Erasmus University Rotterdam

Erasmus School of Economics

Bachelor Thesis

Econometrics and Operations Research

Outlier resistent approach to linear support vector machines

Author:
Martijn Cazemier
411884

Supervisor:
Prof. Dr. P. J. F. (Patrick) Groenen

Second Assessor:
Prof. Dr. D. (Dennis) Fok

Abstract

Support vector machines are widely used for the classification of binary response variables. In this paper, a loss function is introduced that enables support vector machines to be more resistant against outliers. Although outliers may contain a lot of information, it is generally not desirable that these observations play a major role in determining the binary classification. The novel error function that is being introduced, the absolute outlier resistant hinge error, restricts the distorting effect of outliers by treating them differently. It attributes a decreasing marginal impact on the loss function to a support vector as it becomes more deviant. Support vector machines that make use of the novel hinge error have been applied to multiple data sets, including ones that are contaminated with artificial outliers. The experiments show signs of an increased resistance against outliers.

July 2, 2017

1 Introduction

Modeling binary response variables is a topic that has been researched extensively. A variety of methods has been developed that deal with this issue. Logit and probit models, which belong to the class of logistic regression, are often used and have been around since the 1960's. More recently developed models, which are becoming increasingly popular, are neural networks and support vector machines (SVM) see e.g. Steinwart & Christmann (2008). This paper deals with the latter approach, SVM. More specifically, this paper builds upon the research into majorizing the loss function associated with the primal formulation of SVM, conducted by Groenen et al. (2008). It is conventional when implementing SVM to switch to the dual formulation and in a lot of cases, for example, when incorporating kernels, it is more convenient or even required to do so. The primal approach however allows for a nonstandard way of looking at SVM, which has a clear and insightful interpretation.

In Groenen et al. (2008), the loss function associated with the primal approach is constructed in different ways based on three different error functions and is optimized by means of a corresponding majorization algorithm. The error function determines the penalty imposed to the loss function as the result of a misspecified observation. The three error functions that are dealt with by Groenen et al. (2008) are the absolute hinge error, the quadratic hinge error, and the Huber hinge error, all three are convex. This convexity property has benefits regarding the optimization of the loss function. However, convexity can also lead to disproportionately large penalties in case of outliers. In this paper, a new non-convex error function specification is proposed with the purpose of adequately treating outliers and thereby increasing resistance against outliers. This error function will have a negative second derivative on parts of its domain, enforcing a smaller punishment upon the loss function for outlying observations than the three error functions mentioned above.

Brooks (2011) also proposes a method that corrects for outliers introducing two error function specifications. Brooks (2011) shows that SVM that incorporate one of these adjusted error functions, in some cases, perform better than existing methods. The novel error function that is proposed in this paper is different from the ones proposed by Brooks (2011) in two important ways: the first difference is that the novel error function allows for the specification of a threshold and every observation that exceeds the threshold will be classified as an outlier, the second difference is that the magnitude of the observation does matter for observations that are classified as an outlier.

The main question of this research is: Do linear SVM that incorporate the novel hinge error yield better out-of-sample predictions than linear SVM that incorporate the absolute, quadratic, or Huber hinge error, in particular when applied to datasets containing outliers?

In the course of this paper the exact function specification of the novel hinge error will be defined and motivated, a majorization algorithm will be developed, and possible problems that arise due to the non-convexity of the error function will be addressed. In Section 2 the basic concept of SVM will be clarified paving the way for a more thorough analysis of one specific aspect of SVM, the error function. Section 3 provides a careful examination of the newly introduced error function and three existing error functions. Finally Section 4 is dedicated to describing how the numerical optimization method majorization can be applied in the framework of SVM.

2 Support Vector Machines

In this section, a short and intuitive explanation of SVM is presented, while introducing the necessary notation along the way. SVM are used to explain and predict the division of observations into two groups. SVM classify each observation i, $i = 1, \ldots, n$, in one of the two groups, based on the $1 \times m$ vector of predictor variables $\boldsymbol{x}_i'$. The vectors $\boldsymbol{x}_i'$ are the rows of the $n \times m$ matrix $\boldsymbol{X}$. The $n \times 1$ vector $\boldsymbol{y}$ consists of response variables y_i, which can take on the values 1 and -1 indicating the group to which observation i belongs. The variable q_i, which is used as input in the loss function, is a weighted sum of the predictor variables and is defined as

$$q_i = c + \boldsymbol{x}_i'\boldsymbol{w}, \tag{1}$$

where the parameter c is an intercept and $\boldsymbol{w}$ is a vector of weights.

In Figure 1 a number of observations is plotted into an m-dimensional space, $m = 2$. In this figure the plusses represent a group and the circles represent the other group.

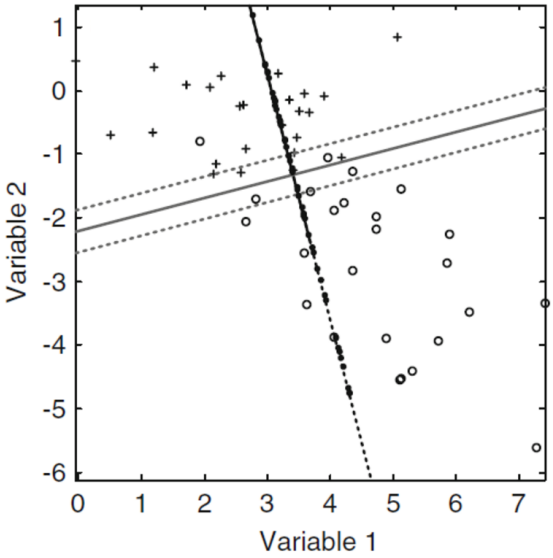


Figure 1: Projections of the observations in groups $1(+)$ and $-1(o)$ onto the line given by w_1 and w_2.

What you see in Figure 1 is a somewhat horizontal line that ought to separate the class 1 objects from the class -1 objects as well as possible, called the separation line. The parameter vector $\boldsymbol{w}$ determines the direction of this separation line and the intercept c determines its location. Hence, the purpose of SVM is to determine c and $\boldsymbol{w}$. The parameters c and $\boldsymbol{w}$ are determined by the minimization of a loss function. The variable q_i that is defined above, is the distance between the point $(x_{i,1}, x_{i,2})$ and the separation line multiplied by the length of vector $\boldsymbol{w}$, $||\boldsymbol{w}|| = (\boldsymbol{w}'\boldsymbol{w})^{1/2}$. Parallel to the separation line there are two dotted lines, called the margin lines. Each point that is on the wrong side of its respective margin line is called a support vector. An error function, $f(q_i)$, takes on positive values for q_i that correspond to support vectors and takes on zero otherwise. Hence, only support vectors contribute to the loss function defined as

$$L_{\text{SVM}} = \sum_{i=1}^{n} f(q_i) + \lambda \boldsymbol{w}'\boldsymbol{w}. \tag{2}$$

Observations that are on the right side of their respective margin line do not contribute to the loss function.

If the error function f is a coercive continuous function, then the loss function of (2) has a global minimum. The values of the parameters c and $\boldsymbol{w}$ that correspond to this global minimum are the optimal ones. The purpose of the last part of (2), λ times the inner product of w, is to control the length of vector $\boldsymbol{w}$. This is done by imposing a punishment to the loss function of size $\lambda||\boldsymbol{w}||$. This punishment is proportional to the length of $\boldsymbol{w}$. This parameterized punishment influences the number of correctly classified observations that contribute to the loss function. Namely, a higher value of $||\boldsymbol{w}||$ corresponds to a higher value of q resulting in less observations in between the margin lines and the separation line and vice versa.

The way in which each support vector contributes to the loss function is determined by the error function, $f(q_i)$. In the next section several existing error functions are discussed and the novel outlier resistant error function is specified and motivated.

3 Error function specifications

The error function is a vital part of SVM as it determines how big the contribution to the loss function of each support vector is. In this section three existing error functions will be discussed and a new error function will be introduced. What all hinge errors that will be discussed have in common is that the further away a support vector is from its respective margin line the higher the corresponding punishment will be i.e. the more this observation will contribute to the loss function. The first error function that we define is the absolute hinge error, which is most commonly used. The absolute hinge error is defined as

$$f_{\mathrm{A}}(q_i) = \max(0, 1 - y_i q_i). \tag{3}$$

The absolute hinge error imposes a punishment that is related linearly to the distance of the support vector to its respective margin line.

Two other hinge errors, which are also discussed by Groenen et al. (2008), are the quadratic hinge and the Huber hinge error. These two error functions are defined as

$$f_{\mathrm{Q}}(q_i) = \max(0, 1 - y_i q_i)^2 \tag{4}$$

and

$$f_{\mathrm{H}}(q_i) = \begin{cases} 1/2(k+1)^{-1} \max(0, 1 - y_i q_i)^2 & \text{if } -y_i q_i \leq k \\ 1 - y_i q_i - (k+1)/2 & \text{if } -y_i q_i > k \end{cases}. \tag{5}$$

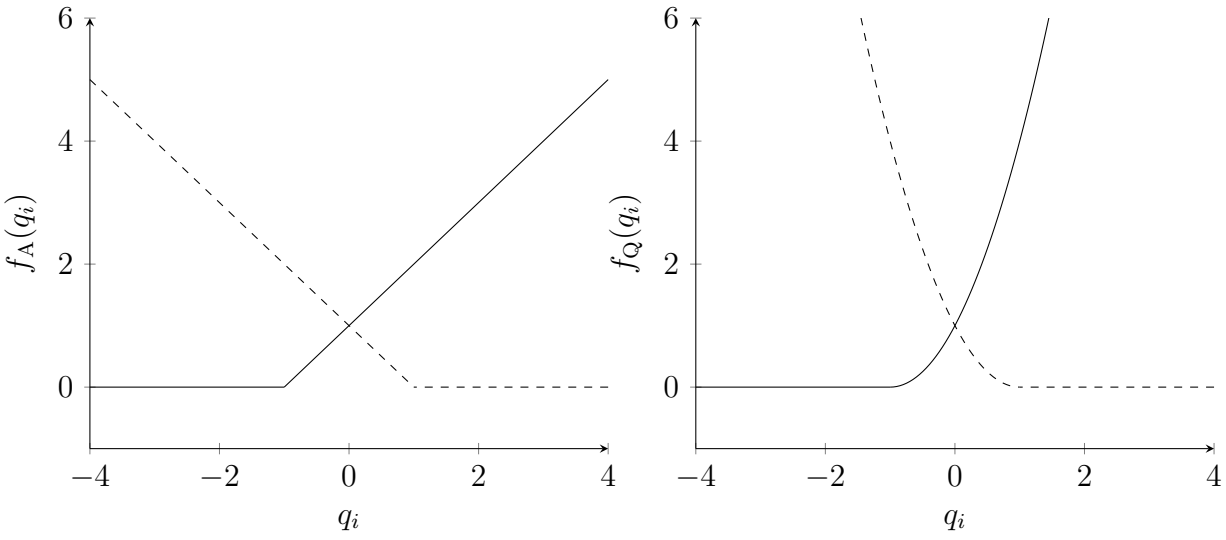
The quadratic hinge error is characterized by a quadratic relationship between the above mentioned distance and the punishment imposed. The Huber hinge error is a smooth hybrid of the absolute and the quadratic hinge error. Up to a certain pre-specified value of k the punishment is quadratic in q_i and for values of q_i higher than k this relationship becomes linear. The novel hinge error function that is introduced in this paper is named the absolute outlier resistant (AOR) hinge error. The AOR hinge error is defined as

$$f_{\mathrm{AOR}}(q_i) = \begin{cases} \max(0, 1 - y_i q_i) & \text{if } -y_i q_i \leq T \\ T + 1 + \ln(1 - y_i q_i - T) & \text{if } -y_i q_i > T \end{cases}, \tag{6}$$

where T is the threshold that separates the AOR hinge error into two parts. The threshold, T, can take values in the interval $[-1, \infty)$. The threshold adds a certain degree of flexibility

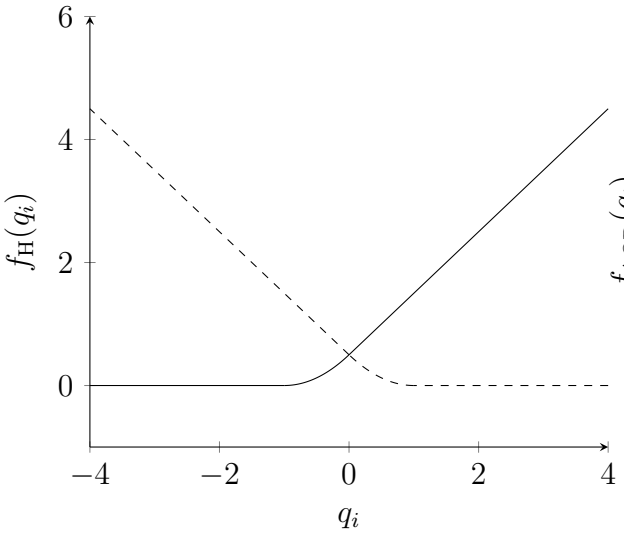
to the AOR hinge error. For example, if T is chosen very high approaching infinity the AOR hinge error is equivalent to the absolute hinge error. Below in Figure 2 the hinge error functions of (3), (4), (5), and (6) are represented by plot a, b, c, and d respectively.

An observation can be considered an outlier if it is on the wrong side of the margin line and relatively far from the separation line. The AOR hinge error is constructed with the aim of reducing the distorting effects of outliers. For this purpose the AOR hinge error of (6) consists of two parts. The first part consists of observations for which $-y_i q_i \leq T$ holds. For this part the AOR hinge error corresponds to the absolute hinge error. The other part of the AOR hinge error consisting of the observations for which $-y_i q_i > T$ holds, is a $\ln(1+x)$ type function which has a negative second derivative. f_{AOR} has a negative second derivative in $-y_i q_i$ for $-y_i q_i > T$.

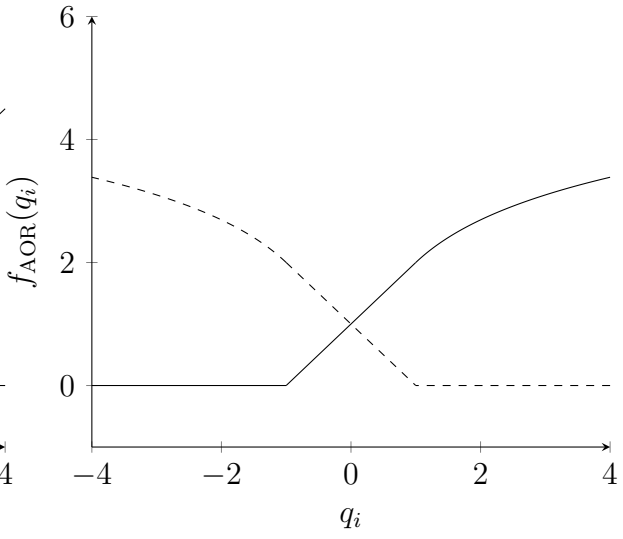


(a) The absolute hinge error function

(b) The quadratic hinge error function



(c) The Huber hinge error function



(d) The AOR hinge error function

Figure 2: Plots of four different hinge error functions including the AOR hinge error with threshold $T = 1$ in (d). For the Huber hinge error in (c) $k = 0$ is chosen.

As you can see from Figure 2 the novel error function for the case where $y_i = -1$ is characterized by a gradual decrease of the gradient of the error function as $|q_i|$ increases after exceeding the threshold T. This decrease of the gradient corresponds to the negative second derivative mentioned above. A decreasing gradient basically means that when the distance between a support vector and its respective margin line increases, the marginal impact of this observation as regards determining the position and the direction of the separation line, decreases.

The absolute hinge error increases linearly in q_i and the Huber hinge error does so too given $q_i > k$. The quadratic hinge error increases quadratically in q_i and the Huber hinge error does so too given $q_i < k$. It follows that for these hinge errors, outliers will contribute substantially to the loss function as none of the second derivatives is negative. Subsequently parameter estimates may be influenced disproportionately by outliers. Although outliers may contain a lot of information it is not always desirable that these observations play a major role in determining the parameter estimates as they display characteristics that deviate from the majority of the observations.

4 Optimizing the loss function

In the previous sections the concept of SVM has been clarified and the associated loss function of (2) is completely specified for each of the four hinge errors. The parameters that minimize this loss function correspond to a possible choice for the separation line.

In this section a numerical optimization method will be discussed to find these parameters that minimize the loss function. The approach to find such parameters that will be discussed in this section is a rather non-standard one. Whereas in existing literature it is often suggested to switch to the dual of the loss function and using quadratic program solvers. Here a majorization approach is presented to minimize the primal formulation of the loss function, (2), directly. In the subsequent subsections, the concept of iterative majorization (IM) is explained briefly and it is demonstrated how this optimization methods can be used in the context of SVM. At the end of this section the majorization algorithm, SVM-Maj, is presented in Algorithm 1 which can be implemented using software like matlab and R.

4.1 Majorization

Let $f(q)$ be a function that needs to be minimized. Then Iterative Majorization (IM) is a minimization method that makes use of an auxiliary function, say $g(q, \bar{q})$, to minimize the original function $f(q)$. The method iterates over values of q till the minimum of $f(q)$ is reached. $\bar{q}$ is defined as the current position at the start of an iteration, the supporting point. Each iteration yields you a new position, q^*, for which holds that $f(q^*) \leq f(\bar{q})$. Preferably the IM algorithm does not need a lot of iterations to converge to the minimum of f. The new position q^* is the minimum of the auxiliary function, $g(q, \bar{q})$, called the majorizing function. The majorizing function should be a simple one, preferably linear or quadratic this way the minimum of g is easy to find. The fact that $f(q^*) \leq f(\bar{q})$ at each iteration follows from the conditions that $g(q, \bar{q})$ has to adhere to listed below.

The first condition is that $g(q, \bar{q})$ should touch f in $\bar{q}$, i.e. $g'(\bar{q}, \bar{q}) = f'(\bar{q})$ and $g(\bar{q}, \bar{q}) = f(\bar{q})$. The second condition is that the majorizing function should never be below f, i.e. $g(q, \bar{q}) \leq f(q)$. These two conditions lead to the following so-called sandwich inequality

$$f(q^*) \leq g(q^*, \bar{q}) \leq g(\bar{q}, \bar{q}) = f(\bar{q}). \tag{7}$$

From the sandwich inequality it can be seen that the update $q^* = \text{argmin}_q g(q, \bar{q})$ is an appropriate update in the sense that $f(q^*) \leq f(\bar{q})$ always holds. By repeating these iterations the loss function will continue to decrease until a local or global minimum is reached. Majorization is more extensively discussed in De Leeuw (1994), Heiser (1995), **?**, Kiers (2002) Hunter & Lange (2004), and Borg & Groenen (2005)

4.2 Majorization applied to SVM

The function that needs to be minimized by means of IM is the loss function of (2). This loss function is a summation of error functions. A desirable property of majorization functions is that the sum of majorization functions, is a majorization function for the sum of the majorized functions. This leads to the insight that to find the majorization function of (2) we only need majorization functions for the hinge errors.

It can be shown that each of the hinge errors can be quadratically majorized by

$$g(q_i, \bar{q}_i) = a_i q_i^2 - 2 b_i q_i + c_i. \tag{8}$$

The specifications of a_i, b_i, and c_i for each of the hinge errors are disclosed in the appendix. A detailed derivation of the majorizing function for the AOR hinge error is to be found in the appendix as well. For the derivation of the majorizing function for the other three hinge errors see Groenen et al. (2008).

It follows from (8) and the desirable property about the sum of majorizing functions that the L_{SVM} can be majorized as

$$L_{SVM}(c, \boldsymbol{w}) \leq \sum_{i=1}^{n} a_i q_i^2 - 2 \sum_{i=1}^{n} b_i q_i + \sum_{i=1}^{n} c_i + \lambda \boldsymbol{w}'\boldsymbol{w}. \tag{9}$$

For the sake of mathematical convenience an extra column of ones is added as the first column of $\boldsymbol{X}$ resulting in an $n \times (m+1)$ matrix $\boldsymbol{X}$. The vector $\boldsymbol{v}$ is defined as $\boldsymbol{v}' = [c\ \boldsymbol{w}']$. Now $q_i = c + \boldsymbol{x}_i'\boldsymbol{w}_i$ can be expressed as $\boldsymbol{q} = \boldsymbol{X}\boldsymbol{v}$ and (9) can be rewritten as

$$\begin{aligned} L_{SVM}(v) &\leq \sum_{i=1}^{n} a_i (x_i'v)^2 - 2 \sum_{i=1}^{n} b_i (x_i'v) + \sum_{i=1}^{n} c_i + \lambda \boldsymbol{w}'\boldsymbol{w} \\ &= \boldsymbol{v}'\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}\boldsymbol{v} - 2\boldsymbol{v}'\boldsymbol{X}'\boldsymbol{b} + c_m + \lambda \boldsymbol{v}'\boldsymbol{P}\boldsymbol{v} \\ &= \boldsymbol{v}'(\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X} + \lambda \boldsymbol{P})\boldsymbol{v} - 2\boldsymbol{v}'\boldsymbol{X}'\boldsymbol{b} + \sum_{i=1}^{n} c_i, \end{aligned} \tag{10}$$

where $\boldsymbol{A}$ is a diagonal matrix with $\boldsymbol{A}_{i,i} = a_i$, $\boldsymbol{b}$ is a vector with elements b_i and $\boldsymbol{P}$ is the identity matrix with $p_{1,1} = 0$. The derivative of the last line of (10) is taken with respect to $\boldsymbol{v}$ and set equal to zero obtaining

$$(\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X} + \lambda \boldsymbol{P})\boldsymbol{v} = \boldsymbol{X}'\boldsymbol{b}. \tag{11}$$

Solving (11) for $\boldsymbol{v}$ yields $\boldsymbol{v}^+$ consisting of the intercept, c, and the direction vector, $\boldsymbol{w}$, that minimize the majorization function. This follows from the definition of $\boldsymbol{w}$ which is

$\boldsymbol{v} = [c, \boldsymbol{w}]'$. $\boldsymbol{v}^+$ can be retrieved by Gaussian elimination or less efficiently by multiplying both sides of (11) by $(\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X} + \lambda\boldsymbol{P})^{-1}$. The iterations can be computed more efficiently for the quadratic and the Huber hinge error because for these errors it holds that $a_i = a$ for all i and it does not depend on $\bar{q}_i$. This means that retrieving update v^+ simplifies to

$$\boldsymbol{v}^+ = (a\boldsymbol{X}'\boldsymbol{X} + \lambda\boldsymbol{P})^{-1}\boldsymbol{X}'\boldsymbol{b}, \tag{12}$$

for these two error functions. The first part of (12), $\boldsymbol{S} = (a\boldsymbol{X}'\boldsymbol{X} + \lambda\boldsymbol{P})^{-1}\boldsymbol{X}'$, does not depend on q_i. Hence this part does not change during the iterative process. Therefore when carrying out the SVM-Maj algorithm, $\boldsymbol{S}$ only needs to be computed once for the first iteration and can be stored in memory for use in all following iterations.

The SVM-Maj algorithm that is summarized in Algorithm 1 on the next page guarantees the loss function to not increase and usually decrease in each iteration. The updates of c and $\boldsymbol{w}$ will come closer to the global minimum after each iteration. At least this is true for a convex error function specifications like the absolute, quadratic, and Huber hinge error. However the AOR hinge error is not convex. Therefore the loss function of (2) which constitutes to the sum of non-convex functions may contain local minima. Note that all error functions are coercive; thus, (2) has a global minimum for every error function. The possible presence of local minima leads to an increase in computation time for two reasons:

1. To cope with the possible presence of local minima the SVM-Maj algortihm needs to be carried out multiple times using a multitude of initial values for c and $\boldsymbol{w}$.

2. because multiple starting points need to be chosen when the AOR hinge error is used the possibility of using smart initial values is excluded. Smart initial values are values that are already close to the values that correspond to the global minimum, such as values from previous cross validation runs. The use of smart initial values, also referred to as a warm start, cuts down computation time. For the SVM-Maj algorithm that is using one of the three existing hinge errors smart initial values can be chosen.

There is another factor to consider when using the AOR hinge error. A threshold, a value for T, needs to be determined. For different choices of t the optimal choice of λ might change. In general the optimal choice of λ can be determined using fivefold cross-validation. Furthermore, the number of random initial values chosen is denoted by R. Note that $R = 1$ suffices for convex error functions.

In the next section experiments will be performed using different hinge errors. Among other things the need for multiple starting points and its influence on the computation time will be investigated.

Algorithm 1: SVM-Maj

Input: $\boldsymbol{y}$, $\boldsymbol{X}$, λ, Hinge, k, T, R
Output: c, $\boldsymbol{w}$

$L_{\min} = 0$;
Set ϵ to a small value;
if *Hinge = Huber or Quadratic* **then**
  if *Hinge = Quadratic* **then** $a = 1$
  if *Hinge = Huber* **then** $a = (1/2)(k+1)^{-1}$
  $S = (a\boldsymbol{X'X} + \lambda\boldsymbol{P})^{-1}\boldsymbol{X'}$
for $r = 1 : R$ **do**
  Set $\boldsymbol{w_0}$ and c_0 to random initial values;
  Compute L_{SVM} as in (2)
  while $t = 0$ *or* $(L_{t-1} - L_{SVM}(c_t, \boldsymbol{w}_t))/L_{SVM}(c_t, \boldsymbol{w}_t) > \epsilon$ **do**
    $t = t + 1$
    $L_{t-1} = L_{\text{SVM}}(c_{t-1}, \boldsymbol{w}_{t-1})$
    Comment: Compute $\boldsymbol{A}$ and $\boldsymbol{b}$ for different hinge errors
    if *Hinge = Absolute* **then**
      Compute a_i by (13)
      Compute b_i by (14)
    else if *Hinge = AOR* **then**
      Compute a_i by (27)
      Compute b_i by (28)
    else if *Hinge = Quadratic* **then**
      Compute b_i by (17)
    else if *Hinge = Huber* **then**
      Compute b_i by (20)
    Make the diagonal matrix $\boldsymbol{A}$ with elements a_i
    Comment: Compute update
    if *Hinge = Absolute or AOR* **then**
      Find $\boldsymbol{v}$ that solves (11): $(\boldsymbol{X'AX} + \lambda\boldsymbol{P})\boldsymbol{v} = \boldsymbol{X'b}$
    else if *Hinge = Huber or Quadratic* **then**
      $\boldsymbol{v} = \boldsymbol{Sb}$
    Set $c_t = v_1$ and $w_{t,j} = v_{j+1}$ for $j = 1, \ldots, m$
  if $r = 1$ *or* $L_{SVM}(c_t, \boldsymbol{w}_t) \leq L_{min}$ **then**
    Set $c = c_t$ and $\boldsymbol{w} = \boldsymbol{w}_t$
    $L_{\min} = L_{\text{SVM}}(c_t, \boldsymbol{w}_t)$
  $t = 0$

5 Experiments

In this section, the results of several experiments that have been carried out are presented. The purpose of these experiments is to determine and evaluate the properties of the novel hinge error function, AOR. The properties of interest are the five-fold cross-validated accuracy and the computational speed of SVM-Maj. These two properties are dealt with in the next two subsections. Finally the last subsection of the results has as its sole purpose the reproduction of several results from Groenen et al. (2008). For the applications three data sets are used that are obtained from the UCI repository (Newman et al., 1998) and the home page of LibSVM software (Chang & Lin, 2006). The properties of these data sets are listed in Table 1 below.

Table 1: The properties of the data sets

Dataset	Source	n	n_1	n_{-1}	m	Sparsity
Australian	LibSVM	690	307	308	14	20.04
Breast_cancer_w	LibSVM/UCI	699	458	241	9	0.00
Heart_statlog	UCI	270	120	150	13	0.00

5.1 Predictive performance

The main concern of this paper is to find out how well SVM that use the AOR hinge error predict out-of-sample compared to SVM that use existing hinge errors. In particular it is of interest how the predictive performance of the AOR compares when applied to a dataset that contains outliers. Hence an experiment has been set-up in which SVM that use the AOR hinge error, with different threshold values, and SVM that use the absolute hinger error are applied to eight datasets. Six of these datasets are contaminated with artificial outliers. The reason that only the predictive performance of the absolute and AOR hinge errors is compared is that according to Groenen et al. (2008) the predictive performance of the absolute, quadratic and Huber hinge errors is quite similar. Therefore it does not seem of added value to include the quadratic and the Huber hinge errors in the comparison.

The four contaminated datasets have been constructed from two original datasets by multiplying the predictor variables of 10% of the observations by either 100, 10, or 5 creating artificial outliers in the data. The predictive performance is measured in terms of accuracy, defined as the total number of correctly predicted observations divided by the total number of predicted observations in fivefold cross-validation. The unknown parameters c and w, that determine the classification of observations into the two groups, are estimated fixing $\lambda = 2^p$ at its optimal value. The optimal value for λ is chosen as the λ that yields the highest fivefold cross-validation accuracy, where p can take any of the values $15, 14.5, 13, \ldots, -7, -7.5, -8$. This means that λ can take values in the interval $[32768, 0.00391]$. In all the experiments $\epsilon = 3 \times 10^{-7}$ is used as the stopping criterion, unless it is explicitly stated differently. For the applications with the AOR hinge error the threshold takes values $T = 0, 0.5,$ and 1. Furthermore, when using the AOR hinge error, a single iteration of the SVM-Maj algorithm may yield estimates for c and w that correspond to a local minimum. Therefore the values for c and w that minimize the loss function of 2, L_{SVM}, are estimated by performing 20 repetitions of the SVM-Maj algorithm with different initial values for c and w. The resulting comparison

Table 2: A comparison of the accuracy of SVM using the absolute and AOR hinge errors.

Dataset	Outlier type	optimal p				accuracy			
		Absolute	AOR			Absolute	AOR		
			$T=1$	$T=0.5$	$T=0$		$T=1$	$T=0.5$	$T=0$
Australian	no	*4.5	3.5	-3.5	-7.5	*85.5*	85.1	84.9	*85.5*
	10% × -1	4.0	4.5	4.5	3.0	83.9	83.7	83.7	*85.5*
	10% × 5	4.0	4.0	3.5	4.0	85.5	85.0	*85.7*	85.3
	10% × -5	-5.5	2.5	3.5	3.0	77.8	*83.5*	*83.5*	*83.5*
	10% × 10	2.0	3.5	-4.0	4.0	*84.8*	*84.8*	84.7	84.5
	10% × -10	-0.5	2.5	3.0	3.0	67.4	*83.5*	*83.5*	*83.5*
	10% × 100	3.0	-6.5	-1.0	2.0	77.9	85.4	85.5	*85.7*
Breast_ cancer_w	no	*6.5	7.0	7.0	7.0	97.0	97.0	*97.3*	97.0
	10% × -1	9.0	5.0	4.5	6.0	92.2	92.9	*93.2*	*93.2*
	10% × 5	-0.5	-2.0	0.0	1.5	93.7	93.7	*94.1*	*94.1*
	10% × -5	10	-3.5	3.0	6.0	88.7	92.9	92.9	*93.2*
	10% × 10	5.5	5.5	7.0	5.5	*93.4*	93.1	93.0	93.1
	10% × -10	5.5	6.0	4.0	6.0	87.1	92.8	92.9	*93.2*
	10% × 100	-1.0	2.5	-1.0	2.5	91.1	92.6	*93.1*	92.8

10% × -1, 5, -5, 10, -10, or 100 means that 10 percent of the data has been multiplied by 5, 10, -10, or 100 respectively. The highest accuracy for a specific data set is displayed in itallics.

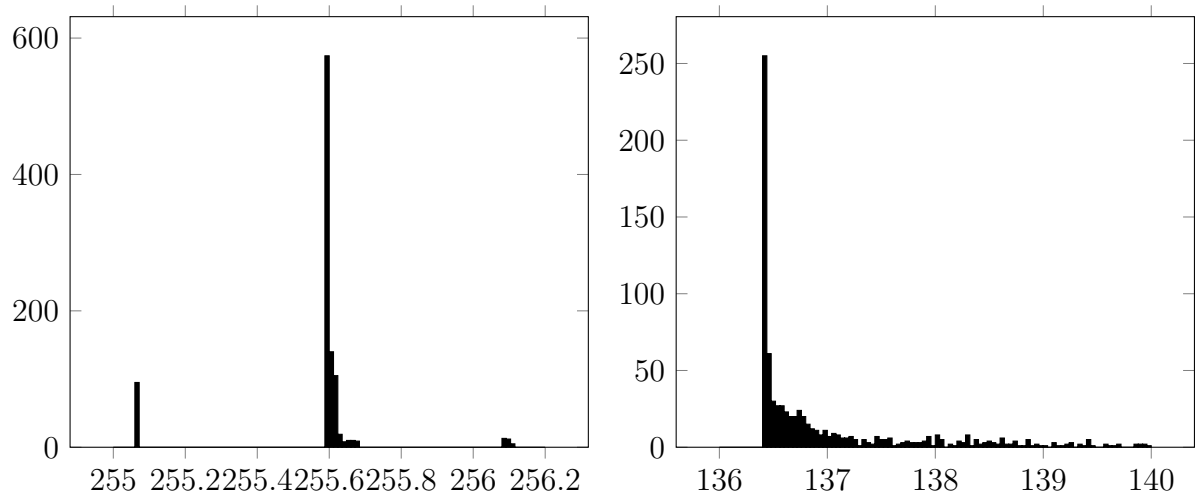
of the predictive performance of SVM using either of both hinge errors is summarized in table 2.

From table 2 it can be seen that there is only one data set for which the absolute hinge error yields a better accuracy than the AOR hinge error. This is the Australian data set with 10%× 10 type outliers. The accuracies of both hinge errors suffers in varying degrees from outliers. The best performance of the AOR hinge error relative to the absolute hinge error is recorded for the Australian dataset with 10% × 100 type outliers. In this case the performance of the SVM with the absolute hinge suffers a lot whereas the AOR hinge error seems to effectively ignore the outliers. Furthermore it can be seen that for the AOR hinge error with different thresholds, different values for λ prove to be optimal.

The accuracies and the optimal values for λ reported in table 2 depend on how the dataset is divided into five groups to perform the fivefold cross validation. In this case a random vector is generated that assigns an equal number of observations to each group, (some groups may contain one observation less than the others in case $\frac{n}{5} \notin \mathbb{Z}$). This explains why in (Groenen et al., 2008) different optimal values for p are found for the absolute hinge error when applied to the Australian and the Breast_cancer_w datasets, these two results are marked with an asterisk in Table 2. Furthermore note that some of the accuracies might be suboptimal as the SVM-Maj algorithm using the AOR hinge error does not guarantee to find a global optimum.

5.2 Computational speed and multiple starting points

It turns out that performing the SVM-Maj algorithm is more time consuming with the AOR hinge error than with the absolute hinge error. In this subsection the results of two



(a) Australian 10% × 5 type outlier (b) Breast_cancer_w 10% × 10 type outlier

Figure 3: Histogram of the L_{SVM} loss function values for 1000 random starting points using the AOR hinge error. plot a.: Australian 10% × 5 outlier type, threshold $T = 0.5$, and p fixed at its optimal value of 3.5. Plot b.: Breast_cancer_w 10% × 10 outlier type, threshold $T = 1$ and p fixed at its optimal value of 5.5.

experiments will be discussed which show that and explain why the computation time is higher. These experiments show two factors that contribute to the increased computation time. The first factor being that one needs to go through the iterative majorization (IM) process multiple times to minimize the loss function instead of just one time. The second factor being that no smart initial values for c and w can be used when performing a majorization, which may require the algorithm to perform a higher number of iterations until convergence.

The first experiment is done to investigate whether or not the global minimum of the loss function of (2) with $f = f_{\text{AOR}}$, will always be found when minimizing by means of IM. To do so IM is applied 1000 times on two different datasets starting from random initial values for c and $\boldsymbol{w}$. The resulting loss function values have been recorded for all of the 1000 replications and all these recordings are summarized in the histograms of Figure 3.

From these two histograms it is clear that local minima are present and that multiple starting points are required when implementing the AOR hinge error. In the histogram of Figure 3a two spikes can be observed. The lowest spike is around 255.07 and the other spike is around 255.59. The lowest 96 out of 1000 observations have a loss function value that is within 0.4×10^{-2} of the observed minimum.

For the histogram of Figure 3b a spike is observed around 136.5. The lowest value observed for L_{SVM} is 136.4141, around five percent of the observations are within 0.2×10^{-3} from this observed minimum. In the panel of Figure 3b, 786 out of 1000 observations are included, the other 224 observations are higher than 140 and are not displayed. The highest 50 observations are between 360 and 600.

The second experiment is designed to test whether or not the fact that smart initial values cannot be used, has an effect on the number of iterations needed to converge. In the experiment the SVM-Maj algorithm is applied to each of the 5 cross-validation samples recording the number of iterations needed to converge and the CPU time need to perform these iterations. For the instances where the AOR hinge error is used, IM is performed starting from random initial values. For the instances where the absolute hinge error is

used, IM is performed starting from smart initial values.

In the two tables below Table 3 and Table 4 the number of iterations needed to converge in a single majorization run are presented for the absolute and the AOR hinge error. For the instance presented in Table 3 it holds that for each cross-validation sample more iterations are needed to converge in case of the AOR hinge compared to the absolute hinge. However the same result does not hold for the instance displayed in Table 4. It can be seen from dividing any of the CPU times by its corresponding number of iterations that an iteration takes about 0.0029 seconds which means that increases (decreases) in CPU time are proportional to increases (decreases) in iterations. For the interpretation of the table entries two circumstances are important to consider. The first is that the number of iterations needed and the CPU time needed are highly dependent on the starting points and for the AOR hinge error these are random. The second is that a different values for p which amount to a different values for λ can amount to differences in the number of iterations needed to converge.

Table 3: Iteration comparison between the absolute and AOR hinge error applied to the Australian dataset without outliers.

	Absolute $p = 4.5$					AOR $p = -3.5$				
cvSample	1	2	3	4	5	1	2	3	4	5
CPU time	0.178	0.151	0.295	0.208	0.215	1.125	0.592	0.920	4.950	0.983
Iterations	66	56	109	77	76	394	207	312	1675	349

The displayed CPU time is measured in seconds and the threshold for the AOR is chosen to be $T = 0.5$. cvSample stands for cross-validation sample.

Table 4: Iteration comparison between the absolute and AOR hinge error applied to the Breast_cancer_w dataset without outliers.

	Absolute $p = 6.5$					AOR $p = 7.0$				
cvSample	1	2	3	4	5	1	2	3	4	5
CPU time	0.283	0.134	0.254	0.148	0.128	0.179	0.173	0.162	0.193	0.237
Iterations	151	71	128	78	51	87	83	79	93	120

The displayed CPU time is measured in seconds and the threshold for the AOR is chosen to be $T = 1$. cvSample stands for cross-validation sample.

5.3 Reproduction

The purpose of this section is reproducing results of Groenen et al. (2008). The results of two different experiments are presented. In the first experiment the predictive performance of the absolute, Huber, and quadratic hinge error is evaluated fixing λ at its optimal value as established in Groenen et al. (2008). The obtained five-fold cross validation accuracies are reported in Table 5.

The second experiment is a minimization of the loss function under different values of the stopping criterion ϵ by by means of SVM-Maj with the absolute hinge error. This is done for three different data sets and the results are displayed in Table 6.

Table 5: Performance of SVM for the absolute (Abs.), Huber (Hub.), and Quadratic (Quad.) hinge error

Data set	Optimal p			Five-fold CV accuracy		
	Abs.	Hub.	Quad.	Abs.	Hub.	Quad.
Australian	-0.5	2.0	3.0	85.5	86.1	86.4
Breast_cancer_w	7.5	6.0	8.0	96.7	96.5	96.7
Heart_statlog	0.0	5.5	7.0	83.5	83.7	83.8

The optimal value for p ($\lambda = 2^p$) has been determined by five-fold cross validation. The predictive performance measured as accuracy (in %) is obtained for 3 different test datasets.

Table 6: Minimal loss function values under different stopping criterion

Dataset	p	L_{SVM}		
		10^{-4}	10^{-5}	10^{-6}
Australian	0	202.78	202.73	202.66
Breast_cancer_w	6	58.16	58.04	58.03
Heart_statlog	0	91.52	91.48	91.48

The value for p ($\lambda = 2^p$) is fixed at a level close to the optimal one of Table 5.

6 Conclusion and discussion

This research is conducted with the aim of finding an error function that enables SVM to be resistant to outliers. The absolute outlier resistant (AOR) hinge error is introduced in this paper as a candidate for such an error function. It turns out that SVM that incorporate the AOR hinge error have an equal or better forecasting performance for five out of the six data sets that have been included in the experiments. Four of these datasets have been contaminated with artificial outliers. The SVM that use the AOR hinge error show increased resistance to outliers in multiple contaminated data sets. An idea would be to create other types of outliers in the data, outliers that harm the performance of the absolute hinge error and they should be mimicking realistic situations. It would be interesting to see how the AOR hinge error performs on those datasets. Ultimately, it would be beneficial to know which outliers the AOR hinge error is resistant to.

On the other hand, in the shadow of a potential better forecasting performance of the AOR hinge error lies a time consuming optimization process. The presence of local minima in the loss function requires multiple repetitions of the SVM-Maj algorithm to be performed instead of just one. For each of these repetitions, warm starts are not an option. The computation time adds up accordingly. It might be worth considering an alternative optimization method.

There is plenty of further research that could be done into the AOR hinge error. For example, it would be helpful for researchers and practitioners to know how many random starting points, i.e. repetitions, is economical to choose. Besides that, an efficient method to simultaneously optimize over λ and the threshold t is desirable.

Appendix: Majorizing the hinge errors

Below the specifications of the quadratic majorizing functions for the absolute, quadratic, and Huber hinge errors i.e. expressions for a, b, and c in (8) are presented. These expressions are obtained from Groenen et al. (2008) in which they are derived. Furthermore, the majorizing function of the novel error function f_{AOR} (6) is derived and specified. For convenience of notation y_i and q_i from here onwards will be denoted as y and q dropping the subscript i.

A.1 Majorizing the absolute hinge error

$$a = \frac{1}{4}|1 - y\bar{q}|^{-1} \tag{13}$$

$$b = y(a + \frac{1}{4}) \tag{14}$$

$$c = a + \frac{1}{2} + \frac{1}{4}|1 - y\bar{q}| \tag{15}$$

A.2 Majorizing the quadratic hinge error

$$a = 1 \tag{16}$$

$$b = \begin{cases} \bar{q} & \text{if } y\bar{q} \geq 1 \\ y & \text{if } y\bar{q} < 1 \end{cases} \tag{17}$$

$$c = \begin{cases} 1 - 2(1 - y\bar{q}) + (1 - y\bar{q})^2 & \text{if } y\bar{q} \geq 1 \\ 1 & \text{if } y\bar{q} < 1 \end{cases} \tag{18}$$

A.3 Majorizing the Huber hinge error

$$a = (1/2)(k + 1)^{-}1 \tag{19}$$

$$b = \begin{cases} a\bar{q} & \text{if } y\bar{q} \geq 1 \\ ya & \text{if } -k < y\bar{q} < 1 \\ a\bar{q} + \frac{1}{2}y & \text{if } y\bar{q} \leq -k \end{cases} \tag{20}$$

$$c = \begin{cases} a\bar{q}^2 & \text{if } y\bar{q} \geq 1 \\ a & \text{if } -k < y\bar{q} < 1 \\ 1 - (k + 1)/2 + a\bar{q}^2 & \text{if } y\bar{q} \leq -k \end{cases} \tag{21}$$

A.4 Majorizing the AOR hinge error

In this section of the appendix a majorizing function for the absolute outlier resistant hinge error will be derived. On the part of the domain of f_{AOR} for which $-yq \leq T$, the majorizing function for the AOR hinge error can be chosen to be the same as the majorizing function of the absolute hinge error. This follows from the fact that $f_a \leq f_{\text{AOR}}$ for $q \in \{-\infty, \infty\}$ and $f_a = f_{\text{AOR}}$ for $-yq < T$, see Figure 4.

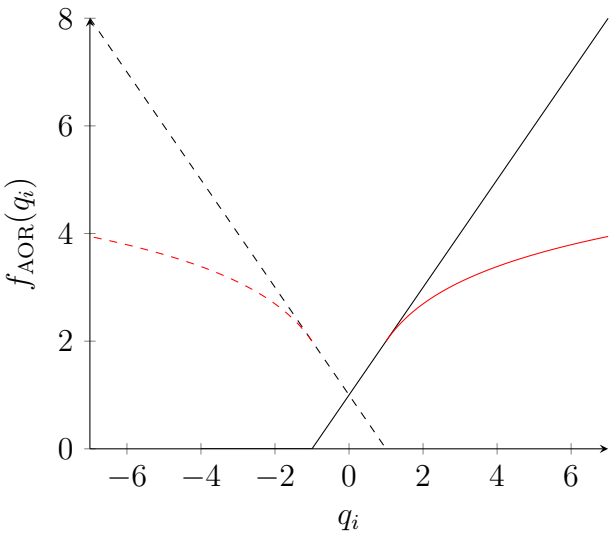


Figure 4: The absolute outlier resistant hinge error function with threshold $T = 1$ and the absolute hinge error function plotted simultaneously.

We are left to define a majorizing function for the part of the domain of f_{AOR} for which $-yq \geq T$. This part of the hinge error functions is defined as

$$e(q) = f_{\text{AOR}}(T) + \ln(1 - yq - T) = T + 1 + \ln(1 - yq - T) \tag{22}$$

and is the represented by the red line in figure 4. To construct a majorizing function the following idea is used. The parabola

$$h(x, \bar{x}) = \frac{\alpha}{4\bar{x}}x^2 + \frac{\alpha}{2}x + \frac{1}{4}\alpha\bar{x} \tag{23}$$

is a majorizing function for the hinged line

$$l(x) = \max(0, \alpha) \tag{24}$$

and touches the function $l(x)$ at the points $\bar{x}$ and $-\bar{x}$. This concept is illustrated in Figure 5.

The above stated idea that (23) is a majorizing function of (24) touching at $\bar{x}$ and $-\bar{x}$ holds for the following reasons:

- $h(\bar{x}, \bar{x}) = l(\bar{x}) = \alpha\bar{x}$
- $h'(\bar{x}, \bar{x}) = l'(\bar{x}) = \alpha$
- $h(-\bar{x}, \bar{x}) = l(-\bar{x}) = 0$
- $h'(-\bar{x}, \bar{x}) = l'(-\bar{x}) = 0$
- $h(x, \bar{x}) \geq l(x)$ which follows from the two observations:
  1. The minimum of the parabola $h(x, \bar{x})$ that opens opens upward is zero by construction, so $h(x, \bar{x}) \geq 0$.
  2. The convex function $h(x, \bar{x})$ touches the concave function αx at $\bar{x}$, so $h(x, \bar{x}) \geq \alpha x$.

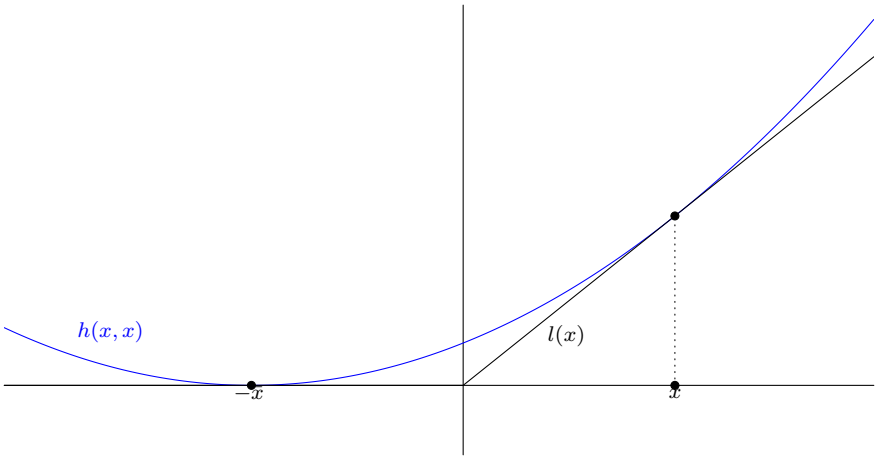


Figure 5: $h(x, \bar{x})$ majorizes $l(x)$, touching $l(x)$ at $\bar{x}$ and $-\bar{x}$.

Note that α and $\bar{x}$ can be smaller than zero in which case the parabola, $h(x, \bar{x})$, and the hinged line, $l(x)$ in Figure 5 would be mirrored in the y-axis.

The idea of majorizing (24) by (23) as illustrated in Figure 5 is used to construct the quadratic majorizing function for $f_{\text{AOR}}(q)$. A tangent line is drawn at $\bar{q}$ creating a similar situation as in Figure 5. From there it is intuitively straightforward to see that the majorization function of $f_{\text{AOR}}(q)$, $g_{\text{AOR}}(q, \bar{q})$, corresponds to the the parabola in Figure 6.

To find $g_{\text{AOR}}(q, \bar{q})$, the quadratic majorizing function, we substitute expressions for α and $\bar{x}$ in (23) and perform a horizontal transformation on the obtained function. α in (23) is substituted by the derivative of the error function at supporting point $\bar{q}$, $f'_{\text{AOR}}(\bar{q})$. $\bar{x}$ is substituted by $\frac{f_{\text{AOR}}(\bar{q})}{f'_{\text{AOR}}(\bar{q})}$ representing the distance from $\bar{q}$ to the point where the tangent line crosses the x-axis. This leaves us with parabola

$$g^*(q, \bar{q}) = a^* q^2 - 2b^* q + c^* \tag{25}$$

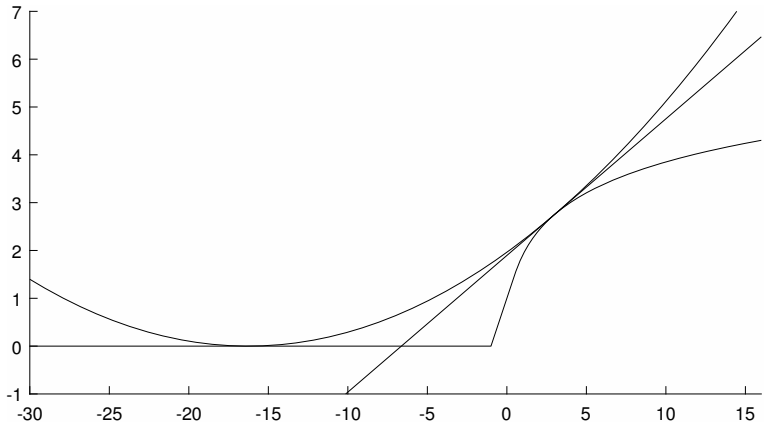


Figure 6: The quadratic majorizing function of f_{AOR} constructed using a tangent line. In this figure $\bar{q} = 3$, $y = -1$ and threshold $T = 0.5$.

where $a^* = \frac{f'_{\text{AOR}}(\bar{q})^2}{4 f_{\text{AOR}}(\bar{q})}$, $b^* = -\frac{f'_{\text{AOR}}(\bar{q})}{4}$, and $c^* = \frac{1}{4} f_{\text{AOR}}(\bar{q})$. Finally g^* is shifted along the x-axis to obtain g_{AOR}. More precisely g^* is shifted $d = \bar{q} - \frac{f_{\text{AOR}}(\bar{q})}{f'_{\text{AOR}}(\bar{q})}$ to the right such that $g_{\text{AOR}}(\bar{q}, \bar{q}) = f_{\text{AOR}}(\bar{q})$. We obtain

$$g_{\text{AOR}}(q, \bar{q}) = aq^2 - 2bq + c = g^*(q - d, \bar{q}) = a^*(q - d)^2 - 2b^*(q - d) + c^*. \tag{26}$$

From (26) it follows that $a = a^*$, $b = a^*d + b^*$, and $c = a^*d^2 + 2b^*d + c^*$. All of the above leads to the result that the majorizing function for f_{AOR} denoted by g_{AOR} is obtained by substituting the below defined a, b, and c into (8).

$$a = \begin{cases} \frac{1}{4}|\bar{q} + 1|^{-1} & \text{if } -y\bar{q} \leq T \\ \frac{1}{4} \frac{f'_{\text{AOR}}(\bar{q})^2}{f_{\text{AOR}}(\bar{q})} & \text{if } -y\bar{q} > T \end{cases} \tag{27}$$

$$b = \begin{cases} y(a + \frac{1}{4}) & \text{if } -y\bar{q} \leq T \\ a\left(\bar{q} - \frac{f_{\text{AOR}}(\bar{q})}{f'_{\text{AOR}}(\bar{q})}\right) - \frac{1}{4} f'_{\text{AOR}}(\bar{q}) & \text{if } -y\bar{q} > T \end{cases} \tag{28}$$

$$c = \begin{cases} a + \frac{1}{2} + \frac{1}{4}|1 - y\bar{q}| & \text{if } -y\bar{q} \leq T \\ a\left(\bar{q} - \frac{f_{\text{AOR}}(\bar{q})}{f'_{\text{AOR}}(\bar{q})}\right)^2 - 2\left(\bar{q} - \frac{f_{\text{AOR}}(\bar{q})}{f'_{\text{AOR}}(\bar{q})}\right) \frac{f'_{\text{AOR}}(\bar{q})}{4} + \frac{1}{4} f_{\text{AOR}}(\bar{q}) & \text{if } -y\bar{q} > T \end{cases} \tag{29}$$

References

Borg, I., & Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media.

Brooks, J. P. (2011). Support vector machines with the ramp loss and the hard margin loss. *Operations research*, *59*(2), 467–479.

Chang, C.-C., & Lin, C.-J. (2006). *LIBSVM a library for support vector machines.* Retrieved from `https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/`

De Leeuw, J. (1994). Block-relaxation algorithms in statistics. In *Information systems and data analysis* (pp. 308–324). Springer.

Groenen, P. J. F., Nalbantov, G., & Bioch, J. C. (2008). SVM-Maj: a majorization approach to linear support vector machines with different hinge errors. *Advances in Data Analysis and Classification*, *2*(1), 17-43.

Heiser, W. J. (1995). Convergent computation by iterative majorization: theory and applications in multidimensional data analysis. *Recent advances in descriptive multivariate analysis*, 157–189.

Hunter, D. R., & Lange, K. (2004). A tutorial on MM algorithms. *The American Statistician*, *58*(1), 30–37.

Kiers, H. A. (2002, 11). Setting up alternating least squares and iterative majorization algorithms for solving various matrix optimization problems. *Computational Statistics and Data Analysis*, *41*(1), 157–170.

Newman, C., Blake, D., & Merz, C. (1998). *UCI repository of machine learning databases.* Retrieved from `http://www.ics.uci.edu/∼mlearn/MLRepository.html`

Steinwart, I., & Christmann, A. (2008). *Support vector machines* (1st ed.). Springer Science & Business Media.