Comparison between the regression depth method and the support vector machine to approximate the minimum number of misclassifications[1]

Andreas Christmann[2], Paul Fischer[3], Thorsten Joachims[4]

[2] University of Dortmund, SFB-475, 44221 Dortmund, Germany.
[3] Informatik 2, University of Dortmund, Germany.
[4] GMD National Research Center for Information Technology, Institute for Autonomous Intelligent Systems, 53754 Sankt Augustin, Germany.

Summary

The minimum number of misclassifications achievable with affine hyperplanes on a given set of labeled points is a key quantity in both statistics and computational learning theory. However, determining this quantity exactly is essentially NP-hard, c.f. Höffgen, Simon and van Horn (1995). Hence, there is a need to find reasonable approximation procedures. This paper compares three approaches to approximating the minimum number of misclassifications achievable with affine hyperplanes. The first approach is based on the regression depth method of Rousseeuw and Hubert (1999) in linear regression models. We compare the results of the regression depth method with the support vector machine approach proposed by Vapnik (1998), and a heuristic search algorithm.

[1] The financial support of the Deutsche Forschungsgemeinschaft (SFB 475, "Reduction of complexity in multivariate data structures") is gratefully acknowledged.

Keywords: Linear discriminant analysis, Logistic regression, Overlap, Regression depth, Separation, Support vector machine.

1 Introduction

Both in statistics and machine learning, generalized linear models for binary responses are among the most popular approaches to modelling the occurrence of an event depending on a vector of explanatory variables, say $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,p-1}) \in \mathrm{I\!R}^{p-1}$. The events can be, for example, the occurrence of a special kind of illness in medical applications or the purchase of a product by a customer in direct marketing. The responses y_i are commonly assumed to be realisations of independent Bernoulli random variables Y_i. From a given set of observations $Z_n = \{(x_{i,1}, \ldots, x_{i,p-1}, y_i);\ i = 1, \ldots, n\} \subset \mathrm{I\!R}^p$, where $y_i \in \{0, 1\}$ for $i = 1, \ldots, n$, one aims to find an affine hyperplane defined via $\theta \in \mathrm{I\!R}^p$ such that a good classification of the responses is possible. A central role concerning the existence and the quality of such estimates plays the quantity n_{co} defined as follows. For the given data set Z_n, n_{co} is the minimum number of misclassifications that any affine hyperplane must incur. In particular, if $n_{\mathrm{co}} = 0$, the data set is *completely separated* so that there exists a vector $\theta \in \mathrm{I\!R}^p$ such that

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

$$(\mathbf{x}_i, 1)\theta' < 0 \quad \text{if} \quad y_i = 0 \tag{2}$$

for $i = 1, \ldots, n$. In other words, n_{co} denotes the smallest number of observations whose removal yields complete separation. What can n_{co} be used for? The following names a few:

- For logistic regression with an intercept term, it is well-known that the classical maximum likelihood estimate of θ does not exist for all data sets. Albert and Anderson (1984) and Santner and Duffy (1986) showed that the maximum likelihood estimate of θ does not exist, if the data is completely separable, i.e. $n_{\mathrm{co}} = 0$.
- The opposite holds true when training a single linear threshold function using the Perceptron (Rosenblatt, 1962) algorithm. The Perceptron algorithm is guaranteed to converge only for data sets with $n_{\mathrm{co}} = 0$ (Novikoff, 1962).
- And finally, when assessing the quality of linear models according to the empirical risk minimization principle (Vapnik, 1998), n_{co} is a parameter in bounds on the prediction error.

Unfortunately, the problem of determining the exact minimum number of misclassifications n_{co} based on an affine hyperplane for arbitrary dimensions is essentially NP-hard, c.f. Höffgen, Simon and van Horn (1995). Hence, there is a need to find reasonable approximation procedures for the minimum number of misclassifications. This paper compares the results of three methods to approximate the number of misclassifications based on an affine hyperplane. We investigate the recently proposed regression depth method (Rousseeuw and Hubert, 1999, and Christmann and Rousseeuw, 1999), the support vector machine (Vapnik, 1998), and a simple heuristic algorithm.

2 Regression depth

Rousseeuw and Hubert (1999) introduced the regression depth approach for linear regression models. Christmann and Rousseeuw (1999) showed that the regression depth method is useful for binary regression models, too. In the following we will consider the logistic 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 $Z_n = \{(x_{i,1}, \dots, x_{i,p-1}, y_i);\ i = 1, \dots, n\} \subset \mathbb{R}^p$ where $y_i \in \{0, 1\}$ for $i = 1, \dots, n$. For simplicity, we will assume that the design matrix has full column rank.

Definition 2.1 *A vector $\theta = (\theta_1, \dots, \theta_p) \in \mathbb{R}^p$ 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 2.2 *The* **regression depth** *of a fit $\theta = (\theta_1, \dots, \theta_p) \in \mathbb{R}^p$ 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 2.1. Equivalently, rdepth(θ, Z_n) is the smallest number of residuals that need to change sign.*

From Definition 2.2 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 $\{(x_{i,1}, \dots, x_{i,p-1}, 1-y_i);\ i = 1, \dots, n\}$. Hence, the regression depth is invariant with respect to different codings of the binary response variable. Let us illustrate the definition of the regression depth by an artificial data set with two explanatory variables x_1 and x_2 and an intercept term:

$$\mathbf{X} = \begin{pmatrix} -1, & 0, & 0, & 1, & 1, & 2\ , & 3, & 3, & 3.5 \\ 3, & 1, & 2, & 2, & 4, & 1.8, & 1, & 3, & 4 \end{pmatrix}', \tag{3}$$

$$\mathbf{y} = (0,\, 0,\, *,\, 0,\, 0,\, 1,\, 1,\, 1,\, 1)'. \tag{4}$$

If the data point y_2 denoted by $*$ in (4) is a failure, i.e. $y_2 = 0$, then the sets $\{y_i = 0; i = 1, \dots, n\}$ and $\{y_i = 1; i = 1, \dots, n\}$ can be separated by an appropriate affine hyperplane, which is indicated as a line in Figure

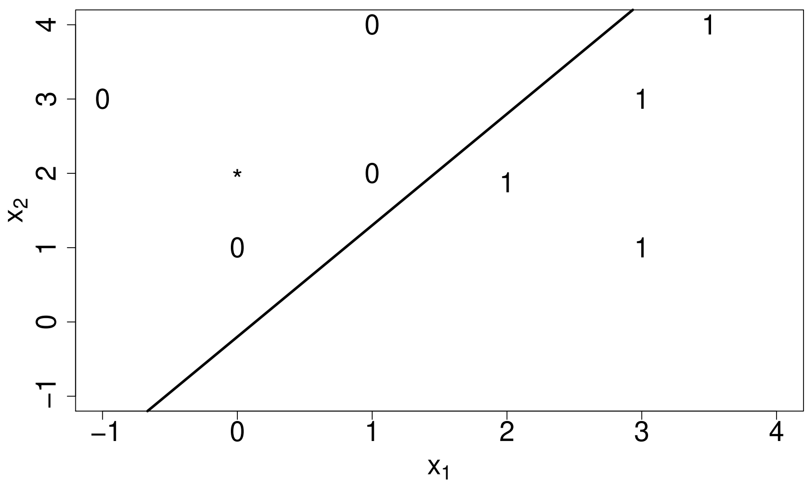

Figure 1: Illustration of complete separation.

1, and hence $n_{\rm co} = 0$. If complete separation is possible, there may exist several affine hyperplanes separating both response groups. The maximum likelihood estimate of θ does not exist in a logistic regression model in that case, due to complete separation.
If the data point denoted by $*$ in (4) has $y_2 = 1$, then the sets $\{y_i = 0; i = 1, \ldots, n\}$ and $\{y_i = 1; i = 1, \ldots, n\}$ cannot be separated by an affine hyperplane, and $n_{\rm co} = 1$. In that case, the maximum likelihood estimate of θ in a logistic regression model does exist.
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 $n_{\rm co} = 0$, and more generally $n_{\rm co} = \mathrm{rdepth}(\theta^*, Z_n)$. This implies that $n_{\rm co}$ can be computed with an algorithm for the regression depth of a given hyperplane, c.f. Christmann and Rousseeuw (1999). 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 (1999) and Rousseeuw and Struyf (1998). For $p \geq 3$, Rousseeuw and Struyf (1998) 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

$$\begin{aligned} n_{\text{co}} &= \text{rdepth}(\theta^*, Z_n) && (5) \\ &= \min_{\theta \in \mathbb{R}^p} \text{rdepth}(\theta^*, \tilde{Z}_n(\theta)) && (6) \\ &\leq \min_{\theta \in B \subset \mathbb{R}^p} \text{rdepth}(\theta^*, \tilde{Z}_n(\theta)) =: n_{\text{co}}(B)\,, && (7) \end{aligned}$$

where

$$\tilde{Z}_n(\theta) = \{((x_i, 1)\theta', y_i); i = 1, \ldots, n\} \subset \mathbb{R}^2,\ \theta \in \mathbb{R}^p. \qquad (8)$$

The set B is determined via projections defined by a large number, say 10^4, of random subsamples of the original data set. Details of the algorithm are described in Christmann and Rousseeuw (1999).
Of course, this approximation algorithm to compute $n_{\text{co}}(B)$ 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.
This motivates us to study the results for other determinations of the set B in (7). A naive alternative to $n_{\text{co}}(B)$ is to use only one *special* vector b in (7). We investigate

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

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 the $\hat{\theta}^{(k)}$. Of course, other programs to compute ML estimates in the logistic regression model can also be used.
The naive method $n_{\text{co}}(b)$ often gives good approximations of n_{co}. Nevertheless, it seems reasonable to find better approximations of n_{co} in an iterative manner as follows.
The heuristic method $n_{\text{co}}(h)$ tries to find good approximations of n_{co} using the vector b as a starting vector for a local search based on the regression depth method and an additional grid search, where sequentially some of the components are set to zero. A description of the heuristic method is given in the appendix.
In the following section a relationship between the regression depth approach and the support vector machine is shown.

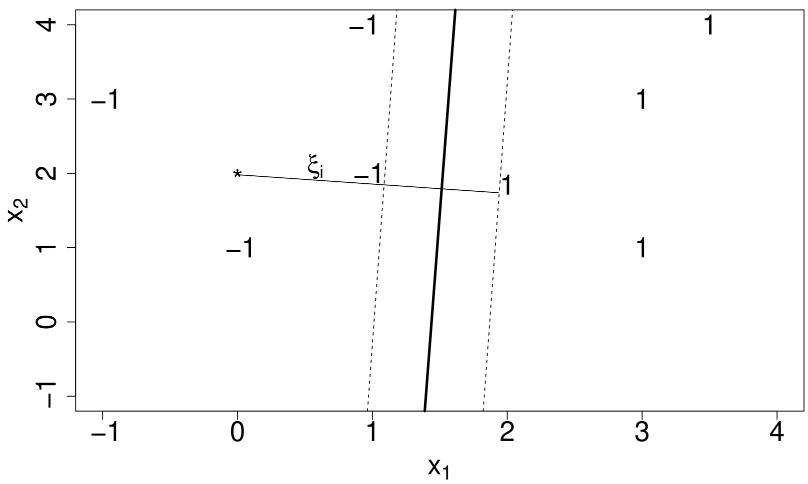

Figure 2: Illustration of the support vector machine.

3 Support vector machine

Vapnik (1998) proposed the support vector machine, which can be described as follows for the case of pattern recognition. As is usual in the literature on the support vector machine, the responses are recoded as $-1/+1$ instead of $0/1$. Decompose θ in the slope part, say $\mathbf{w} = (\theta_1, \ldots, \theta_{p-1})$, and the intercept part θ_p.

To emphasize the connections between the support vector machine and the regression depth method, let us consider the same two-dimensional data set as before, c.f. Figure 2. If the data point marked as $*$ is equal to -1, then the solid line gives a perfect separation of the response groups. The region specified by the dotted border lines is called the margin. It is implicitely defined via the data points on it's 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 perfect 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 penalyzing misclassifications by a large positive extra cost C. Accordingly, the support vector machine solves the following quadratic optimization problem:

$$\text{(P)} \quad \text{minimize (w.r.t. } \mathbf{w}, b, \xi\text{):} \quad \frac{1}{2}\,||\mathbf{w}||^2 + C\sum_i \xi_i$$
$$\text{subject to} \quad \begin{array}{ll} (\mathbf{x}_i, 1)\theta' \geq +1 - \xi_i & \text{for } y_i = +1 \\ (\mathbf{x}_i, 1)\theta' \leq -1 + \xi_i & \text{for } y_i = -1 \\ \xi_i \geq 0 \quad . & \end{array}$$

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, c.f. Burges (1998). Increasing C corresponds to a higher penalty to errors.
In practice, one usually solves the following dual program

$$\text{(D)} \quad \text{minimize (w.r.t. } \alpha\text{):} \quad \frac{1}{2}\,\alpha' Q\alpha - \alpha'\mathbf{1}$$
$$\text{subject to} \quad \begin{array}{l} \alpha'\mathbf{y} = 0 \\ \mathbf{0} \leq \alpha \leq C\mathbf{1} \\ \text{where } (Q)_{ij} = y_i y_j\, x_i' x_j \quad . \end{array}$$

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

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i x_i \,. \tag{9}$$

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 $\mathbf{w}$ (Burges, 1998). This can easily be done after sorting all training points according to their projection on $\mathbf{w}$. This approach is followed in the experiments presented here.
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 (Boser et al., 1992).
If the number of observations n or the dimension p is large, solving the minimization problems (P) or (D) is computer intensive. Please note, that some algorithms require storing the huge matrix $Q \in \mathbb{R}^{n\times n}$ or the whole matrix specifying the constraints. This is true e.g. for PROC NLP or the IML function NLPQUA in SAS (Version 8), which both can be used to solve (D) for small to moderate data sets.
We use SVM^{light} (Joachims, 1999) 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 of (D), the algorithm of SVM^{light} proceeds by decomposing the problem (Osuna et. al, 1997). 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 (Joachims, 1999) is modified for the experiments reported here. 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. To increase the numerical stability of SVM^{light} we use SVM^{light} in conjunction with the PR-LOQO optimizer developed by Smola (1998). Further, the data of all explanatory variables are standardized such that the mean is equal to 0 and the (euclidean) norm is equal to 1. This transformation often avoids numerical problems due to different scalings of the explanatory variables.

4 Data sets

In the next section we will compare the results of $n_{\text{co}}(B)$, $n_{\text{co}}(b)$, $n_{\text{co}}(h)$ and SVM^{light} for the following data sets. They cover a broad range of typical combinations of n and p.
There is complete separation in the banknotes data set (Riedwyl, 1997) and in the hemophilia data set (Hermans and Habbema, 1975).
The vaso constriction data set (Finney, 1947, Pregibon, 1981) and the food stamp data set (Künsch, Stefanski and Carroll, 1989) are well-known in the literature on outlier detection and on robust logistic regression. Christmann and Rousseeuw (1999) show that there are only 3 and 6 observations in these data sets, respectiveley, which can be deleted such that the maximum likelihood estimate $\hat{\theta}$ does not exist for the reduced data sets. Some of these observations are well-known outliers, c.f. Künsch, Stefanski and Carroll (1989).
The cancer remission data set (Lee, 1974) is chosen because $n/p \approx 4$ is small.
The birth weight data set (Hosmer and Lemeshow, 1989) and the coronary heart disease data set (Pires, 1995) are chosen because p is moderate but n is not large.
The toxoplasmosis data set (Efron, 1986) and the IVC data set (Jaeger et al. (1997, 1998), Christmann and Rousseeuw, 1999)) are chosen, because p is small but n is large. The IVC data set is a subset of larger data set from an in vitro experiment to study possible risk factors of the thrombus-capturing efficacy of inferior vena cava (IVC) filters. We focus on the study of a particular conical IVC filter, for which the design consisted of 48 different vectors of the form $(x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4})$. For each vector there were m_i replications with $m_i \in \{50, 60, 90, 100\}$, yielding a total of $n = 3200$.
To also explore a larger data set, we consider a medium-sized text classification problem. The data set WEB collected by Platt (1999) consists of $n = 49749$ WWW-pages represented by their frequency histrogram of 300

words (i.e. $p = 301$). The task is to classify the pages by content.
Finally, the data sets Example 1 and 2 are simulated in the following way to investigate some other complex data sets. Here, $n = 10^4$ and $p = 21$. The first 5 columns are dummy variables from Bernoulli distributions with parameter 0.2, 0.3, 0.4, 0.5, and 0.6, respectively. The last 15 columns of the design matrix X are simulated from a standard normal distribution. The responses Y_i are simulated from a logistic regression model with success probabilities $\Lambda((x_i, 1)\theta')$, where $\theta_j = 1$, $j = 1, \ldots, p$. The data set Example 1 constructed in this way has overlap. The only difference between both data sets is, that a complete separation is artificially constructed in Example 2 by defining $y_i = 1$, if $(x_i, 1)\theta' > 5$, and $y_i = 0$, if $(x_i, 1)\theta' \leq 5$.

5 Results

In this section we compare the results and the computation times given by SVM^{light} and by the regression depth method, i.e. $n_{\text{co}}(B)$, $n_{\text{co}}(b)$, and $n_{\text{co}}(h)$, for the data sets described in the previous section. Our main criterion is a low number of misclassifications. The computation time is our secondary criterion. The computations were done on an IBM/RS6000 Unix workstation, which is approximately twice as fast than a Pentium PC with 166 MHz for our problem. The maximum amount of computation time was set to six hours for each situation.
For the 4 data sets with dimension $p \in \{3, 4\}$ we were able to determine the exact value of n_{co} using the algorithms of Rousseeuw and Struyf (1998). Further, the exact value of n_{co} is known for the banknotes data set and for example 1 because at least one method determines an affine hyperplane which gives a complete separation.
For the first 8 small data sets, none of the considered methods gave consistently better results with respect to the main criterion than the simple heuristic method $n_{\text{co}}(h)$, c.f. Table 1. For all 6 data sets for which the exact value of n_{co} is known, this approximation method yielded the exact value. Whereas $n_{\text{co}}(h)$ is relatively fast for small to moderately large data sets, it can be very slow for high dimensional data sets, c.f. Table 2. E.g., the runtime of $n_{\text{co}}(h)$ on the WEB data for $L = 10^4$ exceeds six hours. On large data sets, the SVM is competitive in term of the approximation quality. The SVM finds better approximations on the IVC and the WEB data, while being less accurate on the artificial examples. An increase of L yielded smaller values of $n_{\text{co}}(h)$ only for two data sets (coronary heart disease data and example 2). The method $n_{\text{co}}(B)$ yield estimates of n_{co} which are not much larger than $n_{\text{co}}(h)$ for small to moderately large data sets and usually better results than the naive approach $n_{\text{co}}(b)$. However, in both simulated examples, which are more complex, the subsampling method $n_{\text{co}}(B)$ does not work well for 10^4 subsamples. For the WEB data the runtime of $n_{\text{co}}(B)$ exceeds six hours. Reducing to 10^3 subsamples, $n_{\text{co}}(B)$ finds only a very loose approximation

Table 1: Values of n_{co} and approximations computed by $n_{\text{co}}(B)$ with 10^4 subsamples, $n_{\text{co}}(b)$, $n_{\text{co}}(h)$ with $L = 10^3$ and $L = 10^4$, and SVM^{light} with options $C = 10^3/(p-1)$ and $C = 10^5/(p-1)$ ($e = 10^{-3}$, $q = 6$, $n = 2$) for several data sets.

	Data set (n, p)	n_{co}	$n_{\text{co}}(B)$	$n_{\text{co}}(b)$	$n_{\text{co}}(h)$		SVM		L_{max}
			10^4		10^3	10^4	10^3	10^5	
1	Banknotes $(200, 7)$	0	0	0	0	0	1	0	100
2	Hemophilia $(52, 3)$	0	0	0	0	0	1	0	22
3	Vaso constriction $(39, 3)$	3	3	3	3	3	3	3	19
4	Food stamp $(150, 4)$	17	17	21	17	17	21	17	24
5	Cancer remission $(27, 7)$	–	3	4	3	3	3	4	9
6	Birth weight $(189, 11)$	–	47	49	41	41	48	49	59
7	CHD $(113, 10)$	–	15	12	10	9	12	13	30
8	Toxoplasmosis $(697, 4)$	284	284	289	284	284	290	290	341
9	IVC $(3200, 5)$	–	458	462	458	458	467	445	748
10	WEB $(49749, 301)$	–	–	596	587	–	685	577	1479
11	Example 1 $(10000, 21)$	0	1396	0	0	0	68	23	2248
12	Example 2 $(10000, 21)$	–	2082	1142	1119	1109	1160	1150	3254

$L_{max} = \min\{\sum y_i, \sum(1 - y_i)\}$

$n_{\text{co}}(B) = 1244$ for the WEB data.
Somewhat astonishing, the naive approach $n_{\text{co}}(b)$ performs much better for large data sets and needs only a fraction of the computation time used by $n_{\text{co}}(B)$. The method $n_{\text{co}}(b)$ is able to detect complete separation in all three separable data sets banknotes, hemophilia, and example 1. By construction, the heuristic method heritates this property from $n_{\text{co}}(b)$.
For the small to moderately large data sets (no. 1 to 9), SVM^{light} performs comparably to $n_{\text{co}}(b)$. The computation time is not of primary interest for such data sets. For large data sets, SVM^{light} finds the best approximation among all methods for the IVC and the WEB data set. Especially for the high dimensional data set WEB, SVM^{light} and $n_{\text{co}}(b)$ are the only algorithms that still exhibit acceptable runtime performance.
Please note that the results of SVM^{light} critically depend on C, c.f. Tables 1 and 2. In general $C = 10^5/(p-1)$ gives a better approximation of n_{co} than $C = 10^3/(p-1)$, but the computation time also increases. While for $C = 10^5/(p-1)$ SVM^{light} successfully detects complete separation for the banknotes and the hemophilia data, it fails for the artificial example 1. This data set is particularly difficult for the SVM, since it has a small margin by construction.

Table 2: Computing times in seconds for n_{co} and its approximations by the programs $n_{co}(B)$ with 10^4 subsamples, $n_{co}(b)$, $n_{co}(h)$ with $L = 10^3$ and $L = 10^4$, and SVM^{light} with options $C = 10^3/(p-1)$ and $C = 10^5/(p-1)$ ($e = 10^{-3}$, $q = 6$, $n = 2$) for several data sets.

	Data set (n, p)	n_{co}	$n_{co}(B)$	$n_{co}(b)$	$n_{co}(h)$		SVM	
			10^4		10^3	10^4	10^3	10^5
1	Banknotes $(200, 7)$	–	0.7	0.2	0.2	0.2	0.1	0.1
2	Hemophilia $(52, 3)$	0.1	0.1	0.1	0.1	0.2	0.1	0.1
3	Vaso constriction $(39, 3)$	0.1	1.1	0.1	0.6	5.2	0.1	0.4
4	Food stamp $(150, 4)$	11.1	4.5	0.1	2.3	21.2	0.1	0.3
5	Cancer remission $(27, 7)$	–	3.4	0.1	0.6	5.1	0.1	1.3
6	Birth weight $(189, 11)$	–	15.6	0.1	4.4	39.5	0.4	6.8
7	CHD $(113, 10)$	–	11.3	0.1	2.5	22.2	0.2	4.8
8	Toxoplasmosis $(697, 4)$	1142.0	2.0	0.1	0.9	7.4	1.8	5.9
9	IVC $(3200, 5)$	–	3.6	0.5	2.0	14.7	3.7	8.5
10	WEB $(49749, 301)$	–	–	396.0	14914.2	–	133.8	700.8
11	Example 1 $(10000, 21)$	–	754.4	33.0	32.9	32.9	23.5	22.9
12	Example 2 $(10000, 21)$	–	753.4	32.1	391.0	3307.0	39.8	55.4

6 Summary

The problem of determining the exact minimum number of misclassifications based on an affine hyperplane is essentially NP-complete, c.f. Höffgen, Simon and van Horn (1995). Hence, there is a need to find reasonable approximation procedures.

In this paper we used the regression depth method introduced by Rousseeuw and Hubert (1999) and the support vector machine approach (Vapnik, 1998) to find such approximations. There are interesting relations between both approaches and the notion of complete separation in the logistic regression model (Albert and Anderson, 1984, Santner and Duffy, 1986).

The results show that it can be helpful in applications to use more than one method to determine a reasonable approximation of the exact minimum number of misclassification based on an affine hyperplane if an exact determination is not possible in a fixed time period.

Summarizing, $n_{co}(h)$ performed well for small to moderate data sets, and the support vector machine and $n_{co}(b)$ performed well for large high dimensional data sets.

The approximation algorithm $n_{co}(B)$ investigated in Christmann and Rousseeuw (1999) gave good – but in general not optimal – results for small to moderate data sets, but can drastically fail for more complex data sets.

The heuristic method based on two-dimensional regression depths seems to be an interesting alternative specially for small and moderately sized data sets, but is slow for high dimensional data sets.
The support vector machine can be an interesting alternative to $n_{\mathrm{co}}(B)$. It's use is most appropriate especially for large and high dimensional data sets, when $n_{\mathrm{co}}(B)$ becomes computationally too inefficient. However, SVM often gave worse upper bounds for the number of misclassifications. Similar to the SVM, for large data sets the naive method $n_{\mathrm{co}}(b)$ can result in a relatively small number of misclassifications and can outperform other methods under considerations.

Acknowledgements

The authors thank Prof. R.J. Carroll for making available the Food Stamp data set, and Dr. H.J. Jaeger for making available the IVC data set.

References

Albert, A. and Anderson, J.A. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika* **71**, 1-10.

Boser, B, Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the 5th Annual Workshop on Computational Learning Theory,* 144-152.

Burges, C.J.C. (1998). A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining.* **2**, 1-43.

Christmann, A. and Roussseeuw, P.J. (1999). *Measuring overlap in logistic regression.* Technical Report, University of Dortmund, SFB 475. http://www.statistik.uni-dortmund.de/sfb475/berichte/tr25-99-software.zip

Efron, B. (1986). Double exponential families and their use in generalized linear regression. *J. Amer. Statist. Assoc.*, **81**, 709-721.

Finney, D.J. (1947). The estimation from individual records of the relationship between dose and quantal response. *Biometrika* **34**, 320-334.

Hermans, J. and Habbema, J.D.F. (1975). Comparison of five methods to estimate posterior probabilities. *EDV in Medizin und Biologie* **6**, 14-19.

Höffgen, K.U., Simon, H.-U., van Horn, K.S. (1995). Robust Trainability of Single Neurons. *J. Computer and System Sciences*, **50**, 114-125.

Hosmer, D.W. and Lemeshow, S. (1989). *Applied Logistic Regression.* Wiley, New York.

Jaeger, H.J., Mair, T., Geller, M., Kinne, R.K., Christmann, A., Mathias, K.D. (1997). A physiologic in vitro model of the inferior vena cava with a computer-controlled flow system for testing of inferior vena cava filters. *Investigative Radiology* **32**, 511-522.

Jaeger, H.J., Kolb, S., Mair, T., Geller, M., Christmann, A., Kinne, R.K., Mathias, K.D. (1998). In vitro model for the evaluation of inferior vena cava filters: effect of experimental parameters on thrombus-capturing efficacy of the Vena Tech-LGM Filter. *Journal of Vascular and Interventional Radiology* **9**, 295-304.

Joachims, T. (1999). Making large-Scale SVM Learning Practical. In: B. Schölkopf, C. Burges, A. Smola (ed.), Advances in Kernel Methods - Support Vector Learning, MIT-Press.
http://www-ai.cs.uni-dortmund.de/svm_light

Künsch, H.R., Stefanski, L.A. and Carroll, R.J. (1989). Conditionally unbiased bounded-influence estimation in general regression models, with applications to generalized linear models. *J. Amer. Statist. Assoc.* **84**, 460-466.

Lee, E.T. (1974). A computer program for linear logistic regression analysis. *Computer Programs in Biomedicine*, 80-92.

Novikoff, A. (1962). On convergence proofs on perceptrons. *Proceedings of the Symposium on the Mathematical Theory of Automata*, Vol XII, pp. 615-622.

Osuna, E., Freund, R., and Griosi, F. (1997). An improved algorithm for training support vector machines. Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, 276-285.

Platt, J. (1999). Fast training of support vector machines using sequential minimal optimization. In: B. Schölkopf, C. Burges, A. Smola (ed.), Advances in Kernel Methods - Support Vector Learning, MIT-Press.

Pires, A.M. (1995). *Análise Discriminante: Novos Métodos Robustos de Estimacão.* Ph.D. thesis, Technical University of Lisbon, Portugal.

Pregibon, D. (1981). Logistic regression diagnostics. *Ann. Statist.* **9**, 705-724.

Riedwyl, H. (1997). *Lineare Regression und Verwandtes.* Birkhäuser, Basel.

Rosenblatt, F. (1962). Principles of Neurodynamics. Spartan. New York.

Rousseeuw, P.J. and Hubert, M. (1999). Regression Depth. *J. Amer. Statist. Assoc.*, **94**, 388-433.

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

Santner, T.J. and Duffy, D.E. (1986). A note on A. Albert and J.A. Anderson's conditions for the existence of maximum likelihood estimates in logistic regression models. *Biometrika* **73**, 755-758.

Smola, A.J. (1998). Learning with Kernels. Ph.D. thesis, TU Berlin, GMD Research Series No. 25.
http://svm.first.gmd.de/software/logosurvey.html

Vapnik, V. (1998). *Statistical Learning Theory.* Wiley, New York.

Appendix

In the following we give pseudo-code for the heuristic method.

1. Read $Z_n = \{(x_{i,1}, \ldots, x_{i,p-1}, y_i);\ i = 1, \ldots, n\}$. Standardize the $x-$variables.

2. Determine the number of different points $(x^a_{j,1}, \ldots, x^a_{j,p-1}, y^a_j)$, say n_a. For each $j \in \{1, \ldots, n_a\}$ count the number t_j of tied data points.

3. Compute b and s(b).
   ncomp $\leftarrow$ n_{co}(b)
   if (ncomp = 0) $\{$ print(b,ncomp); stop $\}$
   L $\leftarrow$ 1000

4. Grid search for the (maximal) first r=10 variables, r < p-1:
   b2 <- b; ncomp4 <- n
   for (all b3=$(d_1, \ldots, d_{p-1}) \neq 0$, where $d_j \in \{0, b_j\}$) $\{$
     if (n_{co}(b3) < ncomp) $\{$
       ncomp4 $\leftarrow$ n_{co}(b3); b4 $\leftarrow$ b3
       if (ncomp = 0) $\{$ b $\leftarrow$ b3; print(b,ncomp); stop $\}$ $\}$

5. Local search starting from b:
   for (c in $\{$ 1, 3, 0.5, 0.1, 0.01 $\}$) $\{$
     b2 <- b
     for (ℓ in $\{$1, ... , L$\}$) $\{$
       b3 $\leftarrow$ b + c s(b) rnorm(p)
       if (n_{co}(b3) < ncomp) $\{$
         ncomp $\leftarrow$ n_{co}(b3); b2 $\leftarrow$ b3 $\}$
       if (ncomp = 0) $\{$ b $\leftarrow$ b2; print(b,ncomp); stop $\}$ $\}$
     if (c = 0.1) ncomplast $\leftarrow$ ncomp
     b $\leftarrow$ b2 $\}$
   Refinement, if neccessary:
   while (ncomp < ncomplast) $\{$
     ncomplast $\leftarrow$ ncomp; c $\leftarrow$ c/2
     for (ℓ in $\{$1, ... , L$\}$) $\{$
       b3 $\leftarrow$ b + c s(b) rnorm(p)
       if (n_{co}(b3) < ncomp) $\{$
         ncomp $\leftarrow$ n_{co}(b3); b2 $\leftarrow$ b3 $\}$
         if (ncomp = 0) $\{$ b $\leftarrow$ b2; print(b,ncomp); stop $\}$ $\}$ $\}$

6. Compare results of grid search and local search:
   if (ncomp4 < ncomp) $\{$ ncomp <- ncomp4; b <- b4 $\}$

7. Print the results: b, ncomp