Regression depth and support vector machine

Andreas Christmann

Abstract. The regression depth method (RDM) proposed by Rousseeuw and Hubert [**RH99**] plays an important role in the area of robust regression for a continuous response variable. Christmann and Rousseeuw [**CR01**] showed that RDM is also useful for the case of binary regression. Vapnik's convex risk minimization principle [**Vap98**] has a dominating role in statistical machine learning theory. Important special cases are the support vector machine (SVM), $\varepsilon-$support vector regression and kernel logistic regression.
In this paper connections between these methods from different disciplines are investigated for the case of pattern recognition. Some results concerning the robustness of the SVM and other kernel based methods are given.

1. Introduction

Binary regression and statistical machine learning play a key role in theoretical and applied statistics. In supervised learning we have a set of variables, say X (the predictors, the explanatory variables, or the inputs) which might have an influence on one or on several response variables, say Y (the dependent variables or the outputs). Then we are mainly interested in the conditional distribution of Y given X. An example is the prediction of claim sizes and of the probability for a claim in the context of motor vehicle insurance companies, *cf.* [**Chr04**]. In contrast to that, in unsupervised learning the distinction between inputs and outputs can not be made in advance such that the joint distribution of all variables is of main interest. The number of variables can sometimes be extremely large in unsupervised statistical learning problems.

In this paper supervised statistical learning will be considered, where the single response variable is discrete. The minimum number of misclassifications achievable with affine hyperplanes on a given set of labeled points is of special importance. The problem to determine this quantity exactly is NP-hard, see [**HSvH95**]. Hence, there is a need to find reasonable and fast approximation procedures. One approach to approximate the minimum number of misclassifications achievable with affine

1991 *Mathematics Subject Classification.* Primary 62G35, 62H30; Secondary 62G05, 47N30.
Key words and phrases. Pattern recognition, kernel logistic regression, regression depth, robustness, statistical machine learning, support vector machine.
This work was partially supported by the Deutsche Forschungsgemeinschaft (SFB 475, "Reduction of complexity in multivariate data structures") and by the Forschungsband DoMuS from the University of Dortmund.

hyperplanes was proposed by Christmann and Rousseeuw [**CR01**]. The approach is based on the regression depth method proposed by Rousseeuw and Hubert [**RH99**].

However, sometimes it is not sufficient to allow only affine hyperplanes for separating two response groups. Therefore, we also treat the support vector machine proposed by Vapnik [**Vap98**]. The SVM can be used with a linear kernel, but it can also be used in combination with universal kernels as the Gaussian RBF kernel which allows more complex structures to separate both response groups. The SVM is one reference method well-known to be effective to fit complex and high dimensional data sets.

The rest of the paper is organized as follows. Section 2 gives the notions of complete separation, quasicomplete separation and overlap from [**AA84**] and [**SD86**]. Section 3 shows that the regression depth approach is useful for binary regression models to check whether the maximum likelihood estimate for the parameter vector exists and how many data points are necessary to guarantee the existence. A connection between regression depth and overlap is shown and some algorithmic considerations are given. Section 4 briefly describes Vapnik's convex risk minimization approach with special emphasis on the support vector machine. Section 5 describes the results of numerical comparisons between SVMs with a linear kernel and RDM. An example is given in Section 6 which shows that a SVM in combination with the classical Gaussian RBF kernel can show an unstable behaviour with respect to training errors and test errors in certain multi-class classification problems. Section 7 contains a brief summary of recent results concerning robustness properties of certain statistical machine learning methods based on kernels for the case of pattern recognition. Kernel logistic regression and the SVM are special cases. Section 8 gives some numerical results about prediction aspects of SVMs. Section 9 contains a discussion.

2. Separation and overlap

Generalized linear models are among the most popular approaches to model the occurrence of an event depending on a vector of explanatory variables, say $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,p-1}) \in \mathbb{R}^{p-1}$. We will assume that there is also an intercept term. Examples are the identification of risk factors for cancer in medical applications, estimating the probability of an insurance claim, the purchase of a product by a customer in direct marketing, or the probability that the price of a stock exceeds a previously defined value within one month taking into account general economic data and indices measuring company performances. The responses y_i are commonly assumed to be realizations of independent Bernoulli random variables Y_i. Consider a given set of observations $Z_n = \{(x_{i,1}, \ldots, x_{i,p-1}, y_i);\ i = 1, \ldots, n\} \subset \mathbb{R}^p$, where $y_i \in \{0, 1\}$ for $i = 1, \ldots, n$. The goal is to find an affine hyperplane defined via $\theta \in \mathbb{R}^p$ such that a good classification of the responses is possible.

For the given data set Z_n, define n_{complete} as the minimum number of misclassifications that any affine hyperplane must incur. In particular, if $n_{\text{complete}} = 0$, the data set is *completely separated* so that there exists a vector $\theta \in \mathbb{R}^p$ such that

$$(\mathbf{x}_i, 1)\theta' > 0 \quad \text{if} \quad y_i = 1 \tag{2.1}$$

$$(\mathbf{x}_i, 1)\theta' < 0 \quad \text{if} \quad y_i = 0 \tag{2.2}$$

for $i = 1, \ldots, n$. A data set which is not completely separated is *quasicompletely separated* if there exists a vector $\theta \in \mathbb{R}^p \backslash \{0\}$ such that

$$(\mathbf{x}_i, 1)\theta' \geq 0 \quad \text{if} \quad y_i = 1 \tag{2.3}$$

$$(\mathbf{x}_i, 1)\theta' \leq 0 \quad \text{if} \quad y_i = 0 \tag{2.4}$$

for all i and if there exists $j \in \{1, ..., n\}$ such that $(\mathbf{x}_j, 1)\theta' = 0$. A data set is said to have *overlap* if there is no complete separation and no quasicomplete separation. The quantity n_{complete} denotes the smallest number of observations whose removal yields complete separation. The quantity n_{overlap} denotes the smallest number of observations whose removal yields complete or quasicomplete separation. For logistic regression with an intercept term, it is well-known that the classical maximum likelihood estimate of θ does not exist if $n_{\text{overlap}} = 0$, see [**AA84**] and [**SD86**].

The opposite holds true when training a single linear threshold function using the Perceptron [**Ros62**] algorithm, which is guaranteed to converge only for data sets with $n_{\text{complete}} = 0$, *cf.* [**Nov62**]. The quantity n_{complete} is a parameter in bounds on the prediction error if one measures the quality of linear models according to the empirical risk minimization principle, see [**Vap98**]. Unfortunately, the problem of determining the exact minimum number of misclassifications n_{complete} based on an affine hyperplane for arbitrary dimensions is NP-hard.

THEOREM 2.1. [**HSvH95**, Theorem 3.1] *Let n disjoint points from $\mathbb{R}^p$, each labelled with response 0 or 1, and a bound $k \geq 1$ be given. The problem to decide whether there is an affine hyperplane such that $n_{\text{complete}} \leq k$ is NP-complete. The problem remains NP-complete if the points are only allowed to have integer coordinates.*

The next section gives the definition of regression depth proposed by [**RH99**] and shows a relationship to the notion of overlap in binary regression models.

3. Regression depth

Rousseeuw and Hubert [**RH99**] introduced the regression depth approach for linear regression models. In the following we will consider the logistic regression model, although the method can be used for other binary regression models in an analogous manner. Data sets analyzed with such models have the form $Z_n = \{(x_{i,1}, \ldots, x_{i,p-1}, y_i);\ i = 1, \ldots, n\} \subset \mathbb{R}^p$ where $y_i \in \{0, 1\}$ for $i = 1, \ldots, n$. For simplicity, we will assume that the design matrix has full column rank. Denote the cumulative distribution function of the logistic distribution by $\Lambda(z) = 1/[1 + e^{-z}]$, $z \in \mathbb{R}$.

DEFINITION 3.1. A vector $\theta = (\theta_1, \ldots, \theta_p) \in \mathbb{R}^p$ is called a **nonfit** to Z_n iff there exists an affine hyperplane V in $\mathbf{x}-$space such that no $\mathbf{x}_i$ belongs to V, and such that the residual $r_i(\theta) = y_i - \Lambda((\mathbf{x}_i, 1)\theta') > 0$ for all $\mathbf{x}_i$ in one of its open halfspaces, and $r_i(\theta) < 0$ for all $\mathbf{x}_i$ in the other open halfspace.

DEFINITION 3.2. The **regression depth** of a fit $\theta = (\theta_1, \ldots, \theta_p) \in \mathbb{R}^p$ relative to a data set $Z_n \subset \mathbb{R}^p$ is the smallest number of observations that need to be removed to make θ a nonfit in the sense of Definition 3.1. Equivalently, $\text{rdepth}(\theta, Z_n)$ is the smallest number of residuals that need to change sign.

From Definition 3.2 it follows for logistic models that the regression depth of a fit θ relative to Z_n is equal to the regression depth of $-\theta$ relative to the data

set $\{(x_{i,1}, \ldots, x_{i,p-1}, 1-y_i);\, i = 1, \ldots, n\}$. Hence, the regression depth is invariant with respect to different codings of the binary response variable.

There exists an interesting connection between regression depth and complete separation. Define the horizontal hyperplane defined by $\theta^* = (0, \ldots, 0, 0.5)$. Then θ^* is a nonfit iff $n_{\text{complete}} = 0$, and more generally $n_{\text{complete}} = \text{rdepth}(\theta^*, Z_n)$. This implies that n_{complete} can be computed with an algorithm for the regression depth of a given hyperplane, *cf.* Christmann and Rousseeuw [**CR01**]. For $p \in \{2, 3, 4\}$ the latter can be computed by the $O(n^{p-1} \log(n))$ time algorithms of Rousseeuw and Hubert [**RH99**] and Rousseeuw and Struyf [**RS98**]. For $p \geq 3$, [**RS98**] constructed a fast approximation algorithm based on appropriate projections for the regression depth. The main idea of the algorithm for $p \geq 3$ is to approximate the $p-$dimensional regression depth by the minimum of certain two-dimensional regression depths. We use

$$n_{\text{complete}} = \text{rdepth}(\theta_{opt}, Z_n) \tag{3.1}$$

$$= \min_{\theta \in \mathbb{R}^p} \text{rdepth}(\theta, \tilde{Z}_n(\theta)) \tag{3.2}$$

$$\leq \min_{\theta \in B \subset \mathbb{R}^p} \text{rdepth}(\theta, \tilde{Z}_n(\theta)) =: n_{\text{complete}}(B)\,, \tag{3.3}$$

where

$$\tilde{Z}_n(\theta) = \{((x_i, 1)\theta', y_i); i = 1, \ldots, n\} \subset \mathbb{R}^2,\ \theta \in \mathbb{R}^p, \tag{3.4}$$

and θ_{opt} is an optimal parameter vector.

The set B is determined via projections defined by a large number, say 10^4, of random subsamples of the original data set. In a similar manner one can also approximate n_{overlap} by $n_{\text{overlap}}(B)$. Details of the algorithms are described in Christmann and Rousseeuw [**CR01**]. Software to compute $n_{\text{overlap}}(B)$ and $n_{\text{complete}}(B)$ written in R (packages `noverlap` and `ncomplete` from the website `http://cran.r-project.org/`) and in FORTRAN is available.

Of course, this approximation algorithm to compute $n_{\text{complete}}(B)$ is computer intensive, if the dimensions n or p or the number of samples to be drawn, i.e. $|B|$, are high. Further, drawing random subsamples of the original data set often result in affine hyperplanes for which the number of misclassifications is much higher than for the desired affine hyperplane.

Other determinations of the set B in (3.3) were investigated by [**CFJ02**]. A naive alternative to $n_{\text{complete}}(B)$ is to use only one *special* vector b in (3.3). Define

$$b = \begin{cases} \hat{\theta}_{ML} & \text{if } \hat{\theta}_{ML} \text{ exists} \\ \hat{\theta}^{(k)} & \text{otherwise,} \end{cases}$$

where $\hat{\theta}^{(k)}$ is the last vector computed by the usual Fisher-scoring algorithm to compute the ML estimate in the logistic regression model after stopping due to detection that there is no overlap in the data set. We compute b by the SAS procedure PROC LOGISTIC. This SAS procedure gives a warning if the data set has complete separation or quasicomplete separation, but stores $\hat{\theta}^{(k)}$ and the linear combinations of $(x_i, 1)$ and $\hat{\theta}^{(k)}$. In the same manner, let $s(b)$ be the asymptotic standard error of the ML estimate, if it exists, or the corresponding quantity evaluated for $\hat{\theta}^{(k)}$. Of course, other programs to compute ML estimates in the logistic regression model can also be used. The naive method $n_{\text{complete}}(b)$ often gives surprisingly good approximations of n_{complete}. Nevertheless, it seems reasonable to find better

approximations of n_{complete} in an iterative manner as follows. The heuristic method $n_{\text{complete}}(h)$ first tries to find a good approximation of n_{complete} using the vector b as a starting vector. Secondly, a grid search is done where sequentially some of the components of b are set to zero. Then a local search again starting from b is performed in the following way. We vary the parameter b componentwise in discrete steps by a factor in the interval 0 to 3 taking into account the variability measured by the quantities $s(b)$. If an improvement occurs an additional refinement is made starting from the best solution got so far. Finally the outcomes of all three methods (starting value, grid search, local search) are compared and the best solution is chosen. Of course, if during the whole procedure the best possible value of $n_{\text{complete}} = 0$ is detected, the algorithms stops and outputs the current solution.

Rousseeuw and Christmann [**RC03**] proposed the hidden logistic regression model. This model is strongly related to the logistic regression model. The advantage of the hidden logistic regression model is that robust estimation in that model is possible and that it circumvents the problem of non-existence of the estimates.

In the following section a relationship between the regression depth approach and the support vector machine is shown.

4. Convex risk minimization and SVM

In modern statistical machine learning theory the convex risk minimization principle plays an important role. Vapnik [**Vap98**] proposed the support vector machine, which is a special case and can be described as follows for the case of pattern recognition. The responses are recoded as $-1/+1$ instead of $0/1$. The empirical regularized risk is defined by

$$\hat{f}_{n,\lambda} = \arg\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda\|f\|_{\mathcal{H}}^2, \tag{4.1}$$

where $\lambda > 0$ is a penalizing constant, $Y = \{-1, +1\}$, $L : Y \times \mathbb{R} \to \mathbb{R}$, is a convex loss function, $\mathcal{H}$ is a reproducing kernel Hilbert space with kernel k, and $f \in \mathcal{H}$ is the function we like to estimate. The term $\lambda\|f\|_{\mathcal{H}}^2$ decreases the generalization error and avoids over-fitting. The convexity of L yields algorithmic advantages and avoids computationally NP-hard problems. Popular loss functions depend on y and f via $v = yf(x)$ or $v = y(f(x) + b)$, where $b \in \mathbb{R}$ is an additional intercept term. The SVM uses the loss function $L(y, f(x) + b) = \max(1 - y[f(x) + b], 0)$ such that points are linearly penalized if $v := y[f(x) + b] < 1$. Other methods based on the convex risk minimization principle are kernel logistic regression, L2-SVM, modified L2-SVM, modied Huber, and AdaBoost, see [**Zha04**].

The optimization problem (4.1) can be interpreted as a stochastic approximation of the minimization of the theoretical regularized risk given in (4.2):

$$f_{\mathrm{P},\lambda} = \arg\min_{f\in\mathcal{H}} \mathrm{E}_{\mathrm{P}}\, L(Y, f(X)) + \lambda\|f\|_{\mathcal{H}}^2\,. \tag{4.2}$$

We denote by $(\hat{f}_{n,\lambda}, \hat{b}_{n,\lambda})$ and $(f_{\mathrm{P},\lambda}, b_{\mathrm{P},\lambda})$ the corresponding quantities if we are modelling $f + b$ instead of f, where $b \in \mathbb{R}$ denotes in intercept term.

Decompose θ in the slope part, say $\mathbf{w} = (\theta_1, \ldots, \theta_{p-1})$, and the intercept part θ_p. To describe the connections between the support vector machine and the regression depth method, let us consider a two-dimensional data set, *cf.* Figure 1. If the data point marked as $*$ is equal to -1, then the solid line gives a complete separation of the response groups. The region specified by the dotted border lines

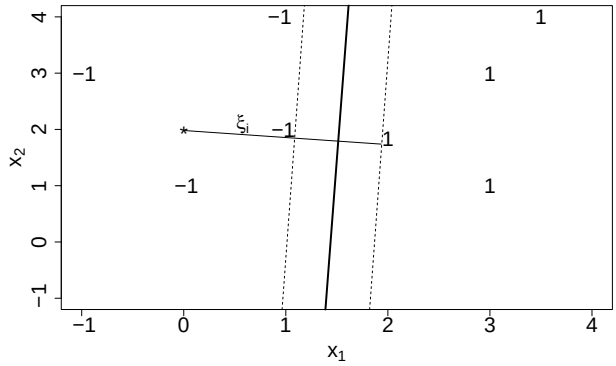

FIGURE 1. Illustration of the support vector machine.

is called the margin. It is implicitly defined via the data points on its boundary. These data points are called support vectors. The margin is defined as the maximum distance between parallel affine hyperplanes which separate both response groups.

However, if the data point marked as $*$ is equal to $+1$, no complete separation is possible by an affine hyperplane. The marked data point lies within the convex hull of the opposite class with a distance proportional to ξ_i minus the margin size.

The aim of the support vector machine is to maximize the width between all possible parallel affine hyperplanes which separate both response groups while penalizing misclassifications by a large positive extra cost C. Define $C = (2\lambda n)^{-1}$. Accordingly, the support vector machine solves the following quadratic optimization problem (with intercept term b):

$$\begin{array}{lll} \text{(P)} & \text{minimize (w.r.t. } \mathbf{w}, b, \xi\text{):} & \frac{1}{2}\,||\mathbf{w}||^2 + C\sum_i \xi_i \\ & \text{subject to} & \text{sign}(y_i)\cdot(x_i, 1)\theta' \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \quad . \end{array}$$

Apart from some degenerate cases, the solution of the optimization is unique due to the fact that the SVM uses a convex loss function. Here, $C > 0$ is a penalty parameter specified by the data analyst to model an extra cost for errors. The quantity ξ_i must exceed unity for a misclassification to occur. Hence, the sum over the slack parameters $\sum_i \xi_i$ is an upper bound on the number of training errors, *cf.* [**Bur98**]. Increasing C corresponds to a higher penalty to errors. In practice, one usually solves the following dual program

$$\begin{array}{lll} \text{(D)} & \text{minimize (w.r.t. } \alpha\text{):} & \frac{1}{2}\,\alpha' Q\alpha - \alpha'\mathbf{1} \\ & \text{subject to} & \alpha'\mathbf{y} = 0 \text{ and } \mathbf{0} \leq \alpha \leq C\mathbf{1} \quad , \end{array}$$

where $(Q)_{ij} = y_i y_j\, x_i' x_j$. Using the Karush-Kuhn-Tucker conditions of the dual program (D), the quantities $\mathbf{w}$, b, and ξ can be computed in the following way. The slope part of θ is given by

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i x_i \,. \tag{4.3}$$

If $0 < \alpha_i < C$ then $b = y_i - x_i^T \mathbf{w}$. While this value of b corresponds to the solution of the primal problem (P), b is commonly selected to directly minimize the number of training errors for the given $\mathbf{w}$ [**Bur98**]. This can easily be done after sorting all training points according to their projection on $\mathbf{w}$.

Of particular interest is the fact that the dual problem (D) depends only on inner products between vectors of explanatory variables. Substituting Mercer kernels for the simple dot product allows SVMs to efficiently estimate not only linear, but also e.g. polynomial functions [**BGV92**]. Of special importance is the Gaussian radial basis function (RBF) kernel

$$k(x, x') = \exp(-\gamma \|x - x'\|^2), \quad \gamma > 0,$$

which is a universal kernel on every compact subset of $\mathbb{R}^p$ in the sense of [**Ste01**].

If the number of observations n or the dimension p is large, solving the minimization problems (P) or (D) is computer-intensive. While some algorithms (e.g. PROC NLP or the IML function NLPQUA in SAS, Version 8) require storing the huge matrix $Q \in \mathbb{R}^{n \times n}$ or the whole matrix specifying the constraints, SVM^{light} [**Joa99**] is useful for solving the SVM optimization problem (D). SVM^{light} is designed to efficiently handle problems with large p (e.g. $30,000$) and large n (e.g. $100,000$). To avoid computing and storing the full Hessian Q of (D), the algorithm of SVM^{light} proceeds by decomposing the problem [**OFG97**]. Only a few variables ($q \approx 10$) are optimized at a time. Their selection is based on a steepest feasible descent strategy. To reduce zig-zagging behavior, the original selection criterion [**Joa99**] can be modified. The working set is updated like a queue, with only two new variables entering in each iteration. Using this decomposition, the algorithm solves only small quadratic programs in each step. This leads to small memory requirements. In particular, memory does not scale $O(n * n)$ like for algorithms requiring the full Hessian, but typically only by $O(n * s)$, where s is the number of support vectors. The number of support vectors is generally much lower than n. The PR-LOQO optimizer developed by Smola [**Smo98**] can be used to increase the numerical stability of SVM^{light}. In general, it is helpful to standardize all explanatory variables in advance.

5. Comparison of SVM and RDM

A fair numerical comparison between the regression depth method and the support vector machine for the case of pattern recognition by affine hyperplanes is not easy. One reason is that the results can depend on the algorithms, on the actual implementations of the algorithms for numerical reasons, and of course on the considered data sets. Further, both approaches offer different options such as the number of subsamples for RDM and the penalyzing constant C for SVM.

There are many software products for the support vector machine and related methods. A good overview is given on the website `www.kernel-machines.org`. Joachims [**Joa99**] showed that the computation time for the SVM depends heavily on the algorithm and its implementation. His software SVM^{light} is fast even for high dimensional data sets and was successfully applied for text mining. Schölkopf and Smola [**SS02**, p. 219] demonstrated that different SVM classifiers can yield very different values for the test errors, ranging from 2.6% to 8.9% for the same data set. Hastie, Tibshirani and Friedman [**HTF01**, p. 384f] compared the SVM using polynomial kernels with degrees d ranging from 1 to 10 with BRUTO and MARS for a simulated data set with 100 observations. It was shown that the test error of the SVM classifiers can depend not only on the kernel but also on the dimensionality of the problem.

Christmann, Fischer and Joachims [**CFJ02**] compared RDM, SVM, and linear discriminant analysis (LDA) for various benchmark data sets from robust logistic regression. SVM^{light} was used to compute the SVM. The main criterion was a low misclassification error w.r.t. to affine hyperplanes. The computation time was the secondary criterion. Summarizing, RDM gave better results for small to moderate sized data sets, say for data sets of dimension $p \leq 10$ and up to $1,000$ observations. LDA showed the worst performance of the three methods. The misclassification error based on LDA was even larger than the trivial upper bound for n_{complete} given by the minimum of the sum of the successes ($y_i = 1$) and the sum of the failures ($y_i = 0$). Especially the heuristic algorithm $n_{\text{complete}}(h)$ gave good results. It was somewhat surprising that the naive algorithm $n_{\text{complete}}(b)$ can outperform the algorithm $n_{\text{complete}}(B)$ in certain situations. For more complex data sets, the SVM usually performed well to approximate n_{complete} and was fast. The results for SVM depended on the penalyzing constant C. In general $C = 10^5/(p-1)$ gave better approximations than $C = 10^3/(p-1)$, but the computation time increased. However, SVM sometimes failed to detect that a data set had complete separation, whereas $n_{\text{complete}}(b)$ was able to detect such situations. This was true also for $n_{\text{complete}}(h)$, which is however more computer-intensive than $n_{\text{complete}}(b)$. There do not yet exist algorithms to use RDM for very high dimensional data sets.

6. SVM and QDA

In this section we show that the SVM can be unstable with respect to training errors and test errors in certain multi-class classification problems. For simplicity, let us consider a two-dimensional problem with up to four classes without noise, although this phenomenon can also occur for more complex data sets. The explanatory variables x_1 and x_2 are simulated independently from a uniform distribution on the interval $[-1, +1]$. The four possible response classes are constructed as follows, see Figure 2:

$y_i = 0$				(complement)
if	$x_1^2 + x_2^2 < 0.15$	then	$y_i = 1$	(ball)
if	$0.15 \leq x_1^2 + x_2^2 < 0.6$	then	$y_i = 2$	(ring)
if	$0.25 + x_1 - x_2 < -0.5$	then	$y_i = 3$	(triangle) .

We consider three situations: {ball, ring, complement}, {ball, triangle, complement}, and {ball, ring, triangle, complement}. For each situation 250 data sets each with $n = 1,000$ observations were generated. Each data set was splitted by random into a training data set with 700 observations and a test data set with 300 observations. An SVM with Gaussian RBF kernel and $C = 10^4$ and a quadratic discriminant analysis (QDA) were done for each training data set. The R library `e1071` [**LDH$^+$03**] was used to train the SVM. Then the corresponding training errors and test errors were computed, see Table 1. The SVM yields much better results than the QDA w.r.t. training errors and test errors for both classification problems with 3 response classes. Sometimes even the test errors based on SVM were zero, i.e. no misclassification happens, which did not happen for the QDA. However, for the classification problem with 4 response classes the SVM performed often *worse* than QDA and the predictions errors had a large variability, see Figure 3. The same figure shows that the SVM gave much smaller prediction errors than the QDA for *some* of the 250 data sets. The predicted responses based on the SVM for the test data sets often consisted of less than four groups, although both

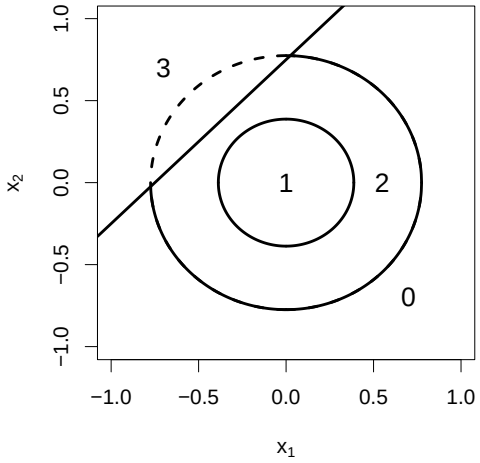

FIGURE 2. Illustration of simulated patterns.

TABLE 1. Averaged training and test errors for SVM and QDA.*

Classes	Training error		Test error	
	SVM	QDA	SVM	QDA
(0,1,2)	0.001 (0.001)	0.267 (0.030)	0.012 (0.007)	0.277 (0.047)
(0,1,3)	0.001 (0.001)	0.275 (0.021)	0.011 (0.007)	0.278 (0.031)
(0,1,2,3)	0.337 (0.225)	0.196 (0.020)	0.360 (0.231)	0.209 (0.034)

* Standard deviations are given in parenthesis.

the training data sets and the test data sets contained observations from all four response groups.

7. Robustness of the SVM

J.W. Tukey, one of the pioneers of robust statistics, already mentioned in 1960 [**HRRS86**, p. 21]:

> *A tacit hope in ignoring deviations from ideal models was that they would not matter; that statistical procedures which were optimal under the strict model would still be approximately optimal under the approximate model. Unfortunately, it turned out that this hope was often drastically wrong; even mild deviations often have much larger effects than were anticipated by most statisticians.*

Different criteria have been proposed to define the notion of robustness in a mathematical way, *e.g.* Huber's minimax approach [**Hub64**], Tukey's sensitivity curve [**Tuk77**], Hampels's approach based on influence functions [**Ham74, HRRS86**], the maxbias curve [**Hub64, HRRS86**], the finite sample breakdown point [**DH83**], and the approach based on least favourable local alternatives [**Rie94**].

In contrast to robust statistics for parametric models such as linear regression or multivariate location and scatter problems the robustness of non-parametric and nonlinear methods as the SVM with Gaussian RBF kernels have not yet been

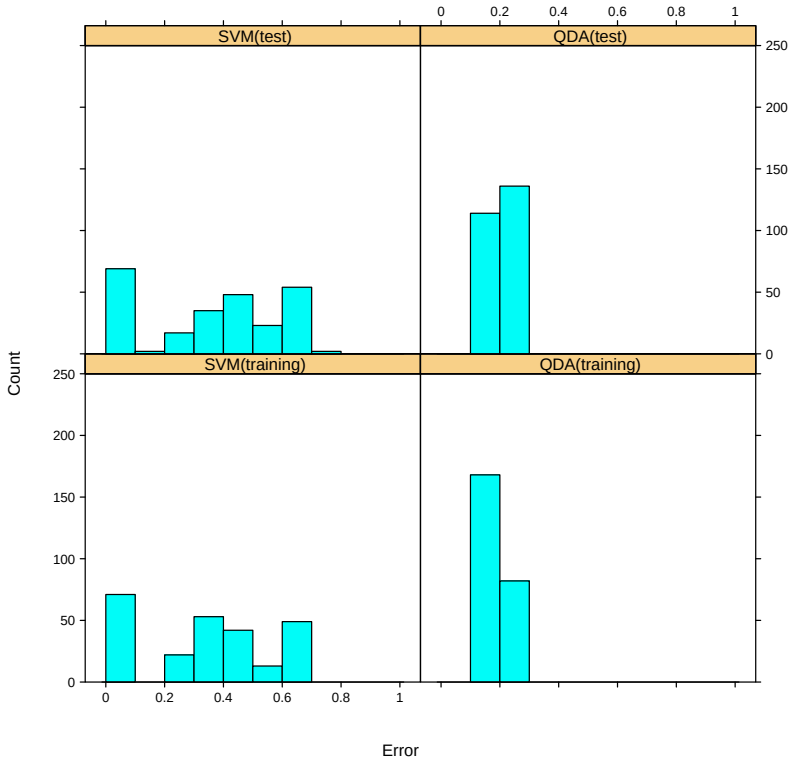

FIGURE 3. Training errors (below) and test errors (above) for SVM (left) and QDA (right) for the case of four response classes.

studied in great detail. One of the difficulties is that the quantity of interest is no longer a parameter vector, say $\theta \in \mathbb{R}^p$, but a function $f \in \mathcal{H}$ or a pair $(f, b) \in \mathcal{H} \times \mathbb{R}$, where $\mathcal{H}$ denotes the reproducing kernel Hilbert space.

For the case of pattern recognition Christmann and Steinwart [**CS04**] showed that some of the general robustness approaches can be successfully applied to a broad class of methods based on Vapnik's convex risk minimization principle. The following theorem gives the influence function of such methods. The Dirac distribution in the point z is denoted by Δ_z.

THEOREM 7.1. [**CS04**] *Let $L : Y \times \mathbb{R} \to [0, \infty)$ be a convex and twice continuously differentiable loss function, where $Y = \{-1, +1\}$. Furthermore, let $X \subset \mathbb{R}^n$ be a closed or open subset, $\mathcal{H}$ be a RKHS of a bounded continuous kernel on X, and P be a distribution on $X \times Y$. We define $G : \mathbb{R} \times \mathcal{H} \to \mathcal{H}$ by*

$$G(\varepsilon, f) := 2\lambda f + \mathrm{E}_{(1-\varepsilon)\mathrm{P}+\varepsilon\Delta_z} L'(Y, f(X))\Phi(X)$$

which implies

$$\frac{\partial G}{\partial \varepsilon}(0, f_{\mathrm{P},\lambda}) = -\mathrm{E}_{\mathrm{P}}[L'(Y, f_{\mathrm{P},\lambda}(X))\Phi(X)] + L'(z_y, f_{\mathrm{P},\lambda}(z_x))\Phi(z_x).$$

Define $S : \mathcal{H} \to \mathcal{H}$ *by*

$$S := \frac{\partial G}{\partial \mathcal{H}}(0, f_{\mathrm{P},\lambda}) = 2\lambda \,\mathrm{id}_{\mathcal{H}} + \mathrm{E}_{\mathrm{P}} L''(Y, f_{\mathrm{P},\lambda}(X))\langle \Phi(X), . \rangle \Phi(X)\,.$$

Then the influence function of the classifiers based on (4.2) exists for all $z = (z_x, z_y) \in X \times Y$ *and is given by*

$$IF(z; T, \mathrm{P}) = -S^{-1} \circ \frac{\partial G}{\partial \varepsilon}(0, f_{\mathrm{P},\lambda})\ . \tag{7.1}$$

REMARK 7.2. The influence function derived in Theorem 7.1 depends on the point $z = (z_x, z_y)$ only by the term $L'(z_y, f_{\mathrm{P},\lambda}(z_x))\Phi(z_x)$. Note the similarity of this term to the weighting scheme used by influence function of Mallows type M-estimators. In our case the weighting of z_x is of course performed in the feature space. This term can be bounded by choosing a loss function with a bounded derivative L' and a bounded and continuous kernel k. An example is kernel logistic regression with the Gaussian RBF kernel.

Summarizing the results given in [**CS04**] for pattern recognition, it turned out that the influence functions of $f_{\mathrm{P},\lambda}$ and $(f_{\mathrm{P},\lambda}, b_{\mathrm{P},\lambda})$ exist, and if bounded and continuous kernels are used in combination with appropriate loss functions, the influence function, the maxbias, and the sensitivity curve can be uniformly bounded.

Note that Theorem 7.1 is a result concerning the robustness of $f_{\mathrm{P},\lambda}$ but not for the prediction of Y, *i.e.* $\mathrm{sign}(f_{\mathrm{P},\lambda}(x))$. In the next section some preliminary numerical results are given for such predictions.

8. Prediction aspects of SVM

The goal of this section is to study the effect which a single data point can have on the prediction areas for the response y computed by the SVM.

We generated a data set with $n = 500$ data points x_i from a bivariate Student's t_3 distribution with location parameter $\mu = (0, 0)$ and scatter matrix Σ. The diagonal elements of Σ were set to 1, whereas the off-diagonal elements were set to 0.25. The responses y_i were generated from a logistic regression model with intercept for the parameter vector $\theta = (-1, 1)$ and $b = 1$, such that $P(Y_i = +1) = [1 + \exp(-[b + x_i'\theta])]^{-1}$ and $P(Y_i = -1) = 1 - P(Y_i = +1)$.

The upper part of Figure 4 shows that the choice of the penalizing constant C can be quite important for making predictions based on a support vector machine with a Gaussian RBF kernel. This holds true especially if predictions are made for a response with an $x-$value outside the bulk of the $x-$values of the training data set. Such observations are often called leverage points. In this sense, the SVM can produce unstable predictions and should be used with some care. In the lower part of Figure 4, corresponding prediction areas are given if two of the data points are moderate outliers. The SVM with a universal Gaussian RBF kernel accommodates outliers due to the consistency property. The corresponding SVM based prediction areas for a linear kernel are quite different, see Figure 5.

Similar sensitivity analyses were done for some other situations: increased sample size, situations where a complete separation of both response groups is possible, and for a multivariate normal distribution instead of a multivariate Student distribution to generate the $x-$values. The results are not given here because the results were qualitatively similar to those described in this section.

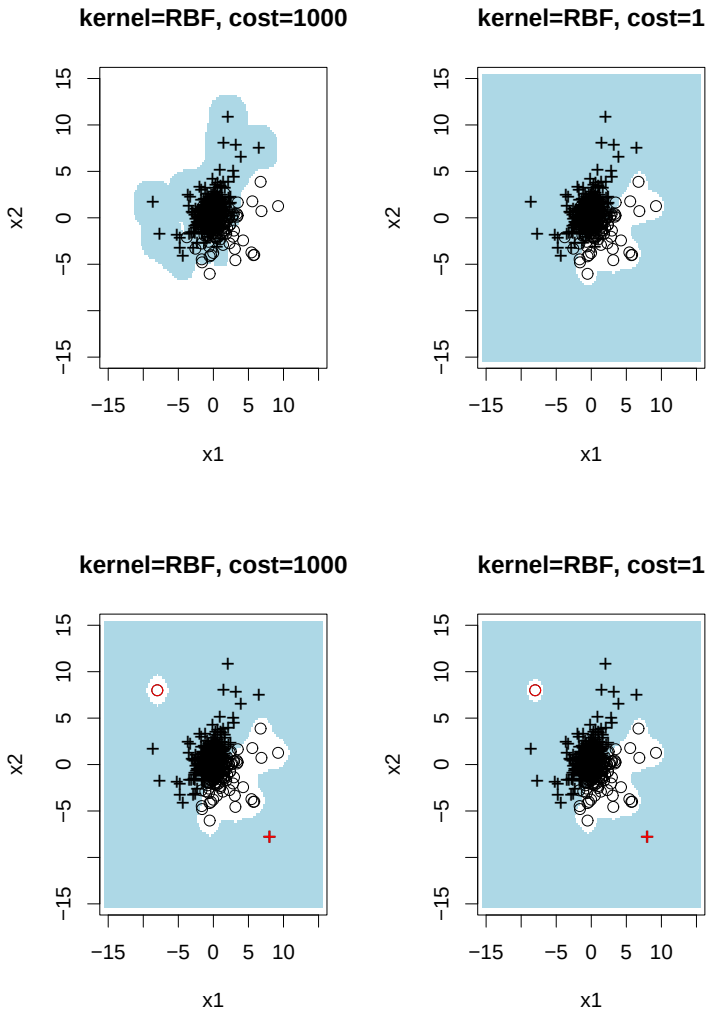

FIGURE 4. SVM based prediction areas using a Gaussian RBF kernel. Upper: simulated data set. Lower: simulated data set with two moderate outliers located at $x_A = (8, -8)$ with $y_A = 1$ and $x_B = (-8, 8)$ with $y_B = -1$. Legend: $+$ for $y = 1$, $\circ$ for $y = -1$. The prediction area for $\hat{y} = 1$ is shaded.

9. Discussion

In this paper pattern recognition problems were considered because they play an important role in many areas for applied statistics.

Firstly, the case was treated that the response groups should be separated by affine hyperplanes. Relationships between the regression depth method [**RH99**], the support vector machine with linear kernels, and the notions of overlap, complete

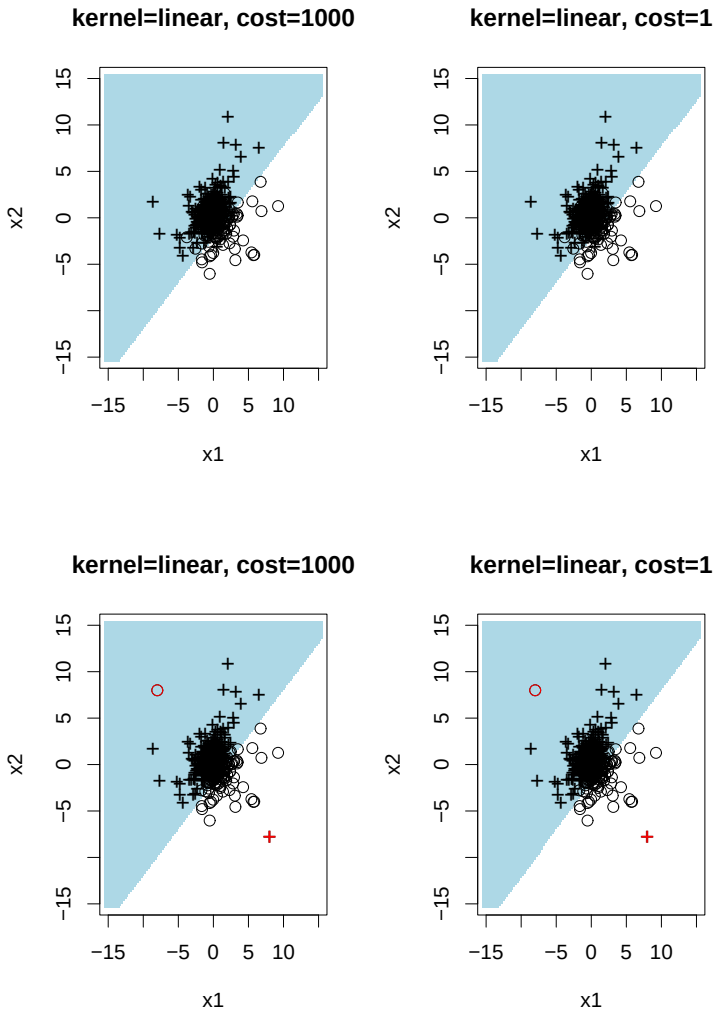

FIGURE 5. SVM based prediction areas using a linear kernel. Upper: simulated data set. Lower: simulated data set with two moderate outliers located at $x_A = (8, -8)$ with $y_A = 1$ and $x_B = (-8, 8)$ with $y_B = -1$. Legend: $+$ for $y = 1$, $\circ$ for $y = -1$. The prediction area for $\hat{y} = 1$ is shaded.

and quasicomplete separation [**AA84, SD86**] in the context of logistic regression were investigated.

We also considered the case that the response groups should be separated by more complex functions f. Therefore, we treated the support vector machine with the more flexible Gaussian RBF kernel. Robustness issues for the estimation of f were considered. Some numerical examples were also given for the case of prediction.

References

[AA84] A. Albert and J.A. Anderson, *On the existence of maximum likelihood estimates in logistic regression models*, Biometrica **71** (1984), 1–10.

[BGV92] B. E. Boser, I. M. Guyon, and V. N. Vapnik, *A training algorithm for optimal margin classifiers*, Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory (Pittsburgh, PA) (D. Haussler, ed.), ACM Press, July 1992, pp. 144–152.

[Bur98] C. J. C. Burges, *A tutorial on support vector machines for pattern recognition*, Data Mining and Knowledge Discovery **2** (1998), 121–167.

[CFJ02] A. Christmann, P. Fischer, and T. Joachims, *Comparison between various regression depth methods and the support vector machine to approximate the minimum number of misclassifications*, Computational Statistics **17** (2002), 273–287.

[Chr04] A. Christmann, *An approach to model complex high-dimensional insurance data*, Allgemeines Statistisches Archiv **4** (2004).

[CR01] A. Christmann and P.J. Rousseeuw, *Measuring overlap in logistic regression*, Computational Statistics and Data Analysis **37** (2001), 65–75.

[CS04] A. Christmann and I. Steinwart, *On robust properties of convex risk minimization methods for pattern recognition*, Journal of Machine Learning Research **5** (2004), 1007–1034.

[DH83] D.L. Donoho and P.J. Huber, *The notion of breakdown point*, A Festschrift for Erich L. Lehmann (Belmont, California, Wadsworth) (P.J. Bickel, K.A. Doksum, and J.L. Hodges Jr., eds.), 1983, pp. 157–184.

[Ham74] F.R. Hampel, *The influence curve and its role in robust estimation*, Journal of the American Statistical Association **69** (1974), 383–393.

[HRRS86] 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.

[HSvH95] K.U. Höffgen, H.-U. Simon, and K.S. van Horn, *Robust trainability of single neurons*, Journal Computer and System Sciences **50** (1995), 114–125.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*, Springer, New York, 2001.

[Hub64] P.J. Huber, *Robust estimation of a location parameter*, Annals of Mathematical Statistics **35** (1964), 73–101.

[Joa99] T. Joachims, *Making large-scale svm learning practical*, Advances in Kernel Methods - Support Vector Learning (MIT Press, Cambridge, Massachusetts) (B. Schölkopf, C. Burges, and A. Smola, eds.), 1999, pp. 41–56.

[LDH+03] F. Leisch, E. Dimitriadou, K. Hornik, D. Meyer, and A. Weingessel, *R package e1071*, 2003, http://cran.r-project.org.

[Nov62] A. Novikoff, *On convergence proofs on perceptrons*, Proceedings of the Symposium on the Mathematical Theory of Automata **XII** (1962), 615–622.

[OFG97] E. Osuna, R. Freund, and F. Girosi, *An improved training algorithm for support vector machines*, Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop (New York) (J. Principe, L. Gile, N. Morgan, and E. Wilson, eds.), IEEE, 1997, pp. 276–285.

[RC03] P.J. Rousseeuw and A. Christmann, *Robustness against separation and outliers in logistic regression*, Computational Statistics & Data Analysis **43** (2003), 315–332.

[RH99] P.J. Rousseeuw and M. Hubert, *Regression depth*, Journal of the American Statistical Association **94** (1999), 388–433.

[Rie94] H. Rieder, *Robust asymptotic statistics*, Springer, New York, 1994.

[Ros62] F. Rosenblatt, *Principles of neurodynamics*, Spartan, New York, 1962.

[RS98] P.J. Rousseeuw and A. Struyf, *Computing location depth and regression depth in higher dimensions*, Statistics and Computing **8** (1998), 193–203.

[SD86] T.J. Santner and D.E. Duffy, *A note on a. albert and j.a. anderson's conditions for the existence of maximum likelihood estimates in logistic regression models*, Biometrika **73** (1986), 755–758.

[Smo98] A. J. Smola, *Learning with kernels*, Ph.D. thesis, Technische Universität Berlin, 1998.

[SS02] B. Schölkopf and A.J. Smola, *Learning with kernels. support vector machines, regularization, optimization, and beyond*, MIT Press, Cambridge, Massachusetts, 2002.

[Ste01] I. Steinwart, *On the influence of the kernel on the consistency of support vector machines*, Journal of Machine Learning Research **2** (2001), 67–93.
[Tuk77] J.W. Tukey, *Exploratory data analysis*, Addison-Wesley, Reading, Massachusetts, 1977.
[Vap98] V.N. Vapnik, *Statistical learning theory*, Wiley, 1998.
[Zha04] T. Zhang, *Statistical behaviour and consistency of classification methods based on convex risk minimization*, Annals of Statistics **32** (2004), 56–134.

Department of Statistics, University of Dortmund, 44221 Dortmund, GERMANY
Current address: Department of Statistics, University of Dortmund, 44221 Dortmund, GERMANY
E-mail address: christmann@statistik.uni-dortmund.de