b) the non-diagonal element are shown at a two-dimensional spherical Gaussian distribution, using univariate MCD estimators for $m(.)$ and $s(.)$ with breakdown value α whose influence function equals

$$IF(z; m_{MCD}, F) = \left(\frac{-2}{1-\alpha} \int_{-\sqrt{q_{\alpha 1}}}^{\sqrt{q_{\alpha 1}}} t^2 g'(t^2) dt \right)^{-1} \frac{z}{1-\alpha} \mathbb{1}(|z| \leq \sqrt{q_{\alpha 1}})$$

$$IF(z; s_{MCD}, F) = \left(\int_{-\sqrt{q_{\alpha 1}}}^{\sqrt{q_{\alpha 1}}} t^2 dF(t) \right)^{-1} \{\mathbb{1}(|z| \leq q_{\alpha 1})(z^2 - q_{\alpha 1}) + (1-\alpha) q_{\alpha 1}\} - 1$$

with $q_{\alpha 1} = F^{-1}(1 - \frac{\alpha}{2})$ [Croux and Rousseeuw, 1992, Butler et al., 1993]. The parameter α was chosen to be 0.25. We see that the influence function is rather small around zero, the mean of the distribution. Away from the mean the influence function in-/decreases rapidly inside the circle $\|z\|^2 \leqslant q_\alpha = \chi^2_{2,0.75}$, the 0.75-quantile of a χ^2-distribution with 2 degrees of freedom. Outside this circle however, the influence function drops again to remain small and bounded. This reflects that contamination placed outside this circle will be part of the 25% probability mass with highest outlyingness. Therefore, it will not be used in calculating the covariance matrix V_{so} and thus its effect will be small.



(a) (b)

Figure 2.2 Influence function of V_{so} (a) of the first diagonal element and (b) of the non-diagonal element, at a two dimensional spherical Gaussian distribution using univariate MCD estimators for $m(.)$ and $s(.)$, with $\alpha = 0.25$

For distributions in more than 2 dimensions we can not plot the influence function anymore, but we can look at the function w_1 as defined in Theorem 2.2. This function is depicted in Figure 2.3, at a 30-dimensional Gaussian distribution for several values of α. It is clearly decreasing, with a jump at $\sqrt{q_\alpha}$. Behind the jump, $w_1 \sim 1/\|z\|^2$ and hence it becomes very small and decreases to zero. Therefore the influence function will also be small for outliers having a large norm.

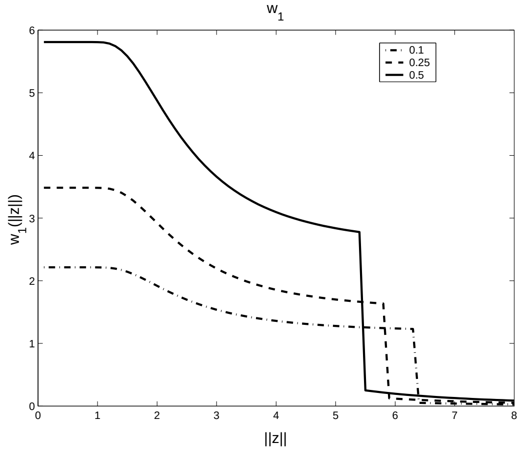


Figure 2.3 Function w_1 appearing in the formula of $IF(z; V_{so}, F)$, for $\alpha = 0.1$ (dashed-dotted line), $\alpha = 0.25$ (dashed line) and $\alpha = 0.5$ (solid line), at a 30-dimensional Gaussian distribution.

Theorem 2.2 thus shows that V$_{so}$ is B-robust, meaning its influence function is bounded [Hampel et al., 1986]. Nevertheless, the influence function might locally attain high peaks. A useful measure of robustness in this respect is the gross error sensitivity

$$\gamma(F) = \sup_{z \in \mathbb{R}^d} \|IF(z; V_{so}, F)\|_{Fr} \tag{2.7}$$

where $\|.\|_{Fr}$ is the Frobenius norm, which is the square root of the sum of all squared matrix elements. We compare the gross error sensitivities of V_{so}, V_{wo} and the MCD estimator at a Gaussian distribution. For V$_{wo}$ we consider two weighting schemes: a truncated exponential weight function

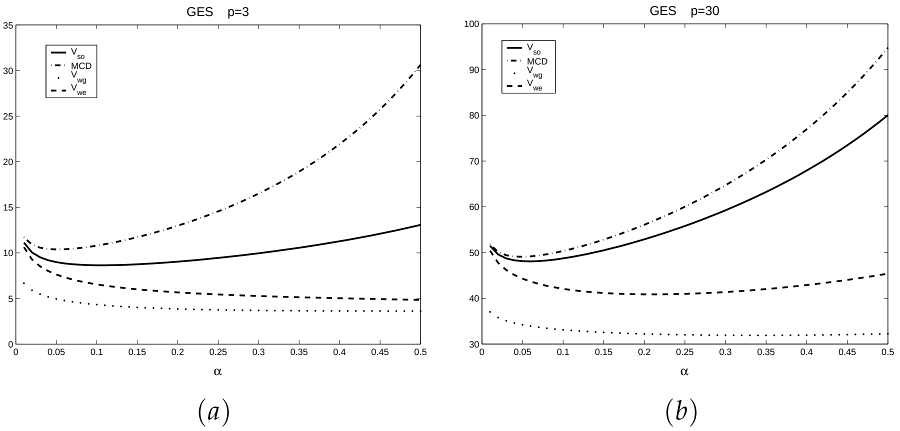
$$we(t^2) = \min\{1, e^{5(1-t^2/c)}\}$$

and a Gaussian weight function

$$wg(t^2) = \frac{\phi(t^2/c)}{\phi(1)}$$

with cut-off value $c = \chi^2_{d,1-\alpha}$ [Gervini, 2002].

Figure 2.4 shows the gross error sensitivities as a function of α, for $d = 3$ and $d = 30$. The estimator V$_{so}$ is clearly situated in between MCD and V$_{wo}$. In higher dimensions it tends more and more towards the MCD. Nevertheless



(a) (b)

Figure 2.4 Gross error sensitivities of V$_{so}$ (solid line) compared to MCD (dashed-dotted line) and weighted Stahel-Donoho (exponential weights: dashed line, Gaussian weights: dotted line), for (a) $d = 3$ and (b) $d = 30$.

V$_{so}$ is uniformly better, since its gross error sensitivity is always smaller than that of the MCD estimator. On the other hand, V$_{so}$ is uniformly worse than weighted Stahel-Donoho with respect to gross error sensitivities.

Remark The influence function of the location part T_{so} can be computed in a similar way. We provide the result in the following theorem, with proof in Section 2.8. In the remainder of the paper we will however restrict ourselves to the covariance estimator V$_{so}$.

Theorem 2.3 *The influence function of the SD$_{so}$ estimator of location at a spherical distribution F is given by*

$$IF(z; T_{so}, F) = \frac{1}{1-\alpha}\mathbb{1}(\|z\|^2 \leq q_\alpha) - \frac{2\pi^{\frac{d-1}{2}}}{\Gamma(\frac{d-1}{2})} g(q_\alpha) \int_0^{\sqrt{q_\alpha}} h(x_1, z) x_1 (q_\alpha - x_1^2)^{\frac{d-3}{2}}$$

using the same notations as in Theorem 2.2.

2.4.3 Asymptotic efficiency

Influence functions also allow us to compute asymptotic efficiencies of estimators. More precisely, assuming asymptotic normality and Fréchet differentiability of the V$_{so}$ estimator, asymptotic variances can be computed by

$$ASV(V_{so}(F)_{ij}) = \int_{\mathbb{R}^d} IF^2(z; V_{so}, F)_{ij} dF(z)$$

for $1 \leqslant i, j \leqslant d$. Note that for the Stahel-Donoho estimator, asymptotic normality has been proven for location [Zuo et al., 2004] and for scatter [Zuo and Cui, 2005]. These asymptotic variances are used to compute relative asymptotic efficiencies by

$$\text{Eff}(V_{so}(F)_{ij}, F) = \frac{ASV(C_{ij}, F)}{ASV(V_{so}(F)_{ij}, F)}$$

where $ASV(C_{ij}, F)$ is the asymptotic variance of the classical covariance matrix (2.4) elements. In case of a Gaussian distribution Φ,

$$ASV(C_{ii}, \Phi) = 2 \quad \text{and} \quad ASV(C_{ij}, \Phi) = 1 \quad \text{for} \quad i \neq j.$$

Table 2.1 shows asymptotic efficiencies of the diagonal resp. off-diagonal elements of the V$_{so}$ estimator, compared to the weighted Stahel-Donoho estimator V$_{wo}$ with Gaussian weights and to the MCD estimator, at a Gaussian distribution. Remember that computing Stahel-Donoho type estimators requires the choice of univariate robust measures $m(.)$ and $s(.)$ of location and spread. For V$_{wo}$ we take $m(.)$ the median and $s(.)$ the median absolute deviation (mad), whereas for V$_{so}$ we additionally consider $m(.)$ and $s(.)$ the univariate MCD estimators of location and spread.

For V$_{so}$ no large differences in efficiency are seen for the two choices of m and s. The deviations are especially small in higher dimensions (e.g. $d = 30$ in Table 2.1). In lower dimensions, V$_{so}$(MCD) is slighter more efficient than $V_{so}(\text{med/mad})$ when $\alpha = 0.25$, whereas the inverse holds for $\alpha = 0.5$. This is not surprising, as the univariate MCD with $\alpha = 0.25$ is more efficient than the median, but not in case $\alpha = 0.5$. Further we see that the performance of V$_{so}$ is better than that of the multivariate MCD at the normal model, but the weighted Stahel-Donoho estimator V$_{wo}$ clearly attains a much higher efficiency.

Based on our previous analysis of the influence function, gross error sensitivities and relative asymptotic efficiencies, we can conclude that V$_{so}$ is a worthy alternative to other robust covariance estimators. It performs a bit better than MCD and slightly worse than weighted Stahel-Donoho. Note however that V$_{so}$ has big computational advantages compared to the others, especially when working in high dimensions. When the number of cases n is smaller than the dimension d, the MCD cannot be computed anymore. Moreover, V$_{so}$ does not require the choice of a weight function or cut-off value.

			$d = 2$	$d = 3$	$d = 5$	$d = 10$	$d = 30$
diag	$\alpha = 0.25$	V$_{so}$ (med/mad)	0.419	0.437	0.481	0.546	0.627
		V$_{so}$ (MCD)	0.455	0.462	0.496	0.553	0.629
		MCD	0.262	0.300	0.366	0.459	0.577
	$\alpha = 0.5$	V$_{so}$ (med/mad)	0.281	0.288	0.313	0.354	0.408
		V$_{so}$ (MCD)	0.252	0.266	0.299	0.347	0.406
		MCD	0.062	0.089	0.134	0.205	0.310
	$c = \chi^2_{d,0.95}$	V$_{wo}$	0.650	0.723	0.782	0.813	0.785
off diag	$\alpha = 0.25$	V$_{so}$ (med/mad)	0.306	0.366	0.441	0.533	0.647
		V$_{so}$ (MCD)	0.354	0.392	0.457	0.543	0.654
		MCD	0.163	0.233	0.324	0.438	0.570
	$\alpha = 0.5$	V$_{so}$ (med/mad)	0.221	0.243	0.283	0.338	0.406
		V$_{so}$ (MCD)	0.205	0.219	0.270	0.332	0.404
		MCD	0.033	0.063	0.113	0.191	0.304
	$c = \chi^2_{d,0.95}$	V$_{wo}$	0.783	0.851	0.906	0.949	0.978

Table 2.1 Relative asymptotic efficiencies for a Gaussian distribution. Comparison between V_{so}, V_{wo} and MCD. Upper half: diagonal element. Lower half: off-diagonal element.

2.4.4 Finite sample comparison

To put our theoretical results in some perspective, we conclude this section by a small finite sample simulation study. We compare the SD$_{wo}$ estimator with Gaussian weights ($c = \chi^2_{d,0.95}$) and the SD$_{so}$ estimator. We computed the outlyingness of each data point by maximizing over directions through d randomly chosen observations. The number of directions needed to get good results turned out to be quite small. In fact, with 1000 directions the outcome was almost the same as with 250 directions. The data was generated from a multivariate standard normal distribution, a multivariate Student distribution with 5 degrees of freedom and a normal distribution with 10% of outliers. These outliers were chosen slightly scattered around the point $(20, 0, 0, \ldots, 0)$. We did the analysis with $n = 150$ and $n = 500$ observations, each time with $d = 5$, $d = 10$ and $d = 30$ variables. Each simulation consisted of $m = 300$ runs.

For the diagonal elements, we use the standardized variance as a measure of comparison (see for example Croux and Haesbroeck [1999]). Denote $\hat{\Sigma}^k_{ii}$ the ith diagonal element of the covariance estimator of the kth sample. Then the standardized variance is defined as

$$\mathrm{StVar}(\hat{\Sigma}_{ii}) = \frac{n\ \mathrm{var}_m(\hat{\Sigma}_{ii})}{(\mathrm{ave}_m(\hat{\Sigma}_{ii}))^2}$$

where $\mathrm{ave}_m(\hat{\Sigma}_{ii})$ and $\mathrm{var}_m(\hat{\Sigma}_{ii})$ are the average and variance over the sequence of m replicates Σ^k_{ii}. For the off-diagonal elements, we use the mean squared error

$$\mathrm{MSE}(\Sigma_{ij}) = \frac{n}{m}\sum_{k=1}^{m}(\hat{\Sigma}^k_{ij})^2.$$

Then the number $\mathrm{ave}_d(\mathrm{StVar}(\Sigma_{ii}))$ is a measure of the error when estimating the diagonal elements. These numbers are reported in Table 2.2. For the off-diagonal elements we report the average $\mathrm{MSE}(\Sigma_{ij})$ over all $1 \leq i \neq j \leq d$.

We can see that the finite sample errors of V_{wo} are much smaller at a normal distribution than that of V_{so}, confirming our theoretical findings. At the t_5 distribution however, the difference becomes a lot smaller, especially for $d = 30$. This indicates that V_{so} is less vulnerable to the heavy tails of the Student distribution. This is confirmed when we look at the normal distribution with contamination. The errors for the V_{so} estimator are about the same as in the case without the outliers. The finite sample errors for weighted Stahel-Donoho V_{wo} on the other hand are clearly larger, even drastically increasing in higher dimensions ($d = 30$).

2.5 Influence function of ROBPCA

An important application of covariance estimators is dimension reduction by PCA, in which the eigenvectors play an important role. In this section we concentrate on the ROBPCA algorithm as briefly explained in Section 2.3. The first step of ROBPCA consists of determining the covariance estimator V_{so}, which was analyzed in Section 2.4. The algorithm then proceeds by projecting all the data onto the subspace spanned by the first k eigenvectors of V_{so}. In this subspace, the MCD method is applied. In this section we will derive the influence function of the resulting eigenvectors and eigenvalues. To this end, we need to introduce some notations as follows.

- Let F be a d-dimensional elliptical distribution with zero mean and covariance matrix $\Sigma = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ with $\lambda_i \neq \lambda_j$ for every $i \neq j$. Assuming Σ a diagonal matrix can be done without loss of generalization, since ROBPCA is orthogonally equivariant.

			SD_{so}			SD_{wo}		
			$d=5$	10	30	$d=5$	10	30
$N(0,I_d)$	diag	$n=500$	3.92	3.51	3.06	2.27	2.05	1.99
		$n=150$	4.16	3.47	3.01	2.43	2.13	2.02
	off-diag	$n=500$	2.14	1.83	1.65	1.07	1.04	1.05
		$n=150$	2.31	1.96	1.62	1.10	1.06	1.02
t_5	diag	$n=500$	5.31	5.32	5.73	3.89	3.72	4.99
		$n=150$	5.43	5.08	5.65	3.93	3.74	4.34
	off-diag	$n=500$	2.44	2.33	2.60	1.81	1.73	2.03
		$n=150$	2.42	2.36	2.51	1.79	1.79	1.96
$N(0,I_d)$	diag	$n=500$	3.49	3.31	3.12	2.41	2.22	6.19
$+$		$n=150$	3.70	3.31	2.98	2.54	2.24	6.33
10%	off-diag	$n=500$	2.40	2.12	1.72	1.18	1.26	1.57
outliers		$n=150$	2.48	1.96	1.73	1.20	1.18	1.53

Table 2.2 Finite sample comparison between SD_{wo} and SD_{so} at the normal distribution, Student distribution and normal distribution with contamination.

- F_k denotes the k-dimensional distribution of the projection of F onto the first k eigenvectors of $V_{so}(F)$. These k eigenvectors are collected in the $k \times d$ matrix $P_k(F)$.

- $V_{\mathrm{robpca}}(F)$ is the MCD estimator of scatter of F_k. Its eigenvalues (in decreasing order) and its corresponding eigenvectors (backtransformed from the k- to the original d-dimensional space) are denoted as $(\lambda_{\mathrm{robpca},j}(F), v_{\mathrm{robpca,j}}(F))$.

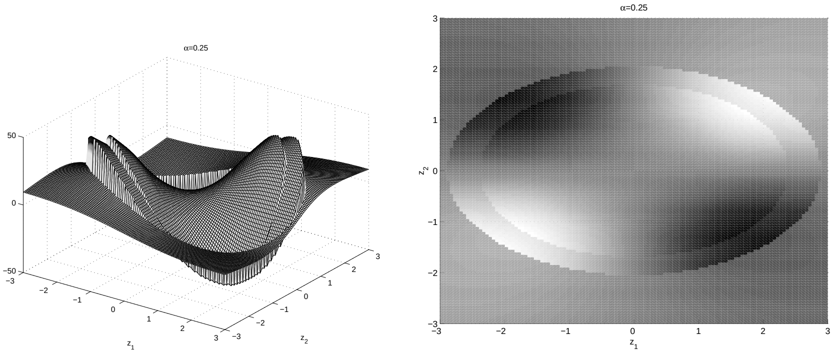
Then we obtain the following results.

Theorem 2.4 *For $j = 1, \ldots, k$, the influence function of $\lambda_{robpca,j}(F)$ at the distribution F is given by*

$$IF(z; \lambda_{robpca,j}, F) = IF(P_k(F)z; MCD, F_k)_{jj}.$$

Thus we can see that the first step of the ROBPCA algorithm has no effect on the influence function of the eigenvalues. Only the MCD estimation in the second step plays a role.

Theorem 2.5 *For $j = 1, \ldots, k$, the influence function of $v_{robpca,j}(F)$ at F is given*



(a) (b)

Figure 2.5 $IF([z_1, z_2, 0]; v_{\text{robpca},1}, F)_2$ at a three dimensional Gaussian distribution, with $\alpha = 0.25$. (a) Three dimensional view and (b) two dimensional view from above. Grey scale indicates the height of the influence function: from black for the smallest values over grey for values around zero to white for the largest values.

by

$$IF(z; v_{robpca,j}, F)_j = 0$$

$$IF(z; v_{robpca,j}, F)_i = IF(P_k(F)z, v_{MCD,j}, F_k)_i - \frac{w_1(\|\Sigma^{-1/2}z\|)z_i z_j(\lambda_i + \lambda_j)}{(\lambda_j - \lambda_i)^2}, \quad i \neq j$$

with w_1 defined as in Theorem 2.2 (see also Figure 2.3).

We clearly see both steps of the algorithm appearing in the formula above. The first term is due to the MCD-step of the algorithm, while the second term depends on the function w_1, reflecting the influence of the first step of the algorithm. Figure 2.5 shows an example with $d = 3$ and $k = 2$. The influence function of the second component of $v_{\text{robpca},1}$ is plotted for all points in the plane $z_3 = 0$, for a Gaussian distribution with covariance matrix $\Sigma = \text{diag}(2, 1, 0.5)$. Part (a) of Figure 2.5 is the three dimensional view, whereas part (b) is the same plot seen from above, with a greyscale indicating the value of the influence function. Values around zero are grey. White on the other hand means a very large positive effect. Large negative values are depicted in black. One can clearly distinguish two ellipses, which reflect the two estimators in play: the three-dimensional V_{so} estimator of which the first two eigenvectors provide a dimension reduction to a two-dimensional subspace, and then the two-dimensional MCD estimator applied in this subspace. Moreover the influence function is clearly bounded, with large outliers outside the ellipses having minimal effect.

2.6 Influence function of RSIMPLS

As a second application of the V_{so} estimator and the resulting ROBPCA method, we consider robust Partial Least Squares regression. PLSR is a widely used technique in various fields, including chemometrics and econometrics. It tries to link two sets of variables by means of a linear model. The first set of variables contains the d covariates or predictor variables, denoted by $X_1, \ldots, X_d$. The second set consists of q response variables $Y_1, \ldots, Y_q$. PLSR assumes the following linear model:

$$Y_i = \beta_0 + Bx_i + e_i$$

with i.i.d. errors satisfying $E(e_i) = 0$ and $cov(e_i) = \Sigma_e$. When $q > 1$ the intercept term β_0 is a vector, and the slope B is a $q \times d$ matrix. Estimates for β_0, B and Σ_e can be obtained via classical multiple linear regression (MLR). However, this leads to estimates with large variance when the covariates are highly correlated, which typically occurs in high dimensional data applications. PLSR tries to solve these inconveniences by applying MLR to $k < d$ latent uncorrelated variables $T_1, \ldots, T_k$. Several algorithms have been proposed to extract these latent variables from the data. We consider the SIMPLS algorithm [de Jong, 1993] and its robust version RSIMPLS [Vanden Branden and Hubert, 2004].

In the SIMPLS algorithm, weight vectors r_i and q_i (for $i = 1, \ldots, k$) are obtained as the vectors that maximize the covariance between X and Y components

$$\max_{\|r_i\|=1, \|q_i\|=1} \operatorname{cov}(\tilde{X}r_i, \tilde{Y}r_i) = \max_{\|r_i\|=1, \|q_i\|=1} r_i^t S_{xy} q_i \tag{2.8}$$

under the additional restrictions that the components $T_i = \tilde{X}r_i$ be uncorrelated. In (2.8) $\tilde{X}$ and $\tilde{Y}$ represent the mean-centered data matrices and $S_{xy} = \tilde{X}^t\tilde{Y}/(n-1)$ is the empirical cross-covariance matrix between the X- and Y-variables.

In practice, the key step of the algorithm is to determine eigenvectors from estimators of the covariance matrix Σ of the joint $Z = [X, Y]$ variables, so

$$\Sigma = E(Z-\mu)(Z-\mu)^t = \begin{pmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_y \end{pmatrix}$$

where μ is the center of the $[X, Y]$. The SIMPLS algorithm estimates Σ by the empirical covariance matrix. As explained in the previous sections, outliers can have a devastating effect on the results. Therefore, one can also use a robust estimator of covariance. This was done in Vanden Branden and Hubert [2004], where the MCD method and ROBPCA were proposed as alternatives. The latter approach was named RSIMPLS. The resulting estimator for the slope can be expressed as

$$\hat{B} = R_k(R_k^t\hat{\Sigma}_x R_k)^{-1}R_k^t\hat{\Sigma}_{xy}$$

with R_k the matrix of weight vectors r_i.

In Vanden Branden and Hubert [2004] the influence function of $\hat{B}$ was derived (see Theorem A.4) and it was shown that this influence function depends on the influence function of the weight vectors which for their part depend on $\hat{\Sigma}$. Results were derived for $\hat{\Sigma}$ the MCD estimator of scatter.

Using theorem 2.2 we can now derive analogously the influence function of $\hat{B}$ when $\hat{\Sigma}$ is the Stahel-Donoho scatter estimate with smallest outlyingness V_{so} (For ROBPCA the computations would become very complicated). Similar to Theorem 1 in Vanden Branden and Hubert [2004], we obtain the following theorem.

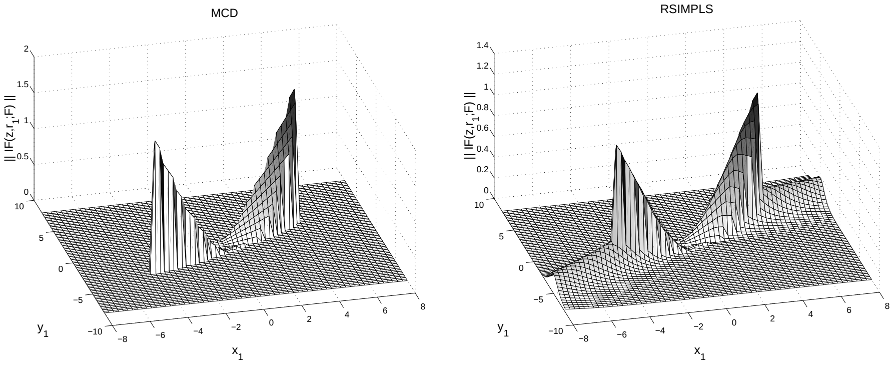
Theorem 2.6 *Denote R_k the matrix of weight vectors r_i as in (2.8). Denote $A = (R^t\Sigma_x R_k)^{-1}R_k^t\Sigma_{xy}$. The influence function of the slope $\hat{B}$ at an elliptical distribution F of the joint $[X, Y]$ variables in a point $z = (x, y),\ x \in \mathbb{R}^d,\ y \in \mathbb{R}^q$ equals*

$$\begin{aligned} IF(z; \hat{B}, F) = IF(z; R_k, F)A &- R_k(R_k^t\Sigma_x R_k)^{-1}\{IF(z; R_k, F)\Sigma_x B \\ &+ R_k^t\Sigma_x IF(z; R_k, F)A - IF(z; R_k^t, F)\Sigma_{xy}\}) \\ &+ w_1(d(z))R_k(R_k^t\Sigma_x R_k)^{-1}R_k^t(x-\mu_x)\{y-\mu_y)^t - (x-\mu_x)^t B\} \end{aligned}$$

with w_1 defined as in Theorem 2.2.

The influence function of the weight vectors $IF(z; R_k, F)$ can be obtained analogously to Theorem A.4 in Vanden Branden and Hubert [2004].

We illustrate the results by means of a simple example. We take $q = 1$ and $d = 2$. For F a Gaussian distribution with mean 0 and $\Sigma = \begin{pmatrix} 5 & \frac{1}{2} & 3 \\ \frac{1}{2} & 2 & \frac{1}{3} \\ 3 & \frac{1}{3} & 2 \end{pmatrix}$, we compute the norm of $IF(z; r_1, F)$, i.e. the influence function of the first RSIMPLS weight vector.

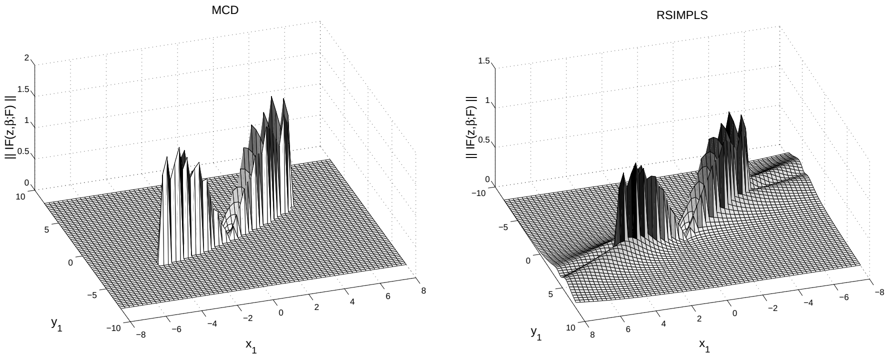


(a) (b)

Figure 2.6 Norm of the influence function of the RSIMPLS weight vector r_1, (a) using MCD and (b) using Stahel-Donoho with smallest outlyingness, both with $\alpha = 0.25$.

The result is shown in Figure 2.6. Part (a) shows the result using the MCD estimator and part (b) when using the Stahel-Donoho estimator with smallest outlyingness, both with $\alpha = 0.25$. In both plots an ellipse is visible containing 75% of the probability mass. Inside this ellipse the influence function increases, reaching its maximum at the border of the ellipse. Outside this ellipse however both influence functions become very small. This indicates that far outliers do not have a big effect, showing the robustness of these methods. Note that the influence function with MCD becomes 0 immediately outside the ellipse, since the corresponding data points are not used at all. For the Stahel-Donoho estimator, these points are neither used in computing the covariance matrix, but they do influence the calculation of the outlyingness (see also Figure 2.2). Therefore, small effects can still be seen for large values of the covariate variables.

Figure 2.7 shows the norm of the influence function of the slope vector, (a) using MCD and (b) using SD_{so}. Again it is clear that the influence functions are bounded, with a smaller maximum obtained by SD_{so}. As the slope has two



(a) (b)

Figure 2.7 Norm of the influence function of the slope vector, (a) using MCD and (b) using Stahel-Donoho with smallest outlyingness, both with $\alpha = 0.25$.

components, two 'regions' with lower influence become visible.

2.7 Conclusion

We investigated the influence function of the Stahel-Donoho estimator of covariance based on smallest outlyingness. An explicit expression was found from which it follows that the influence function is bounded, allowing us to say that the estimator is B-robust. We calculated gross error sensitivities and asymptotic efficiencies. They show that Stahel-Donoho with smallest outlyingness is a worthy alternative to other robust covariance estimators such as MCD and weighted Stahel-Donoho. Moreover the difficult task of choosing an appropriate weight function is not needed. Instead we can choose a number α, which has a clear interpretation as the fraction of most outlying observations.

Next, we considered two applications of these covariance estimators in the context of robust PCA and robust PLSR. Again we derived explicit expressions for the influence functions of ROBPCA and RSIMPLS and showed that they are bounded. This yields a theoretical justification of the robustness of ROBPCA and RSIMPLS.

2.8 Proofs

Proof of Lemma 2.1.

The proof is similar to the proof of Theorem 2 in Gervini [2002]. Take $x \in \mathbb{R}^d$. By definition

$$r^2(x; F_{\epsilon,z}) = \max_{a \in \mathbb{R}^d} \left(\frac{a^t x - m(F^a_{\epsilon,z})}{s(F^a_{\epsilon,z})} \right)^2$$

Assume this maximum is reached at $a = a(\epsilon)$ with $\|a(\epsilon)\| = 1$. If $s(F_{\epsilon,z})$ is twice differentiable at $(0, z)$, then $a(\epsilon$ is differentiable as well. Moreover

$$\frac{\partial}{\partial \epsilon} r^2(x; F_{\epsilon,z})|_{\epsilon=0} = 2\, \frac{a(0)^t x - m(F_0^{a(0)})}{s(F_0^{a(0)})} \times \\ \left(\frac{\frac{\partial}{\partial \epsilon} a(\epsilon)^t|_{\epsilon=0}\, x - \frac{\partial}{\partial \epsilon} m(F^{a(\epsilon)}_{\epsilon,z})|_{\epsilon=0}}{s(F_0^{a(0)})} - \frac{a(0)^t x - m(F_0^{a(0)})}{s(F_0^{a(0)})^2} \frac{\partial}{\partial \epsilon} s(F^{a(\epsilon)}_{\epsilon,z})|_{\epsilon=0} \right). \tag{2.9}$$

Furthermore

$$\frac{\partial}{\partial \epsilon} m(F^{a(\epsilon)}_{\epsilon,z})|_{\epsilon=0} = \frac{\partial}{\partial \epsilon} m(F^{a(0)}_{\epsilon,z})|_{\epsilon=0} + \frac{\partial}{\partial \epsilon} m(F^{a(\epsilon)}_{0})|_{\epsilon=0}.$$

The second term is zero because of the Fisher consistency of m, whereas the first term is equal to $IF(a(0)^t z, m, F^1)$. Similarly

$$\frac{\partial}{\partial \epsilon} s(F^{a(\epsilon)}_{\epsilon,z})|_{\epsilon=0} = IF(a(0)^t z; s, F^1).$$

Finally, we use that $a(0) = \pm x/\|x\|$ [Gervini, 2002] and that $a'(0)^t a(0) = 0$ (as $a(\epsilon)^t a(\epsilon) = 1$).

□

Proof of Theorem 2.2.

Consider the contaminated distribution $F_{\epsilon,z} = (1-\epsilon)F + \epsilon \Delta_z$. Then

$$\begin{aligned} V_{so}(F_{\epsilon,z}) &= c_\alpha \left(\frac{\int_{A(F_{\epsilon,z})} x x^t dF_{\epsilon,z}(x)}{1-\alpha} - T(F_{\epsilon,z}) T(F_{\epsilon,z})^t \right) \\ &= c_\alpha \left(\frac{1-\epsilon}{1-\alpha} \int_{A(F_{\epsilon,z})} x x^t dF(x) + \frac{\epsilon}{1-\alpha} \int_{A(F_{\epsilon,z})} x x^t d\Delta_z(x) - T(F_{\epsilon,z}) T(F_{\epsilon,z})^t \right) \\ &= c_\alpha \left(\frac{1-\epsilon}{1-\alpha} \int_{A(F_{\epsilon,z})} x x^t dF(x) + \frac{\epsilon}{1-\alpha} \mathbb{1}(z \in A(F_{\epsilon,z})) z z^t - T(F_{\epsilon,z}) T(F_{\epsilon,z})^t \right). \end{aligned}$$

To find the influence function, we derive the latter expression to ϵ and evaluate at $\epsilon = 0$.

$$\begin{aligned} IF(z; V_{so}, F) = -\frac{c_\alpha}{1-\alpha}\int_{A(F)} xx^t dF(x) + \frac{c_\alpha}{1-\alpha}\frac{\partial}{\partial \epsilon}\int_{A(F_{\epsilon,z})} xx^t dF(x)|_{\epsilon=0} \\ + \frac{c_\alpha}{1-\alpha}\mathbb{1}(z \in A(F))zz^t \end{aligned}$$

or

$$IF(z; V_{so}, F) = -I + \frac{c_\alpha}{1-\alpha}\frac{\partial}{\partial \epsilon}\int_{A(F_{\epsilon,z})} xx^t dF(x)|_{\epsilon=0} + \frac{c_\alpha}{1-\alpha}\mathbb{1}(\|z\|^2 \leq q_\alpha)zz^t. \tag{2.10}$$

To further evaluate the second term, we assume z lying on the x_1-axis. Then

$$\frac{c_\alpha}{1-\alpha}\frac{\partial}{\partial \epsilon}\int_{A(F_{\epsilon,z})} xx^t dF(x)|_{\epsilon=0} \tag{2.11}$$

$$= \frac{c_\alpha}{1-\alpha}\frac{\partial}{\partial \epsilon}\int_{x1(\epsilon)}^{x2(\epsilon)} dx_1 \int_{C(\epsilon,x_1)} dx_2 \ldots dx_d \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix} \begin{pmatrix} x_1 & \ldots & x_d \end{pmatrix} g(\|x\|^2)|_{\epsilon=0} \tag{2.12}$$

for certain $x_1(\epsilon)$ and $x_2(\epsilon)$. Because of symmetry, $C(F_{\epsilon,z}, x_1)$ is a $(d-1)$-dimensional ball. Denote its radius with $r = \sqrt{q_\alpha(F_{\epsilon,z}, x_1)}$ and transform to polar coordinates as follows: $\begin{pmatrix} x_2 \\ \vdots \\ x_d \end{pmatrix} \longrightarrow re(\theta)$. Denote the Jacobian of this transformation as $J(\theta, r)$. Then the right-hand size of (2.12) can be written as

$$\begin{aligned} \frac{c_\alpha}{1-\alpha}\int_{-\sqrt{q_\alpha}}^{\sqrt{q_\alpha}} dx_1 \frac{\partial}{\partial \epsilon}\int_0^{\sqrt{q_\alpha(F_{\epsilon,z},x_1)}} dr \\ \int_\Theta d\theta\, J(\theta, r) \begin{pmatrix} x_1 \\ re(\theta) \end{pmatrix} \begin{pmatrix} x_1 & re(\theta)^t \end{pmatrix} g(x_1^2 + r^2)|_{\epsilon=0} \end{aligned}$$

which by Leibniz' rule equals

$$\begin{aligned} \frac{c_\alpha}{1-\alpha}\int_{-\sqrt{q_\alpha}}^{\sqrt{q_\alpha}} dx_1 \frac{\partial}{\partial \epsilon}\sqrt{q_\alpha(F_{\epsilon,z}, x_1)}|_{\epsilon=0}\int_\Theta d\theta\, J(\theta, \sqrt{q_\alpha(F, x_1)}) \times \\ \begin{pmatrix} x_1 \\ \sqrt{q_\alpha(F, x_1)}e(\theta) \end{pmatrix} \begin{pmatrix} x_1 & \sqrt{q_\alpha(F, x_1)}e(\theta)^t \end{pmatrix} g(x_1^2 + q_\alpha(F, x_1)). \end{aligned} \tag{2.13}$$

We thus need to evaluate $\frac{\partial}{\partial\epsilon}\sqrt{q_\alpha(F_{\epsilon,z},x_1)}|_{\epsilon=0}$. By definition of $A(F_{\epsilon,z})$ we have

$$\int_{A(F_{\epsilon,z})} dF_{\epsilon,z}(x) = 1-\alpha.$$

Deriving both sides of this equality we find after some calculations that

$$\begin{aligned}&-\int_{A(F)} dF(x) + \mathbb{1}(z\in A(F))\\ &\qquad+\int_{-\sqrt{q_\alpha}}^{\sqrt{q_\alpha}} dx_1 \frac{\partial}{\partial\epsilon}\sqrt{q_\alpha(F_{\epsilon,z},x_1)}|_{\epsilon=0}\int_\Theta d\theta\, J(\theta,\sqrt{q_\alpha(F,x_1)}\, g(q_\alpha) = 0\end{aligned} \tag{2.14}$$

Now take a point $x(\epsilon)$ on the boundary of $A(F_{\epsilon,z})$ with first coordinate x_1. Then

$$q_\alpha(F_{\epsilon,z}) = r^2(x(\epsilon),F_{\epsilon,z}).$$

This yields

$$\begin{aligned}\frac{\partial}{\partial\epsilon}q_\alpha(F_{\epsilon,z})|_{\epsilon=0} &= \frac{\partial}{\partial\epsilon}r^2(x(0),F_{\epsilon,z})|_{\epsilon=0}+\frac{\partial}{\partial\epsilon}r^2(x(\epsilon),F_{0,z})|_{\epsilon=0}\\ &= \frac{\partial}{\partial\epsilon}r^2(x(0),F_{\epsilon,z})|_{\epsilon=0}+\frac{\partial}{\partial\epsilon}\|x(\epsilon)\|^2|_{\epsilon=0}\\ &= \frac{\partial}{\partial\epsilon}r^2(x(0),F_{\epsilon,z})|_{\epsilon=0}+\frac{\partial}{\partial\epsilon}q_\alpha(F_{\epsilon,z},x_1)|_{\epsilon=0}\end{aligned} \tag{2.15}$$

and so we find

$$\frac{\partial}{\partial\epsilon}\sqrt{q_\alpha(F_{\epsilon,z},x_1)}|_{\epsilon=0} = \frac{\frac{\partial}{\partial\epsilon}q_\alpha(F_{\epsilon,z})|_{\epsilon=0}-\frac{\partial}{\partial\epsilon}r^2(x(0),F_{\epsilon,z})|_{\epsilon=0}}{2\sqrt{q_\alpha(F,x_1)}}.$$

Substituting this in (2.14), we find an expression for $\frac{\partial}{\partial\epsilon}q_\alpha(F_{\epsilon,z})|_{\epsilon=0}$. Using equality (2.15), we find after some calculations that

$$\begin{aligned}\frac{\partial}{\partial\epsilon}q_\alpha(F_{\epsilon,z},x_1)|_{\epsilon=0} &= \frac{1-\alpha-\mathbb{1}(\|z\|^2\le q_\alpha)}{\frac{\int_{-\sqrt{q_\alpha}}^{\sqrt{q_\alpha}}dx_1\sqrt{q_\alpha(F,x_1)}^{d-3}}{2}\int_\Theta d\theta\, J_\theta(\theta)\, g(q_\alpha)}\\ &+\frac{\int_{-\sqrt{q_\alpha}}^{\sqrt{q_\alpha}}dx_1\, \frac{\partial}{\partial\epsilon}r^2(x(0),F_{\epsilon,z})|_{\epsilon=0}\sqrt{q_\alpha(F,x_1)}^{d-3}}{\int_{-\sqrt{q_\alpha}}^{\sqrt{q_\alpha}}dx_1\sqrt{q_\alpha(F,x_1)}^{d-3}}-\frac{\partial}{\partial\epsilon}r^2(x(0),F_{\epsilon,z})|_{\epsilon=0}\end{aligned} \tag{2.16}$$

from which $\frac{\partial}{\partial\epsilon}\sqrt{q_\alpha(F_{\epsilon,z},x_1)}|_{\epsilon=0}$ can be computed. Plugging in this expression in (2.13) and simplifying the integrals (thereby using Lemma 2.1), finally gives us following expression for the second term in (2.10):

$$\begin{aligned}&\frac{c_\alpha}{1-\alpha}\frac{q_\alpha}{d}\left(1-\alpha-\mathbb{1}(\|z\|^2\le q_\alpha)\right)I+\frac{c_\alpha}{1-\alpha}g(q_\alpha)\left(\frac{q_\alpha d_1}{2d}\right)I\\ &-\frac{c_\alpha}{1-\alpha}g(q_\alpha)\frac{d_2}{2(d-1)}I-\frac{c_\alpha}{1-\alpha}g(q_\alpha)\frac{d_3}{2z_1^2}\begin{pmatrix} z_1^2 & 0_{1,d-1}\\ 0_{d-1,1} & 0_{d-1,d-1}\end{pmatrix}\end{aligned}$$

Thus if z lies on the x_1-axis we obtain

$$
\begin{aligned}
IF(z; V_{so}, F) = & -I + \frac{c_\alpha}{1-\alpha}\mathbb{1}(\|z\|^2 \leq q_\alpha) \begin{pmatrix} z_1^2 & 0_{1,d-1} \\ 0_{d-1,1} & 0_{d-1,d-1} \end{pmatrix} \\
& + \frac{c_\alpha}{1-\alpha}\frac{q_\alpha}{d}\Big(1-\alpha-\mathbb{1}(\|z\|^2 \leq q_\alpha)\Big) I + \frac{c_\alpha}{1-\alpha} g(q_\alpha)\left(\frac{q_\alpha d_1}{2d}\right) I \\
& - \frac{c_\alpha}{1-\alpha} g(q_\alpha)\frac{d_2}{2(d-1)} I - \frac{c_\alpha}{1-\alpha} g(q_\alpha)\frac{d_3}{2z_1^2} \begin{pmatrix} z_1^2 & 0_{1,d-1} \\ 0_{d-1,1} & 0_{d-1,d-1}. \end{pmatrix}
\end{aligned}
$$

The affine equivariance of V_{so} then leads to the result in Theorem 2.2 for general z.

□

Proof of Theorem 2.3

In a completely analogue way as in the previous proof, one can show that

$$
\begin{aligned}
IF(z; T_{so}, F) = & -\frac{1}{1-\alpha}\int_{A(0)} x dF(x) + \frac{1}{1-\alpha}\frac{\partial}{\partial \epsilon}\int_{A(F_{\epsilon,z})} x dF(x)|_{\epsilon=0} \\
& + \frac{1}{1-\alpha}\mathbb{1}(z \in A(F))z.
\end{aligned}
$$

The first term is zero due to Fisher-consistency, while the second term equals

$$
\frac{1}{1-\alpha}\int_{-\sqrt{q_\alpha}}^{\sqrt{q_\alpha}} dx_1 \quad \tfrac{\partial}{\partial \epsilon}\sqrt{q_\alpha(F_{\epsilon,z}, x_1)}|_{\epsilon=0} \int_\Theta d\theta \; J(\theta, \sqrt{q_\alpha(F, x_1)}) x_1 g(x_1^2 + q_\alpha(F, x_1))
$$

using the same notation as in (2.13). The derivative $\frac{\partial}{\partial \epsilon}\sqrt{q_\alpha(F_{\epsilon,z}, x_1)}|_{\epsilon=0}$ was already obtained in (2.16). Substituting and using the symmetry of F, one finds after some calculations the result in Theorem 2.3.

□

Proof of Theorems 2.4 and 2.5

Assume F a d-dimensional elliptical distribution with center 0 and covariance matrix $\Sigma = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$. Let us first derive $\frac{\partial}{\partial \epsilon} V_{\mathrm{robpca}}(F_{\epsilon,z})|_{\epsilon=0}$. So first we project the contaminated distribution $F_{\epsilon,z}$ onto a k-dimensional subspace. The distribution of this projection is then given by

$$
F_{\epsilon,z}^{proj} = (1-\epsilon)F_\epsilon + \epsilon \Delta_{P_k(F_{\epsilon,z})z}
$$

where F_ϵ is a k-dimensional elliptical distribution with covariance matrix $P_k(F_{\epsilon,z})\Sigma P_k(F_{\epsilon,z})^t$. Now $V_{\text{robpca}}(F_{\epsilon,z})$ is nothing but the MCD estimator of covariance applied to $F_{\epsilon,z}^{proj}$ and thus one can write (with $V_r = V_{\text{robpca}}$):

$$\begin{aligned} V_r(F_{\epsilon,z}) &= c_\alpha\{\frac{1}{1-\alpha}\int_{B(F_{\epsilon,z})} xx^t dF_{\epsilon,z}(x) - T_r(\epsilon)T_r(\epsilon)^t\} \\ &= c_\alpha\{\frac{1-\epsilon}{1-\alpha}\int_{B(F_{\epsilon,z})} xx^t dF_\epsilon(x) + \frac{\epsilon}{1-\alpha}\mathbb{1}(P_k(F_{\epsilon,z})z \in B(F_{\epsilon,z}))\, P_k(F_{\epsilon,z})zz^tP_k(F_{\epsilon,z})^t \\ &\qquad - T_r(\epsilon)T_r(\epsilon)^t\} \end{aligned}$$

with $T_r(\epsilon)$ the MCD estimator of location of $F_{\epsilon,z}^{proj}$ and $B(F_{\epsilon,z}) = \{x \in \mathbb{R}^d : (x - T_r(\epsilon))^t V_r(F_{\epsilon,z})^{-1}(x - T_r(\epsilon)) \le q_\alpha(\epsilon)\}$ for a certain positive number $q_\alpha(\epsilon)$. From the above expression, one can proceed similarly to the proof of Theorem 1 in Croux and Haesbroeck [1999]. After some calculations one then finds

$$IF(z; V_r, F) = IF(P_k(F)z; MCD, F) + \frac{\partial}{\partial\epsilon}(P_k(F_{\epsilon,z})\Sigma P_k(F_{\epsilon,z}))|_{\epsilon=0} \tag{2.17}$$

Denote $\lambda_{r,j}(F)$ and $v_{r,j}(F)$ the jth eigenvalue and eigenvector. Then

$$\begin{aligned} IF(z; \lambda_{r,j}, F) &= IF(z; V_r, F)_{jj} \\ IF(z; v_{r,j}, F)_i &= \frac{IF(z; V_r, F)_{ij}}{\lambda_j - \lambda_i}, \end{aligned}$$

see for example [Croux and Haesbroeck, 2000].
Combining (2.17) with Theorem 2.2 to calculate $\frac{\partial}{\partial\epsilon}(P_k(F_{\epsilon,z}))$ yields

$$\begin{aligned} &IF(z; \lambda_{r,j}, F) = IF(P_k(F)z; MCD, F_k)_{jj} \\ &IF(z; v_{r,j}, F)_j = 0 \\ &IF(z; v_{r,j}, F)_i = IF(P_k(F)z, v_{MCD,j}, F_k)_i - \frac{w_1(\|\Sigma^{-1/2}z\|)z_iz_j(\lambda_j + \lambda_i)}{(\lambda_j - \lambda_i)^2} \quad \text{for } i \neq j \end{aligned}$$

leading to the expressions for λ_{robpca} and v_{robpca} as in Theorems 2.4 and 2.5.

Chapter 3

Robustness and stability of reweighted kernel based regression

3.1 Introduction

Kernel Based Regression (KBR) is a popular method belonging to modern machine learning and is based on convex risk minimization. An objective function is optimized consisting of the sum of a data term and a complexity term. The data term represents the loss at the given data points. Recently the robustness of these methods was investigated with respect to outlying observations [Christmann and Steinwart, 2006]. It was found that KBR with a loss function with unbounded first derivative can be heavily affected by the smallest amount of outliers. As such a least squares loss is not a good choice from a robustness point of view, contrary to e.g. an L_1 loss or Vapnik's ϵ-insensitive loss function. From a computational point of view on the other hand, a least squares loss leads to faster algorithms solving a linear system of equations [Wahba, 1990, Evgeniou et al., 2000, Suykens et al., 2002b], whereas an L_1 loss involves solving a quadratic programming problem. In this chapter we investigate the possibility of stepwise reweighting Least Squares KBR (LS-KBR) in order to improve its robustness. This is already proposed in Suykens et al. [2002a], where data experiments show how reweighting steps reduce the effect of outliers, whereas the algorithm still only requires solving linear systems of equations.

Reweighting ideas are very common in traditional linear regression models. A theoretical analysis of the robustness of reweighted linear least squares regression is provided by Dollinger and Staudte [1991], where the influence functions of successive steps are calculated. A short summary of their results is given in Section 3.2.

Section 3.3 contains some explanation about KBR. Section 3.4 contains the main results of this chapter:

- We introduce the weighted regularized risk and show a representer theorem for its minimizer (Theorem 3.6).

- We calculate the influence function of one-step reweighted KBR (Theorem 3.7).

- Theorem 3.7 shows that the influence function after performing a reweighting step depends on a certain operator evaluated at the influence function before the reweighting step. Since our goal is to reduce the influence function (thereby improving robustness), it is important that the norm of this operator is smaller than one. Assuming a signal plus noise distribution, we are able to determine conditions on the weight function such that an operator norm smaller than one is guaranteed. This provides some practical guidelines on how to choose the weights.

- If the weight function is well chosen, it is shown that reweighted KBR with a bounded kernel converges to an estimator with a bounded influence function, even if the initial estimator is LS-KBR, which is not robust. This is an important difference compared to linear reweighted LS regression, which converges to an estimator with an unbounded influence function.

Throughout the chapter the influence function is used as a tool to assess the robustness of the methods under consideration. It reflects how an estimator changes when a tiny amount of contamination is added to the original distribution. As such it can also be seen as a measure of stability at continuous distributions: it shows how the result changes when the distribution changes slightly. This is very similar to some stability measures that were recently defined. Poggio et al. [2004] for example show that it is very important for a method not to change too much when an additional point is added to a sample. However, these stability measures typically add a point which is generated i.i.d. from the same distribution as the other points. In robust statistics the added contamination can be any possible outcome, even a very unlikely one under the generating distribution. Thus in a way robustness is a stronger requirement than stability. A robust method should give stable results when adding *any* possible point, even an extremely unlikely one.

In Section 3.5 we explore these links and differences between robustness and stability a little bit further. We show how the influence function can be used to approximate traditional stability measures by evaluating it at sample points. A smaller influence function leads to methods that are more stable. Therefore, since reweighting steps reduce the influence function, they also improve the stability of the initial LS-KBR estimator. When the error distribution is Gaussian, this effect is rather small. At heavy tailed distributions on the other hand the stability can improve quite drastically.

In Section 3.6 we discuss some practical consequences of our theoretical results. Some weight functions traditionally used in linear regression are examined. It is shown that some weight functions, e.g. Hampel weights, do not satisfy the necessary conditions. We construct an example showing that this can indeed lead to bad results in practice, in contrast to e.g. a logistic weight function satisfying all conditions.

In the same section we provide some results on the convergence speed. As explained the influence function is reduced at each step. An upper bound for this reduction factor is found. Unfortunately this factor depends on the error distribution, such that this upper bound is not distribution free. In Table 3.3 results are shown for several error distributions. Again logistic weights give good results in the most common cases.

Finally we analyze some specific data sets. The robustness of reweighted KBR is demonstrated on a data example from astronomy. A small simulation study demonstrates that reweighting leads to better stability at heavy tailed distributions.

3.2 Linear least squares regression

In this section we gather some results concerning linear regression. Consider n observations $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R},\ \ i = 1, \ldots, n$. We assume a linear relationship between the d-dimensional covariates and the 1-dimensional response variable:

$$y_i = v_0^t x_i + e_i \tag{3.1}$$

with independent errors e_i. The unknown parameter $v_0 \in \mathbb{R}^d$ can be estimated in numerous ways. The most traditional way is least squares regression, which minimizes the sum of the squared residuals. Thus, define $r_i(v) = y_i - v^t x_i$, then the least squares estimator is obtained as

$$\underset{v}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} r_i(v)^2. \tag{3.2}$$

It is however well known that this estimator is very sensitive to deviations from the model. This is also reflected in its influence function (cfr. Definition 1). Given a $(d+1)-$dimensional distribution P, we can define the functional T_l as

$$T_l(P) = \underset{v}{\operatorname{argmin}} \mathbb{E}_P (Y - v^t X)^2 \tag{3.3}$$

with (X, Y) a pair of random variables with joint distribution P. Plugging in a finite sample distribution P_n, this indeed reduces to (3.2). The influence function of T_l at the point $z = (z_x, z_y) \in \mathbb{R}^d \times \mathbb{R}$ equals (Cook and Weisberg [1982], Hinkley [1977])

$$IF(z; T_l, P) = (\mathbb{E}_P(XX^t))^{-1}(z_y - T_l(P)^t z_x) z_x. \tag{3.4}$$

This expression is clearly unbounded, meaning that an infinitesimal amount of outliers can have an arbitrarily large effect. The influence function increases both with the component in x-space, z_x, as with the residual $z_y - T_l(P)^t z_x$, showing that linear least squares regression is sensitive to vertical outliers as well as leverage points.

In an attempt to correct this bad behavior in the presence of contamination, reweighted regression can be considered. The idea is to downweight observations that have a high residual with respect to an initial estimator T_l^0. Then the one step reweighted least squares estimator T_l^1 is defined as

$$T_l^1(P) = \underset{v}{\operatorname{argmin}} \mathbb{E}_P[w(X, Y - T_l^0(P)X)(Y - v^t X)^2] \tag{3.5}$$

with $w(.,.) : (\mathbb{R}^d, \mathbb{R}) \to \mathbb{R}$ a properly chosen weight function. The influence function of this estimator was proven to be [Dollinger and Staudte, 1991]

$$IF(z; T_l^1, P) = \Sigma_w^{-1} w(z_x, z_y - v_0^t z_x)(z_y - v_0^t z_x) z_x + \Sigma_w^{-1} C_w IF(z; T_l^0, P) \tag{3.6}$$

with

$$\Sigma_w = \mathbb{E}_P[w(X, Y - v^t X)XX^t] \text{ and } C_w = -\mathbb{E}_P[(Y - v^t X)\frac{\partial w}{\partial r}(X, Y - v^t X)XX^t]$$

where $\frac{\partial w}{\partial r}$ denotes the derivative of $w(.,.)$ with respect to its second argument. In equation (3.6) the influence of the initial estimator T_l^0 is still present. However, iteratively repeating the reweighting process, convergence is shown in Dollinger and Staudte [1991] under appropriate conditions. Their final result yields

$$IF(z; T_l^\infty, P) = (\Sigma_w - C_w)^{-1} w(z_x, z_y - v_0^t z_x)(z_y - v_0^t z_x)z_x. \tag{3.7}$$

In order to obtain a bounded influence function in (3.7), the weight function $w(.,.)$ should decrease in both its arguments:

$$\|w(x,r)xr\| \text{ bounded for all } (x,r) \in \mathbb{R}^d \times \mathbb{R}. \tag{3.8}$$

Thus commonly used reweighting schemes based on the residuals only, are not sufficient to bound the influence function. It is absolutely necessary to downweight observations outlying in x-space as well, which can be quite hard in high dimensions $d \to \infty$ due to the curse of dimensionality.

Moreover, even when condition (3.8) is satisfied, it is well known that the resulting estimator has a low breakdown point, meaning that it can only resist small fractions of outliers. This led to the insight that reweighted linear least squares regression is insufficiently robust. A vast list of robust regression methods was proposed ever since, combining bounded influence functions with high break down values, e.g. Least Median of Squares [Rousseeuw, 1984], Least Trimmed Squares [Rousseeuw and Leroy, 1987], S-estimators of regression [Rousseeuw and Yohai, 1984], MM-estimators [Yohai, 1987] and τ-estimators [Yohai and Zamar, 1988].

3.3 Kernel based regression

The goal of this chapter is to extend the results of Dollinger and Staudte [1991] from the previous section to the case of kernel based regression (KBR). These methods estimate a functional relationship between a covariate random variable X and a response variable Y, using a sample of n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^d \times \mathbb{R}$ with joint distribution P.

Let $L : \mathcal{Y} \times \mathbb{R} \to [0, \infty)$ be a function which is convex with respect to its second argument. Then KBR methods minimize the *empirical regularized risk*

$$\hat{f}_{n,\lambda} := \underset{f \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda \|f\|_{\mathcal{H}}^2 \tag{3.9}$$

where λ is a regularization parameter and $\mathcal{H}$ is the reproducing kernel Hilbert space of a kernel K as in Definition 2, see for example Wahba [1990] or Evgeniou et al. [2000].

Results about the form of the solution of KBR methods are known as representer theorems. A well known result in the literature of statistical learning shows that

$$\hat{f}_{n,\lambda} = \sum_{i=1}^{n} \alpha_i \Phi(x_i). \tag{3.10}$$

The form of the coefficients α_i however strongly depends on the loss function. For the squared loss $L(y,t) = (y-t)^2$, Tikhonov and Arsenin [1977] already characterized the coefficients α_i as solutions of a system of linear equations. For arbitrary convex differentiable loss functions, like the logistic loss $L(y,t) = -\log(4) + |r| + 2\log(1 + e^{-|r|})$, the α_i are the solution of a system of algebraic equations (Girosi [1998], Wahba [1999], Schölkopf et al. [2001]). For arbitrary convex, but possibly non differentiable loss functions, extensions were obtained by Steinwart [2003] and DeVito et al. [2004].

In practice the variational problem (3.9) and its representation (3.10) are closely related to the methodology of Support Vector Machines. This method formulates a primal optimization problem and solves it via a corresponding dual formulation. Vapnik [1995] extended this approach to the regression setting introducing Support Vector Regression (SVR) using the ϵ-insensitive loss function. A dual problem similar to (3.10) is solved, where the coefficients α_i are obtained from a quadratic programming problem. A least squares loss function however leads to a linear system of equations, generally easier to solve (see e.g. Suykens et al. [2002b], where primal-dual problems are formulated, including a bias term as well). On the other hand, real data examples showed that Least Squares Support Vector Regression is more affected by outlying observations. Suykens et al. [2002a] therefore introduced a reweighting step, trying to downweight malicious effects outliers can have.

In this chapter we want to investigate the theoretical properties of such reweighting steps by means of the *theoretical regularized risk*, an extension of (3.9) for continuous distributions. We introduce a reweighted version of this theoretical regularized risk and analyze convergence and robustness properties. An important aspect is the choice of the weights. We find some conditions that need to be satisfied in order to obtain good robustness. Some data examples show that these conditions should also be kept in mind when modelling data with reweighted LS-SVR.

3.4 Robustness

First we recall some results about ordinary unweighted KBR.

3.4.1 Unweighted KBR

For our theoretical results we will look at the minimization of the theoretical regularized risk

$$f_{P,\lambda} := \underset{f\in\mathcal{H}}{\operatorname{argmin}}\, \mathbb{E}_P L(Y, f(X)) + \lambda \|f\|^2_{\mathcal{H}}. \tag{3.11}$$

It is clear that the empirical regularized risk (3.9) is a stochastic approximation of the theoretical regularized risk.

Before we give some theoretical results, we need two definitions. Firstly we describe the growth behavior of the loss function [Christmann and Steinwart, 2006].

Definition 3.1 *Let $L : \mathcal{Y} \times \mathbb{R} \to [0, \infty)$ be a loss function, $a : \mathcal{Y} \to [0, \infty)$ be a measurable function and $p \in [0, \infty)$. We say that L is a loss function of type (a, p) if there exists a constant $c > 0$ such that*

$$L(y, t) \leq c\,(a(y) + |t|^p + 1)$$

for all $y \in \mathcal{Y}$ and all $t \in \mathbb{R}$. Furthermore we say that L is of strong type (a, p) if the first two partial derivatives $L' := \partial_2 L$ and $L'' := \partial_{22} L$ of L with repect to the second argument of L exist and L, L' and L'' are of (a, p)-type.

Secondly we recall the notion of subdifferentials (see e.g. Phelps [1986]).

Definition 3.2 *Let $\mathcal{H}$ be a Hilbert space, $F : \mathcal{H} \to \mathbb{R} \cup \{\infty\}$ be a convex function and $g \in \mathcal{H}$ with $F(g) \neq \infty$. Then the subdifferential of F at g is defined by*

$$\partial F(g) := \{g^* \in \mathcal{H} : \langle g^*, u - g^* \rangle \leq F(u) - F(g) \text{ for all } u \in \mathcal{H}\}.$$

Furthermore, if L is a convex loss function, we denote the subdifferential of L with respect to the second variable by $\partial_2 L$.

Finally we also need the following definition about the distribution P.

Definition 3.3 *Let P be a distribution on $\mathcal{X} \times \mathcal{Y}$ with total variation $|P|$ and $a : \mathcal{Y} \to [0, \infty)$ be a measurable function. Then we write*

$$|P|_a := \int_{\mathcal{X}\times\mathcal{Y}} a(y) dP(x, y).$$

In DeVito et al. [2004] the following representation of the theoretical regularized risk was proven.

Proposition 3.4 *Let $p \geq 1$, L be a convex loss function of type (a, p), and P be a distribution on $\mathcal{X} \times \mathcal{Y}$ with $|P|_a < \infty$. Let $\mathcal{H}$ be the RKHS of a bounded, continuous kernel K over $\mathcal{X}$, and $\Phi : \mathcal{X} \to \mathcal{H}$ be the feature map of $\mathcal{H}$. Then there exists an $h \in L_{p'}(P)$ such that $h(x, y) \in \partial_2 L(y, f_{P,\lambda}(x))$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and*

$$f_{P,\lambda} = -\frac{1}{2\lambda} \mathbb{E}_P [h\Phi]. \tag{3.12}$$

Note that the notion of subdifferential reduces to the derivative if the loss function is differentiable. Moreover the least squares loss function is of type $(y^2, 2)$ and thus the previous proposition basically simplifies to

$$f_{P,\lambda} = \frac{1}{\lambda} \mathbb{E}_P [(Y - f_{P,\lambda}(X))\Phi(X)] \tag{3.13}$$

for all distributions P with finite second moment.

Now consider the map T which assigns to every distribution P on a given set Z, the function $T(P) = f_{P,\lambda} \in \mathcal{H}$. Then the following expression for the influence function of T was proven in Christmann and Steinwart [2006].

Proposition 3.5 *Let $\mathcal{H}$ be a RKHS of a bounded continuous kernel K on $\mathcal{X}$ with feature map $\Phi : \mathcal{X} \to \mathcal{H}$, and $L : \mathcal{Y} \times \mathbb{R} \to [0, \infty)$ be a convex loss function of some strong type (a, p). Furthermore, let P be a distribution on $\mathcal{X} \times \mathcal{Y}$ with $|P|_a < \infty$. Then the influence function of T exists for all $z := (z_x, z_y) \in \mathcal{X} \times \mathcal{Y}$*

and we have

$$IF(z;T,P) = S^{-1}\left(\mathbb{E}_P[L'(Y,f_{P,\lambda}(X))\Phi(X)]\right) - L'(z_y,f_{P,\lambda}(z_x))S^{-1}\Phi(z_x)$$

where $S : \mathcal{H} \to \mathcal{H}$ *is defined by*

$$S(f) = 2\lambda f + \mathbb{E}_P\left[L''(Y,f_{P,\lambda}(X))\langle\Phi(X),f\rangle\Phi(X)\right].$$

Note that the influence function only depends on z through the term

$$-L'(z_y,f_{P,\lambda}(z_x))S^{-1}\Phi(z_x).$$

From a robustness point of view it is important to bound the influence function. The previous proposition shows that this can be achieved using a bounded kernel, e.g. the Gaussian RBF kernel, and a loss function with bounded first derivative, e.g. the logistic loss. The least squares loss function on the other hand leads to an unbounded influence function, meaning that an infinitesimal amount of contamination can totally ruin the estimator.

However, reweighting might improve the robustness of LS-KBR. In the next section we will extend the previous results to the case of reweighted KBR.

Remark: For the special case of the least squares loss function, we provide a slight extension of Proposition 3.5, including an intercept term. For reasons of simplicity we will however not include this intercept term anymore further on and continue working with the functional part only, as in Christmann and Steinwart [2006].

3.4.2 Reweighted KBR

Let $f_{P,\lambda}^{(0)} \in \mathcal{H}$ be an initial fit, e.g. obtained by ordinary unweighted LS-KBR. Let $w(x, y - f_{P,\lambda}^{(0)}(x)) : \mathbb{R}^d \times \mathbb{R} \to [0,1]$ be a weight function, depending on the covariate x and the residual $y - f_{P,\lambda}^{(0)}(x)$ with respect to the initial fit. We will make the following assumptions about w from now on:

(w_1) $w(x,r)$ a non-negative bounded Borel measurable function on $\mathbb{R}^d \times \mathbb{R}$.

(w_2) w an even function of r.

(w_3) w continuous and differentiable with respect to its second argument r, denoting this derivative by $\dfrac{\partial}{\partial r}w$.

Then we can minimize the weighted regularized risk, a weighted version of (3.11).

$$f_{P,\lambda}^{(1)} := \operatorname*{argmin}_{f\in\mathcal{H}} \mathbb{E}_P \left[w(X, Y - f_{P,\lambda}^{(0)}(X)) L(Y, f(X)) \right] + \lambda \|f\|_{\mathcal{H}}^2. \tag{3.14}$$

The following theorem can be derived from Proposition 3.4 (for full proofs we refer to Section 3.8).

Theorem 3.6 *Let $p \geq 1$, L be a convex loss function of type (a, p), and P be a distribution on $\mathcal{X} \times \mathcal{Y}$ with $|P|_a < \infty$. Let $\mathcal{H}$ be the RKHS of a bounded, continuous kernel K on $\mathcal{X}$, and $\Phi : \mathcal{X} \to \mathcal{H}$ be the feature map of $\mathcal{H}$. Then there exists an $h \in L_{p'}(P)$ such that $h(x, y) \in \partial_2 L(y, f_{P,\lambda}^{(1)}(x))$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and*

$$f_{P,\lambda}^{(1)} = -\frac{1}{2\lambda} \mathbb{E}_P \left[w(X, Y - f_{P,\lambda}^{(0)}(X)) h\Phi \right]. \tag{3.15}$$

Using this representation we can now calculate the influence function of one step reweighted KBR. Denote by T_1 the map $T_1(P) = f_{P,\lambda}^{(1)}$.

Theorem 3.7 *Denote T_0 the map $T_0(P) = f_{P,\lambda}^{(0)} \in \mathcal{H}$ with $\mathcal{H}$ a RKHS of a bounded continuous kernel K on $\mathcal{X}$ with feature map $\Phi : X \to \mathcal{H}$, and $L : \mathcal{Y} \times \mathbb{R} \to [0, \infty)$ be a convex loss function of some strong type (a, p). Furthermore, let P be a distribution on $\mathcal{X} \times \mathcal{Y}$ with $|P|_a < \infty$ and $\int_{\mathcal{X}\times\mathcal{Y}} w(x, y - f_{P,\lambda}^{(0)}(x)) dP(x, y) > 0$. Then the influence function of T_1 exists for all $z := (z_x, z_y) \in \mathcal{X} \times \mathcal{Y}$ and we have*

$$\begin{aligned} IF(z; T_1, P) &= S_w^{-1}(\mathbb{E}_P w(X, Y - f_{P,\lambda}^{(0)}(X)) L'(Y, f_{P,\lambda}^{(1)}(X)) \Phi(X)) \\ &+ S_w^{-1}(C_w(IF(z; T_0, P))) - w(z_x, z_y - f_{P,\lambda}^{(0)}(z_x)) L'(z_y, f_{P,\lambda}^{(1)}(z_x)) S_w^{-1}(\Phi(z_x)). \end{aligned}$$

with $S_w : \mathcal{H} \to \mathcal{H}$,

$$S_w(f) = 2\lambda f + \mathbb{E}_P[w(X, Y - f_{P,\lambda}^{(0)}(X)) L''(Y, f_{P,\lambda}^{(1)}(X)) \langle f, \Phi(X) \rangle \Phi(X)]$$

and $C_w : \mathcal{H} \to \mathcal{H}$,

$$C_w(f) = -\mathbb{E}_P[\frac{\partial}{\partial r} w(X, Y - f_{P,\lambda}^{(0)}(X)) L'(Y, f_{P,\lambda}^{(1)}(X)) \langle f, \Phi(X) \rangle \Phi(X)].$$

Note that the expression for $IF(z; T_1, P)$ consists of three terms. The first one is a constant function independent of z, i.e. it does not depend on the position z where we plug in the contamination. The second term $(S_w^{-1} \circ C_w)(IF(z; T_0, P))$

reflects the influence of the initial KBR estimate. Since S_w and C_w are operators independent of z, this term can be unbounded if the influence function of the initial estimator is unbounded, which is the case for e.g. LS-KBR. However, it is possible that this influence of the initial fit is reduced because the operator $S_w^{-1} \circ C_w$ is applied on it. In that case, the second term might vanish if we keep reweighting until convergence. To investigate this iterative reweighting, we will however make some simplifying assumptions. First of all we will restrict ourselves to the case of a least squares loss function, taking $L(y,t) = (y-t)^2$, $y, t \in \mathbb{R}$. Next we examine three cases. In a first case we assume that the regularization parameter $\lambda = 0$ and we investigate the effect of the kernel on the reweighting. Secondly we will consider the case $\lambda > 0$, but only for special distributions P. Nevertheless this gives us good insight in the effect of the regularization on the reweighting. Finally we spend some time discussing the general case $\lambda \geq 0$ for arbitrary distributions P.

Case 1: $\lambda = 0$

In this section we assume that the distribution P follows a classical regression setting. This means that (i) a function $f_P \in \mathcal{H}$ exists such that the conditional mean $\mathbb{E}_P(Y|x)$ of the response Y given $x \in \mathbb{R}^d$ equals $f_P(x)$, (ii) the error $e = Y - f(X)$ is independent of X and (iii) the distribution P_e of these errors is symmetric about 0 with finite second moment. For such distributions P, it is easy to see that LS-KBR with $\lambda = 0$ is Fisher consistent, meaning that $f_{P,0} = f_P$ with (see equation (3.11))

$$f_{P,0} = \operatorname*{argmin}_{f \in \mathcal{H}} \mathbb{E}_P (Y - f(X))^2.$$

Moreover one step reweighted LS-KBR is also Fisher consistent (see Section 3.8 for proof):

$$f_{P,0}^{(1)} = \operatorname*{argmin}_{f \in \mathcal{H}} \mathbb{E}_P [w(X, Y - f_{P,0}(X))(Y - f(X))^2] = f_P. \tag{3.16}$$

Now we can define $k+1$-step reweighted LS-KBR:

$$f_{P,0}^{(k+1)} = \operatorname*{argmin}_{f \in \mathcal{H}} \mathbb{E}_P \left[w(X, Y - f_{P,0}^{(k)}(X))(Y - f(X))^2 \right]$$

which is Fisher consistent as well for every $k \in \mathbb{N}$, thus we have that $f_{P,0}^{(k)} = f_P$ for every $k \in \mathbb{N}$. Then we have the following theorem.

Theorem 3.8 *Denote T_0 the map $T_0(P) = f_{P,0} \in \mathcal{H}$ and $T_{k+1}(P)$ the map $T_{k+1}(P) = f_{P,0}^{(k+1)} \in \mathcal{H}$ with $\mathcal{H}$ a RKHS of a bounded continuous kernel K on X with feature map $\Phi : \mathcal{X} \to \mathcal{H}$. Then the influence function of T_{k+1} exists for all $z := (z_x, z_y) \in X \times Y$ and we have*

$$\begin{aligned} IF(z; T_{k+1}, P) = {} & S_w^{-1}(C_w(IF(z; T_k, P))) \\ & + w(z_x, z_y - f_P(z_x))(z_y - f_P(z_x))S_w^{-1}(\Phi(z_x)) \end{aligned}$$

with

$$S_w : \mathcal{H} \to \mathcal{H},\ \ S_w(f) = \mathbb{E}_P[w(X, Y - f_P(X))\langle f, \Phi(X)\rangle \Phi(X)]$$

and

$$C_w : \mathcal{H} \to \mathcal{H},\ \ C_w(f) = -\mathbb{E}_P[\frac{\partial}{\partial r} w(X, Y - f_P(X))(Y - f_P(X))\langle f, \Phi(X)\rangle \Phi(X)].$$

Theorem 3.8 gives a first order recursive relationship between $IF(z; T_{k+1}, P)$ and $IF(z; T_k, P)$, which is easy to solve. Denote $h(z)$ the second part of the expression for $IF(z; T_{k+1}, P)$ in Theorem 3.8, independent of $IF(z; T_k, P)$. Thus

$$h(z) := w(z_x, z_y - f_P(z_x))(z_y - f_P(z_x))S_w^{-1}(\Phi(z_x)).$$

Then we see that

$$IF(z; T_{k+1}, P) = \sum_{i=0}^{k} (S_w^{-1} \circ C_w)^i h(z) + (S_w^{-1} \circ C_w)^{k+1} IF(z; T_0, P).$$

If the operator norm of $(S_w^{-1} \circ C_w)$ is smaller then one, the second term will vanish as $k \to \infty$. The first term will converge to $(1 - (S_w^{-1} \circ C_w))^{-1}(h(z)) = (S_w - C_w)^{-1} S_w(h(z))$. This leads to the following Corollary.

Corollary 3.9 *Let P, $\mathcal{H}$, T_k, S_w and C_w be as in Theorem 3.8. Denote $T_\infty = \lim_{k\to\infty} T_k$. Then, if $\|S_w^{-1} \circ C_w\| < 1$, the influence function of T_∞ at P exists and equals*

$$IF(z; T_\infty, P) = (S_w - C_w)^{-1}(w(z_x, z_y - f_P(z_x))(z_y - f_P(z_x))\Phi(z_x)).$$

A first important conclusion concerns the boundedness of this expression. Since the operators S_w and C_w are independent of the contamination z, the influence function $IF(z; T_\infty, P)$ is bounded if

$$\|w(x, r) r \Phi(x)\|_{\mathcal{H}} \text{ is bounded } \forall (x, r) \in \mathbb{R}^d \times \mathbb{R}. \tag{3.17}$$

If we take Φ the feature map of a linear kernel, this corresponds to condition (3.8) for ordinary linear least squares regression. In that case, the weight function should decrease with the residual r as well as with x, to obtain a bounded influence.

This is also true for other unbounded kernels, e.g. polynomial, but not for non-linear function estimation using a bounded kernel, like the popular RBF kernel for instance. The latter only requires downweighting the residual, as the influence in x-space is controlled by the kernel. This shows that LS-KBR with a bounded kernel is much more suited for iterative reweighting than linear least squares regression, see also Theorem 4 in Christmann and Steinwart [2004] for classification and Corollary 19 in Christmann and Steinwart [2006] for the regression case.

Let us now restrict ourselves to a weight function independent of x.

$$w(x,r) = \frac{\psi(r)}{r} \text{ with } \psi : \mathbb{R} \to \mathbb{R} \text{ a bounded, real, odd function.}$$

From Corollary 3.9 we know that this is sufficient to bound the influence function of iteratively reweighted LS-KBR with a bounded kernel, if convergence takes place, that is if $\|S_w^{-1} \circ C_w\| < 1$.

From its definition in Theorem 3.8, we know that

$$S_w = \mathbb{E}_P[w(X, Y - f_P(X))\langle ., \Phi(X)\rangle\Phi(X)].$$

Using the assumed regression structure of P, we can decompose P in the error distribution P_e of the errors $e = Y - f_p(X)$ and the distribution P_X of X such that $dP = dP_X dP_e$. This yields

$$S_w = \mathbb{E}_P[\frac{\psi(e)}{e}\langle ., \Phi(X)\rangle\Phi(X)].$$

Defining $d := \mathbb{E}_{P_e}\frac{\psi(e)}{e}$ we have that

$$S_w = d\ \mathbb{E}_{P_X}[\langle ., \Phi(X)\rangle\Phi(X)].$$

Note that d always exists since we assumed errors with finite second moment. Some analogous calculations give a similar result for the operator C_w.

$$C_w = c\ \mathbb{E}_{P_X}\langle ., \Phi(X)\rangle\Phi(X) \text{ with } c := d - E_{P_e}\psi'(e).$$

Thus, denoting $\mathrm{id}_{\mathcal{H}}$ the identity operator in $\mathcal{H}$ such that $\mathrm{id}_{\mathcal{H}}(f) = f$ for all $f \in \mathcal{H}$, we obtain

$$S_w^{-1} \circ C_w = \frac{c}{d}\ \mathrm{id}_{\mathcal{H}} \tag{3.18}$$

showing that the condition $\|S_w^{-1} \circ C_w\| < 1$ is satisfied if $c < d$, meaning that

$$\mathbb{E}_{P_e} \psi'(e) > 0.$$

Since this condition depends on the error distribution P_e, a stronger but distribution free assumption might be useful, for example taking ψ a strictly increasing function, as is often used in the context of M-estimates in linear regression.

Summarizing the previous results we can state: in case of distributions P with a regression structure as defined in the beginning of this section, the influence function of iteratively reweighted LS-KBR with bounded kernel, $\lambda = 0$ and weight function $w(x,r) = \frac{\psi(r)}{r}$ converges to a bounded function if

$$\begin{aligned}
&(c1)\ \ \psi : \mathbb{R} \to \mathbb{R} \text{ is a measurable, real, odd function.}\\
&(c2)\ \ \psi \text{ is continuous and differentiable.}\\
&(c3)\ \ \psi \text{ is bounded.}\\
&(c4)\ \ \mathbb{E}_{P_e}\psi'(e) > 0. \qquad (c4')\ \ \psi \text{ is strictly increasing.}
\end{aligned} \tag{3.19}$$

When using unbounded kernels such as linear or polynomial, this is not sufficient: the weight function $w(x,r)$ should also decrease with x.

Case 2: $f_P = 0$

The previous results were obtained for LS-KBR with $\lambda = 0$. In general LS-KBR with $\lambda > 0$ is not necessarily Fisher-consistent and therefore the theorems from the previous section are hard to extend. Nevertheless, to gain some insight into these methods with $\lambda > 0$, we will have a look at the situation where $f_P \equiv 0$. For such distributions, we also have that $f_{P,\lambda}^{(k)} = f_P$ for every k. Through similar calculations as in the previous section, one can see that Theorem 3.8 and Corollary 3.9 are still valid, if we define S_w as follows:

$$S_w : \mathcal{H} \to \mathcal{H},\ \ S_w(f) = \lambda \,\mathrm{id}_{\mathcal{H}} + \mathbb{E}_P[w(X,Y)\langle f, \Phi(X)\rangle \Phi(X)].$$

Comparing with Theorem 3.8 in the case $\lambda = 0$, we see that a term $\lambda \,\mathrm{id}_{\mathcal{H}}$ comes into play. Due to this term, equation (3.18) is not valid anymore. However, after a non-trivial calculation using a spectral theorem (see Section 3.8), the operator norm of $S_w^{-1} \circ C_w$ is found to be equal to

$$\|S_w^{-1} \circ C_w\| = \frac{c}{d + \lambda} \tag{3.20}$$

where c and d are the same constants as in (3.18). Since λ is positive, we see that this norm is smaller than 1 if $c < d$, which is exactly the condition found in the case $\lambda = 0$. Now we observe that taking $\lambda > 0$ only relaxes this condition, at least to $c \leq d$. We can thus relax condition $(c4)$ from (3.19) as well.

$$(c4)\ \ E_{P_e}\psi'(e) > -\lambda \qquad (c4'')\ \ \psi \text{ is increasing.} \tag{3.21}$$

We conclude that a positive generalization parameter λ improves the convergence of iteratively reweighted LS-KBR. This is plausible, since higher values of λ will lead to smoother fits. Then the method will be less attracted towards an outlier in y-direction, indeed leading to better robustness.

General case

Consider an arbitrary distribution P and $\lambda \geq 0$ satisfying all conditions in Theorem 3.7. From Theorem 3.7 it follows that

$$IF(z;T_{k+1},P) = S_{w,k}^{-1}(\mathbb{E}_P w(X,Y-f_{P,\lambda}^{(k)}(X))L'(Y,f_{P,\lambda}^{(k+1)}(X))\Phi(X))$$
$$+ S_{w,k}^{-1}(C_{w,k}(IF(z;T_0,P))) - w(z_x,z_y-f_{P,\lambda}^{(k)}(z_x))L'(z_y,f_{P,\lambda}^{(k+1)}(z_x))S_{w,k}^{-1}\Phi(z_x)$$

with $S_{w,k} : \mathcal{H} \to \mathcal{H}$,

$$S_{w,k}(f) = 2\lambda f + \mathbb{E}_P[w(X,Y-f_{P,\lambda}^{(k)}(X))L''(Y,f_{P,\lambda}^{(k+1)}(X))\langle f,\Phi(X)\rangle\Phi(X)]$$

and $C_{w,k} : \mathcal{H} \to \mathcal{H}$,

$$C_{w,k}(f) = -\mathbb{E}_P[\frac{\partial}{\partial r}w(X,Y-f_{P,\lambda}^{(k)}(X))L'(Y,f_{P,\lambda}^{(k+1)}(X))\langle f,\Phi(X)\rangle\Phi(X)].$$

Note that the operators $S_{w,k}$ and $C_{w,k}$ depend on the subscript k. Thus the operator $S_{w,k}^{-1} \circ C_{w,k}$ that acts on the influence function of the initial estimator can be different in each step. In the previous sections we were able to get rid of this subscript due to our assumptions, and more specifically because in those cases $f_{P,\lambda}^{(k)}$ was constant over k. In general this is not the case. We can however assume that $f_{P,\lambda}^{(k)}$ converges to a certain function $f_{P,\lambda}^{(\infty)}$. If this assumption does not hold, it means that the reweighted KBR does not converge at the uncontaminated distribution P. Hence there is not much scope for convergence results at contaminated distributions. If the series $f_{P,\lambda}^{(k)}$ indeed converges, then the series of operators $S_{w,k}^{-1} \circ C_{w,k}$ also converges as k goes to infinity. For k large enough results such as (3.20) extend for $S_{w,k}^{-1} \circ C_{w,k}$. Thus, in the first few reweighting steps, one might see a behavior that contradicts the results from the previous sections. After enough reweighting steps however they should remain valid.

3.5 Stability

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, denote P_n the empirical distribution associated with a sample $Z_n = \{z_j = (x_j, y_j) \in \mathbb{R}^d \times \mathbb{R},\ j = 1, \ldots, n\}$ of size n, then one can consider

$$D_i = |L(y_i, T(P_n)(x_i)) - L(y_i, T(P_n^i)(x_i))|$$

with P_n^i the empirical distribution of the sample Z_n without the ith observation z_i. Poggio et al. [2004] call the map T CV$_{loo}$-stable if

$$\sup_{i=1,\ldots,n} D_i \to 0 \tag{3.22}$$

for $n \to \infty$. They show under mild conditions that CV$_{loo}$-stability is required to achieve generalization, meaning that the empirical error of the estimate converges to the expected error [Poggio et al., 2004].

The influence function actually measures something very similar. Recall that this function is defined as

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

It measures how the result of a learning map changes as the original distribution P is changed by adding a small amount of contamination at the point z. In robust statistics it is important to bound the influence function over *all possible points* z in the support of P. This is a major difference with stability, where the supremum is taken over n points *sampled i.i.d. from the distribution* P (as in (3.22)).

This however suggests a possible approach to analyze stability using the influence function: by evaluating it at n sample points only. For an easy heuristic argument, take $z = z_i$, $P = P_n^i$ and $\epsilon = 1/n$ in the definition of the influence function above. Then for large n we have that

$$IF(z_i; T, P) \approx \frac{T(P_n) - T(P_n^i)}{1/n}.$$

Then it is easy to see that

$$|L(y_i, T(P_n)(x_i) - L(y_i, T(P_n^i)(x_i))| \approx |L'(y_i, T(P)(x_i))| \frac{|IF(z_i; T, P)|}{n}. \quad (3.23)$$

As such the influence function can be used in a first order approximation to the quantity D_i which is so important in the concept of CV$_{loo}$-stability. The influence function *evaluated at the sample point* z_i should be small for every i in order to obtain stability. Assume that the loss has a bounded first derivative. From equation (3.23) one might define a new stability criterion, in the spirit of (3.22) but based on the influence function, as follows:

$$\sup_{i \in \{1,\ldots,n\}} \frac{|IF(z_i; T, P)|}{n} \to 0. \quad (3.24)$$

If a method is robust, then its influence function is bounded over *all possible points* z in the support of P and thus (3.24) is obviously satisfied. As such robustness is in a sense a strictly stronger requirement than stability. Robustness can be interpreted as adding *any* point, even points that are very unlikely under the sampling distribution P.

Consider for example unweighted KBR. Recall from Proposition 3.5 that for any $z = (z_x, z_y)$

$$IF(z; T, P) = S^{-1} \left(\mathbb{E}_P[L'(Y, f_{P,\lambda}(X))\Phi(X)] \right) - L'(z_y, f_{P,\lambda}(z_x)) S^{-1} \Phi(z_x).$$

If the first derivative of the loss function L is bounded, this influence function is bounded as well and KBR is then automatically stable as well. For KBR with a squared loss, the influence function is unbounded. Despite this lack of robustness, LS-KBR is stable as long as the distribution P is not too heavy tailed. For example in case of a signal plus noise distribution with Gaussian distributed noise, $\sup_{i=1,\ldots,n}(y_i - T(P)(x_i))$ converges to ∞ as n grows larger. For Gaussian distributed noise however, this convergence will only be at logarithmic speed. Thus the convergence of (3.24) is of the order $O(\frac{\log(n)}{n})$ and (3.24) obviously still holds. For a more heavy tailed noise distribution on the other hand, the rate of stability might be much slower than $O(\frac{\log(n)}{n})$.

Since reweighted KBR has a bounded influence function, its rate of stability is always $O(\frac{1}{n})$. Reweighting steps are thus not only helpful when outliers are present in the data. They also lead to a more stable method, especially at heavy tailed distributions.

Table 3.1 links some of these concepts from robustness and stability. The

	Influence function	Leave-one-out
Robustness	$\sup_z \lvert IF(z;T,P)\rvert$ bounded	$\sup_i\{\sup_z D_i^z\} \to 0$
	$\Downarrow$	$\Downarrow$
Stability	$\sup_{z_i} \lvert IF(z_i;T,P)\rvert/n \to 0$	$\sup_i D_i \to 0$

Table 3.1 Overview of some robustness and stability concepts

influence function originated in robust statistics as a tool to assess the robustness of statistical methods (upper left cell of Table 3.1). The leave-one-out error on the other hand is often used in statistical learning to assess the stability of a learning map (lower right cell of Table 3.1). In equation (3.24) we combined both ideas using the influence function to assess stability (lower left cell of the table). In order to complete the table, the question raises whether a leave-one-out criterion can be formulated to assess robustness. Define $P_n^{z,i}$ the sample P_n where the point z_i is replaced by z and

$$D_i^z = |L(y_i, T(P_n^{z,i})(x_i)) - L(y_i, T(P_n^i)(x_i))|.$$

Then of course $D_i^{z_i} = D_i$, since taking $z = z_i$ returns the original sample P_n. Thus CV$_{loo}$ stability (3.22) can be written as

$$\sup_{i=1,\dots,n} D_i^{z_i} \to 0.$$

Now since robustness is concerned with the effect of adding any point z, not only sample points, a possible definition of robustness is

$$\sup_{i=1,\dots,n} \{\sup_z D_i^z\} \to 0.$$

This could be a sample counterpart for the classical approach of 'bounding the influence function' in robust statistics, completing Table 3.1 with the upper right cell.

In this chapter we restricted ourselves to the first column of Table 3.1 where it is clear that small influence functions lead to more stable methods. We showed that reweighting steps bound the influence function of LS-KBR and as a consequence the stability is improved as well.

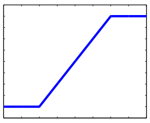
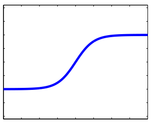
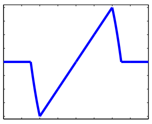
	Huber	logistic	Hampel
$w(r)$	$1 \quad \text{if } \lvert r\rvert < \beta$ $\frac{\beta}{\lvert r\rvert} \quad \text{if } \lvert r\rvert \geq \beta$	$\frac{\tanh(r)}{r}$	$1 \quad \text{if } \lvert r\rvert < b_1$ $\frac{b_2-\lvert r\rvert}{b_2-b_1} \quad \text{if } b_1 \leq \lvert r\rvert \leq b_2$ $0 \quad \text{if } \lvert r\rvert > b_2$
$\psi(r)$			
$L(r)$	$r^2 \quad \text{if } \lvert r\rvert < \beta$ $\beta\lvert r\rvert \quad \text{if } \lvert r\rvert \geq \beta$	$r\tanh(r)$	$r^2 \quad \text{if } \lvert r\rvert < b_1$ $\frac{b_2 r^2-\lvert r^3\rvert}{b_2-b_1} \quad \text{if } b_1 \leq \lvert r\rvert \leq b_2$ $0 \quad \text{if } \lvert r\rvert > b_2$

Table 3.2 Definitions for Huber, logistic and Hampel weight functions. Only the logistic weight function satisfies all conditions (c1)-(c4).

3.6 Examples

3.6.1 Weight functions

Many weight functions have been described in the literature, especially for linear regression. We show three of them in Table 3.2, with corresponding functions $w(r)$, $\psi(r)$ and loss function $L(r)$. Note that only the logistic weight function satisfies all conditions $(c1) - (c4)$ in (3.19). Huber's weight function does not satisfy $(c4)$ as ψ is not strictly increasing. Simpson et al. [1992] show that this can lead to unstable behavior of M-estimators in linear models. It does however satisfy condition $(c4'')$ in (3.21). The third weight function in Table 3.2 is Hampel's suggestion for linear least squares. These weights were also used in the context of least squares support vector regression by Suykens et al. [2002a]. In this case ψ satisfies condition $(c4')$ nor $(c4'')$, but condition $(c4)$ is valid for common error distributions, i.e. normally distributed errors. Also note that the resulting loss function is not convex anymore for these Hampel weights. Although this still leads to satisfactory results in many examples, bad fits may occur occasionally. In Figure 3.1 some data points were simulated including three outliers. Ordinary LS-SVR (dashed curve) is clearly affected by the outlying observations. Reweighting using a logistic weight function (solid

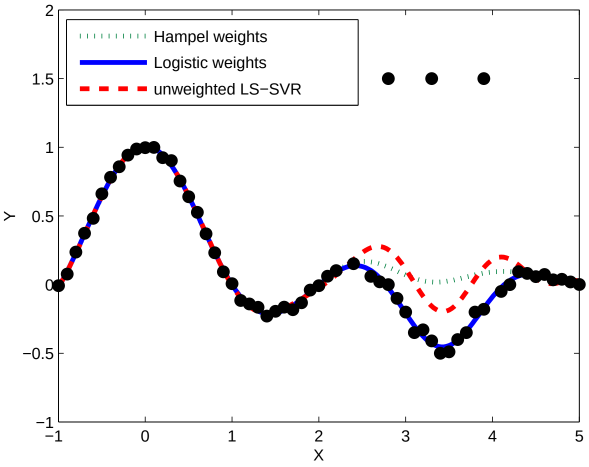


Figure 3.1 Simulated data example. Dashed curve: original LS-SVR. Dotted curve: wLS-SVR using Hampel weights. Solid curve: wLS-SVR using logistic weights.

curve) improves the fit remarkably well. Using Hampel's weight function (dotted curve) however does not improve the original estimate in this example. In that case all points in the region $x \in [2.5, 4.2]$ receive a weight exactly equal to zero. Thus, locally the outliers do not have a smaller weight than the neighboring "good" data. With logistic weights, all these good data points with $x \in [2.5, 4.2]$ receive a small weight as well, but the outliers get an even smaller weight. Therefore they are also locally recognized as outliers and thus wLS-SVR with logistic weights performs a lot better in this example. This example clearly shows that it is not trivial to choose a good weight function. Conditions (c1)-(c4) are not just technical. They should be kept in mind in practice as well.

3.6.2 Convergence

In equations (3.18) and (3.20), an upper bound is established on the reduction of the influence function at each step. In Table 3.2 we calculated this upper bound at a normal distribution, a Student distribution with five degrees of freedom and at a Student distribution with three degrees of freedom. We compare Huber's weight function with several cutoff values β, as well as logistic weights. Note that the convergence of the influence functions is pretty

		$N(0,1)$			t_5			Cauchy		
		c	d	$\frac{c}{d}$	c	d	$\frac{c}{d}$	c	d	$\frac{c}{d}$
Huber	$\beta = 0.5$	0.32	0.71	**0.46**	0.31	0.67	**0.46**	0.28	0.63	**0.47**
	$\beta = 1$	0.22	0.91	**0.25**	0.23	0.87	**0.27**	0.22	0.78	**0.28**
	$\beta = 1.5$	0.11	0.97	**0.11**	0.14	0.94	**0.15**	0.16	0.89	**0.18**
	$\beta = 2$	0.04	0.99	**0.04**	0.08	0.98	**0.08**	0.10	0.94	**0.11**
Logistic		0.22	0.82	**0.26**	0.22	0.79	**0.28**	0.22	0.73	**0.30**

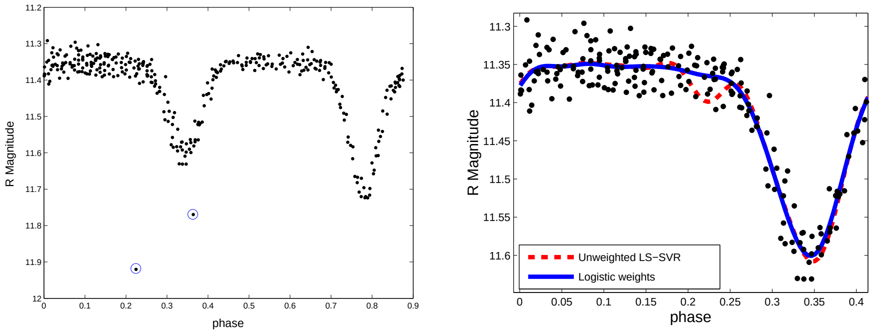
Table 3.3 Values of the constants c, d and $\frac{c}{d}$ for the Huber weight function with cutoff $\beta = 0.5, 1, 1.5, 2$ and for the logistic weight function, at a standard normal distribution, Student distribution with 5 degrees of freedom, and Student distribution with 3 degrees of freedom. The values of c/d (bold) represent an upper bound for the reduction of the influence function at each step.

fast, even at heavy tailed distributions. For Huber weights, the convergence rate (3.20) decreases rapidly as β increases. This is quite expected, since the larger β is, the less points are downweighted. Also note that the upper bound on the convergence rate approaches 1 as β goes to 0. The Huber loss function converges to an L_1 loss as β convergence to 0. Thus when reweighting LS-KBR to obtain L_1-KBR no fast convergence is guaranteed by our results, since the upper bound on the reduction factor approaches 1. When β is exactly 0, no results can be given at all, because then the ψ function is discontinuous.

Logistic weights are doing quite well. Even at heavy tailed noise distributions such as a t_3, the influence function is reduced to 0.30 of the value at the previous step. This means for example that after k steps, at most 0.30^k is left of the influence of the initial estimator, so fast convergence can be expected.

3.6.3 Star data

Variable stars are stars whose brightness periodically changes over time. Such a variable star was analyzed in Oh et al. [2004]. A plot of its brightness versus its phase (with period 0.8764, as found in Oh et al. [2004]) is shown in Figure 3.2(a). It concerns an eclipsing binary star, with both stars orbiting each other in the plane of the earth. Therefore, if one member of the pair eclipses the other, the combined brightness decreases. This explains the two peaks that are clearly present in the picture. Our goal is now to estimate the light curve, i.e. the functional relationship between brightness and phase, which is useful



(a) (b)

Figure 3.2 Star data. (a) Brightness (expressed in stellar magnitude R) of the binary star versus the phase (with a period of 0.8764 days). The two outliers in the data are circled. (b) Plot of the fit in the region: phase $\in [0, 0.4]$. Initial LS-SVR fit (dashed line), wLS-SVR with Huber weights and one reweighting step (solid line). The fit after four reweighting steps is practically coinciding with the solid line.

for classification of stars. In this case for example, the light curve is flat in between two peaks. This feature is associated with the detached type of eclipsing stars.

From Figure 3.2(a) it is obvious that two outliers are part of the data. When using classical LS-SVR to fit the light curve, these two data points have quite an impact on the result. In Figure 3.2(b) (dashed line) the LS-SVR fit shows an extra bump at phases in $[0.15, 0.25]$. The solid line represents the one step reweighted LS-SVR with Hubers weight function (b=1.5). The effect of the outliers is reduced, leading to quite a nice fit. The two step reweighted LS-SVR is plotted as well (dotted line), but the difference with the one step reweighting is practically invisible. After six steps, all residuals were the same as after five steps up to 0.001, showing the fast convergence properties of weighted LS-SVR.

3.6.4 Artificial data

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

- Sinc curve ($d = 1$): $y(x) = \sin(x)/x$.
- Friedman 1 ($d = 10$): $y(x) = 10\sin(\pi x_1 x_2) + 20(x_3 - 1/2)^2 + 10x_4 +$

$5x_5 + \sum_{i=6}^{10} 0.x_i$.

- Friedman 2 ($d = 4$): $y(x) = (x_1^2 + (x_2x_3 - 1/x_2x_4)^2))^{1/2}$.

In each replication 100 data points were generated. For the sinc curve, the inputs were taken uniformly on $[-5, 5]$. For the Friedman data sets [Friedman, 1991] inputs were generated uniformly from the unit hypercube. Noise was added to $y(x)$ from two distributions: first, Gaussian with unit variance and second, Student with 2 degrees of freedom.

For each data set, unweighted LS-SVR with RBF kernel was performed. The hyperparameters λ and σ were obtained by 10-fold cross validation using the Mean Squared Error (MSE) as cost criterion. Reweighted LS-SVR with RBF kernel and logistic weights was performed as well, using the same hyperparameters as found in the unweighted case. To compare both methods, the MSE was calculated over 200 noisefree test points. This procedure was repeated in 100 replications. Figure 3.3 shows boxplots of these 100 MSE's for the six cases.

First consider the left panel of Figure 3.3 containing the results with Gaussian noise. In that case the difference between reweighting or not is rather small. For Friedman 1, the median MSE is slightly smaller in the case of reweighting, whereas the sinc curve and Friedman 2 give slightly bigger median MSE's.

At the right panel of Figure 3.3 boxplots are shown for Student distributed noise. In that case reweighting clearly offers an improvement of the results. Not only is the median MSE smaller in all three settings. Also the right skewness of the MSE's clearly diminishes after reweighting, indicating that the method is more stable. This is exactly what we concluded in our theoretical analysis from Section 3.5, where it was demonstrated that reweighting improves stability at heavy tailed distributions.

Here we see in practice that reweighting leads to improved results at heavy tailed error distributions but retains the quality of unweighted LS-KBR at others such as the Gaussian distribution. Also note that we kept the hyperparameters fixed at their optimal value in the unweighted case, since we also treat the hyperparameters fixed in our theoretical results. Nevertheless, re-optimizing them at each reweighting step might possibly lead to even better results.

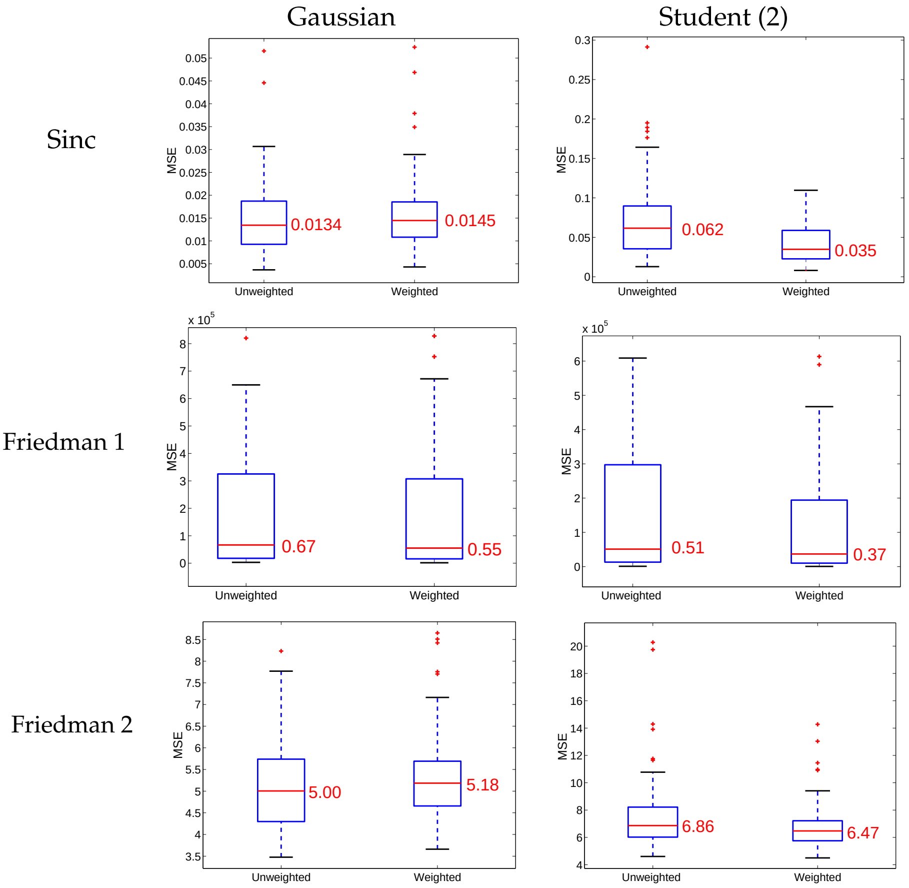


Figure 3.3 Simulation results for three data sets (sinc, Friedman 1 and Friedman 2). On the left: Gaussian errors. On the right: Student with 2 degrees of freedom. Each time boxplots of 100 MSE's are shown for unweighted LS-SVR and reweighted LS-SVR with logistic weights. For Gaussian errors no clear winner can be seen between unweighted versus reweighted. For Student errors reweighting leads to improvement.

3.7 Conclusion

We analyzed the series of influence functions of reweighted LS-KBR. It was found that a weight function $w(r) = \psi(r)/r$ with ψ increasing and bounded guarantees convergence to a bounded influence function if the kernel is bounded. This means for example that reweighted LS-KBR with logistic weights using a RBF-kernel is a robust estimator, even if the initial estimator is obtained by ordinary (non-robust) LS-KBR.

An upper bound for the convergence rate at each step shows that the convergence is generally quite fast. It is also interesting to note that this upper bound scales as $1/\lambda$.

The good robustness properties of reweighted LS-KBR depend on the kernel. If the kernel is not bounded, e.g. the linear kernel, then the previous weight functions do not lead to robust estimators. One can however normalize kernels. For the linear kernel one can take

$$K(x,z) = \frac{x^t z}{||x||_2||z||_2}.$$

Then all nice robustness properties hold, since the normalization obviously makes the linear kernel bounded. Further research exploiting normalized kernels could therefore be very interesting from a robustness point of view.

Finally it is important to notice that reweighting does not only improve robustness against outliers or gross errors. It also improves the stability of LS-KBR, especially at heavy-tailed distributions.

3.8 Proofs

Remark on Proposition 3.5 for least squares

As in the empirical case (3.10), it is also possible to include an intercept term $b_{P,\lambda} \in \mathbb{R}$ in the theoretical expressions, next to the functional part $f_{P,\lambda}$. For any distribution P and $\lambda > 0$ we denote

$$T(P) = (f_{P,\lambda}, b_{P,\lambda}) \in \mathcal{H} \times \mathbb{R}$$

minimizing the regularized risk:

$$(f_{P,\lambda}, b_{P,\lambda}) = \min_{(f,b)\in\mathcal{H}\times\mathbb{R}} \left(\mathbb{E}_P L(Y, f(X) + b) + \lambda||f||_{\mathcal{H}}\right).$$

The solution of this minimization problem is characterized in DeVito et al. [2004] (main theorem pp. 1369). If the loss function L is the least squares loss function, then this theorem provides us the following equations:

$$f_{P,\lambda} = \frac{1}{\lambda}\mathbb{E}_P[(Y - f_{P,\lambda}(X) - b_{P,\lambda})\Phi(X)] \tag{3.25}$$

$$b_{P,\lambda} = \mathbb{E}_P(Y - f_{P,\lambda}(X)). \tag{3.26}$$

Now we consider the contaminated distribution $P_{\epsilon,z} = (1-\epsilon)P + \epsilon\Delta_z$ with Δ_z a Dirac distribution with all probability mass located at the point z. Then by definition the influence function of the intercept term at $z \in \mathcal{X} \times \mathcal{Y}$ equals

$$IF(z;b,P) = \lim_{\epsilon \downarrow 0} \frac{b_{P_{\epsilon,z},\lambda} - b_{P,\lambda}}{\epsilon}.$$

Using equation (3.26) for both $b_{P_{\epsilon,z},\lambda}$ and $b_{P,\lambda}$ yields

$$\begin{aligned} IF(z;b,P) &= \lim_{\epsilon \downarrow 0} \frac{\mathbb{E}_{P_{\epsilon,z}}(Y - f_{P_{\epsilon,z},\lambda}(X)) - \mathbb{E}_P(Y - f_{P,\lambda}(X))}{\epsilon} \\ &= \lim_{\epsilon \downarrow 0} \frac{(1-\epsilon)\mathbb{E}_P(Y - f_{P_{\epsilon,z},\lambda}(X)) + \epsilon(z_y - f_{P_{\epsilon,z},\lambda}(z_x)) - \mathbb{E}_P(Y - f_{P,\lambda}(X))}{\epsilon}. \end{aligned}$$

Rearranging terms in the nominator we have

$$\begin{aligned} IF(z;b,P) &= \lim_{\epsilon \downarrow 0} \frac{\mathbb{E}_P(Y - f_{P_{\epsilon,z},\lambda}(X)) - \mathbb{E}_P(Y - f_{P,\lambda}(X))}{\epsilon} \\ &\quad - \lim_{\epsilon \downarrow 0} \frac{\epsilon\mathbb{E}_P(Y - f_{P_{\epsilon,z},\lambda}(X)) + \epsilon(z_y - f_{P_{\epsilon,z},\lambda}(z_x))}{\epsilon} \\ &= \lim_{\epsilon \downarrow 0} \frac{\mathbb{E}_P(f_{P,\lambda}(X) - f_{P_{\epsilon,z},\lambda}(X))}{\epsilon} - \mathbb{E}_P(Y - f_{P,\lambda}(X)) + (z_y - f_{P,\lambda}(z_x)). \end{aligned}$$

Thus for the intercept term we obtain the following expression.

$$IF(z;b,P) = -\mathbb{E}_P IF(z;f,P) - \mathbb{E}_P(Y - f_{P,\lambda}(X)) + (z_y - f_{P,\lambda}(z_x)). \tag{3.27}$$

For $f_{P,\lambda}$ we have

$$IF(z;b,P) = \lim_{\epsilon \downarrow 0} \frac{f_{P_{\epsilon,z},\lambda} - f_{P,\lambda}}{\epsilon}.$$

Plugging in equation (3.25) for both $f_{P_{\epsilon,z},\lambda}$ and $f_{P,\lambda}$, it is clear that

$$\begin{aligned} &\lambda IF(z;f,P) + \mathbb{E}_P[IF(z;f,P)(X)\Phi(X)] + \mathbb{E}_P[IF(z;b,P)(X)\Phi(X)] \\ &= -\mathbb{E}_P(Y - f_{P,\lambda}(X) - b_{P,\lambda})\Phi(X) + (z_y - f_{P,\lambda}(z_x) - b_{P,\lambda})\Phi(z_x). \end{aligned} \tag{3.28}$$

Thus, combining (3.27) and (3.28) in matrix notation, we have

$$\begin{pmatrix} \lambda \mathrm{id}_{\mathcal{H}} + \mathbb{E}_P[\langle ., \Phi(X)\rangle \Phi(X)] & \mathbb{E}_P\Phi(X) \\ \mathbb{E}_P\langle ., \Phi(X)\rangle & 1 \end{pmatrix} \begin{pmatrix} IF(z; f, P) \\ IF(z; b, P) \end{pmatrix}$$
$$= \begin{pmatrix} -\mathbb{E}_P[(Y - f_{P,\lambda}(X) - b_{P,\lambda})\Phi(X)] + (z_y - f_{P,\lambda}(z_x) - b_{P,\lambda})\Phi(z_x) \\ -\mathbb{E}_P(Y - f_{P,\lambda}(X) - b_{P,\lambda}) + (z_y - f_{P,\lambda}(z_x) - b_{P,\lambda}) \end{pmatrix}. \tag{3.29}$$

When not considering the intercept term, the previous expression indeed corresponds to the one already obtained by Christmann and Steinwart [2006]. Also note the similarities to the results obtained in classification [Christmann and Steinwart, 2004]. However, since this intercept term is not essential in explaining the robustness principles of kernel based regression, we will not include it anymore furtheron.

Proof of Theorem 3.6

Let P be a distribution on $\mathcal{X} \times \mathcal{Y}$ with $|P|_a < \infty$ and define $\xi = \int_{\mathcal{X}\times\mathcal{Y}} w(x, y - f^{(0)}_{P,\lambda}(x))dP(x, y)$. Assume $\xi > 0$. Then we can define a distribution P_w by $dP_w(x, y) = \xi^{-1} w(x, y - f^{(0)}_{P,\lambda}(x))dP(x, y)$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Since $\xi > 0$ and w is continuous, P_w is well defined and one can easily see that $f^{(1)}_{P,\lambda} = f_{P_w,\lambda/\xi}$. Moreover $|P_w|_a < \infty$ if $|P|_a < \infty$ and Proposition 3.4 yields

$$f^{(1)}_{P,\lambda} = f_{P_w,\lambda/c} = -\frac{\xi}{2\lambda}\mathbb{E}_{P_w} h\Phi = -\frac{1}{2\lambda}\mathbb{E}_P w(X, Y - f^{(0)}_{P,\lambda}(X))h\Phi.$$

For the least squares loss function we obtain

$$f^{(1)}_{P,\lambda} = \frac{1}{\lambda}\mathbb{E}_P w(X, Y - f^{(0)}_{P,\lambda}(X))(Y - f_{P,\lambda}(X))\Phi(X).$$

If $\xi = 0$ then $\mathbb{E}_P[w(X, Y - f^{(0)}_{P,\lambda}(X))L(X, Y - f(X))] = 0$. Thus

$$f^{(1)}_{P,\lambda} = \operatorname*{argmin}_{f\in\mathcal{H}} \lambda ||f||_{\mathcal{H}} = 0.$$

But if $\xi = 0$ we also have that $\frac{1}{2\lambda}\mathbb{E}_P w(X, Y - f^{(0)}_{P,\lambda}(X))h\Phi = 0$ and therefore Theorem 1 still holds.

Proof of Theorem 3.7

We can use the representation from Theorem 3.6 to calculate the influence function in a point $z \in X \times Y$

$$\begin{aligned}
IF(z;T_1,P) &= \frac{\partial}{\partial\epsilon} T_1(P_{\epsilon,z})|_{\epsilon=0} \\
&= -\frac{1}{2\lambda}\frac{\partial}{\partial\epsilon}\mathbb{E}_{P_{\epsilon,z}} w(Y,Y-f^{(0)}_{P_{\epsilon,z},\lambda}(X))L'(Y,f^{(1)}_{P_{\epsilon,z},\lambda}(X))\Phi(X)|_{\epsilon=0} \\
&= \frac{1}{2\lambda}\mathbb{E}_P w(Y,Y-f^{(0)}_{P,\lambda}(X))L'(Y,f^{(1)}_{P,\lambda})\Phi(X) \\
&\quad -\frac{1}{2\lambda} w(z_x,z_y-f^{(0)}_{P,\lambda}(z_x))L'(Y,f^{(1)}_{P,\lambda}(X))\Phi(z_x) \\
&\quad -\frac{1}{2\lambda}\frac{\partial}{\partial\epsilon}\mathbb{E}_P[w(X,Y-f^{(0)}_{P_{\epsilon,z},\lambda}(X))L'(Y,f^{(1)}_{P_{\epsilon,z},\lambda}(X))\Phi(X)]|_{\epsilon=0}.
\end{aligned}$$

The last term equals

$$\begin{aligned}
&\frac{1}{2\lambda}\mathbb{E}_P[IF(z;T_0,P)\frac{\partial}{\partial r} w(X,Y-f^{(0)}_{P,\lambda}(X))L'(Y,f^{(1)}_{P,\lambda}(X))] \\
&-\frac{1}{2\lambda}\mathbb{E}_P[w(X,Y-f^{(0)}_{P,\lambda}(X))IF(z;T_1,P)L''(Y,f^{(1)}_{P,\lambda}(X))].
\end{aligned}$$

Thus defining S_w and C_w as in Theorem 3.7, we have

$$\begin{aligned}
S_w(IF(z;T_1,P)) &= \mathbb{E}_P w(X,Y-f^{(0)}_{P,\lambda}(X))L'(Y,f^{(1)}_{P,\lambda}(X))\Phi(X) \\
&\quad + C_w(IF(z;f^{(0)}_{P,\lambda},P)) - w(z_x,z_y-f^{(0)}_{P,\lambda}(z_x))L'(z_y,f^{(1)}_{P,\lambda}(z_x))\Phi(z_x).
\end{aligned}$$

Now it suffices to show that S_w is invertible. In Christmann and Steinwart [2006] this was already proven for the operator S as defined in Proposition 3.5. However, we can again consider the distribution P_w such that $dP_w(x,y) = \xi^{-1}w(x,y-f^{(0)}_{P,\lambda}(x))dP(x,y)$ for all $(x,y) \in \mathbb{R}^d \times \mathbb{R}$, with $\xi = \int_{X\times Y} w(x,y-f^{(0)}_{P,\lambda}(x))dP(x,y) > 0$. Then the operator S_w using distribution P and generalization factor λ is equivalent to the operator S using the distribution P_w and generalization factor λ/ξ. Thus, using Christmann and Steinwart [2006] (more specific their proof of Theorem 18), we see that S_w is invertible.

Proof of equation (3.16)

Denote P_w the distribution such that $dP_w(x,y) = w(x,y-f_P(x))dP(x,y)$ for any $(x,y) \in \mathcal{X} \times \mathcal{Y}$. Then

$$\operatorname*{argmin}_{f\in\mathcal{H}} \mathbb{E}_P\left[w(X,Y-f_P(x))(Y-f(x))^2\right] = \operatorname*{argmin}_{f\in\mathcal{H}} \mathbb{E}_{P_w}(Y-f(x))^2 = \mathbb{E}_{P_w}(Y|x)$$

since unweighted least squares KBR is Fisher consistent if $\lambda = 0$. Now

$$\mathbb{E}_{P_w}(Y|x) = \mathbb{E}_{P_w}(Y - f_P(X) + f_P(X)|x) = \mathbb{E}_{P_w}(Y - f_P(X)|x) + f_P(x)$$

and the first term of this sum equals zero because the weight function w is even in its second argument and the errors are independent of x and symmetric around zero.

Proof of equation (3.20)

We need two propositions from operator theory in Hilbert spaces.

Proposition 3.10 *(Spectral Theorem)*
Let T be a compact and self-adjoint operator on a Hilbert space $\mathcal{H}$. Then $\mathcal{H}$ has an orthonormal basis (e_n) consisting of eigenvectors for T. If $\mathcal{H}$ is infinite dimensional, the corresponding eigenvalues (different from 0) (γ_n) can be arranged in a decreasing sequence $|\gamma_1| \geq |\gamma_2| \geq \ldots$ where $\gamma_n \to 0$ for $n \to \infty$, and for $x \in \mathcal{H}$

$$T(x) = \sum_n \gamma_n \langle x, e_n \rangle e_n.$$

Proposition 3.11 *(Fredholm Alternative)*
Let T be a compact and self-adjoint operator on a Hilbert space $\mathcal{H}$, and consider the equation

$$(T - \gamma \; id_{\mathcal{H}})x = y.$$

If γ is not an eigenvalue of T, then the equation has a unique solution $x = (T - \gamma id_{\mathcal{H}})^{-1} y$.

Recall that we assumed that the distribution P could be decomposed in an error distribution P_e and a distribution in x-space P_x such that $dP = dP_e dP_x$. Using this regression structure of P we can write

$$S_w = \lambda \, \mathrm{id}_{\mathcal{H}} + E_{P_e} \frac{\psi(e)}{e} E_{P_X} \langle ., \Phi(X) \rangle \Phi(X).$$

Denote $T = E_{P_X} \langle ., \Phi(X) \rangle \Phi(X)$, then

$$S_w = \lambda \, \mathrm{id}_{\mathcal{H}} + d \; T$$

with the constant $d = E_{P_e} \frac{\psi(e)}{e}$. In the same way we find

$$C_w = cT$$

with $c = d - E_{P_e}\psi'(e)$.
Now we know T is compact (proven in Christmann and Steinwart [2006]) and self-adjoint. Moreover, T is positive and thus its eigenvalues are positive. As such, $-\frac{\lambda}{d}$ cannot be an eigenvalue, and by the Fredholm alternative, $T - (-\frac{\lambda}{d})\mathrm{id}_{\mathcal{H}}$ is invertible. Thus for any $g \in \mathcal{H}$ the equation

$$\left(T - (-\frac{\lambda}{d})\mathrm{id}_{\mathcal{H}}\right)(f) = \frac{c}{d}T(g)$$

has a unique solution in terms of $f \in \mathcal{H}$. Moreover, from the spectral theorem we know that T has an orthonormal basis (f_i) with corresponding eigenvalues λ_i and we can write our equation as

$$\left(T - (-\frac{\lambda}{d})\mathrm{id}_{\mathcal{H}}\right)(f) = \sum_{i=1}^{\infty}(\lambda_i + \frac{\lambda}{d})\langle f, f_i\rangle f_i = \sum_{i=1}^{\infty}\frac{c}{d}\lambda_i\langle g, f_i\rangle f_i$$

Thus we see that

$$\langle f, f_i\rangle = (\lambda_i + \frac{\lambda}{d})^{-1}\lambda_i\frac{c}{d}\langle g, f_i\rangle$$

and so we find

$$(S_w^{-1} \circ C_w)(g) = f = \sum_{i=1}^{\infty}(\lambda_i + \frac{\lambda}{d})^{-1}\lambda_i\frac{c}{d}\langle g, f_i\rangle f_i$$

Since the operator norm of a compact operator equals the supremum of its eigenvalues, we have that

$$\|S_w^{-1} \circ C_w\| = \sup_i \frac{\lambda_i\frac{c}{d}}{\lambda_i + \frac{\lambda}{d}} = \frac{c}{d}\frac{1}{1+\frac{\lambda}{d}},$$

proving equation (3.20). Since $c = d - E_{P_e}\psi'(e)$,

$$\frac{c}{d}\frac{1}{1+\frac{\lambda}{d}} < 1 \Leftrightarrow 1 - \frac{E_{P_e}\psi'(e)}{d} < 1 + \frac{\lambda}{d}$$

or

$$E_{P_e}\psi'(e) > -\lambda.$$

Chapter 4

Robust model selection for kernel based regression using the influence function

4.1 Introduction

In the previous chapter we considered the influence function of reweighted kernel based regression. We obtained some conditions on the weight function in order to bound the influence function, thus preventing unlimited effects of outliers. However, apart from this outlier interpretation coming from robust statistics, analyzing 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. In Section 3.5 we discussed the concept of stability using such leave-one-out idea to assess the generalization capacity. In this chapter we further explore some heuristic links between the influence function and other concepts in Section 4.2. Emphasis in this section lies on exploration rather than formal proofs.

In its original form the influence function works on continuous distributions. If we want to use some of the stability- and other links in practice, we have to approximate these influence functions at the sample distribution. This can be done fairly easy and some expressions are obtained in Section 4.3.

All analysis from Chapter 3 considered the case where both λ and the kernel K with its possible kernel parameters are fixed and chosen independently from P. In practice this is of course not true. Both the regularization parameter λ and for example the bandwidth of a Gaussian RBF kernel are chosen in a data driven way, e.g. by leave-one-out or 10-fold-cross validation, or a model selection criterion such as Generalized Cross Validation or Akaike's AIC. However, these criteria themselves can be affected by outliers. Although reweighting can provide a robust estimation procedure, this does not automatically guarantee robust hyperparameter selection. In this chapter we investigate a criterion that is not as affected by outliers as some of the classical procedures. The idea is to use finite sample influence function as an approximation of the leave-one-out error. This is done in Section 4.4. Some examples are shown in Section 4.5.

4.2 Asymptotic variance and generalization

4.2.1 Asymptotic variance

A classical way [Hampel et al., 1986, Huber, 1981] to relate the influence function to the performance of an estimator T is through the concept of asymptotic variance.

If T is sufficiently regular, it can be linearized near P in terms of the influence function. If the distribution $\tilde{P}$ is near P, then the leading terms of a Taylor expansion are

$$T(\tilde{P}) = T(P) + \int IF(z;T,P)[\tilde{P}(dz) - P(dz)] + \dots. \tag{4.1}$$

For example, taking $\tilde{P} = P_{\epsilon,z}$ we have that

$$T(P_{\epsilon,z}) = T(P) + \epsilon \int IF(z;T,P) + \dots$$

which is also immediately clear from Definition 4. Now one can also take $\tilde{P} = P_n$ the sample distribution. Since

$$\int IF(z;T,P)P(dx) = 0$$

we obtain that

$$\begin{aligned} T(P_n) - T(P) &= \int IF(z;T,P)P_n(dz) + \dots \\ &= \frac{1}{n}\sum_{i=1}^{n} IF(z_i;T,P) + \dots. \end{aligned} \tag{4.2}$$

If the remaining terms are asymptotically negligible, the central limit theorem immediately shows that $\sqrt{n}(T(P_n) - T(P))$ is asymptotically normal with mean 0 and variance

$$ASV(T,P) = \int IF^2(z;T,P)P(dz). \tag{4.3}$$

Again note that the influence function should not attain too large values *at points z where the density of P is not too small*. This is very similar to the stability in (3.24). In the context of kernel methods we can apply these concepts to obtain pointwise asymptotic variances. If we fix a point $x \in \mathcal{X}$, then we can define the operator $f_{\lambda,x} : P \to f_{P,\lambda}(x)$ with the same notation as (3.11). The influence function of $f_{\lambda,x}$ is easily obtained by evaluating the expression from Proposition 3.5 (which is an element of $\mathcal{H}$) in the point x. For the reweighted case this is of course completely similar using Lemma 3.7. In the next section we will show how to evaluate these variances on a specific sample P_n, but for now it is interesting to compare again the unweighted to the reweighted estimator.

Assume for example a regression setting where most of the uncertainty is captured in the distribution of the errors. For the unweighted least squares kernel estimator, the influence function in Proposition 3.5 is unbounded as it depends on the residual $z_y - f_{P,\lambda}(z_x)$. For the asymptotic variance we will have to average the square of the influence function over all z coming from P and thus this means that the second moment of the error distribution is the main factor determining the variance. At a Gaussian error distribution, there is again no problem and a least squares method is an excellent choice. But at a heavy-tailed distribution the asymptotic variance can be very high and even infinity if the second moment of the errors does not exist. For estimators with a bounded influence function this problem does not occur, since obviously $ASV < \sup\{IF(z;T,P)\}^2$ which is always finite in that case. Thus a reweighted least squares estimator guarantees a finite asymptotic variance for any distribution P.

4.2.2 Generalization

In the previous paragraphs we analyzed the variance of the kernel estimator at a fixed point $x \in \mathcal{X}$. Usually one is however rather interested in the average performance of the estimator with respect to a certain loss function L. Define for any two distributions Q and P.

$$G(P,Q) = \mathbb{E}_P L(Y - f^{(k)}_{Q,\lambda}(X)).$$

The following lemma holds.

Lemma 4.1 *Denote $G(P,.)$ the functional that maps any distribution Q onto $G(P,Q)$. Then for any $z \in \mathcal{X} \times \mathcal{Y}$*

$$IF(z; G(P,.), P) = \mathbb{E}_P[L'(Y - f^{(k)}_{P,\lambda}(X)) IF(z; f^{(k)}_{P,\lambda})(X)].$$

Proof. By definition we have for any distribution Q

$$IF(z; G(P,.), Q) = \lim_{\epsilon \downarrow 0} \frac{\mathbb{E}_P[L(Y - f^{(k)}_{Q_{\epsilon,z},\lambda}(X))] - \mathbb{E}_P[L(Y - f^{(k)}_{Q,\lambda}(X))]}{\epsilon}.$$

Since the loss function L is differentiable, assuming that limit and integral can be switched gives

$$\begin{aligned} IF(z; G(P,.), Q) &= -\mathbb{E}_P[L'(Y - f^{(k)}_{Q,\lambda}(X)) \lim_{\epsilon \downarrow 0} \frac{f^{(k)}_{Q_{\epsilon,z},\lambda}(X) - f^{(k)}_{Q,\lambda}(X)}{\epsilon}] \\ &= -\mathbb{E}_P[L'(Y - f^{(k)}_{Q,\lambda}(X))] IF(z; f^{(k)}_{Q,\lambda})(X). \end{aligned}$$

Now take $Q = P$ to obtain the result.

□

The quantity $G(P, P_n)$ equals the generalization error and is very important, since this is what one often wants to minimize. Cucker and Smale [2002] discuss a bias-variance tradeoff formulation to the generalization error starting from

$$G(P, P_n) \leq |G(P, P_n) - G(P, P)| + |G(P, P)|.$$

Via (4.2) we get that for n large enough

$$G(P, P_n) \leq |\sum_{i=1}^{n} \frac{IF(z_i; G(P,.), P)}{n}| + |G(P, P)|.$$

and by Theorem 4.1 we find

$$G(P, P_n) \leq |\mathbb{E}_P[L'(Y - f^{(k)}_{P,\lambda}(X)) \sum_{i=1}^{n} \frac{IF(z_i; f^{(k)}_{\lambda}, P)(X)}{n}]| + |G(P, P)|.$$

Once more it becomes clear that the influence function at points z_i sampled from the distribution P, should not be too large in order not to blow up the variance term. One should however be aware that it is of course easy to obtain a small influence function. E.g. taking λ extremely large will lead to an influence function that is almost 0 everywhere, and thus a very small variance. However, the bias term $|G(P,P)|$ will be almost maximal in that case. Thus the key for regularized methods is to find an estimator with both a small influence function and a small bias.

4.3 Estimating influence functions from the data

4.3.1 Unweighted case

In the previous sections we discussed some interesting links that suggest an important role of the influence function with respect to the stability of an estimator. So far however all results were formulated for continuous distributions P. In order to make practical use of the influence function, it is of course important to have specific expressions estimating these quantities from the data.

First consider the expression for unweighted least squares in Proposition 3.5. The influence function there is an element of the Reproducing Kernel Hilbert Space $\mathcal{H}$. We now approximate this function in the n sample points $x_1, \ldots, x_n$. First consider the operator $S : \mathcal{H} \to \mathcal{H}$ defined by

$$S(f) = \lambda f + \mathbb{E}_P \left[\langle \Phi(X), f \rangle \Phi(X) \right]$$

for any $f \in \mathcal{H}$. According to this definition an approximating $n \times n$ matrix S_n should satisfy for any $f \in \mathcal{H}$ that

$$S_n \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} = \lambda \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} + \frac{1}{n} \begin{pmatrix} K(x_1,x_1) & \ldots & K(x_1,x_n) \\ \vdots & & \\ K(x_n,x_1) & & K(x_n,x_n) \end{pmatrix} \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}$$

which means that

$$S_n = \lambda I_n + \frac{\Omega}{n}$$

with I_n the identity matrix and Ω the kernel matrix $\Omega_{ij} = K(x_i, x_j)$. Thus one can calculate a finite sample version of the influence function in the n sample points as

$$\begin{pmatrix} IF(z;f_\lambda,P_n)(x_1) \\ \vdots \\ IF(z;f_\lambda,P_n)(x_n) \end{pmatrix} = S_n^{-1}\Bigg((z_y - f_{P_n,\lambda}(z_x))\begin{pmatrix} K(z_x,x_1) \\ \vdots \\ K(z_x,x_n) \end{pmatrix} - \lambda \begin{pmatrix} f_{P_n,\lambda}(x_1) \\ \vdots \\ f_{P_n,\lambda}(x_n) \end{pmatrix}\Bigg). \tag{4.4}$$

The value of $f_{P_n,\lambda}$ at a point $x \in \mathcal{X}$ is given by

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

which is a classical result going back to Tikhonov and Arsenin [1977].

4.3.2 Reweighted case

Now assume again that we have a weight function $w : \mathbb{R} \to \mathbb{R}^+$ satisfying conditions (3.19). For the one step reweighted estimator we can perform similar calculations. Define the $n \times n$ weight matrix W as $W_{ii} = w(y_i - f_{P_n,\lambda}(x_i))$ and $W_{ij} = 0$ for $i \neq j$. Define the $n \times n$ matrix C as $C_{ii} = w'(y_i - f_{P_n,\lambda}(x_i))(y_i - f^{(1)}_{P,\lambda}(x_i))$ and $C_{ij} = 0$ for $i \neq j$. Then

$$\begin{pmatrix} IF(z;f^{(1)}_\lambda,P_n)(x_1) \\ \vdots \\ IF(z;f^{(1)}_\lambda,P_n)(x_n) \end{pmatrix} = (\lambda I_n + \frac{\Omega W}{n})^{-1}\Bigg(\psi(z_y - f^{(1)}_{P_n,\lambda}(z_x))\begin{pmatrix} K(z_x,x_1) \\ \vdots \\ K(z_x,x_n) \end{pmatrix} - \Omega C \begin{pmatrix} IF(z;f_\lambda,P_n)(x_1) \\ \vdots \\ IF(z;f_\lambda,P_n)(x_n) \end{pmatrix} - \lambda \begin{pmatrix} f_{P_n,\lambda}(x_1) \\ \vdots \\ f_{P_n,\lambda}(x_n) \end{pmatrix}\Bigg).$$

For $k+1$-steps one can proceed recursively. Define the $n \times n$ weight matrix $W^{(k)}$ as $W^{(k)}_{ii} = w(y_i - f^{(k)}_{P_n,\lambda}(x_i))$ and $W^{(k)}_{ij} = 0$ for $i \neq j$. Define the $n \times n$ matrix $C^{(k)}$ as $C^{(k)}_{ii} = w'(y_i - f^{(k)}_{P_n,\lambda}(x_i))(y_i - f^{(k+1)}_{P,\lambda}(x_i))$ and $C^{(k)}_{ij} = 0$ for $i \neq j$.

Then

$$\begin{pmatrix} IF(z;f_\lambda^{(k+1)},P_n)(x_1) \\ \vdots \\ IF(z;f_\lambda^{(k+1)},P_n)(x_n) \end{pmatrix} = \left(\lambda I_n + \frac{\Omega W^{(k)}}{n}\right)^{-1}$$
$$\Bigg(\psi(z_y - f_{P_n,\lambda}^{(k+1)}(z_x)) \begin{pmatrix} K(z_x,x_1) \\ \vdots \\ K(z_x,x_n) \end{pmatrix}$$
$$- \Omega C^{(k)} \begin{pmatrix} IF(z;f_\lambda^{(k)},P_n)(x_1) \\ \vdots \\ IF(z;f_\lambda^{(k)},P_n)(x_n) \end{pmatrix} - \lambda \begin{pmatrix} f_{P_n,\lambda}^{(k)}(x_1) \\ \vdots \\ f_{P_n,\lambda}^{(k)}(x_n) \end{pmatrix}\Bigg). \tag{4.5}$$

Here the value of $f_{P_n,\lambda}^{(k)}$ at a point $x \in \mathcal{X}$ is given by

$$\frac{1}{n}\sum_{i=1}^{n} \alpha_i^{(k)} K(x_i,x), \text{ with } \begin{pmatrix} \alpha_1^{(k)} \\ \vdots \\ \alpha_n^{(k)} \end{pmatrix} = W^{(k)} \left(\lambda I_n + \frac{\Omega W^{(k)}}{n}\right)^{-1} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}. \tag{4.6}$$

With these formulas the influence function can be evaluated in any point $z \in \mathcal{X} \times \mathcal{Y}$. In Sections 3.5 and 4.2 we argued that the influence function is also very useful when it comes to stability and variance, if we evaluate the influence function at the data points themselves. To this end we introduce the following definition.

Definition 4.2 *Let P_n be the empirical distributions of n observations $\{z_i = (x_i, y_i)\}$. Then the influence matrix $[IFM]^{(k)}$ of the k-step reweighted kernel estimator at P_n is defined as*

$$[IFM]_{i,j}^{(k)} = [IF(z_i; f_\lambda^{(k)}, P_n)(x_j)]^2$$

These entries of [IFM]$^{(k)}$ can be calculated from (4.5) *taking $z = z_i$.*

4.3.3 Estimating variance

Now fix a datapoint x_j. A straightforward finite sample approximation of the asymptotic variance at the point x_j consists of (compare to (4.3))

$$V_n^{(k)}(x_j) = \frac{1}{n}\sum_{i=1}^{n}[IF(z_i; f_\lambda^{(k)}, P_n)(x_j)]^2$$

which is the average of the elements in the jth column of the influence matrix $[\mathrm{IFM}]^{(k)}$. Denote $f^{(k)}_{\lambda,x_j} : P \rightarrow f^{(k)}_{P,\lambda}(x_j)$. Applying (4.2) and (4.3) to $f^{(k)}_{\lambda,x_j}$ gives that

$$f^{(k)}_{P_n,\lambda}(x_j) - f^{(k)}_{P,\lambda}(x_j) \longrightarrow_{n\rightarrow\infty} N(0, \mathrm{ASV}(f^{(k)}_{\lambda,x_j}, P)/n)$$

Estimating $\mathrm{ASV}(f^{(k)}_{\lambda,x_j}, P)$ by $V^{(k)}_n(x_j)$ we can produce the following confidence bands for the k-step reweighted estimator.

Corollary 4.3 *Denote* $z_{\alpha/2}$ *the* $(1-\frac{\alpha}{2})$*-quantile of a standard normal distribution. Then*

$$\textit{Prob}\left(f^{(k)}_{P,\lambda}(x_j) \in \left[f^{(k)}_{P_n,\lambda}(x_j) - z_{\alpha/2}\sqrt{V^{(k)}_n(x_j)/n}, f^{(k)}_{P_n,\lambda}(x_j) + z_{\alpha/2}\sqrt{V^{(k)}_n(x_j)/n}\right]\right)$$

converges to $1-\alpha$ *as* $n \rightarrow \infty$*.*

Note that these confidence bands are constructed around $f^{(k)}_{P,\lambda}(x_j)$, which is the theoretical estimate given λ and K. They give information about the variance that occurs when estimating $f^{(k)}_{P,\lambda}(x_j)$ by $f^{(k)}_{P,\lambda}(x_j)$. The bias however is not taken into account. For example as $\lambda \rightarrow \infty$ these bands become more and more tight, but around an estimator that is severely biased. In the next section we provide the main definition providing a bias-variance trade-off using the influence matrix defined in the previous paragraphs.

4.4 Model selection

4.4.1 Main definition

Define the sample P^{-i}_n as the sample P_n minus the ith observation. We start from the weighted leave-one-out criterion $\sum_{i=1}^n w_i\ (y_i - f^{(k)}_{P^{-i}_n,\lambda}(x_i))^2/n$. We can rewrite this expression as

$$\sum_{i=1}^n w_i\left(y_i - f^{(k)}_{P_n,\lambda}(x_i) + f^{(k)}_{P_n,\lambda}(x_i) - f^{(k)}_{P^{-i}_n,\lambda}(x_i)\right)^2 \Big/ n \tag{4.7}$$

where it becomes clear that one looks for a small training error $(y_i - f^{(k)}_{P_n,\lambda}(x_i))$ and good stability (small value for $f^{(k)}_{P_n,\lambda}(x_i) - f^{(k)}_{P^{-i}_n,\lambda}(x_i)$).

By definition the influence function measures the change of the estimator when some probability mass of size ϵ at z is added. Now take $\epsilon = -1/(n-1)$ and $z = z_i$ in Definition 1. Since

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

we have that

$$f^{(k)}_{P_n^{-i},\lambda} - f^{(k)}_{P_n,\lambda} \approx -\frac{1}{n-1}\mathrm{IF}(z_i; f^{(k)}_{P_n,\lambda}, P_n).$$

Thus we get

$$\sum_{i=1}^{n} w_i \Big(y_i - f^{(k)}_{P_n,\lambda}(x_i) + \frac{1}{n-1}\mathrm{IF}(z_i; f^{(k)}_{P_n,\lambda}, P_n)(x_i) \Big)^2 \Big/ n$$

Evaluating the square gives

$$\sum_{i=1}^{n} w_i \Big((y_i - f^{(k)}_{P_n,\lambda}(x_i))^2 + (\frac{1}{n-1}\mathrm{IF}(z_i; f^{(k)}_{P_n,\lambda}, P_n)(x_i))^2 + \frac{2}{n-1}(y_i - f^{(k)}_{P_n,\lambda}(x_i))\mathrm{IF}(z_i; f^{(k)}_{P_n,\lambda}, P_n)(x_i) \Big) \Big/ n$$

Using the expression for the influence function from 4.5 we define the following criterion.

Definition 4.4 *Given a sample P_n and a weight function w satisfying all conditions* (3.19)*. Let $f^{(k)}_{P_n,\lambda}$ be as in* (4.6)*. Define $w_i^{(k)} = w(y_i - f^{(k-1)}_{P_n,\lambda}(x_i))$. Denote $W^{(k)}$ the weight matrix containing the weights $w_i^{(k)}$ on its diagonal elements and Ω the kernel matrix. Denote [IFM]$^{(k)}$ the influence function matrix as in Definition 4.2. Given a kernel K and regularization parameter λ, define the function $C^{(k)}_{IFM}$ as*

$$C^{(k)}_{IFM}(\lambda, K) = \frac{1}{n}\sum_{i=1}^{n} w_i^{(k)}(y_i - f^{(k)}_{P_n,\lambda}(x_i))^2(1 + \frac{2}{n-1}w_i^{(k)}((\lambda I_n + \frac{\Omega W^{(k)}}{n})^{-1}\Omega)_{i,i}) + \frac{1}{n}\mathit{trace}(W^{(k)}\,\frac{[IFM]^{(k)}}{(n-1)^2}).$$

4.4.2 Link with GCV

In the unweighted case Generalized Cross Validation is given by

$$GCV(\lambda, K) = \frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - f_{P_n,\lambda}(x_i))^2}{(1 - \frac{1}{n}\mathrm{trace}((\lambda I_n + \frac{\Omega}{n})^{-1}\Omega))^2}.$$

If the diagonal elements of $S_n^{-1}\Omega$ are small, then we can use the approximation $(1-x)^{-2} \approx 1 + 2x$ to get

$$GCV(\lambda, K) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_{P_n,\lambda}(x_i))^2 \left(1 + \frac{2}{n}\mathrm{trace}((\lambda I_n + \frac{\Omega}{n})^{-1}\Omega)\right)^2$$

which is very similar to the first term in $C^{(0)}_{IFM}(\lambda, K)$. If the trace of $S_n^{-1}\Omega$ tends to n then the previous approximation does not hold at all. This happens in case of overfitting, for example when λ and the bandwidth σ in case of a Gaussian kernel are chosen too small. Then the residuals of the training data tend to zero. For the GCV criterion this is penalized since this training error that tends to zero is multiplied by a factor $\frac{1}{(1-\frac{1}{n}\text{trace}(S_n^{-1}\Omega))^2}$ that tends to infinity. For $C^{(0)}_{IFM}(\lambda, K)$ the residuals are multiplied by a bounded factor $(1 + \frac{2}{n-1}w_i(S_{w,n}^{-1}\Omega)_{i,i})$ and thus the first term tends to zero as overfitting occurs. In this case however the second term comes in to play. As this is a sum of squared influences it represents the amount of variance. When there is overfitting, this term will thus become larger and penalize too small choices for λ and the kernel parameters.

4.4.3 Link with Mallows' Cp

Assume data is generated from a model $g(x_i) + e_i$ with 'true regression function g and i.i.d. errors e_i independent from x_i and with constant variance γ^2. Then Mallows' Cp criterion corresponds to

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f_{P_n,\lambda}(x_i))^2 + \frac{1}{n}\hat{\gamma}^2\text{trace}((\lambda I_n + \frac{\Omega}{n})^{-1}\Omega))^2$$

where $\hat{\gamma}$ is an estimate of γ. If one would use $\hat{\gamma} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_{P_n,\lambda}(x_i))^2$ we again find something very similar to the first term of $C^{(0)}_{IFM}(\lambda, K)$. It is however obvious that this choice of $\hat{\gamma}$ is bad. There is no problem as long as λ and K are chosen such that $f_{P_n,\lambda}(x_i)$ approximates the true regression g well. But in case of overfitting $\hat{\gamma}$ decreases to zero and is a bad estimate of γ. In the $C^{(0)}_{IFM}(\lambda, K)$ criterion however, this is compensated for by the second term. If λ and K are chosen such that $\hat{\gamma}$ is too small, then this is penalized by a variance term consisting of a sum of squared influences.

4.4.4 Correction for small data sets

Since we make an asymptotic approximation of the leave-one-out error by the influence function, we implicitly assume that n is large enough. Unfortunately we often encounter problems for small samples and higher dimensional samples when calculating $C^{(k)}_{IFM}(\lambda, K)$ at small values of λ and σ. In such a situation we calculate an explicit leave-one-out version of our criterion, mean-

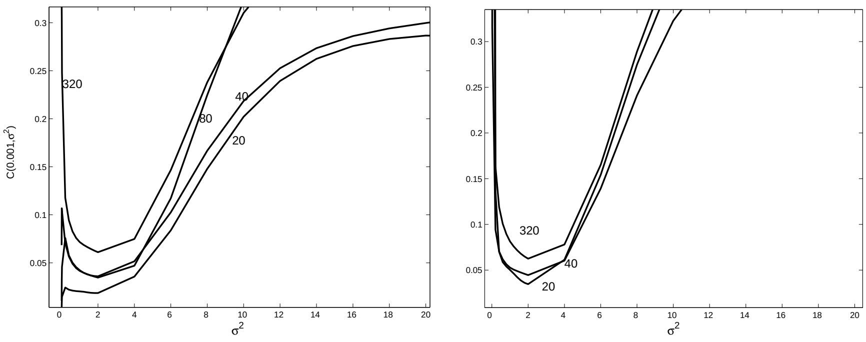


Figure 4.1 On the left the basic criterion $C_{IFM}(0:001;\sigma)$, on the right the cross validated correction, both for different sample sizes.

ing that we use $IF(z_i;f_\lambda,P_n^{-i})(x_i)$ on the diagonal of the influence matrix instead of $IF(z_i;f_\lambda,P_n)(x_i)$. In Figure 1 we show this for a simple sine function $y_i = \sin(x_i) + e_i$ with the errors e_i normally distributed. In the left the original criterion is used to calculate $C^{(k)}_{IFM}(\lambda,\sigma)$ as a function of σ. Note that the result is quite bad for small σ and small n, especially $n = 20$. The crossvalidated criterion on the right performs much better, but of course the computational cost is much higher. However we see that even at $n = 20$ a local minimum occurs in the region around $\sigma = 2$. Our strategy is thus to find the local minima of $C^{(k)}_{IFM}(\lambda,\sigma)$. If many occur, we select the best among these minima using the crossvalidated criterion.

4.5 Empirical results

4.5.1 Toy example

As a toy example 100 data points were generated with x_i uniformly distributed on the interval $[-5,5]$ and $y_i = \sin(x_i) + e_i$ with e_i Gaussian distributed noise with standard deviation 0.2. To illustrate robustness of the proposed criterion, an (extreme) outlier was added at position $(-2,20)$ (not visible on the plot). The dotted curve in Figure 4.2 is the result when an ordinary least squares method is used with *fixed* $\lambda = 0.005$ and a RBF kernel with bandwidth $\sigma = 1.4$. The outlier clearly has a huge effect around $x = -2$. If we now perform one-step reweighting with a logistic weight function $(w(r) = \tanh(r)/r)$, then the solid curve is the result. This confirms the theoretical results in Chapter 3:

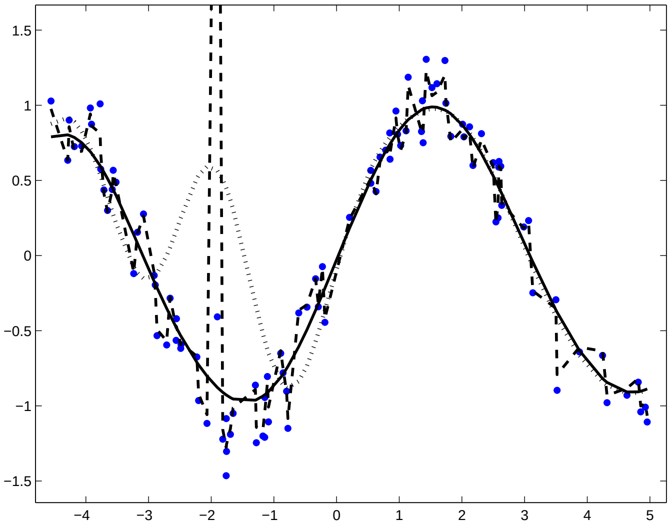
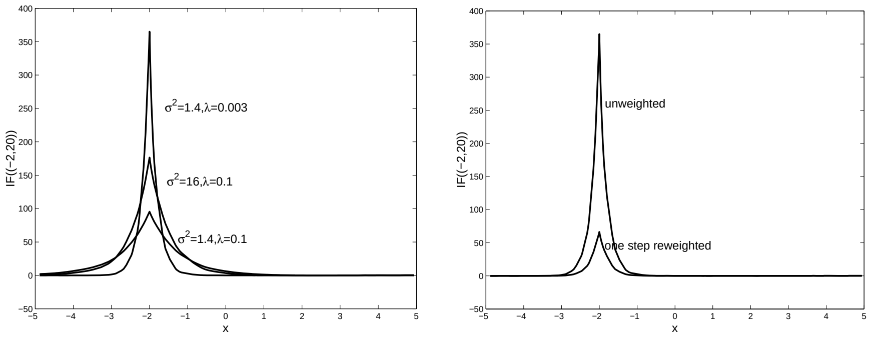


Figure 4.2 Sine curve toy example. Dotted: least squares kernel based regression, solid: one step reweighted estimator, both with the same a priori chosen λ and σ. The reweighted estimator is more robust. However, if λ and σ are chosen by an ordinary leave-one-out criterion, a completely wrong choice is made for λ and σ, even with the reweighted estimator (dashed).

for a least squares method the effect of 1 outlier can be arbitrary large. With reweighting this is not the case. Next we chose λ and σ in a data driven way. We used a leave-one-out criterion with the reweighted kernel estimator. Then the optimal values were around $\lambda = 0.0004$ and $\sigma = 0.05$ resulting in the dashed curve in Figure 4.2. Clearly there is severe overfitting. The presence of the outlier has completely distorted the choice of the optimal hyperparameters. This illustrates that the effect of contamination should be neutralized not only in the estimation procedure, but also in the model selection step.

Next we calculated the finite sample influence function at the outlier through expressions (4.4) and (4.5) taking $z = (-2, 20)$ and plotted this function in Figure 4.3(a). This means that we see the effect of adding a small amount of probability mass at the place $(-2, 20)$. We see that this effect is the largest around the region $x = -2$, where a large positive value is obtained (indicating that the estimator will increase at these points when adding probability mass). Further away from -2, the effect rapidly decreases, which is of course due to the local nature of the RBF kernel. The influence function also depends on the regular-



(a) (b)

Figure 4.3 Finite sample approximations of the influence function at the outlier $(-2, 20)$: (a) unweighted case for several values of λ, σ^2. (b) unweighted versus reweighted, both with $\lambda = 0.003$ and $\sigma^2 =$

ization constant λ and the RBF kernel bandwidth σ. For large values of λ and σ, the height of the peak rapidly decreases, indicating that the stability of the estimator increases as one expects. A good choice of λ and σ should result in a good balance between a small training error and a smooth influence function (good stability).

Next we can calculate pointwise confidence bands as in Corollary 4.3. These are depicted in Figure 4.4. With $\alpha = 0.05$ these error bands are plotted as dashed lines in Figure 4.4. Again the outlier has a big effect. Not only around the outlier at $x = -2$, but also in other areas the variances of the reweighted estimator is much lower. Consider for example the region around $x = 0.5$ where the reweighted estimator has a narrow error band, while the ordinary least squares still suffers from the outlier effects.

Instead of estimating $V_n(x_j)$ pointwise at every x_j, it is also instructive to look at an averaged variance over the entire sample. In Figure 4.5 we plot $\frac{1}{n}\sum_{j=1}^{n}\sqrt{V_n(x_j)}$ as a function of the bandwidth in 4 different situations: the one-step reweighted and unweighted estimator on the example data set with and without the outlier. If we exclude the outlier then both the weighted and unweighted estimators give more or less the same curve. Note that the data was generated with Gaussian errors with standard deviation 0.2. This value of 0.2 is indeed what we estimate for $\frac{1}{n}\sum_{j=1}^{n}\sqrt{V_n(x_j)}$ if we choose the bandwidth for which the variance is the lowest (σ^2 between 1 and 3). If we keep the outlier in the data set then the curve for the weighted estimator remains similar,

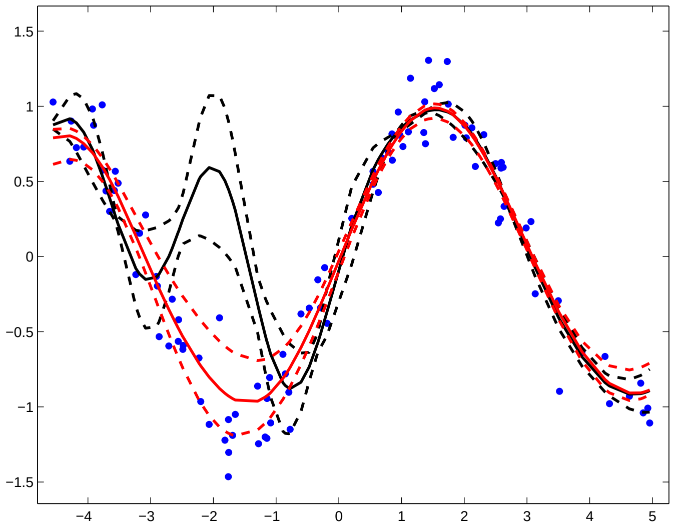


Figure 4.4 Least squares estimator (solid black) and reweighted estimator (solid gray) with 95% confidence bands (dashed) based on the influence function matrix.

although the least varying σ^2 would be chosen slightly smaller (between 1 and 1.5) albeit still in an acceptable region. In the unweighted case however we get a completely different curve. First the variance is blown up by the outlier. Second, the shape of the curve changes completely as well. It indicates that the best stability would be obtained either at very big or very small bandwidths which does not make sense.

For fixed λ it might thus be possible to choose a good bandwidth based on these variance-curves. However, variance is obviously not the only consideration. As a function of λ for example $\frac{1}{n}\sum_{j=1}^{n}\sqrt{V_n(x_j)}$ would be monotonically decreasing, since estimates are more and more stable as the regularization is stronger. It is of course important to the consider the bias as well, which was done in the main criterion from Definition 4.4

Figure 4.6 visualizes the resulting curves with one reweighting step as a function of σ for 4 different values of λ. From this plot a value of $\sigma = 2$ and $\lambda = 0.004$ would be chosen. This is also more or less the choice one would make with GCV or cross-validation *if the outlier were discarded*, but not if the outlier is part of the data set (compare to Figure 4.2).

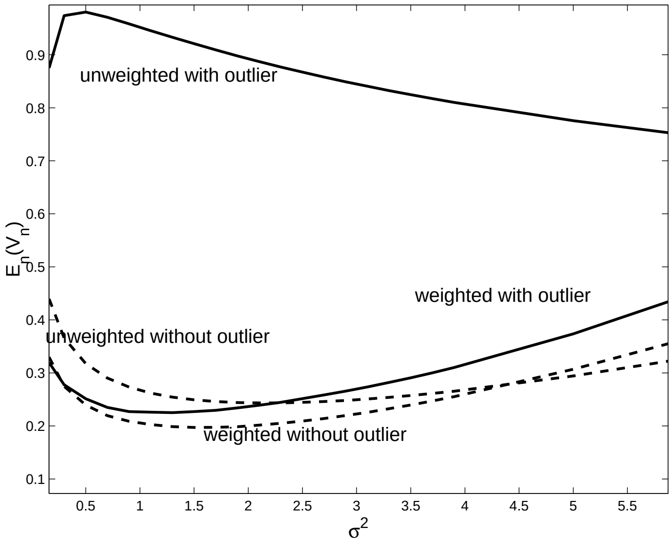


Figure 4.5 Average asymptotic variance as a function of σ^2 for the weighted/unweighted estimator with/without the outlier.

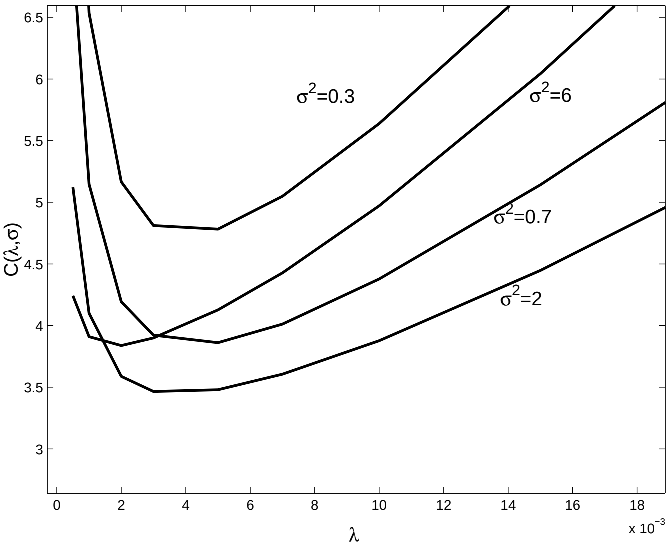


Figure 4.6 Sine example. The minimum of $C^{(1)}_{IFM}(\lambda,\sigma)$ is around (0.004,2).

Data	GCV	$C^{(1)}_{IFM}$
Friedman 1 (Gaussian)	2.93 (0.46)	3.10 (0.69)
Friedman 1 (Student)	4.45 (1.86)	4.31 (2.10)
Friedman 2 (Gaussian)	4090 (1242)	4267 (1681)
Friedman 2(Student)	8111 (7158)	7934 (10321)
Boston Housing	14.45 (5.86)	17.32 (5.81)
Contaminated Boston Housing	80.12 (78.76)	19.76 (6.13)

Table 4.1 MSE's (standard deviations) for several data sets.

4.5.2 Other examples

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

- Friedman 1 ($d = 10$): $y(x) = 10\sin(\pi x_1 x_2) + 20(x_3 - 1/2)^2 + 10x_4 + 5x_5 + \sum_{i=6}^{10} 0.x_i$.
- Friedman 2 ($d = 4$): $y(x) = (x_1^2 + (x_2x_3 - 1/x_2x_4)^2))^{1/2}$.
- Boston Housing Data from the UCI machine learning depository with 506 instances and 13 covariates.

In each replication 200 data points were generated. For the Friedman data sets [Friedman, 1991] inputs were generated uniformly from the unit hypercube. Noise was added to $y(x)$ from two distributions: first, Gaussian with unit variance and second, Student with 2 degrees of freedom. MSE's were calculated at 200 noisefree test points. The Boston Housing data set was split in 450 training points and 56 test points. A final data set was constructed from the Boston Housing data by randomly permuting the 13 variables for 10 training points, making these 10 points outliers. The average MSE's over 10 replicates are summarized in Table 1. In the first column we searched the optimal parameters with the GCV criterion and then performed least squares kernel regression with these parameters. The values in the second column were found by applying the one-step $C^{(1)}_{IFM}(\lambda, \sigma)$ criterion with logistic weight function $w(r) = \tanh(r)/r$. For both criteria we used a grid search over the parameter space. For GCV we took the global minimum. For our criterion we often encountered that the global minimum was not meaningful. Therefore we selected one of the local minimum with the corrected leave-one-out criterion (see Section 4.4.4). Then reweighted kernel regression with the same weight function was performed. It is clear that our criterion performs slightly worse than GCV at the original Boston Housing and both Friedman data sets with Gaussian errors. However for the Friedman data with Student errors reweighting has a positive effect. For the contaminated Boston Housing data the reweighted estimator performs a lot better. This illustrates again that the proposed method is robust against outliers, contrary to classical methods.

4.6 Conclusion

The influence function is a mathematical tool from robust statistics assessing the robustness of a method. In this chapter several links were studied with concepts of stability and variance. We defined the influence matrix. Its columns are closely related to asymptotic variance. We proposed a GCV type criterion using the trace of this influence matrix, in order to perform model selection for reweighted kernel regression estimators. The main advantage over the GCV criterion is the robustness of our proposal. If a small percentage of outliers is present in the data, this does not affect the choice of hyperparameters too much. A disadvantage on the other hand is the bad behavior when the hyperparameters λ and σ are small. In such areas a local minimum can be selected with a leave-one-out corrected criterion.

Chapter 5

Spherical Kernel PCA

5.1 Introduction

In Chapter 2 we considered Principal Component Analysis (PCA) as a technique designed to reduce the dimension of a data set by projecting onto a lower dimensional subspace. In this chapter we study Kernel PCA [Schölkopf et al., 1998], which is an extension of PCA using the same kernel based framework from Chapters 3 and 4 for regression. The data are first mapped into a high dimensional feature space. Then ordinary PCA is performed in this feature space. A remarkable aspect is that the explicit feature vectors are not needed to compute the resulting scores. Only the inner products between feature vectors are required. This makes it possible to apply the kernel trick: one replaces all inner products by a kernel function that is chosen beforehand. A more detailed description of Kernel PCA is provided in Section 5.2.

In particular this chapter addresses some questions about influential observations in Kernel PCA. In Chapter 2 we noticed that linear PCA is not robust. It is possible that one or a small fraction of observations in the data set almost fully determines the principal components. Sometimes this is not desirable, since then the structure of the majority of the data is not learned anymore.

In Section 5.3 we make an analysis of these effects for Kernel PCA with an arbitrary kernel. We show that the influence function of Kernel PCA is bounded if the kernel itself is bounded. For unbounded kernels the influence function can be arbitrary large.

The influence function is a functional analytic tool working at continuous distributions. In practice one deals of course with samples. An idea to assess the influence of individual observations in a sample, would be to simply plug in the sample Kernel PCA estimates into the expression for the influence functions. It is important to realize that this approach sometimes completely fails. Observations can be so influential that the resulting fits and diagnostics become unreliable, not revealing the true influences. In the field of robust statistics, this is known as the masking effect. We explain this effect at the end of Section 5.3 with a small example.

To overcome this problem robust methods should be used. These methods are constructed in such a way that no small group of observations can overrule the majority. For linear PCA many robust algorithms have been proposed, for instance ROBPCA from Chapter 2 but also in Hubert et al. [2002], Croux and Ruiz-Gazen [2005], Maronna [2005]. A robustification of Kernel PCA was presented by Alzate and Suykens [2005], but their approach mainly focuses on bounded kernels. Here we present Spherical Kernel PCA as a robust alternative to Kernel PCA for any type of kernel. It is a generalization of the linear method from Locantore et al. [1999]. The first step is a robust centering of the data, which is explained in Section 5.4. The Spherical KPCA procedure is given in Section 5.5.

To assess the influence of observations in a sample on the results of ordinary KPCA, we plug in the robust estimates from Spherical KPCA into the influence functions obtained for ordinary Kernel PCA. For linear PCA such an idea proved successful in Pison and Van Aelst [2004]. In Section 5.6 we show how to construct this diagnostic tool for KPCA. Section 5.7 illustrates everything on some examples.

5.2 Kernel PCA

Assume that we have a sample of n observations in some non-empty set $\mathcal{X}$: $x_i \in \mathcal{X}$, $i = 1, \ldots, n$. Suppose a kernel K is fixed with corresponding feature space $\mathcal{H}$ as in Definition 2. Then Kernel PCA basically performs linear PCA in this feature space $\mathcal{H}$ instead of the original space $\mathcal{X}$. Schölkopf et al. [1998] show that the solution of this problem can be obtained only in terms of the inner products between feature vectors. The score function $f_k : \mathcal{X} \to \mathbb{R}$ associated to the kth principal component can be found as follows:

$$f_k(x) = \sum_{i=1}^{n} \alpha_i \left(\langle \Phi(x_i), \Phi(x) \rangle - \frac{1}{n} \sum_{k=1}^{n} \langle \Phi(x_i), \Phi(x) \rangle \right).$$

The vector $\alpha = (\alpha_1, \ldots, \alpha_n)$ is the eigenvector belonging to the kth largest eigenvalue λ_k of the $n \times n$ matrix with entry i, j equal to

$$\left\langle \left(\Phi(x_i) - \frac{1}{n} \sum_{k=1}^{n} \Phi(x_k) \right), \left(\Phi(x_j) - \frac{1}{n} \sum_{k=1}^{n} \Phi(x_k) \right) \right\rangle$$

and $||\alpha_k||^2 = 1/\lambda_k$. Now the so-called kernel trick can be applied. This means that all inner products $\langle \Phi(u), \Phi(v) \rangle$ are replaced by $K(u, v)$, cfr. Definition 2. Making this substitution the kth score function of kernel PCA can be computed as

$$f_k(x) = \sum_{i=1}^{n} \alpha_i \left(K(x_i, x) - \frac{1}{n} \sum_{k=1}^{n} K(x_k, x) \right) \tag{5.1}$$

with α the eigenvector belonging to the kth largest eigenvalue λ_k of the mean centered kernel matrix $\Omega_c^{L_2}$. Entry i, j of this matrix equals

$$\left(\Omega_c^{L_2} \right)_{i,j} := \left(K(x_i, x_j) - \frac{1}{n} \sum_{k=1}^{n} K(x_k, x_j) - \frac{1}{n} \sum_{k=1}^{n} K(x_k, x_i) + \frac{1}{n^2} \sum_{k=1}^{n} \sum_{l=1}^{n} K(x_k, x_l) \right). \tag{5.2}$$

Retaining the $m \in \mathbb{N}_0$ scores corresponding to the m largest eigenvalues one finds a new set of m variables describing the data. These m scores can then be used to proceed the analysis, for instance by applying regression, classification, clustering etc. The idea is that for a well chosen kernel and m, these m scores contain all essential information, whereas all unimportant information is filtered out. In this spirit it is no surprise that image analysis is an area where KPCA is particularly popular. Images can sometimes be quite blurry. Using KPCA to filter out the noise often results in a much clearer image. A typical example is the data set with handwritten digits [Schölkopf and Smola, 2002]

5.3 Characterizing influential observations

Suppose that we have a distribution P on the input space $\mathcal{X}$. Define the operator

$$C_P : \mathcal{H} \to \mathcal{H} : f \to C_P(f) = \mathbb{E}_P f(X)\Phi(X) - \mathbb{E}_P f(X) \mathbb{E}_P \Phi(X).$$

If $\mathbb{E}_P||\Phi(X)||^2 < \infty$, then the operator C_P is a well-defined, compact, positive and self-adjoint Hilbert-Schmidt operator [Blanchard et al., 2007]. Therefore it has a countable spectrum of positive eigenvalues $\lambda_{P,1} > \lambda_{P,2} > \dots$ with an associated orthonormal basis of eigenfunctions $\{e_{P,i}\}$. Thus for any function $f \in \mathcal{H}$ we have that

$$f = \sum_{i=1}^{\infty} \langle f, e_{P,i} \rangle e_{P,i} \qquad \text{and} \qquad C_P(f) = \sum_{i=1}^{\infty} \lambda_{P,i} \langle f, e_{P,i} \rangle e_{P,i}.$$

Note that plugging in the empirical distribution P_n returns the eigenvalues $\lambda_{P_n,i}$ and eigenfunctions $e_{P_n,i}$ corresponding to the eigenvalues and scores of KPCA discussed in the previous section. Convergence results for $n \rightarrow \infty$ were obtained by Blanchard et al. [2007], Shawe-Taylor et al. [2002]. In this section we work with continuous distributions rather then empirical distributions since we would like to assess properties of the corresponding statistical functionals through the concept of the influence function. The following definition introduces the necessary tools.

Definition 5.1 *Given a distribution P with $\mathbb{E}_P||\Phi(X)||^2 < \infty$, then the statistical functionals C, λ_i resp. e_i map P onto $C(P) = C_P$, $\lambda_i(P) = \lambda_{P,i} \in \mathcal{H}$ resp. $e_i(P) = e_{P,i} \in \mathcal{H}$.*

We prove the following theorem for KPCA.

Theorem 5.2 *With the notation of Definition 5.1, the influence functions of λ_i and e_i equal*

$$IF(z; \lambda_i, P) = \langle e_{P,i}, \Phi(z) \rangle^2 - \lambda_i.$$
$$IF(z; e_i, P) = \langle e_{P,i}, \Phi(z) \rangle \sum_{j=1, j \neq i}^{\infty} \frac{\langle e_{P,j}, \Phi(z) \rangle}{\lambda_i - \lambda_j} e_j.$$

Proof

By definition we have that

$$\langle e_i(P_{\epsilon,z}), e_i(P_{\epsilon,z}) \rangle = 1.$$

Taking the derivative with respect to ϵ on both sides yields

$$\langle IF(z; e_i, P), e_i \rangle = 0.$$

Denote $\mathcal{H}^{\perp,i}$ the subspace of $\mathcal{H}$ orthogonal to the ith component. Then the previous equation means that $IF(z; e_i, P) \in \mathcal{H}^{\perp,i}$. Furthermore we have that

$$\begin{aligned}\lambda_i(P_{\epsilon,z})e_i(P_{\epsilon,z}) &= \mathbb{E}_{P_{\epsilon,z}}\langle e_i(P_{\epsilon,z}), \Phi(X)\rangle\Phi(X)\\ &= (1-\epsilon)\mathbb{E}_P\langle e_i(P_{\epsilon,z}), \Phi(X)\rangle\Phi(X) + \epsilon\langle e_i(P_{\epsilon,z}), \Phi(z)\rangle\Phi(z).\end{aligned}$$

Taking the derivative with respect to ϵ yields

$$\begin{aligned}&IF(z;\lambda_i, P)e_i(P) + \lambda_i(P)IF(z; e_i, P) \qquad (5.3)\\ &= -\mathbb{E}_P\langle e_i(P), \Phi(X)\rangle\Phi(X) + \mathbb{E}_P\langle IF(z; e_1, P), \Phi(X)\rangle\Phi(X) + \langle e_i(P), \Phi(z)\rangle\Phi(z)\end{aligned}$$

Now take the inner product of both sides with respect to $e_i(P)$. Then

$$IF(z;\lambda_i, P) = -\lambda_i + \langle e_{P,i}, \Phi(z)\rangle^2$$

proving the first statement. Using this result (5.3) can be rewritten as

$$(C_P - \lambda_i \mathrm{id}_{\mathcal{H}})(IF(z; e_i, P)) = \langle e_i(P), \Phi(z)\rangle^2 e_i(P) - \langle e_i(P), \Phi(z)\rangle\Phi(z).$$

The operator $(C_P - \lambda_i \mathrm{id}_{\mathcal{H}})$ does not have an eigenvalue equal to λ_i in $\mathcal{H}^{\perp,i}$. Thus the Fredholm alternative (Proposition 3.11)) shows that this operator is invertible and

$$IF(z; e_i, P) = (C_P - \lambda_i \mathrm{id}_{\mathcal{H}})^{-1}\left(\langle e_i(P), \Phi(z)\rangle^2 e_i(P) - \langle e_i(P), \Phi(z)\rangle\Phi(z)\right). \qquad (5.4)$$

Moreover we have that

$$(\lambda_j - \lambda_i)\langle IF(z; e_i, P), e_j(P)\rangle = \langle e_i(P), \Phi(z)\rangle\langle e_j(P), \Phi(z)\rangle$$

for any $j \neq i$. Thus

$$IF(z; e_i, P) = \langle e_{P,i}, \Phi(z)\rangle \sum_{j=1, j\neq i}^{\infty} \frac{\langle e_{P,j}, \Phi(z)\rangle}{\lambda_i - \lambda_j} e_j.$$

proving the second statement.

□

This theorem reveals two interesting properties. First we see that estimating an eigenfunction is very sensitive to small distributional changes when other eigenvalues are very close to its corresponding eigenvalue. This is well known for instance in linear PCA. As a limit case consider a spherical distribution.

Then a first principal component is not well defined, since all directions give raise to the same projected variance. This changes of course if an arbitrary small amount of Dirac probability mass is put at any point z except for the center of the distribution. Then the direction through z and the center of the distribution will be the first principal component. In this case, an infinitesimally small amount of probability mass fully determines the first component, which is reflected in an infinitesimally large influence function.

Now suppose again that all eigenvalues are different. An important question is whether the influence function can become arbitrary large in such a case as well. Theorem 5.2 tells us exactly how to choose z in order to achieve this: both the score with respect to its own component and the sum of the scores with respect to the other components should be large. Figure 5.1 shows a classical example of a very influential point (denoted as observation 11) having a large influence on the components of a two-dimensional Gaussian distribution, in case of a linear kernel.

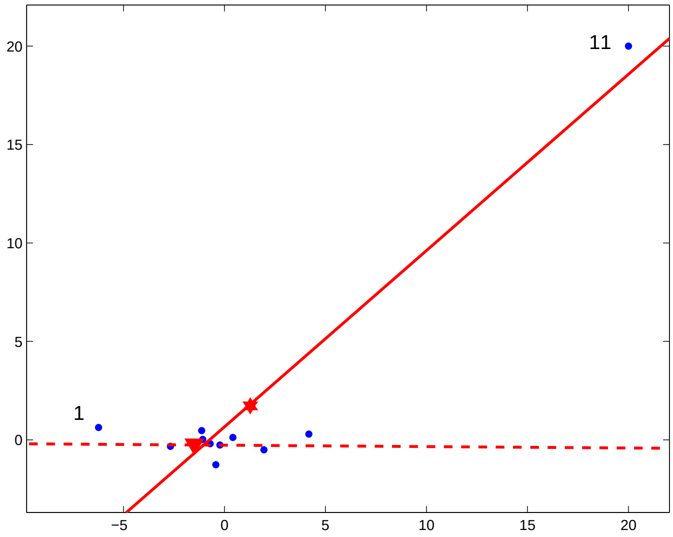


Figure 5.1 The influence of a single observation on the mean and on the first principal component of linear PCA can be arbitrary large.

For a bounded kernel however this is different. From the previous theorem upper bounds on the influence function in terms of bounds on the kernel can be derived.

Theorem 5.3 *With the notation of Definition 5.1 and a feature map with bounded* *norm, i.e. there exists $M > 0$ such that $||\Phi(z)|| \leq M$, the following bounds hold:*

$$|IF(z; \lambda_i, P)| \leq M^2 + \lambda_i.$$
$$||IF(z; e_i, P)|| \leq \frac{2M^2}{\min_j |\lambda_i - \lambda_j|}.$$

Proof

The Cauchy-Schwarz theorem guarantees that

$$|\langle e_{P,i}, \Phi(z) \rangle|^2 \leq ||e_{P,i}||^2 ||\Phi(z)||^2 = ||\Phi(z)||^2 < M^2$$

for any $i \in \mathbb{N}$. Together with Theorem 5.2 this immediately gives the upper bound for the influence function of the eigenvalues. For the eigenfunctions equation (5.4) shows that

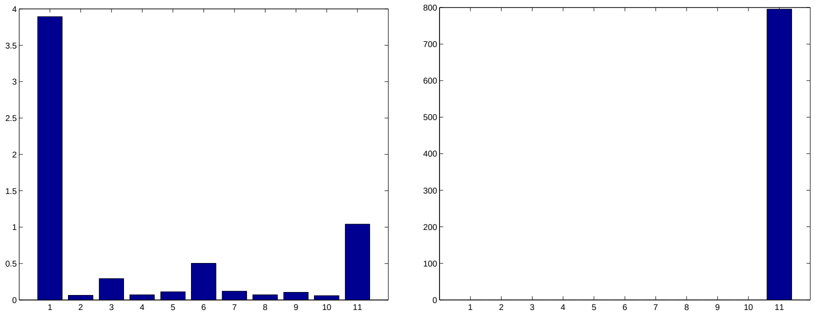
$$||IF(z; e_i, P)|| \leq ||(C_P - \lambda_i \mathrm{id}_{\mathcal{H}})^{-1}|| \; ||\langle e_i(P), \Phi(z) \rangle^2 e_i(P) - \langle e_i(P), \Phi(z) \rangle \Phi(z)||.$$

The norm of the operator in the first term is bounded by its largest eigenvalue which equals $\left(\min_j |\lambda_j - \lambda_i|\right)^{-1}$. Cauchy-Schwarz bounds the second term by $2M^2$.

□

This indicates a crucial difference between bounded and unbounded kernels in terms of robustness of kernel PCA. Similar conclusions were obtained for classification [Christmann and Steinwart, 2004] and regression (Christmann and Steinwart [2006] and Chapter 3, see the discussion of Proposition 3.5, p.60 and Corollary 3.9, p.64). For an RBF kernel for example, we can take $M = 1$ showing a bounded influence. On the other hand the spacings between eigenvalues play an important role as well. For instance as the bandwidth σ of the RBF kernel reaches infinity, the kernel matrix converges to the matrix with all entries equal to 1. Thus all eigenvalues converge to the same value and the upper bound becomes arbitrary large as $\sigma \to \infty$. This is quite expected, since an RBF kernel with large σ approaches linear PCA for which the influence function is indeed unbounded. Therefore observations might still be rather influential even for an RBF, especially if the parameter σ is chosen in a data-driven way. Theorem 5.3 shows however that the worst cases appear for unbounded kernels.

The main problem with unbounded kernels is that the effect of one or few observations can be so big that their influence is not easily detected anymore. One could for instance think about plugging in the sample eigenvalues and scores from Kernel PCA in the expressions for the influence function. Taking the norm of the resulting influence vector gives a data based diagnostic tool assessing the influence of each observation in the sample. In Figure 5.2 (a) one



(a) (b)

Figure 5.2 Estimated influences based on (a) classical linear PCA, (b) robust linear PCA.

sees however that such an approach completely fails even for the simple example from Figure 5.1. If one point would be considered an outlier, it would be observation 1 for which the influence is the highest. The really influential observation 11 is only considered moderately influential. Looking back to Figure 5.1 one understands why. According to Theorem 5.2 influential points are characterized by the product of the first and second principal component scores. For observation 1 both terms are rather large leading to a large product. For observation 11 the first score is very large, but the second score is small giving only a moderately large product. In robust statistics such a phenomenon is called the *masking effect*: the influence of point 11 is so huge that it affects estimates and diagnostics so heavily that its influence is actually hidden!

One way around this problem is to use a robust method. Then the principal components are constructed in a such a way that a small fraction of the data can never demolish an entire fit. In Figure 5.1 the spherical PCA method of Locantore et al. [1999] was used to find the dashed line as first principal component. When we plug in these results into Theorem 5.2, we get Figure 5.2 (b) as resulting diagnostic plot. This provides clearly a much better graphical display of the influence of each observation, revealing the huge domination of point 11 over the others. In this paper we propose Spherical KPCA as a general robust KPCA method that can be used for any kernel. It is an extension of the ideas from Locantore et al. [1999] for linear PCA to KPCA in a kernel-induced feature space.

5.4 Robust centering

5.4.1 Spatial median

The first step of PCA consists of centering the data, usually around the mean. However, the mean is not a robust measure of the center. Again one observation can have an arbitrary large influence. In Figure 5.1 for example the mean (pictured as a star) is clearly influenced a lot by observation 11. A first logical step in a robust PCA procedure consists of a more robust centering. In this section we propose to use the L_1 M-estimate of location, which is a multivariate extension of the univariate median and which has been around for a long time (see for instance Huber [1981] and Haldane [1948]). This location measure is also known as the spatial median. It has a nice geometrical interpretation [Small, 1990]: take a point θ in $\mathbb{R}^d$ and project all observation onto a sphere around the center θ. If the mean of these projection equals θ, then θ equals the spatial median. Figure 5.3 shows this for the previous two-dimensional example. The mean of the data projected on the sphere (these projections are pictured as crosses) around the asterisk does not equal the asterisk at all. The asterisk is a bad estimator of location indeed. The mean of the data projected on the sphere around the triangle does equal the triangle itself. Thus the triangle indicates the position of the spatial median. Note how the sphering reduces the influence of observation 11, such that it does not affect the spatial median more than any of the other observations.

Definition 5.4 *Given a sample of inputs $x_i \in \mathbb{R}^d$, $i = 1, \ldots, n$, the spatial median θ is defined as the solution of*

$$\sum_{i=1}^{n} \frac{x_i - \theta}{||x_i - \theta||} = 0.$$

For the computation of this center, the following simple iterative algorithm exists [Gower, 1974, Huber, 1981]. Given an initial guess $\theta^{(0)} \in \mathbb{R}^d$, iteratively

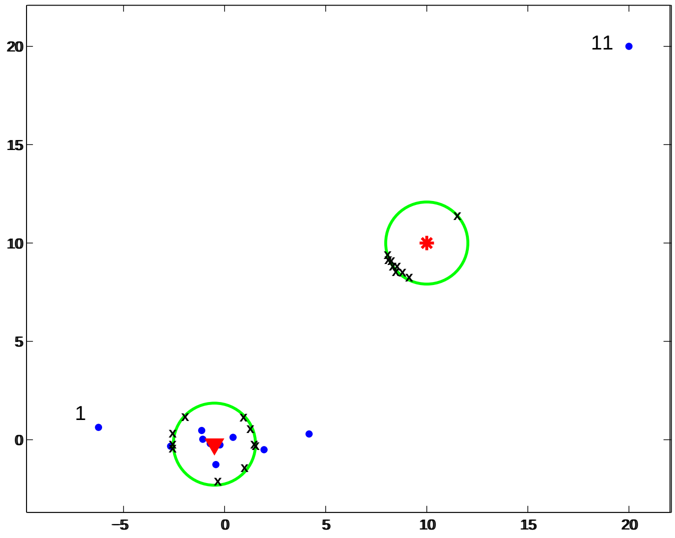


Figure 5.3 When projecting all data on a sphere around the star, the mean of these projection (depicted as crosses) does not equal the center of the sphere. For the triangle, it does. By definition, the triangle equals the spatial median. Note the moderate influence of observation 11.

define

$$\theta^{(k)} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

where

$$w_i = \frac{1}{||x_i - \theta^{(k-1)}||}.$$

Other algorithms exist as well, see for example Hössjer and Croux [1995], but we stick to the former since it allows an extension to a kernelized version.

5.4.2 Spatial median in feature space

Assume again that the inputs x_i are mapped into a high- (possibly infinite) dimensional feature space $\mathcal{H}$. Applying Definition 5.4 in feature space means that we want to find $\theta \in \mathcal{H}$ such that

$$\sum_{i=1}^{n} \frac{\Phi(x_i) - \theta}{||\Phi(x_i) - \theta||} = 0.$$

This is equivalent to demanding that

$$\left\| \sum_{i=1}^{n} \frac{\Phi(x_i) - \theta}{||\Phi(x_i) - \theta||} \right\|^2 = 0$$

or if we write out the norms as inner products

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \left\langle \frac{\Phi(x_i) - \theta}{||\Phi(x_i) - \theta||}, \frac{\Phi(x_j) - \theta}{||\Phi(x_j) - \theta||} \right\rangle = 0$$

which is equivalent to

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\langle \Phi(x_i), \Phi(x_j) \rangle - \langle \theta, \Phi(x_j) \rangle - \langle \theta, \Phi(x_i) \rangle + \langle \theta, \theta \rangle}{\sqrt{A_i}\sqrt{A_j}} = 0. \tag{5.5}$$

with the notation

$$A_i = \langle \Phi(x_i), \Phi(x_i) \rangle - 2\langle \Phi(x_i), \theta \rangle + \langle \theta, \theta \rangle.$$

If the mapping Φ is explicitly known, one could use this equation to find the center θ. In most kernel applications however, this is of course not the case. The following lemma provides a way out.

Lemma 5.5 *The spatial median in feature space can always be written as a linear combination of the n observations in feature space: $\theta = \sum_{k=1}^{n} \gamma_k \Phi(x_k)$.*

This is simply because the spatial median naturally lies in the space spanned by the n inputs, and any point in this $\min(n, d)$-dimensional space can be parametrized as a linear combinations of the inputs.

Using this representation in (5.5) we find

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \left\{ \frac{\langle \Phi(x_i), \Phi(x_j) \rangle - \sum_{k=1}^{n} \gamma_k \langle \Phi(x_k), \Phi(x_j) \rangle}{\sqrt{A_i}\sqrt{A_j}} \right. \tag{5.6}$$

$$\left. - \frac{\sum_{k=1}^{n} \gamma_k \langle \Phi(x_k), \Phi(x_i) \rangle + \sum_{k=1}^{n} \sum_{l=1}^{n} \gamma_k \gamma_l \langle \Phi(x_k), \Phi(x_l) \rangle}{\sqrt{A_i}\sqrt{A_j}} \right\} = 0$$

with

$$A_i = \langle \Phi(x_i), \Phi(x_i) \rangle - 2 \sum_{k=1}^{n} \gamma_k \langle \Phi(x_i), \Phi(x_k) \rangle + \sum_{k=1}^{n} \sum_{l=1}^{n} \gamma_k \gamma_l \langle \Phi(x_k), \Phi(x_l) \rangle.$$

In this representation the spatial median can be expressed in terms of inner products only. Therefore this center can be just as well defined in a kernel-induced feature space, using the same kernel trick as in kernel PCA replacing $\langle \Phi(u), \Phi(v) \rangle$ by $K(u, v)$.

Definition 5.6 *Given a sample of inputs $x_i \in \mathcal{X}$, $i = 1, \ldots, n$ and a kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R} : (u, v) \to K(u, v)$. Define the $n \times n$ kernel matrix as $\Omega_{i,j} = K(x_i, x_j)$. Denote $\Omega_{.,j}$ as the jth column of this matrix. Then the vector of coefficients $\gamma \in \mathbb{R}^n$ determining the spatial median in the kernel induced features space is defined by*

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\Omega_{i,j} - \gamma^t \Omega_{.,j} - \gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma}{\sqrt{\Omega_{i,i} - 2\gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma}\sqrt{\Omega_{j,j} - 2\gamma^t \Omega_{.,j} + \gamma^t \Omega \gamma}} = 0.$$

To compute the vector γ the iterative algorithm in Section 5.4.1 can easily be modified to be computed in a kernel-induced feature space, only using the kernel inner product.

Given an initial guess $\gamma^{(0)} \in \mathbb{R}^n$, iteratively define

$$\gamma^{(k)} = \frac{w}{\sum_{i=1}^{n} w_i}$$

where $w \in \mathbb{R}^n$ has components

$$w_i = \frac{1}{\sqrt{\Omega_{i,i} - 2\gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma}}.$$

For the starting point we take the coefficients corresponding to the mean: $\gamma^{(0)} = (1/n, \ldots, 1/n) \in \mathbb{R}^n$. In any data set we tried, the algorithm took 20 or less step to converge to the solution giving 0 in the expression of Definition 5.4, indicating similar good behavior as the original algorithm.

5.4.3 Centering the kernel matrix around the spatial median

The resulting center in the kernel feature space can of course not be computed. We do find the n coefficients γ_k such that the spatial median equals $\sum_{k=1}^{n} \gamma_k \Phi(x_k)$, but the feature map Φ is unknown. However, operations involving distances and inner products between feature vectors and the center often can be computed. A well known operation is for instance centering of the data. Suppose we want to center the data in feature space around the spatial median. We define a new feature map as

$$\tilde{\Phi}(x) = \Phi(x) - \sum_{i=1}^{n} \gamma_i \Phi(x_i).$$

The corresponding centered kernel function K_c becomes

$$\begin{aligned}K_c(x,z) &= \langle \tilde{\Phi}(x), \tilde{\Phi}(z)\rangle \\ &= \langle \Phi(x) - \sum_{i=1}^{n} \gamma_i \Phi(x_i), \Phi(z) - \sum_{i=1}^{n} \gamma_i \Phi(z_i)\rangle \\ &= K(x,z) - \sum_{i=1}^{n} \gamma_i K(x,x_i) - \sum_{i=1}^{n} \gamma_i K(z,x_i) + \sum_{i=1}^{n}\sum_{j=1}^{n} \gamma_i \gamma_j K(x_i,x_j)\end{aligned}$$

or expressed in terms of matrix operations on the kernel matrix:

$$\Omega_c^{L_1} = \Omega - \gamma 1_n^t \Omega - \Omega 1_n \gamma^t + \gamma^t \Omega \gamma 1_n 1_n' \tag{5.7}$$

where 1_n is a vector containing 1 in its n entries.

Thus first computing γ as in Definition 5.4 with the algorithm from the previous paragraph and then computing (5.7), gives a robustly centered kernel matrix centered around the spatial median instead of the mean.

5.5 Spherical KPCA

5.5.1 Spherical PCA

Once the data is centered in an appropriate robust way, we can continue estimating the kernel principal components. We use the idea first mentioned in Locantore et al. [1999]. Basically they project the data on a sphere around the L_1 median. Then the traditional components are computed for these projected data. Scores are computed by projecting the original, unsphered data on the principal directions. Figure 5.4 shows the algorithm in practice. Due to the sphering the influence of the outlier is obviously heavily reduced leading to principal components capturing the structure of the majority of the data much better. Marden [1999] shows that these spherical principal components are exactly equal to the original ones at population level for a rather large class of distributions.

5.5.2 Spherical PCA in feature space

Assume that $\gamma \in \mathbb{R}^n$ is the vector of coefficients determining the spatial median in feature space $\sum_{k=1}^{n} \gamma_k \Phi(x_k)$. In the first step we project all feature vectors onto the unit sphere around the spatial median, giving us new feature vectors

$$\Phi^*(x_i) = \frac{\Phi(x_i) - \sum_{k=1}^{n} \gamma_k \Phi(x_k)}{||\Phi(x_i) - \sum_{k=1}^{n} \gamma_k \Phi(x_k)||}. \tag{5.8}$$

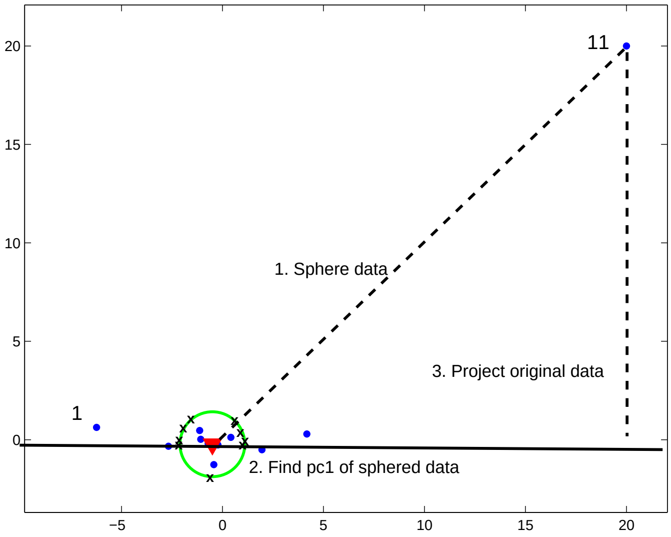


Figure 5.4 Spherical PCA in a simple 2-dimensional example.

This means that

$$\langle \Phi^*(x_i), \Phi^*(x_j) \rangle = \left\langle \frac{\Phi(x_i) - \sum_{k=1}^{n} \gamma_k \Phi(x_k)}{||\Phi(x_i) - \sum_{k=1}^{n} \gamma_k \Phi(x_k)||}, \frac{\Phi(x_j) - \sum_{k=1}^{n} \gamma_k \Phi(x_k)}{||\Phi(x_j) - \sum_{k=1}^{n} \gamma_k \Phi(x_k)||} \right\rangle.$$

In terms of the original and uncentered kernel matrix Ω, this leads to a new kernel matrix Ω^* with entries

$$\begin{aligned} \Omega^*_{i,j} &:= \langle \Phi^*(x_i), \Phi^*(x_j) \rangle \\ &= \frac{\Omega_{i,j} - \gamma^t \Omega_{.,j} - \gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma}{\sqrt{\Omega_{i,i} - 2\gamma^t \Omega_{.,i} + \gamma^t \Omega \gamma}\sqrt{\Omega_{j,j} - 2\gamma^t \Omega_{.,j} + \gamma^t \Omega \gamma}}. \end{aligned} \tag{5.9}$$

Thus once the spatial median is found, it is easy to compute the new kernel matrix Ω^* belonging to the sphered data based on the kernel matrix Ω of the original data.

In the second step, ordinary KPCA is applied to the sphered data. This means that we compute the eigenvectors and eigenvalues of Ω^* which we denote by α^*_k resp. λ^*_k where $\lambda^*_1 > \lambda^*_2 > \ldots$ and $||\alpha^*_k||^2 = 1/\lambda^*_k$.

Thirdly the score $f^*_k(x)$ of any point x for the kth component is computed by

$$f^*_k(x) = \sum_{i=1}^{n} (\alpha^*_k)_i \Phi^*(x_i)^t \left(\Phi(x) - \sum_{k=1}^{n} \gamma_k \Phi(x_k) \right)$$

or equivalently

$$f_k^*(x) = \sum_{i=1}^{n} (\alpha_k^*)_i K^*(x_i, x) \left(K(x,x) - 2\sum_{k=1}^{n} \gamma_k K(x_k, x) + \sum_{k=1}^{n}\sum_{l=1}^{n} \gamma_k \gamma_l K(x_k, x_l) \right). \tag{5.10}$$

These are the Spherical KPCA scores that provide a robust alternative to the classical KPCA scores from (5.1).

5.6 Finding influential observations

The spherical KPCA scores themselves can be useful in many applications. The example in Figure 5.1 shows for instance that spherical KPCA with a linear kernel (dashed line) produces scores that capture the structure of the majority of the data much better than ordinary KPCA (solid line). However, in high dimensions it is more difficult to visualize the difference between both methods. In this section we propose a simple graphical display to assess the influence of observations with respect to ordinary KPCA. Our strategy is to use the spherical KPCA estimates in the expressions for the influence function in Theorem 5.2. For linear PCA this idea was applied by Pison and Van Aelst [2004]. We use the score function $f_k^*(x)$ as a sample estimate of $e_{P,k}$. However, as explained by Marden [1999], the spherical eigenvalues λ_k^* are not always good estimates of $\lambda_{P,k}$. But since $\lambda_{P,k}$ equals the variance of the score function, we can re-estimate these eigenvalues by a measure of spread of the scores at the data points. Of course this measure of spread should not be influenced too much by individual observations either, so taking the variance would not be such a good idea. We use the more robust Median Absolute Deviation (MAD) to define

$$\lambda_k^{**} = \left(\text{median}(|f_k^*(x_i) - \text{median}(x_i)|)\right)^2. \tag{5.11}$$

Other robust scale estimators, such as the Q_n-estimator [Rousseeuw and Croux, 1993], are possible as well. Now observe from Theorem 5.2 that

$$||IF(z; e_k, P)|| = |\langle e_{P,k}, \Phi(z)\rangle| \sqrt{\sum_{j=1, j\neq k}^{\infty} \frac{\langle e_{P,j}, \Phi(z)\rangle^2}{(\lambda_k - \lambda_j)^2}}$$

which gives us the following sample based influence diagnostic equal to the norm of the empirical influence function at z of the kth component:

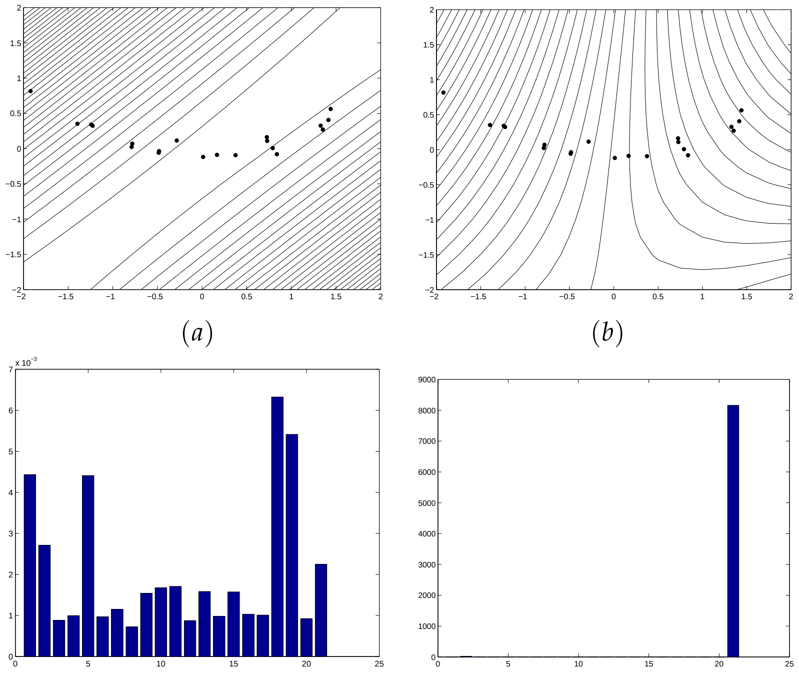
$$||EIF_k(z)|| = |f_k^*(z)|\sqrt{\sum_{j=1,j\neq k}^{n} \frac{f_j^*(z)^2}{(\lambda_k^{**} - \lambda_j^{**})^2}}. \tag{5.12}$$

To obtain the influence of an observation x_i, just take $z = x_i$. A bar plot of these values for all 11 observations in the sample from the example in Figure 5.1 is shown in Figure 5.2 (b), for a linear kernel and the first component ($k = 1$). Again note how using the classical PCA scores fails to provide an accurate description of the data (Figure 5.2 (a)). For linear PCA other robust methods would be able to give good results as well. Our method however can deal with any type of kernel. In the next section we show some examples where spherical KPCA is able to detect influential observations in more general kernel based frameworks.

5.7 Examples

5.7.1 Toy example

We explained the methodology on a toy example for the linear kernel (see Figure 5.1). Consider now a second toy example where the underlying structure of the data is not linear, but generated from the model $x_2 = x_1^2/4$ with random Gaussian noise with standard deviation 0.1. In Figure 5.5 we generated 20 data points showing a quadratic curvature, together with 1 outlier at $(-5, 5)$ (not visible on the plot). If we construct the score-contours corresponding to the first principal component of ordinary KPCA with a polynomial kernel of degree 2, we get Figure 5.5(a). Clearly the quadratic structure is lost completely due to the single outlier. For Spherical KPCA, Figure 5.5(b) depicts the corresponding score-contours. Now the quadratic structure of the majority of observations is learned, despite the outlier. Also note how the influence of outlying observation 21 on ordinary KPCA is not detected using ordinary KPCA itself (diagnostic plot Figure 5.5 (c)), whereas Spherical KPCA easily spots it (Figure 5.5 (d)).



(a) (b)

(b) (d)

Figure 5.5 Score-contours and estimated influences for $(a) - (c)$: classical KPCA, $(b) - (d)$: spherical KPCA.

5.7.2 String kernel

Consider a situation where the inputs are no vectors, but strings. Then many kernels exist that can be used to identify patterns in this set of strings. Here we concentrate on one example, i.e. the all-subsequence kernel. Then the strings are represented by feature vectors of which each component represents a possible substring. For the three strings ″gca″, ″cag″ and ″ggc″ for instance the corresponding feature vectors are:

	∅	a	c	g	ag	ca	cg	ga	gc	gg	gca	cag	ggc
gca	1	1	1	1	0	1	0	1	1	0	1	0	0
cag	1	1	1	1	1	1	1	0	0	0	0	1	0
ggc	1	0	1	2	0	0	0	0	1	1	0	0	1

So in this case every string can be represented as a vector with 13 components, and thus analysis could proceed in a 13-dimensional space. However, this example is extremely simple, since there are only three possible characters (a,c,g) and only strings of size three are considered. Unfortunately the dimension of the feature space increases exponentially with the size of the strings. For longer strings the explicit computation of the feature vectors thus becomes infeasible. However, when using a kernel method these explicit representations are not necessary. All we need are the inner products between any two feature vectors. For the all-subsequence kernel the kernel matrix containing these inner products can be computed with fast recursive algorithms [Shawe-Taylor and Cristianini, 2004]. Since spherical KPCA does not require explicit feature vectors either, but only the kernel matrix, applying the methodology from the previous sections is straightforward.

As an example take the first 20 DNA sequences in the ′Primate splice-junction gene sequences′ database from the UCI machine learning database. This gives us 20 observations, all strings of size 60 composed out of 4 characters (A,C,G,T). The first 3 elements are shown below.

′CCAGCTGCATCACAGGAGGCCAGCGAGCAGGTCTGTTCCAAGGGCCTTCGAGCCAGTCTG′,

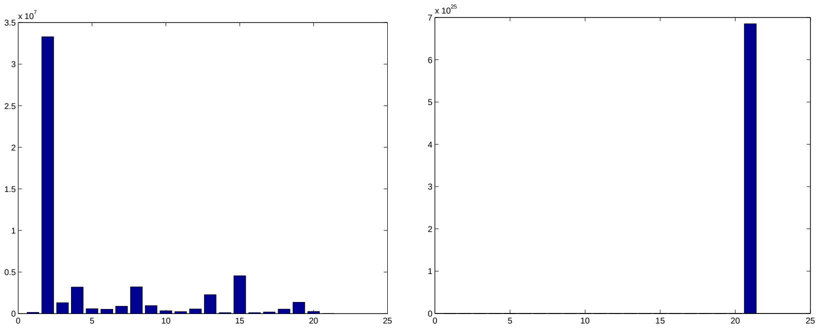
′AGACCCGCCGGGAGGCGGAGGACCTGCAGGGTGAGCCCCACCGCCCCTCCGTGCCCCCGC′,

′GAGGTGAAGGACGTCCTTCCCCAGGAGCCGGTGAGAAGCGCAGTCGGGGGCACGGGGATG′.

As an example we add one strange string to the data set, observation 21, which is the following sequence:

’CCCCCCCCCCCCCCCAAAAAAAAAAAAAATTTTTTTTTTTTTTGGGGGGGGGGGGGGGGG’.

Although the length of this string and the number of A’s,C’s,G’s and T’s are both similar as for the other strings, this new observation 21 is clearly different due to the specific order of the characters. Next we perform KPCA and spherical KPCA on this data set with the all-subsequence kernel. For each string we compute its influence measure as in (5.12) with respect to the first principal component. Figure 5.6(a) shows the result if we use the original KPCA scores and eigenvalues. String number 2 comes out as the most influential observation. Nevertheless it does not look extremely dominating and one would probably not suspect big problems. One would certainly not detect that observation 21 is an exceptional string, since its influence measure is very small. The results using spherical KPCA are depicted in Figure 5.6(b). Then it is immediately clear that we have the same effect we discussed for the simple toy example in Figures 5.1 and 5.2. Observation 21 is in reality extremely influential, dominating the estimation of the ordinary first kernel principal component completely. This first principal component is completely attracted by string 21. Therefore using this component results in a misleading plot of the influences. Only by using the spherical kernel principal components a correct assessment can be made about observations deviating from the mainstream. Also note that robust linear PCA methods cannot be used. They require the explicit feature vectors corresponding to the strings. However, according to Shawe-Taylor and Cristianini [2004], the dimension of these feature vectors would be likely to exceed 4^{30} in this example, which is obviously infeasible.



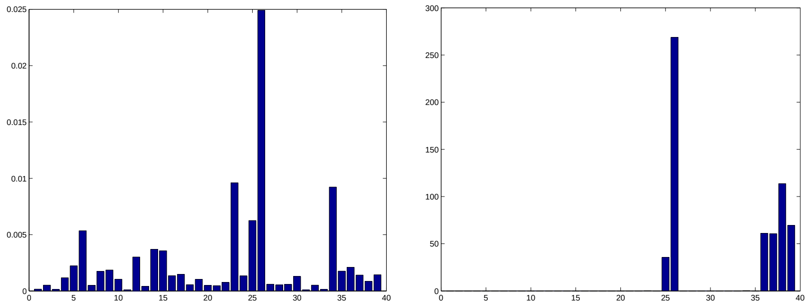
(a) (b)

Figure 5.6 Estimated influences based on (a) KPCA, (b) spherical KPCA.

5.7.3 Octane data

The next example is the octane data set described in Esbensen et al. [1994]. It contains near-infrared (NIR) absorbance spectra over 226 wavelengths of $n = 39$ gasoline samples with certain octane numbers. It is known that six of the samples $(25, 26, 36 - 39)$ contain added alcohol. The data set was also analyzed in Hubert et al. [2005], where it was shown that the robust linear PCA method ROBPCA was able to detect the six outlying samples in contrast to ordinary linear PCA. Now suppose that we increase the difficulty of the problem by using a polynomial kernel of degree 2. In theory the corresponding feature vectors could be computed by taking appropriately weighted squares and cross-products for all 226 variables. In practice the resulting dimension of these feature vectors will again be way too high. Explicitly calculating quadratic forms and then applying a robust method such as ROBPCA in feature space is thus infeasible.

Using a kernel method avoids this problem. All we need is the 39×39 dimensional kernel matrix, both for ordinary as spherical kernel PCA. The resulting diagnostic plots are shown in Figure 5.7. Part (a) of this plot depicts



(a) (b)

Figure 5.7 Octane data: estimated influences based on (a) KPCA, (b) spherical KPCA.

the results using ordinary KPCA. Of course some points seem more influential than others, but no dramatic effects would be detected. Part (b) shows the influence measures using spherical KPCA. Now we see what is really happening: six observations are extremely influential, dominating all others. These six observations are exactly the outlying samples that contain alcohol.

5.8 Conclusion

This chapter characterized the influence of data points on the results of Kernel PCA by calculating the influence function. It turns out that this depends on the spacings between eigenvalues, the kernel and the scores themselves. For bounded kernels we provided a bound on the influence function.

Secondly, for any type of kernel, Spherical KPCA was introduced as an alternative to ordinary KPCA spreading the influence more evenly over every observation.

Thirdly, the results from Spherical KPCA were used as plug-in estimates in the expression for the influence function. As such an easy graphical display was obtained to detect influential observations. Some examples demonstrated that this approach can produce good results where ordinary KPCA fails.

Conclusion

In this dissertation robustness was discussed in several settings. In Chapter 1 an algorithm was proposed to perform robust quantile regression dealing with right censored observations. Compared to L_1-quantiles our deepest regression based method is more time-consuming and less efficient at traditional regression setups, especially at higher dimensions and quantiles closer to the boundaries 0 and 1. In return however, much better protection against malicious outlier effects is obtained.

In Chapter 2 some theoretical results were obtained about the covariance estimator called Stahel-Donoho with smallest outlyingness. It was shown that this estimator has a bounded influence function (Theorem 2.2). Gross error sensitivities and asymptotic efficiencies were computed (Table 2.1), showing robustness and performance in between the weighted Stahel-Donoho estimator and the MCD estimator. Next we considered influence functions of ROBPCA (Theorem 2.5) and RSIMPLS (Theorem 2.6). Again these functions are bounded, showing robustness indeed.

From linear PCA and regression analysis we turned to kernel methods. In Chapter 3 we discussed reweighted KBR. An expression for the influence function of stepwise reweighted KBR was obtained at every step. In order to bound the influence function of the iteratively reweighted estimator until convergence, some sufficient conditions on the weight function were obtained (3.19). Not all frequently used weight functions satisfy these terms, i.e. Hampel's suggestion. In Figure 3.1 we were able to construct a data example where Hampel weights fail indeed. We put logistic weights in the spotlight since they obey all conditions, provide good results and fast convergence.

Up to that point, the influence function was mainly used as a tool to assess the theoretical robustness of an estimator looking at worst case scenarios. More generally one can see the influence function as a way to analyze the effects of small changes in a distribution. Some links with stability and variance were explained. In Section 3.5 we shortly reviewed the concept of stability and used it to motivate that reweighting also improves results when heavy-tailed noise occurs.

Chapter 4 continued discussing some heuristic links between influence function and stability, variance and the leave-one-out error. More specifically we tried to use the influence function as an asymptotic approximation of the leave-one-out error. A fast model selection criterion was obtained to select both regularization and kernel parameters. Since the same reweighting scheme from Chapter 4 is applied, the resulting model selection criterion is more robust than traditional methods. However, we also noted that the asymptotic approximation can become problematic for small values of the hyperparameters as well as in high dimensions. Several local minima occur, and thus it is necessary to complement this criterion with explicitly performing cross-validation, in order to select the correct local minimum.

Chapter 5 concerned Kernel PCA. The influence function was derived (Theorem 5.2). We bounded this function in case the kernel is bounded. For unbounded kernels the influence function can be unbounded. Therefore we proposed Spherical Kernel PCA as a more robust Kernel PCA method. Finally we proposed a simple graphical display to assess the influence of observations in a sample on the ordinary Kernel PCA components. Some examples in string-analysis and chemometrics show the relevance of this method.

Challenges for future work are omnipresent. One could wonder whether it is possible to kernelize results from the first Chapter. Takeuchi et al. [2006] already introduced kernel based quantile regression. Possibly an extension dealing with right-censored observations along the lines of Chapter 1 can be obtained as well.

In Chapter 2 we obtained expressions for the influence function of robust covariance and PCA estimators. It would be nice to make use of these formulas. To select the number of principal components to be retained, a leave-one-out criterion is often used. The influence function could provide a fast approximation. Some preliminary results show that this is true to a certain extent, but that this approximation is not always very reliable in high dimensions. Since PCA is obviously most useful specifically for such high-dimensional data, this approach needs some extra investigation.

In Chapters 3 and 4 some properties of reweighted Kernel Based Regression were examined. An obvious question is whether similar results can be obtained in other settings. Reweighting could be useful in classification or PCA as well. It would be interesting to investigate what types of weight functions are to be preferred in these situations. Furthermore the main focus was put on bounding the influence function, essentially preventing the effects of infinitely small amounts of contamination. Of course, other robustness criteria exist as well. The breakdown value for example measures how much contamination a method can resist. In the ROBPCA method from Chapter 2 for instance, this breakdown value is explicitly controlled by the value of α. However, for kernel methods no results in this direction have been established so far.

In Chapter 5 we proposed Spherical KPCA. On a theoretical level, it would be interesting to calculate the influence function for this method as well. On the practical side, KPCA often is the first step in a data analysis. One could use the resulting scores for instance in clustering or classification. An analysis of the robustness in such combined situations could be potential research. An extension towards Kernel PLS regression is worth investigating too. Finally the spatial median based centering of the kernel matrix could be useful itself outside a PCA framework, for example in classification.

Bibliography

J. Adrover, R.A. Maronna, and V.J. Yohai. Relationships between maximum depth and projection regression estimates. *Journal of Statistical Planning and Inference*, 105:363–375, 2002.

J. Adrover, R.A. Maronna, and V.J. Yohai. Robust regression quantiles. *Journal of Statistical Planning and Inference*, 122:187–202, 2004.

C. Alzate and J.A.K Suykens. Extending kernel principal component analysis to general underlying loss functions. In *Proc. of the International Joint Conference on Neural Networks (IJCNN'05)*, pages 214–219, Montreal, Canada, 2005.

Z.D. Bai and X. He. Asymptotic distributions of the maximal depth estimators for regression and multivariate location. *The Annals of Statistics*, 27:1616–1637, 2000.

G. Blanchard, O. Bousquet, and L. Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 33:259–294, 2007.

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

R.W. Butler, P.L. Davies, and M. Jhun. Asymptotics for the Minimum Covariance Determinant estimator. *The Annals of Statistics*, 21:1385–1400, 1993.

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.

A. Christmann and I. Steinwart. Consistency and robustness of kernel based regression. *Bernoulli*, 2006. to appear.

R.D. Cook and S. Weisberg. *Residuals and influence in regression.* Chapman & Hall, New York, 1982.

C. Croux and G. Haesbroeck. Influence function and efficiency of the Minimum Covariance Determinant scatter matrix estimator. *Journal of Multivariate Analysis*, 71:161–190, 1999.

C. Croux and G. Haesbroeck. Principal components analysis based on robust estimators of the covariance or correlation matrix: influence functions and efficiencies. *Biometrika*, 87:603–618, 2000.

C. Croux and P.J. Rousseeuw. A class of high-breakdown scale estimators based on subranges. *Communications in Statistics-Theory and Methods*, 21:1935–1951, 1992.

C. Croux and A. Ruiz-Gazen. High breakdown estimators for principal components: the projection-pursuit approach revisited. *Journal of Multivariate Analysis*, 95:206–226, 2005.

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

S. de Jong. SIMPLS: an alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18:251–263, 1993.

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.

M. B. Dollinger and R. G. Staudte. Influence functions of iteratively reweighted least squares estimators. *Journal of the American Statistical Association*, 86:709–716, 1991.

D.L. Donoho. Breakdown properties of multivariate location estimators. Qualifying paper, Harvard University, Boston, 1982.

S. Engelen, M. Hubert, and K. Vanden Branden. A comparison of three procedures for robust PCA in high dimensions. *Austrian Journal of Statistics*, 34: 117–126, 2005.

K.H. Esbensen, S. Schönkopf, and T. Midtgaard. *Multivariate Analysis in Practice*. Camo, Trondheim, 1994.

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

J.H. Friedman. Multivariate adaptive regression splines. *Annals of Statistics*, 19: 1–14, 1991.

D. Gervini. The influence function of the Stahel–Donoho estimator of multivariate location and scatter. *Statistics and Probability Letters*, 60:425–435, 2002.

F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10:1455–1480, 1998.

J.C. Gower. The mediancentre. *Applied Statistics*, 23:466–470, 1974.

J.B.S. Haldane. Note on the median of a multivariate distribution. *Biometrika*, 35:414–415, 1948.

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.

X. He and F. Hu. Markov chain marginal bootstrap. *Journal of the American Statistical Association*, 97:783–795, 2002.

D. Hinkley. Jackknifing in unbalanced situations. *Technometrics*, 19:285–292, 1977.

B. Honoré, S. Khan, and J. Powell. Quantile regression under random censoring. *Journal of econometrics*, 109:67–105, 2002.

O. Hössjer and C. Croux. Generalizing univariate signed rank statistics for testing and estimating a multivariate location parameter. *Non-parametric statistics*, 4:293–308, 1995.

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

M. Hubert and S. Engelen. Robust PCA and classification in biosciences. *Bioinformatics*, 20:1728–1736, 2004.

M. Hubert and S. Engelen. Fast cross-validation for high-breakdown resampling algorithms for PCA. *Computational Statistics and Data Analysis*, 51:5013–5024, 2007.

M. Hubert, P.J. Rousseeuw, and K. Vanden Branden. ROBPCA: a new approach to robust principal components analysis. *Technometrics*, 47:64–79, 2005.

M. Hubert, P.J. Rousseeuw, and S. Verboven. A fast robust method for principal components with applications to chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 60:101–111, 2002.

M. Hubert and K. Vanden Branden. Robust methods for Partial Least Squares Regression. *Journal of Chemometrics*, 17:537–549, 2003.

M. Hubert and S. Verboven. A robust PCR method for high-dimensional regressors. *Journal of Chemometrics*, 17:438–452, 2003.

R. Koenker. *Quantile regression*. Cambridge University Press, Cambridge, 2005.

R. Koenker and G.W. Bassett. Regression quantiles. *Econometrica*, 46:33–50, 1978.

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.

N. Locantore, J.S. Marron, D.G. Simpson, N. Tripoli, J.T. Zhang, and K.L. Cohen. Robust principal component analysis for functional data. *Test*, 8:1–73, 1999.

J.I. Marden. Some robust estimates of principal components. *Statistics and Probability Letters*, 43:349–359, 1999.

R.A. Maronna. Principal components and orthogonal regression based on robust scales. *Technometrics*, 47:264–273, 2005.

R.A. Maronna and V.J. Yohai. The behavior of the Stahel-Donoho robust multivariate estimator. *Journal of the American Statistical Association*, 90:330–341, 1995.

T. Neocleous, K. Vanden Branden, and S. Portnoy. Correction to censored regression quantiles. *Journal of the American Statistical Association*, 101:860–861, 2006.

H.S. Oh, D. Nychka, T. Brown, and P. Charbonneau. Period analysis of variable stars by robust smoothing. *Journal of the Royal Statistical Society C (Applied Statistics)*, 53:15–30, 2004.

J. Park and J. Hwang. Regression depth with censored and truncated data. *Communications in Statistics: Theory and Methods*, 32:997–1008, 2003.

R. Phelps. *Convex functions, monotone operators and differentiability, volume 1364 of Lecture notes in math.* Springer, 1986.

G. Pison and S. Van Aelst. Diagnostic plots for robust multivariate methods. *Journal of Computational and Graphical Statistics*, 13:310–329, 2004.

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

S. Portnoy. Censored regression quantiles. *Journal of the American Statistical Association*, 98:1001–1012, 2003.

B. Rambali, S. Van Aelst, L. Baert, and D.L. Massart. Using deepest regression method for optimization of fluidized bed granulation on semi-full scale. *International Journal of Pharmaceutics*, 258:85–94, 2003.

P.J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79:871–880, 1984.

P.J. Rousseeuw and C. Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88:1273–1283, 1993.

P.J. Rousseeuw and M. Hubert. Regression depth. *Journal of the American Statistical Association*, 94:388–402, 1999.

P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. Wiley-Interscience, New York, 1987.

P.J. Rousseeuw and K. Van Driessen. A fast algorithm for the Minimum Covariance Determinant estimator. *Technometrics*, 41:212–223, 1999.

P.J. Rousseeuw and V.J. Yohai. Robust regression by means of S-estimators. In J. Franke, W. Härdle, and R.D. Martin, editors, *Robust and Nonlinear Time Series Analysis*, pages 256–272, New York, 1984. Lecture Notes in Statistics No. 26, Springer-Verlag.

M. Salibian-Barrera and R.H. Zamar. Bootstrapping robust estimates of regression. *The Annals of Statistics*, 30:556–582, 2002.

B. Schölkopf, R. Herbrich, and A. Smola. A generalized representer theorem. In D. Helmblod and B. Williamson, editors, *Neural Networks and Computational Learning Theory*, pages 416–426, Berlin, 2001. Springer.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

B. Schölkopf, A. Smola, and K-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, Cambridge, 2004.

J. Shawe-Taylor, C. Williams, N. Cristianini, and J. Kandola. Eigenspectrum of the gram matrix and its relationship to the operator eigenspectrum. In *Algorithmic learning theory: 13th international conference, ALT2002 of lecture notes in computer science*, volume 2533, pages 23–40. Springer-Verlag, 2002.

D.G. Simpson, D. Ruppert, and R.J. Carroll. On one-step GM-estimates and stability of inferences in linear regression. *Journal of the American Statistical Association*, 87:439–450, 1992.

D. Singh, P.G. Febbo, K. Ross, D.G. Jackson, J. Manola, C. Ladd, P. Tamayo, A.A. Renshaw, A.V. D'Amico, J.P. Richie, E.S. Lander, M. Loda, P.W. Kantoff, T.R. Golub, and W.R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1:203–209, 2002.

C.G. Small. A survey of multidimensional medians. *International Statistical Review*, 58:263–277, 1990.

W.A. Stahel. *Robuste Schätzungen: infinitesimale Optimalität und Schätzungen von Kovarianzmatrizen*. PhD thesis, ETH Zürich, 1981.

I. Steinwart. Sparseness of support vector machines. *Journal of Machine Learning Research*, 4:1071–1105, 2003.

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.

I. Takeuchi, V.Q. Le, T.D. Sears, and A.J. Smola. Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7:1231–1264, 2006.

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

S. Van Aelst and P.J. Rousseeuw. Robustness of deepest regression. *Journal of Multivariate Analysis*, 73:82–106, 2000.

S. Van Aelst, P.J. Rousseeuw, M. Hubert, and A. Struyf. The deepest regression method. *Journal of Multivariate Analysis*, 81:138–166, 2002.

K. Vanden Branden and M. Hubert. Robustness properties of a robust PLS regression method. *Analytica Chimica Acta*, 515:229–241, 2004.

K. Vanden Branden and M. Hubert. Robust classification in high dimensions based on the SIMCA method. *Chemometrics and Intelligent Laboratory Systems*, 79:10–21, 2005.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New-York, 1995.

S. Verboven and M. Hubert. LIBRA: a Matlab library for robust analysis. *Chemometrics and Intelligent Laboratory Systems*, 75:127–136, 2005.

G. Wahba. Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 69–88, Cambridge, MA, 1999. MIT Press.

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

V.J. Yohai. High breakdown point and high efficiency robust estimates for regression. *The Annals of Statistics*, 15:642–656, 1987.

V.J. Yohai and R.H. Zamar. High breakdown point estimates of regression by means of the minimization of an efficient scale. *Journal of the American Statistical Association*, 83:406–413, 1988.

Y. Zuo and H. Cui. Depth weighted scatter estimators. *The Annals of Statistics*, 33:381–413, 2005.

Y. Zuo, H. Cui, and X. He. On the Stahel-Donoho estimator and depth-weighted means of multivariate data. *The Annals of Statistics*, 32:167–188, 2004.

Nederlandse samenvatting

Klassieke statistische methodes steunen vaak op aannames die in de praktijk niet altijd correct zijn. Zo wordt regelmatig verondersteld dat de gegevens een welbepaalde verdeling volgen. Zelfs indien deze onderliggende verdeling niet gespecifieerd wordt, gaat men meestal uit van een steekproef van onderling onafhankelijke en identiek verdeelde observaties. Zulke methodes kunnen echter zeer matig presteren indien uitschieters aanwezig zijn in de data set. Robuuste statistiek bestudeert methodes die niet willekeurig beïnvloed kunnen worden door uitschieters. Het doel is om de structuur te leren van de meerderheid van de data punten, zelfs als een minderheid dit patroon verstoort.

Deze thesis bestudeert robuustheid in twee verschillende contexten: regressie en Principaal Component Analyse (PCA). Regressie analyse modelleert het verband tussen een respons variabele en een aantal onafhankelijke variabelen (ook wel covariaten genoemd). Doorgaans is men geïnteresseerd in de voorwaardelijke verdeling van de respons gegeven een waarde voor de covariaten. Men kan zich concentreren op bepaalde kenmerken van deze voorwaardelijke verdeling, bvb. het gemiddelde wat leidt tot kleinste kwadraten regressie.

In sommige toepassingen is een meer gedetailleerd beeld wenselijk. Kwantielregressie [Koenker, 2005] schat alle voorwaardelijke kwantielen, waardoor de voorwaardelijke verdeling volledig gekarakteriseerd wordt. Uitgaande van een lineair model kan men kwantielregressie uitvoeren met behulp van een L_1 kostfunctie [Koenker and Bassett, 1978]. Hoewel dat minder uitschietergevoelig is dan kleinste kwadraten regressie, kunnen robuustheidsproblemen nog steeds voorkomen. Een robuustere aanpak met de naam ‘deepest regression’ wordt beschreven door Rousseeuw and Hubert [1999]. In Hoofdstuk 1 hernemen we kort beide methodes met hun relevante eigenschappen. Vervolgens beschouwen we het geval waar de steekproef rechts-gecensureerde observaties bevat. Dit betekent dat de respons waarde niet exact gemeten werd voor elke observatie, maar enkel een ondergrens. Dergelijke data komen frequent voor in geneeskunde, bijvoorbeeld wanneer een patiënt de deelname aan een studie vroegtijdig beëindigt, vooraleer de eindresultaten gemeten kunnen worden. Een methode om met zo'n data kwantielregressie te kunnen uitvoeren, werd voorgesteld door Portnoy [2003] voor de L_1 schatter. In Hoofdstuk 1 passen we een gelijkaardig idee toe op de deepest regression methode. We leiden het bijhorend optimisatiecriterium af en stellen een grid-algoritme voor om de berekeningen uit te voeren. Robuustheid wordt aangetoond in een kleine simulatiestudie en in twee toepassingen.

Het tweede onderwerp van deze thesis betreft Principaal Component Analyse (PCA). Deze techniek reduceert de dimensie van multivariate data. Traditionele lineaire PCA bekomt zo'n reductie door projectie op een lager dimensionale lineaire deelruimte, opgespannen door de eigenvectoren van de klassieke covariantiematrix. Deze eigenvectoren zijn echter zeer gevoelig aan de aanwezigheid van uitschieters. Een meer robuuste methode met de naam ROBPCA werd voorgesteld door Hubert et al. [2005]. In Hoofdstuk 2 bestuderen we enkele theoretische eigenschappen van de onderliggende covariantieschatter. We berekenen de asymptotische efficiëntie en vergelijken met een aantal andere robuuste schatters. Bovendien berekenen we de invloedsfunctie waardoor we de effecten van uitschieters wiskundig kunnen analyseren. We tonen aan dat de invloedsfunctie van ROBCPA begrensd is, wat een belangrijke eis is voor robuuste methodes.

Zowel hoofstukken 1 and 2 gaan uit van een lineaire onderliggende structuur. In de praktijk komen ook meer gecompliceerde situaties voor. Hoofdstukken 3, 4 and 5 passen in het onderzoeksgebied rond kernel methodes. Algemeen geformuleerd passen kernel methodes lineaire methodes toe in een zogenaamde feature ruimte $\mathcal{H}$, in plaats van in de originele ruimte. Indien deze methode in $\mathcal{H}$ enkel geformuleerd kan worden in termen van scalaire producten $\langle \Phi(u), \Phi(v) \rangle$, kan men een kernel functie K gebruiken en $K(u, v)$ evalueren. Dit laat data analyse toe in een mogelijk zeer hoog dimensionale feature ruimte $\mathcal{H}$, zonder de expliciete features te berekenen of zelfs te kennen.

Dergelijke kernels in een regressie context, samen met ideeën uit optimisatie en regularisatie, vormen de kern tot Kernel Based Regression (KBR). Christmann and Steinwart [2006] bewezen dat de invloedsfunctie van Least Squares KBR (LS-KBR) onbegrensd is, in tegenstelling tot KBR met kostfuncties met begrensde eerste afgeleide. Suykens et al. [2002a] stelden een herwogen versie van LS-KBR voor ter verbetering van de robuustheid van LS-KBR, met behoud van de kleinste kwadraten methodologie en bijhorende voordelen. In hoofdstuk 3 onderzoeken we een aantal eigenschappen van deze herwogen LS-KBR methode. We berekenen de invloedsfunctie van de $k-$stap gewogen schatter. Onder enkele voorwaarden analyseren we het gedrag van de rij invloedsfuncties als men iteratief blijft herwegen ($k \rightarrow \infty$). Een belangrijk resultaat stelt dat de invloedsfunctie van iteratief herwogen LS-KBR begrensd is indien de kernel begrensd is en indien de gewichtsfunctie van de vorm is $w(r) = \psi(r)/r$ met r het residu en ψ een begrensde en stijgende functie. Deze voorwaarde is niet triviaal. Hampel gewichten bijvoorbeeld voldoen niet.

We besluiten hoofdstuk 3 door de invloedsfunctie te linken aan concepten van stabiliteit Poggio et al. [2004]. Op deze manier wordt beargumenteerd dat herweging niet alleen nuttig is om uitschieters te neutraliseren, maar ook in het geval van zwaarstaartige ruis. In hoofdstuk 4 bestuderen we de invloedsfunctie verder. Een modelselectiecriterium wordt voorgesteld waarmee goede waardes kunnen bekomen worden voor de modelparameters, zoals de regularisatieparameter en de bandbreedte in het geval van een RBF-kernel.

In hoofdstuk 5 keren we terug naar PCA analyse. Ook hier kunnen kernels geïncorporeerd worden om complexere structuren te ontdekken. De invloedsfunctie van Kernel PCA Schölkopf et al. [1998] wordt afgeleid. Net als in het regressiegeval leidt een begrensde kernel tot een begrensde invloedsfunctie. Met een onbegrensde kernel kunnen uitschieters echter een willekeurig grote invloed uitoefenen. Een nieuwe techniek wordt voorgesteld met de naam Spherical KPCA, waarmee robuustere scores kunnen bekomen worden. Tenslotte construeren we een diagnostische plot waar de invloed van punten in een steekproef gevisualiseerd wordt, zodat invloedrijke punten gedetecteerd kunnen worden.