Multiclass Bounded Logistic Regression – Efficient Regularization with Interior Point Method

Ribana Roscher, Wolfgang Förstner

rroscher@uni-bonn.de, wf@ipb.uni-bonn.de

TR-IGG-P-2009-02

July 23, 2009

Technical Report Nr. 2, 2009

Department of Photogrammetry
Institute of Geodesy and Geoinformation
University of Bonn

Available at
http://www.ipb.uni-bonn.de/technicalreports/

Multiclass Bounded Logistic Regression – Efficient Regularization with Interior Point Method

Ribana Roscher, Wolfgang Förstner
rroscher@uni-bonn.de, wf@ipb.uni-bonn.de

Abstract

Logistic regression has been widely used in classification tasks for many years. Its optimization in case of linear separable data has received extensive study due to the problem of a monoton likelihood. This paper presents a new approach, called bounded logistic regression (BLR), by solving the logistic regression as a convex optimization problem with constraints. The paper tests the accuracy of BLR by evaluating nine well-known datasets and compares it to the closely related support vector machine approach (SVM).

1.1 Introduction

Logistic regression makes optimal decisions regarding class labels and at the same time efficiently estimates a posteriori probabilities. It is particularly suitable for a large number of data samples with a relatively low number of features. If the dimension of the feature space is higher than the number of samples, the data is separable and the problem of monotone likelihood occurs, as Albert and Anderson [1] and Santner and Duffy [2] have shown. In this case the standard logistic regression strives towards the global optimum at infinity, which provides no practical solution.

To overcome this problem, a modification of the logistic regression by introducing a regularization term is possible, leading to a unique minimum. Its optimization with the Newton-Raphson procedure is standard and often faster compared to the Gradient Descent procedure. In addition, we do not have to choose a fixed steplength and so we are not confronted with the typical problems of the gradient descent. However, the Newton procedure is only reliable for good starting values near the minimum and if the data points are not separable. Otherwise, the process achieves huge steplengths

and cannot be solved precisely. Therefore in the separable case, the logistic regression with regularization is not suitable in combination with the Newton-Raphson optimization. Related to this idea of regularization, there exist several works e.g. Lin et. al [3], who discuss a Trust Region optimization method and Kim et. al [4], who discuss an Interior Point method. The main question besides the kind of the regularization term is the choice of the regularization parameter, which computation is mostly laborious.

A possible alternative, especially for low sample size with high feature space dimension, is using the support vector machine approach (SVM, Vapnik [5]). In contrast to the logistic regression, the SVM does not provide a posteriori probabilities for the class memberships of the data points. The calculation of these probabilities is possible by extending the SVM to a Relevance Vector Machine (RVM, Tipping [6]).

The following paper presents a new approach to the logistic regression called bounded logistic regression (BLR) as an alternative to the SVM and without any regularization term, being applicable for both separable and non separable data. In section 1.2 we will describe the theoretical background in detail. We will derive the reformulated optimization problem for the logistic regression with additional constraints, before solving it with a customized Newton procedure. Section 1.3 will demonstrate the accuracy of our algorithm by a evaluation of nine well-known datasets from the UCI Machine Learning Repository and a comparison to classification results achieved by the SVM. Finally we will give an outlook to further analyses of the approach.

1.2 Theory

This section provides an efficient algorithm for learning the parameters of a logistic regression model also in case the data a linearly separable. Section 1.2.1 introduces the basic model, for which section 1.2.2 provides the standard solution based on a Newton iteration scheme.

1.2.1 Logistic regression

We assume to have a training set $(\boldsymbol{x}_n, \mathcal{C}_n), n = 1, ..., N$ of N labeled samples with vectors $\boldsymbol{x}_n$ of observations and classes $\mathcal{C}_n \in \{1, ..., K\}$. For classification we use a vector $\boldsymbol{\phi}_n = \boldsymbol{\phi}_n(\boldsymbol{x}_n) \in \mathbb{R}^M$ of M features derived from the observations. The a posteriori probability of a test feature vector $\boldsymbol{\phi} \in \mathbb{R}^M$ is assumed to follow the discriminative multiclass logistic regression model

$$P(\mathcal{C}_k|\boldsymbol{\Phi}, \mathbf{w}) = \frac{\exp(\mathbf{w}_k^\mathsf{T}\boldsymbol{\Phi})}{\sum_j \exp(\mathbf{w}_j^\mathsf{T}\boldsymbol{\Phi})} \tag{1.1}$$

with the extended vectors

$$\boldsymbol{\phi}^\mathsf{T} = [1, \boldsymbol{\phi}^\mathsf{T}] \in \mathbb{R}^{M+1}\,, \qquad \mathbf{w}_k^\mathsf{T} = [w_{k0}, \boldsymbol{w}_k^\mathsf{T}] \in \mathbb{R}^{M+1} \tag{1.2}$$

for the features $\boldsymbol{\phi}$ and the class related parameters $\mathbf{w}_k$ containing a bias w_{k0} and the weight vector $\boldsymbol{w}_k$. The complete parameter vector is $\mathbf{w}^\mathsf{T} = [\mathbf{w}_1^\mathsf{T}, ..., \mathbf{w}_K^\mathsf{T}] \in \mathbb{R}^{K(M+1)}$ collecting the individual parameter vectors.

The model only contains $(K-1)(M+1)$ independent parameters, as the ratio may be reduced e. g. by $\exp(\mathbf{w}_1^\mathsf{T}\boldsymbol{\phi})$ without changing the posteriors, leading to $P(\mathcal{C}_k|\boldsymbol{\phi}, \mathbf{w}) = 1/(1 + \sum_{j=2}^{K} \exp((\mathbf{w}_j - \mathbf{w}_1)^\mathsf{T}\boldsymbol{\phi})$. However, because of its symmetric properties, we prefer (1.1), taking into account that $M+1$ parameters are not identifiable.

The task is to derive optimal parameters $\mathbf{w}$ from the training data.

1.2.2 Basic iteration scheme

The classical procedure is based on a minimizing the negative logarithm of the complete probability

$$\mathcal{Q}(\mathbf{w}) = -\log \prod_n P(\mathcal{C}_n|\boldsymbol{\phi}_n, \mathbf{w}) = -\sum_k \sum_n t_{nk} \log \frac{\exp(\mathbf{w}_k^\mathsf{T}\boldsymbol{\phi}_n)}{\sum_l \exp(\mathbf{w}_l^\mathsf{T}\boldsymbol{\phi}_n)} \tag{1.3}$$

with respect to $\mathbf{w}$. The indicator vector $\boldsymbol{t}_n = [t_{nk}] = \boldsymbol{e}_{\mathcal{C}_n}$ is the $\mathcal{C}_n$-th unit vector. The iteration process needs the gradient and the Hessian

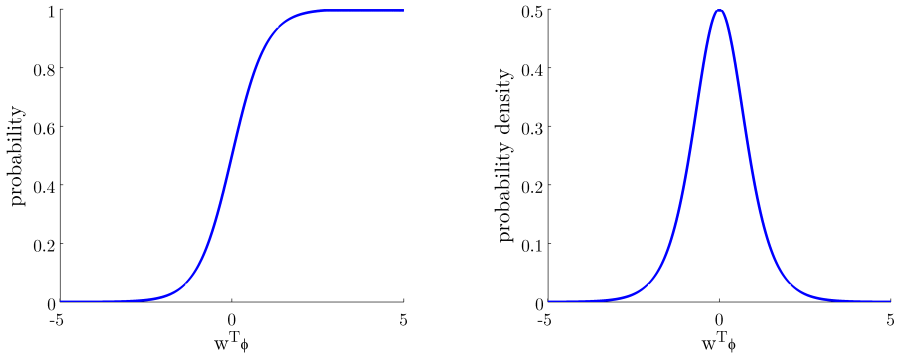
$$\underset{K(M+1)\times 1}{\nabla \mathcal{Q}(\mathbf{w})} = \sum_n (\boldsymbol{t}_n - \boldsymbol{p}_n(\mathbf{w})) \otimes \boldsymbol{\phi}_n \tag{1.4}$$

$$\underset{K(M+1)\times K(M+1)}{\nabla^2 \mathcal{Q}(\mathbf{w})} = -\sum_n \left(\mathrm{Diag}(\boldsymbol{p}_n(\mathbf{w})) - \boldsymbol{p}_n(\mathbf{w})\boldsymbol{p}_n(\mathbf{w})^\mathsf{T}\right) \otimes \boldsymbol{\phi}_n\boldsymbol{\phi}_n^\mathsf{T} \tag{1.5}$$

where the vector $\boldsymbol{p}_n(\mathbf{w}) = [P(\mathcal{C}_k|\boldsymbol{\phi}_n, \mathbf{w})]$ contains the posterior probabilities for the n-th feature vector belonging to class $\mathcal{C}_k$, given the parameters $\mathbf{w}$.

The Hessian is negative semi-definite. It has a rank deficiency of $M+1$, as the left factor of the Kronecker product has rank $K-1$ and eigenvector $\mathbf{1}_K$, reflecting that $M+1$ parameters are not identifiable. Therefore the iteration scheme could be simply $\mathbf{w}^{(i+1)} = \mathbf{w}_{(i)} - [\nabla^2 \mathcal{Q}(\mathbf{w}_{(i)})]^+ \nabla \mathcal{Q}(\mathbf{w}_{(i)})$.

As already discussed in Albert and Anderson [1] the problem has no unique solution in case the data is separable, thus also in the important case that the feature space is larger than the number of training samples. Moreover, the optima sit at infinity. This can easily seen with an example in a one-dimensional feature space with two separable classes: (1) There is an interval (a, b) of finite length for the point x_0 of separation. (2) Setting



(a) Sigmoid function

(b) Derivative of the sigmoid function

Figure 1.1: The relation between the sigmoid function and the Gaussian can be shown by derivation of the sigmoid function to achieve an approximate Gaussian.

$\mathbf{w}_1 = 0$ any $\mathbf{w}_2 = \lim_{\lambda\to\infty}(\lambda[w_{20}, w_{21}]^\mathsf{T})$ with $x_0 = -w_{20}/w_{21} \in (a, b)$ is a solution.

In case of separability, classical Newton-Raphson schemes do not converge. Even so, in order to use this optimization method by reason of fast convergence we will solve the logistic regression as a convex optimization problem with constraints, illustrated in the following.

1.2.3 The optimization problem with constraints

We have observed slow convergence and overflows, due to large weights $|\boldsymbol{w}_k|$'s, also in case of overlapping classes. Therefore we propose to limit the length of the individual weight vectors $\boldsymbol{w}_k$, which is equivalent to limiting the sharpness posterior at the boundaries. That is reasonable, because the weights tends to an unlimited precision if there is no barrier. Like Rennie[7] already mentioned even the regularization with L2-norm assumes unlimited precision, so we have to choose an appropriate barrier to limit the weights. Assuming the feature values ϕ_{nm} have an uncertainty σ of a Gaussian, then the sharpness can be related to the uncertainty measured by $\frac{\pi}{\sqrt{3}w_{max}}$.

Figure 1.1 shows the relation between the sigmoid function and the Gaussian. The derivative of $P(\mathcal{C}|\boldsymbol{\Phi}, w_{max})$ with respect to $\boldsymbol{\phi}$ is

$$\frac{\partial P(\mathcal{C}|\boldsymbol{\Phi}, w_{max})}{\partial \boldsymbol{\phi}} = \boldsymbol{w}\frac{\exp\left(-w_{max}^\mathsf{T}\boldsymbol{\phi} + w_0\right)}{\left(1 + \exp\left(-w_{max}^\mathsf{T}\boldsymbol{\phi} + w_0\right)\right)^2}. \tag{1.6}$$

So the σ of the Gaussian is

$$\sigma = \sqrt{\int \boldsymbol{\phi}^2 \frac{\partial P(\mathcal{C}|\boldsymbol{\Phi}, w_{max})}{\partial \boldsymbol{\phi}} d\boldsymbol{\phi}} = \frac{\pi}{\sqrt{3} w_{max}}, \tag{1.7}$$

which can be reordered to get the sharpness barrier $r = \frac{\pi}{\sqrt{3}\boldsymbol{\sigma}}$.

Therefore we propose to solve the following convex minimization problem with K constraints on the weights for learning the parameters $\mathbf{w}$:

$$\text{minimize} \quad \mathcal{Q}(\mathbf{w}), \tag{1.8}$$

$$\text{subject to} \quad h_k(\mathbf{w}) = |\boldsymbol{w}_k|^2 = \sum_{m=1}^{M} w_{km}^2 \leq r^2 \qquad k = 1, \ldots, K. \tag{1.9}$$

We choose the Interior-point method for solving the problem, because it has been proven to be useful for each kind of optimization. The following remarks are based on the work of Antoniou and Lu [8] as well as Pang and Mangasarian [9].

First of all we convert the inequalities (1.9) into equalities

$$h_k - s_k = -\left(|\boldsymbol{w}_k|^2 - r^2\right) - s_k = 0 \tag{1.10}$$

with $h_k = h(\mathbf{w}_k)$, by introducing slack variables $\{s_k\}$ leading to the nonnegativities

$$s_k \geq 0. \tag{1.11}$$

We can incorporate the contraints into the objective function using a logarithmic barrier function:

$$\text{minimize} \quad \mathcal{Q}_\tau(\mathbf{w}, \boldsymbol{s}) = \mathcal{Q}(\mathbf{w}) - \tau \sum_k \log s_k, \tag{1.12}$$

$$\text{subject to} \quad \boldsymbol{h} - \boldsymbol{s} = \boldsymbol{0}. \tag{1.13}$$

The barrier parameter $\tau \geq 0$ is the weight for the penalization of small slack variables. Smaller values for the slack variables reflect smaller distances to the barrier. Figure 1.2 shows a profile of the height of the penalization by the use of the barrier term $\tau \log\left(-|\boldsymbol{w}_k|^2 + r^2\right)$ in a two-dimensional feature space under variation of τ. Feasible points for the optimization problem lie within a cylinder with radius r, since the barrier term prevents a solution out of its border.

For $\tau \to 0$ the problem (1.12), (1.13) converges to the solution of the original problem (1.8), (1.9). We establish the Lagrangian for (1.12) and (1.13):

$$\mathcal{L}_\tau(\mathbf{w}, \boldsymbol{s}, \boldsymbol{\lambda}) = \mathcal{Q}(\mathbf{w}) - \tau \sum_k \log s_k - \sum_k \lambda_k (h_k - s_k). \tag{1.14}$$

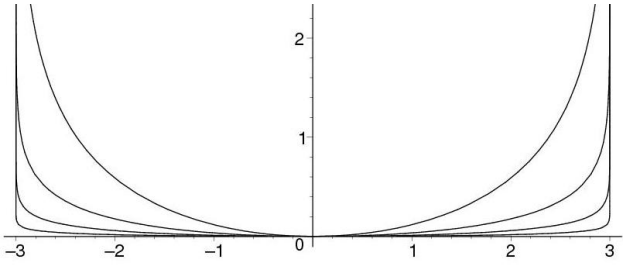


Figure 1.2: The barrier term $\tau \log(-|\boldsymbol{w}_k|^2 + r^2)$ limits the size of $|\boldsymbol{w}_k|$. For large τ the barrier is soft, for values of τ approaching 0 the barrier becomes harder. The plot shows the barrier for $r = 3$ for values $\tau = 1.0,\ 0.3,\ 0.1,\ 0.03$.

The Karush-Kuhn-Tucker conditions are given by

$$\nabla_{\mathbf{w}} \mathcal{L}_\tau = \nabla \mathcal{Q}(\mathbf{w}) - \nabla_w^{\mathsf{T}} \boldsymbol{h} \boldsymbol{\lambda} = \boldsymbol{0}, \tag{1.15}$$

$$\nabla_{\boldsymbol{s}} \mathcal{L}_\tau = -\tau \mathbf{1}_K + \boldsymbol{S} \boldsymbol{\Lambda} \mathbf{1}_K = \boldsymbol{0}, \tag{1.16}$$

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L}_\tau = \boldsymbol{h} - \boldsymbol{s} = \boldsymbol{0}, \tag{1.17}$$

where $\nabla_w^{\mathsf{T}} \boldsymbol{h}$ is the $(M+1)K \times K$ Jacobian of the constraints $\boldsymbol{h}$, $\boldsymbol{S}$ is the $K \times K$-dimensional diagonal matrix of the slack variables , $\boldsymbol{\Lambda}$ is the $K \times K$-dimensional diagonal matrix of the lagrange multipliers and $\mathbf{1}_K$ is a K-vector with all entries equal to one.

Starting from a point $\left\{\mathbf{w}_{(i)}, \boldsymbol{s}_{(i)}, \boldsymbol{\lambda}_{(i)}\right\}$ with $i = 0$ on the convex function (1.14) we can reach the minimum in I iterations by updating the point at each iteration:

$$\mathbf{w}_{(i+1)} = \mathbf{w}_{(i)} + \alpha_{(i)} \boldsymbol{\Delta} \mathbf{w}_{(i)}, \tag{1.18}$$

$$\boldsymbol{s}_{(i+1)} = \boldsymbol{s}_{(i)} + \alpha_{(i)} \boldsymbol{\Delta} \boldsymbol{s}_{(i)}, \tag{1.19}$$

$$\boldsymbol{\lambda}_{(i+1)} = \boldsymbol{\lambda}_{(i)} + \alpha_{(i)} \boldsymbol{\Delta} \boldsymbol{\lambda}_{(i)}. \tag{1.20}$$

We determine $\alpha_{(i)}$ in a line search and choose the Newton-direction for $\left\{\boldsymbol{\Delta} \mathbf{w}_{(i)}, \boldsymbol{\Delta} \boldsymbol{s}_{(i)}, \boldsymbol{\Delta} \boldsymbol{\lambda}_{(i)}\right\}$. Referring to Antoniou and Lu [8] and Pang and Mangasarian [9], the update for the parameters of the decision surfaces is

$$\boldsymbol{\Delta} \mathbf{w}_{(i)} = \boldsymbol{N}_{(i)}^{-1} \left[\nabla \mathcal{Q}_{(i)} - \tau \nabla^{\mathsf{T}} \boldsymbol{h}_{(i)} \boldsymbol{S}_{(i)}^{-1} \mathbf{1}_K + \nabla^{\mathsf{T}} \boldsymbol{h}_{(i)} \boldsymbol{S}_{(i)}^{-1} \boldsymbol{\Lambda}_{(i)} \left(\boldsymbol{s}_{(i)} - \boldsymbol{h}_{(i)} \right) \right] \tag{1.21}$$

with the updates for the slack variables and lagrange multipliers:

$$\begin{aligned} \boldsymbol{\Delta} \boldsymbol{s}_{(i)} = & -\nabla \boldsymbol{h}_{(i)} \boldsymbol{S}_{(i)}^{-1} \nabla \mathcal{Q}_{(i)} + \tau - \nabla \boldsymbol{h}_{(i)} \boldsymbol{N}_{(i)}^{-1} \nabla^{\mathsf{T}} \boldsymbol{h}_{(i)} \boldsymbol{S}_{(i)}^{-1} \mathbf{1}_K \\ & - \left(\boldsymbol{I}_{MK \times MK} - \nabla \boldsymbol{h}_{(i)} \boldsymbol{N}_{(i)}^{-1} \nabla^{\mathsf{T}} \boldsymbol{h}_{(i)} \boldsymbol{S}_{(i)}^{-1} \boldsymbol{\Lambda}_{(i)} \right) \left(\boldsymbol{s}_{(i)} - \boldsymbol{h}_{(i)} \right) \end{aligned} \tag{1.22}$$

$$\boldsymbol{\Delta} \boldsymbol{\lambda}_{(i)} = \boldsymbol{S}_{(i)}^{-1} \boldsymbol{\Lambda}_{(i)} \left(\left(\boldsymbol{s}_{(i)} - \boldsymbol{h}_{(i)} \right) - \nabla \boldsymbol{h}_{(i)} \boldsymbol{\Delta} \mathbf{w}_{(i)} \right) + \tau \boldsymbol{S}_{(i)}^{-1} \mathbf{1}_K - \boldsymbol{\lambda}_{(i)}. \tag{1.23}$$

For solving the equations we need the inverse of

$$N_{(i)} = \nabla^2 \mathcal{Q}_{(i)} - \sum_K \lambda_{(i)k} \nabla^2 h_{(i)k} + \nabla^\mathsf{T} \boldsymbol{h}_{(i)} S_{(i)}^{-1} \boldsymbol{\Lambda}_{(i)} \nabla \boldsymbol{h}_{(i)}. \tag{1.24}$$

The inversion is computational very intensive, so we use the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method to approximate $N_{(i)}$ at every iteration (i). Because we do not explicitly calculate the Hessian from the second derivatives, this approach is a so-called quasi-Newton method. For further information concerning the theory see [10]. We derive the approximated matrix $\widehat{N}_{(i+1)}$ with the help of (1.21) by

$$\widehat{N}_{(i+1)}^{-1} = \widehat{N}_{(i)}^{-1} + \frac{\left(\boldsymbol{\Delta}\mathbf{w}_{(i)} \boldsymbol{\Delta}\mathbf{w}_{(i)}^\mathsf{T}\right)\left(\boldsymbol{\Delta}\mathbf{w}_{(i)}^\mathsf{T} \mathbf{y}_{(i)} + \mathbf{y}_{(i)}^\mathsf{T} \widehat{N}_{(i)}^{-1} \mathbf{y}_{(i)}\right)}{(\boldsymbol{\Delta}\mathbf{w}_{(i)}^\mathsf{T} \mathbf{y}_{(i)})^2} \tag{1.25}$$

$$- \frac{\widehat{N}_{(i)}^{-1} \mathbf{y}_{(i)} \boldsymbol{\Delta}\mathbf{w}_{(i)}^\mathsf{T} + \boldsymbol{\Delta}\mathbf{w}_{(i)} \mathbf{y}_{(i)}^\mathsf{T} \widehat{N}_{(i)}^{-1}}{\boldsymbol{\Delta}\mathbf{w}_{(i)}^\mathsf{T} \mathbf{y}_{(i)}} \tag{1.26}$$

and

$$\begin{aligned} \boldsymbol{y}_{(i)} = &\left[-\nabla \mathcal{Q}_{(i+1)} - \tau \nabla^\mathsf{T} \boldsymbol{h}_{(i+1)} S_{(i+1)}^{-1} \mathbf{1}_K + \nabla^\mathsf{T} \boldsymbol{h}_{(i+1)} S_{(i+1)}^{-1} \boldsymbol{\Lambda}_{(i+1)} \left(\boldsymbol{s}_{(i+1)} - \boldsymbol{h}_{(i+1)}\right)\right] \\ &- \left[-\nabla \mathcal{Q}_{(i)} - \tau \nabla^\mathsf{T} \boldsymbol{h}_{(i)} S_{(i)}^{-1} \mathbf{1}_K + \nabla^\mathsf{T} \boldsymbol{h}_{(i)} S_{(i)}^{-1} \boldsymbol{\Lambda}_{(i)} \left(\boldsymbol{s}_{(i)} - \boldsymbol{h}_{(i)}\right)\right] \end{aligned} \tag{1.27}$$

as the difference of the parenthetic term in (1.21) between iteration (i) and $(i+1)$. We update $\widehat{N}_{(i)}^{-1}$ with an initial choice of $\widehat{N}_0 = 0.01 \cdot I_{MK \times MK}$ until convergence.

Regarding Antoniou and Lu [8] and Pang and Mangasarian [9] for convex optimization problems, which are not quadratic or linear, further reduction of the aforementioned steplength $\alpha_{(i)}$ may be necessary to guarantee convergence. The choice of $\alpha_{(i)}$ is determined through a L_2-merit function

$$\mathcal{Q}_{\tau,\beta}(\mathbf{w}, \boldsymbol{s}) = \mathcal{Q}(\mathbf{w}) - \tau \sum_K \log s_k + \frac{\beta}{2} \|\boldsymbol{s} - \boldsymbol{h}\|^2, \quad \beta \geq 0 \tag{1.28}$$

ensuring that the steplength along the Newton direction is shortened sufficiently.

The merit function is differentiable with respect to $\mathbf{w}$ and $\boldsymbol{s}$. Minimizing (1.28) with large enough β reduces the objective function (1.12) and sets the point closer to the feasible region due to the term $\frac{\beta}{2} \|\boldsymbol{s} - \boldsymbol{h}\|^2$.

The parameter β is set to zero as long as $\{\mathbf{\Delta w}, \mathbf{\Delta} \boldsymbol{s}\}$ is a descent direction for the merit function. This is true if

$$a_{(i)} = \begin{bmatrix} \nabla_{\mathbf{w}} \mathcal{Q}_{\tau,\beta}\left(\mathbf{w}_{(i)}, \boldsymbol{s}_{(i)}\right) \\ \nabla_{\boldsymbol{s}} \mathcal{Q}_{\tau,\beta}\left(\mathbf{w}_{(i)}, \boldsymbol{s}_{(i)}\right) \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mathbf{\Delta w}_{(i)} \\ \mathbf{\Delta} \boldsymbol{s}_{(i)} \end{bmatrix} \tag{1.29}$$

$$= -\boldsymbol{\xi}_{(i)}^{\mathsf{T}} N_{(i)}^{-1} \boldsymbol{\xi}_{(i)} + \tau \mathbf{1}_K^{\mathsf{T}} N_{(i)}^{-1} \left(\boldsymbol{s}_{(i)} - \boldsymbol{h}_{(i)}\right) + \boldsymbol{\xi}_{(i)}^{\mathsf{T}} S_{(i)}^{-1} \nabla^{\mathsf{T}} \boldsymbol{h}_{(i)} S_{(i)}^{-1} \mathbf{\Lambda}_{(i)} \left(\boldsymbol{s}_{(i)} - \boldsymbol{h}_{(i)}\right)$$

$$- \beta \left\| \boldsymbol{s}_{(i)} - \boldsymbol{h}_{(i)} \right\|^2 < 0 \tag{1.30}$$

with $\boldsymbol{\xi}_{(i)} = \nabla \mathcal{Q}_{(i)} - \tau \nabla^{\mathsf{T}} \boldsymbol{h}_{(i)} S_{(i)}^{-1} \mathbf{1}_K$. If this is not the case, β has to assume a value, which ensures that (1.30) is negative. In practice, one usually computes $\beta_{\min}$ and then chooses $\beta = 10\beta_{min}$

The steplength α can be calculated with exact line search using (1.28) and β determined as described above. The search intervall is $[\epsilon, \alpha_{max}]$ with

$$\alpha_{max} = 0.95 \left[\max_{1 \le k \le K} \left(-\frac{\mathbf{\Delta} s_{(i)k}}{s_{(i)k}}, -\frac{\mathbf{\Delta} \lambda_{(i)k}}{\lambda_{(i)k}} \right) \right]^{-1} \tag{1.31}$$

sufficing $\boldsymbol{s}_{(i)} + \alpha_{max} \mathbf{\Delta s}_{(i)} > 0$ and $\boldsymbol{\lambda}_{(i)} + \alpha_{max} \mathbf{\Delta} \boldsymbol{\lambda}_{(i)} > 0$.

For the choice of τ – regardless of the data – a transformation of every datapoint to zero mean and unit variance, e.g. with a principal component analysis, is necessary. Because the problem (1.12) subject to (1.13) converges to the solution of the original problem for $\tau \to 0$, we reduce the barrier parameter in each iteration. We choose the starting value of $\tau_0 = 1$ and reduce τ in every iteration by 90 %.

1.3 Experiments and results

1.3.1 Data and implementations

We choose eight well-known datasets from the UCI (University of California at Irvine) Repository of machine learning databases [11], which makes also available several classification problems from the StatLog collection [12]. From the UCI Repository we use IRIS, WINE, GLASS, VOWEL, WISCONSIN BREAST CANCER and PIMA INDIAN DIABETES and from the StatLog collection we choose the datasets VEHICLE SILHOUETTE [13] and IMAGE SEGMENTATION. Additionally, we use the COLON dataset, a high dimensional dataset with low sample size data, already discussed in Alon et. al [14]. The datasets with their characteristics are collected in table 1.1. We transform all training points to zero mean and unit variance with principal component analysis and use the computed projection matrix P, the mean and the maximum value to

transform the test points in the same way. The COLON dataset is reduced to its centroid and transformed into a range of $[-1, 1]$. In order to determine the classification rate, we conduct a ten-fold crossvalidation and train with nine subsets and test with the one remaining subset. We randomly assign the datapoints to the subsets and report the best and the average crossvalidation rate and their standarddeviation. The experiments have been realized on a Intel Dual Core 3.0 GHz CPU with 8 GB RAM and a 64 bit Windows platform. For comparison we use our BLR approach with linear discrimi-

dataset	# datapoints	# features	# classes
iris	150	4	3
wine	178	13	3
glass	214	13	6
vowel	528	10	11
vehicle	846	18	4
segment	2310	19	7
wisconsin	683	10	2
Pima	768	9	2
colon	62	2000	2

Table 1.1: Problem characteristics

nants and several SVM approaches, whose classification rates are available in the literature. These are in particular: Leave-One-Out Machine (LOOM, Weston [15]), designed output code with SVM (DOC, Crammer and Singer [16]), one-against-all (OAA), one-against-one (OAO) and directed acyclic graph SVM (DAG) with evaluation rates reported in Hsu and Lin [17]. All these approaches use linear kernels. LOOM and DOC are multiclass SVM approaches, while the other approaches are binary classification methods. We also use one support vector machine approach to Decision Trees called Global Tree Optimization SVM (GTO) and a standard SVM implementation with rates reported in Bennett and Winther [18], a SVM implementation with classification results reported in Opper and Winther [19] and a SVM implementation used in Zhang et. al [20].

1.3.2 Results and discussion

Table 1.2 presents the comparison of BLR and SVM. For each dataset, we compute the best and the average classification rate and their standarddeviation of randomly chosen ten-fold crossvalidation sets hundred times and report the best rate. For comparision we choose the best classification rate

	BLR			SVM		
dataset	best	average	σ_{rate}	algorithm	best rate	reference
iris	**97.3**	93.3	1.7	OAO, DAG, LOOM	**97.3**	Hsu and Lin [17]
wine	**99.6**	98.7	0.3	OAO, DOC	99.4	Hsu and Lin [17]
glass	**92.1**	88.6	1.2	OAO	66.4	Hsu and Lin [17]
vowel	64.2	62.0	0.9	OAO	**83.0**	Hsu and Lin [17]
vehicle	80.7	79.6	0.5	DAG	**80.9**	Hsu and Lin [17]
segment	91.9	91.3	0.6	OAO	**96.0**	Hsu and Lin [17]
Wisconsin	**97.2**	96.7	0.2	SVM	97.0	Opper and Winther [19]
Pima	**78.5**	77.3	2.0	SVM	77.6	Bennett and Blue [18]
colon	**78.8**	67.9	3.9	SVM	78.1	Zhang et. al [20]

Table 1.2: A comparison of classification rates between BLR and different SVM approaches (best rates bold-faced)

of SVM, which is reported in the literature. In all references the best crossvalidation rate is reported, so we also compare the best achieved result.

In case of low feature dimension we choose the Hessian and not the approximation with the BFGS algorithm for optimization. We choose an appropriate uncertainty for the data, which prevents overfitting but also gives a sufficient discrimination, because their is no such information about the data. Since the normalization of the features also influence σ, this value is transformed with $\widetilde{\sigma} = P\sigma I_{N\times N}P^{\mathsf{T}}$ with P as the projection matrix computed from the training data. The restricting of the sharpness of the weights is done with $\widetilde{\sigma}$. It is necessary to determine the σ before the normalization to preserve the isotropic character of the uncertainty of the data.

The results in table 1.2 show that the BLR performs well with the datasets and can handle separable and non separable datasets. The classification rates are often as well as for the several SVM implementations. Lim and Loh [21] compared error rates of 33 classification algorithms, especially decision trees, statistical and neural network algorithms and evaluated among other the datasets Wisconsin breast cancer, Pima Indian diabetes, StatLog image segmentation and StatLog vehicle silhouette due to a ten-fold crossvalidation. Lim and Loh [21] report classification rates between $[91.5, 97.2]$ for the Wisconson breast cancer dataset, rates between $[69.0, 77.9]$ for the Pima Indian diabetes dataset and rates between $[51.3, 85.5]$ for the StatLog vehicle silhouette dataset. We achieved the best result for the first two dataset and the third best result for the last dataset.

The table also shows that the BLR is a stable algorithm, because the standarddeviation is mostly slow.

1.4 Conclusions

We suggested a new approach to the logistic regression, which handles both separable and non separable data. By introducing a barrier for the sharpness of the weights we could avoid overfitting by preventing the boundaries assuming unlimited precision. In case of known precision σ for the feature values we can bound the values for the weights and achieve an exact regularization, because the weights will never become more precise than the feature values. In the separable case the weights will converge against the barrier.

The proposed approach showed good performance on standard datasets from the UCI Repository compared to nine well-known SVM implementations. A future work is to compare the L2-regularized logistic regression with assumed Gaussian prior, mentioned in Rennie([7]) and our approach by the usage of uncertain data e.g. features from images. We will further obtain a fast implementation and to test the effiency of the BLR approach and analyse the influence of the number of datapoints, classes and features and the data character to it and compare it to the SVM and the L2-regularized logistic regression. We will also extend the BLR by the usage of different kernels.

Bibliography

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

[2] Santner, T., Duffy, D.: A note on A. Albert and JA Anderson's conditions for the existence of maximum likelihood estimates in logistic regression models. Biometrika **73**(3) (1986) 755–758

[3] Lin, C., Weng, R., Keerthi, S.: Trust region Newton method for logistic regression. The Journal of Machine Learning Research **9** (2008) 627–650

[4] Kim, S., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D.: An interior-point method for large-scale l1-regularized least squares. Selected Topics in Signal Processing. IEEE Journal of **1**(4) (2007) 606–617

[5] Vapnik, V.: The nature of statistical learning theory. Springer (2000)

[6] Tipping, M.: Relevance vector machine (October 14 2003) US Patent 6,633,857.

[7] Rennie, J.D.M.: On L2-norm regularization and the Gaussian prior. http://people.csail.mit.edu/jrennie/writing (May 2003)

[8] Antoniou, A., Lu, W.: Practical optimization: algorithms and engineering applications. Springer (2007)

[9] Pang, J., Mangasarian, O.: Computational Optimization: A Tribute to Olvi Mangasarian. Springer (1999)

[10] Nocedal, J., Wright, S.: Numerical Optimization. Springer (1999)

[11] Asuncion, A., Newman, D.J.: UCI machine learning repository (2007)

[12] Michie, D., Spiegelhalter, D.J., Taylor, C.C., Campbell, J., eds.: Machine learning, neural and statistical classification. Ellis Horwood, Upper Saddle River, NJ, USA (1994)

[13] Setiono, R., Leow, W.K.: Vehicle recognition using rule based methods. Turing Institute Research Memorandum TIRM-87-018 **121** (1987)

[14] Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences **96**(12) (1999) 6745–6750

[15] Weston, J., Watkins, C.: Multi-class support vector machines technical report. Technical report, CSD-TR-98-04, 1998 (1998)

[16] Crammer, K., Singer, Y.: On the learnability and design of output codes for multiclass problems. Machine Learning **47**(2) (2002) 201–233

[17] Hsu, C., Lin, C.: A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks **13**(2) (2002) 415–425

[18] Bennett, K.P., Blue, J.A.: A support vector machine approach to decision trees. In: Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on. Volume 3. (1998)

[19] Opper, M., Winther, O.: Gaussian process classification and SVM: Mean field results and leave-one-out estimator. In Smola, A.J., Bartlett, P., Schlkopf, B., Schuurmans, D., eds.: Advances in large margin classifiers, MIT Press (2000) 311–326

[20] Zhang, C., Fu, H., Jiang, Y., Yu, T.: High-dimensional pseudo-logistic regression and classification with applications to gene expression data. Computational Statistics and Data Analysis **52**(1) (2007) 452–470

[21] Lim, T., Loh, W., Shih, Y.: A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning **40**(3) (2000) 203–228