Artificial Intelligence

UNI FREIBURG

Albert-Ludwigs-Universität Freiburg

Thorsten Schmidt

Abteilung für Mathematische Stochastik

www.stochastik.uni-freiburg.de
thorsten.schmidt@stochastik.uni-freiburg.de
SS 2017

Our goal today

Generalized linear models

- Regularization
- LASSO
- Logistic regression

Maximum-likelihood

Support vector machines

- The statistical classification problem
  - Support Vector Classifier
  - The kernel trick

Literature (incomplete, but growing):

- Ian Goodfellow, Yoshua Bengio und Aaron Courville (2016). **Deep Learning**. http://www.deeplearningbook.org. MIT Press
- D. Barber (2012). **Bayesian Reasoning and Machine Learning**. Cambridge University Press
- Richard S. Sutton und Andrew G. Barto (1998). **Reinforcement Learning : An Introduction**. MIT Press
- Gareth James u. a. (2014). **An Introduction to Statistical Learning: With Applications in R**. Springer Publishing Company, Incorporated. ISBN: 1461471370, 9781461471370
- T. Hastie, R. Tibshirani und J. Friedman (2009). **The Elements of Statistical Learning**. Springer Series in Statistics. Springer New York Inc. URL: https://statweb.stanford.edu/~tibs/ElemStatLearn/

Generalized Linear Models

We already saw that transforming the input variables suitable might be helpful. This is the idea of a generalized linear model (GLM), see Casella & Berger (2002).

Definition

A GLM consists of three components:

1. Response variables (random) $Y_1, \dots, Y_n$,
2. a systematic component of the form $\alpha + \boldsymbol{\beta}^\top \boldsymbol{x}_i$, $i = 1, \dots, n$,
3. a link function g satisfying

$$\mathbb{E}[Y_i] = g(\alpha + \boldsymbol{\beta} \boldsymbol{x}_i), \quad i = 1, \dots, n.$$

Regularization of multiple linear regression

- One problem in practice is parsimony of a linear regression: suppose you have many covariates and you want to include only those which are relevant.
- It would be possible to iteratively throw out those parameters which are not significant. This procedure, however is not optimal. Many others have been proposed.
- We concentrate on **continuous** subset selection methods: it is better to introduce a penalty for including two many parameters, which we call regularization. This is moreover a standard procedure for ill-posed problems. We will consider a famous example: the **LASSO** introduced in Robert Tibshirani (1996). „Regression Shrinkage and Selection via the Lasso“. In: **Journal of the Royal Statistical Society. Series B (Methodological)** 58.1, S. 267–288.

LASSO

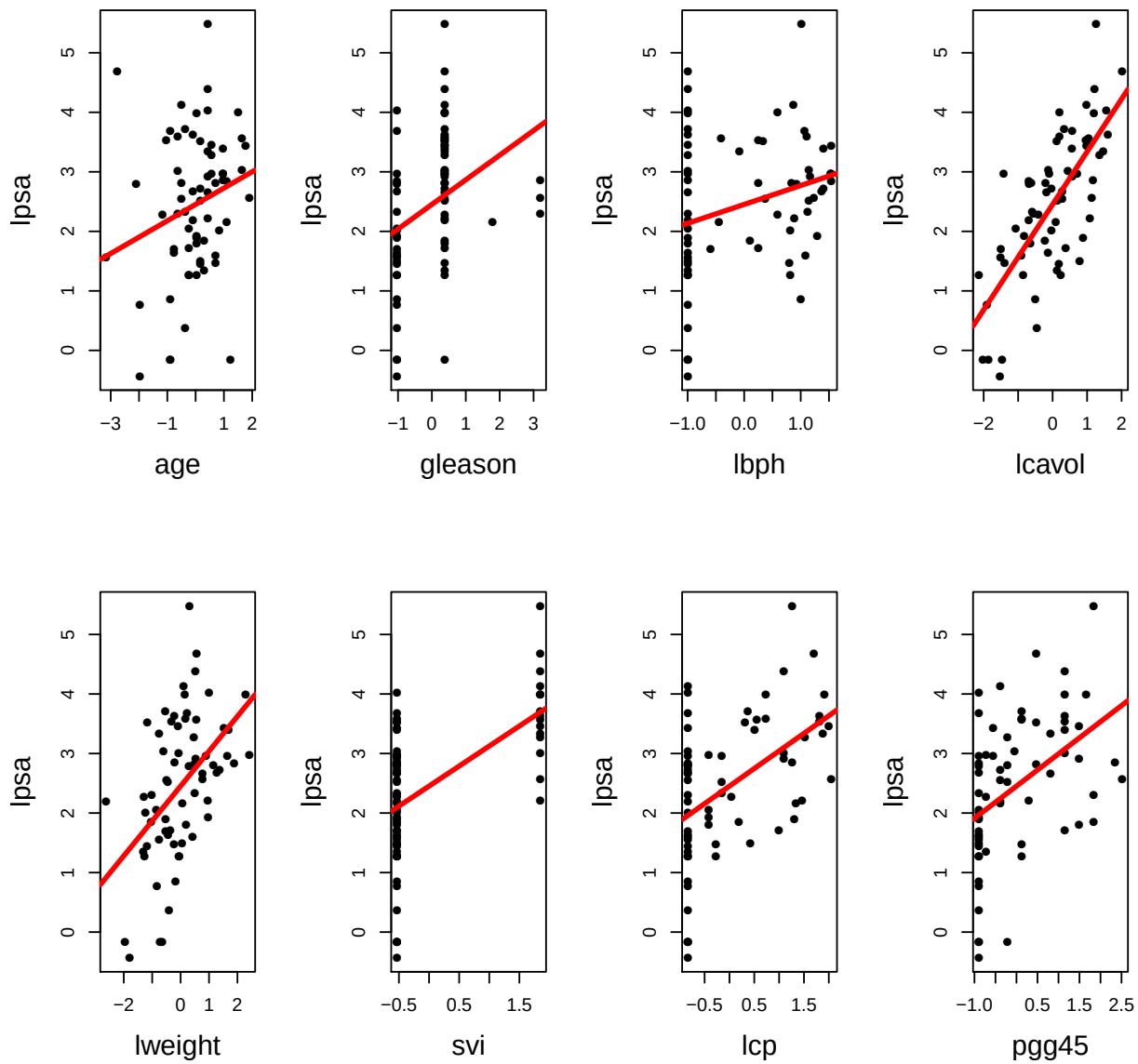
- The **least absolute shrinkage and selection operator** minimizes the following function

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ \frac{1}{n} \| \boldsymbol{Y} - \boldsymbol{x}\boldsymbol{\beta} \|_2^2 + \lambda \| \boldsymbol{\beta} \|_1 \right\}.$$

  The parameter λ has to be chosen and allows to vary the level of regularization. Clearly this model prefers to set non-significant parameters to zero.
- Let us illustrate the lasso with an example taken from Chris Franck, `http://www.lisa.stat.vt.edu/?q=node/5969`. The data stems from Stamey et.al.[1].
- The data describes clinical measures from 97 men about to undergo radical prostatectomy. It is of interest to estimate the relation between the clinical measures and the prostate specific antigen (measures are: lcavol - log (cancer volume), lweight - log(prostate weight volume), age, lbph - log (benign prostatic hyperplasia), svi - seminal vesicle invasion, lcp - log(capsular penetration), Gleason (score), ppg45 - percent Gleason scores 4 or 5, $Y =$lpsa - log(prostate specific antigen))

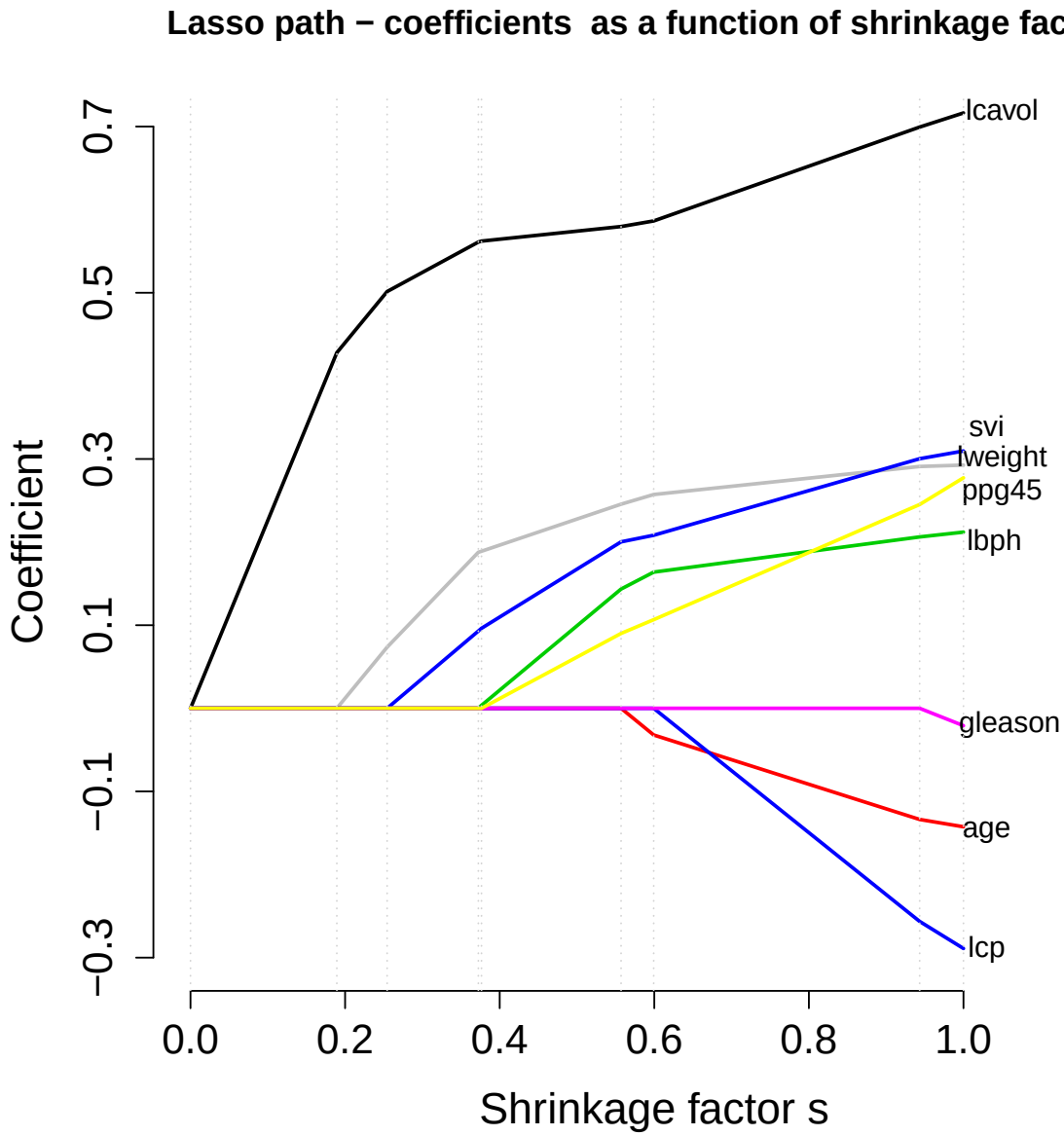
[1] Thomas A Stamey u. a. (1989). „Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. II. Radical prostatectomy treated patients.“ In: **The Journal of urology** 141.5, S. 1076–1083.

We start by examining bi-variate regressions.

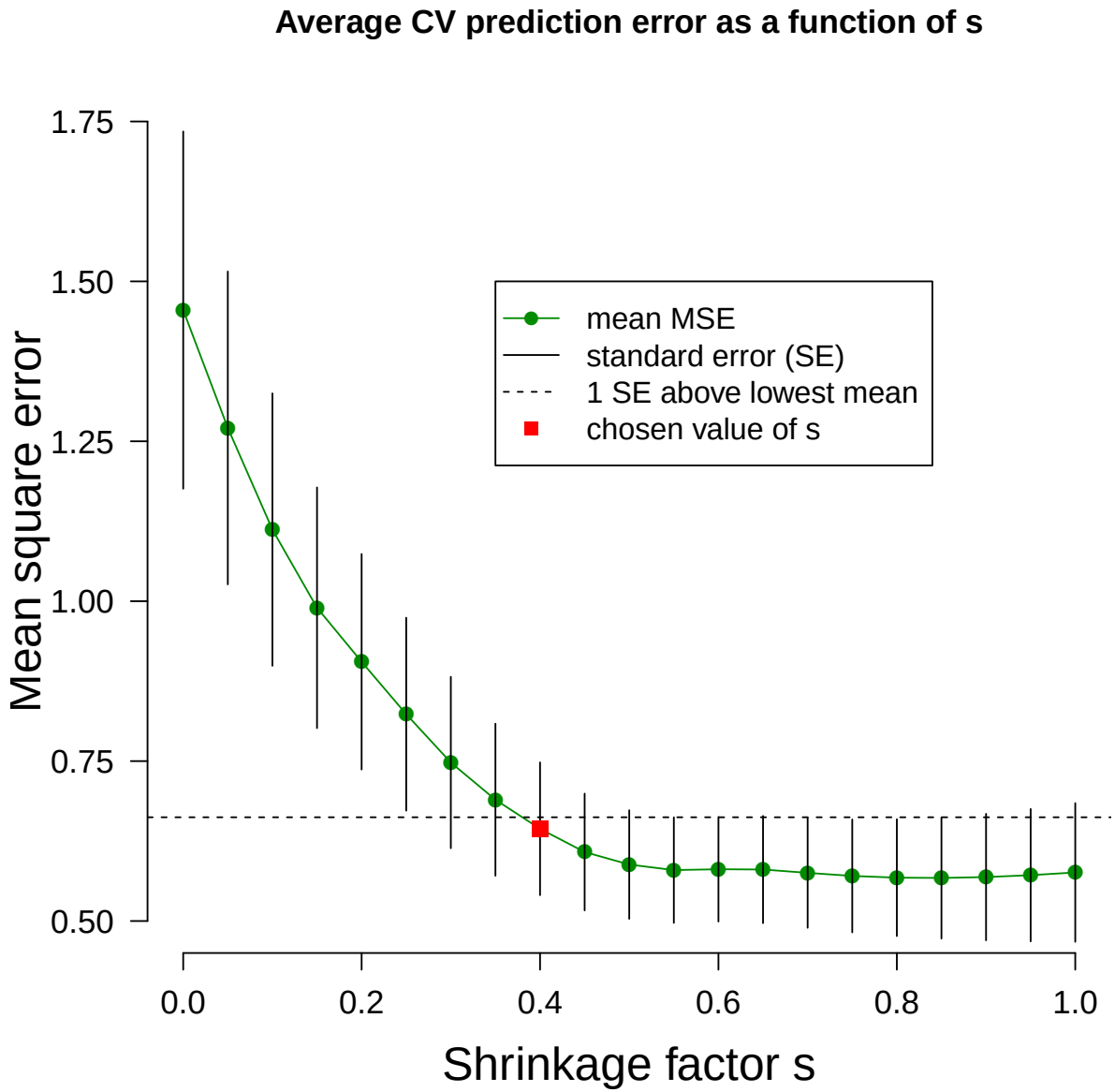


- It is obvious that some variables have fewer impact and some others seem to be more important. The question is how to effectively select those.
- We illustrate how cross-validation may be used in this case. This means we separate the data into a training set and a validation set. The tuning parameter λ is chosen based on the training set and validated on the validation set.
- We use a 10-fold cross validation, ie. the set is split into 10 pieces. Iteratively, each piece is chosen as the validation set while the remaining 9 sets are used to estimate the model.

This is the so-called lasso path. The shrinkage factor is antiproportional to λ.



This is the cross-validation result. A rule of thumb is to select that value of s that is within 1 standard error of the lowest value.



- We see that the optimal choice of λ is far from trivial. Alternative approaches are at hand, compare the recent results by Johannes Lederer and coauthors, J. Lederer und C. Müller (2014). „Don't Fall for Tuning Parameters: Tuning-Free Variable Selection in High Dimensions With the TREX“. In: **ArXiv e-prints**. eprint: 1404.0541 (stat.ME).

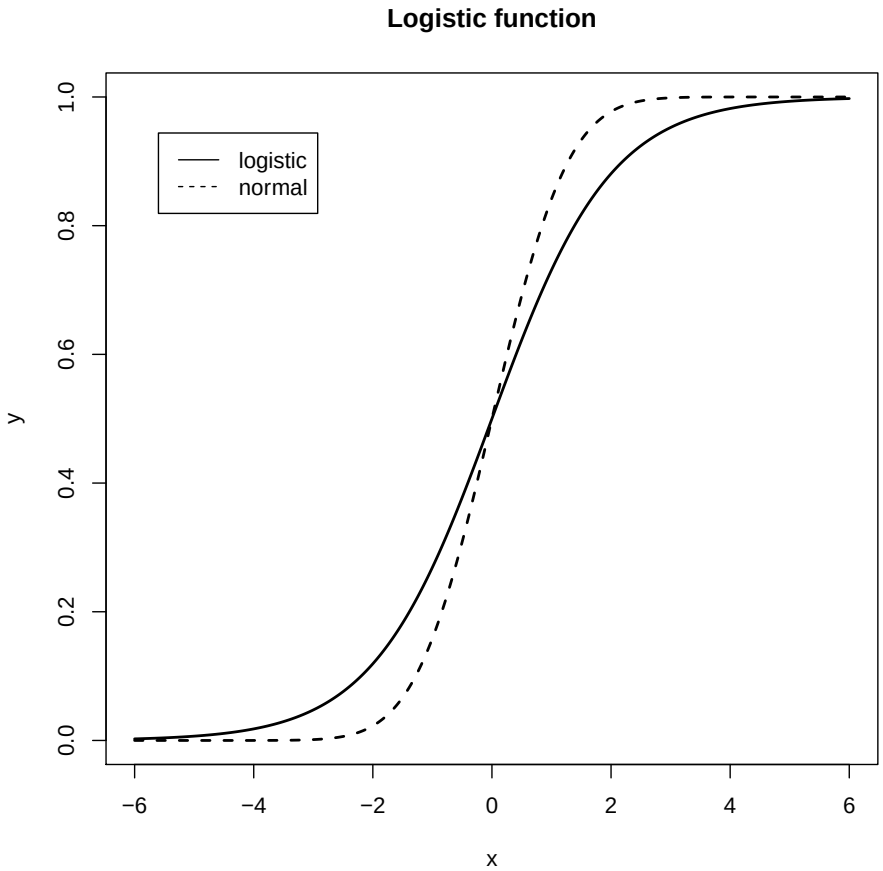
Logistic regression

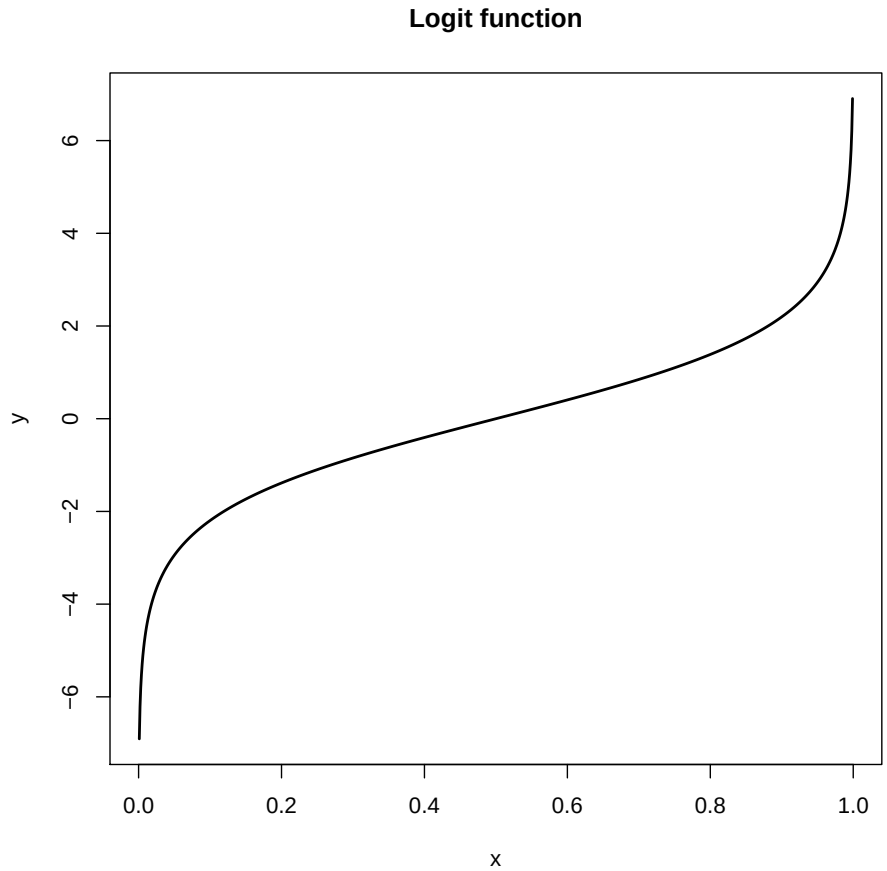
One important regression approach for classification is logistic regression. In this case, the response is always binary. One therefore needs to transform the whole real line to $[0,1]$ and two approaches are common: first, via the logistic function

$$\sigma(x) = \frac{e^x}{1+e^x},$$

(leading to $g = \sigma^{-1}$, the so-called **logit** function) by a cumulative distribution function (when this is Φ - standard normal - this approach is called **probit** model).

The most common estimation method used is **maximum-likelihood**. We take a small detour towards this exciting statistical concept going back to Sir Ronald Fisher.


Logistic function
logistic
normal
y
x
0.0
0.2
0.4
0.6
0.8
1.0
−6
−4
−2
0
2
4
6


Logit function
y
x
−6
−4
−2
0
2
4
6
0.0
0.2
0.4
0.6
0.8
1.0

Maximum-likelihood

- A **statistical model** is given by a family of probability measures $(P_\theta)_{\theta\in\Theta}$ on a common measurable space $(\Omega,\mathscr{F})$. It is typically called **parametric**, if Θ is of finite dimension.
- The **likelihood**-function for the observation E is given by

$$L(\theta) = P_\theta(E)$$

If $P_\theta(E) = 0$ for all $\theta \in \Theta$ one proceeds via the density: assume $P_\theta \ll P^*$ for all $\theta \in \Theta$ and denote the densities by $f_\theta := {}^{dP_\theta}/_{dP^*}$. Then, for the observation x,

$$L(\theta) = f_\theta(x).$$

- This looks complicated, but is in most cases quite simple: consider i.i.d. random variables $X_1,\dots,X_n$ with common density f_θ. Then P^* is clearly the Lebesgue-measure. Due to the i.i.d.-property,

$$L(\theta) = \prod_{i=1}^{n} f_\theta(x_i).$$

Definition

Any maximizer $\hat{\theta}$ of the likelihood-function is called maximum-likelihood estimator for the model $(P_\theta)_{\theta\in\Theta}$.

In the above example, we need to maximize $\prod_{i=1}^n f_\theta(x_i)$, which is typically infeasible. One therefore considers the log-likelihood function

$$\ell(\theta) := \ln L(\theta)$$

which is often much easier to maximize. Typically one can apply first-order conditions or needs to solve numerically.

Example (ML for the normal distribution)

Consider $X_i \sim \mathcal{N}(\mu, 1)$. Then the density is

$$f_\theta(x) = \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{1}{2}(x-\mu)^2 \Big).$$

We obtain the log-likelihood function

$$l(\theta) = \text{const.} - \frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^2.$$

The first derivative is

$$\partial_\mu l(\theta) = \sum_{i=1}^{n} x_i - n\mu \stackrel{!}{=} 0$$

and we obtain the maximum-likelihood estimator (second derivative is < 0)

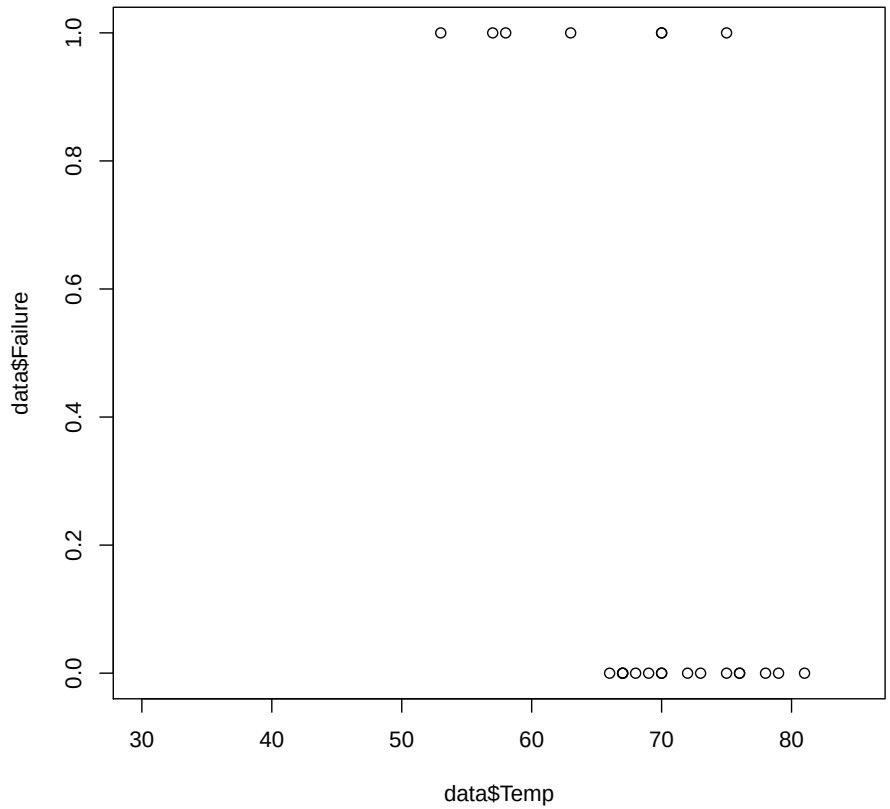
$$\hat{\mu} = \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}.$$

Exercise: compute the ML estimator for σ ! Read Czado & Schmidt (2011) on ML-estimation and further estimation procedures.

Back to logistic regression. We look at the by now infamous Challenger O-ring data set (taken from Caslla & Berger (2002))

1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0
53	57	58	63	66	67	67	67	68	69	70	70	70	70	72	73	75	75	76	76

The table reports failures with associated temperature.



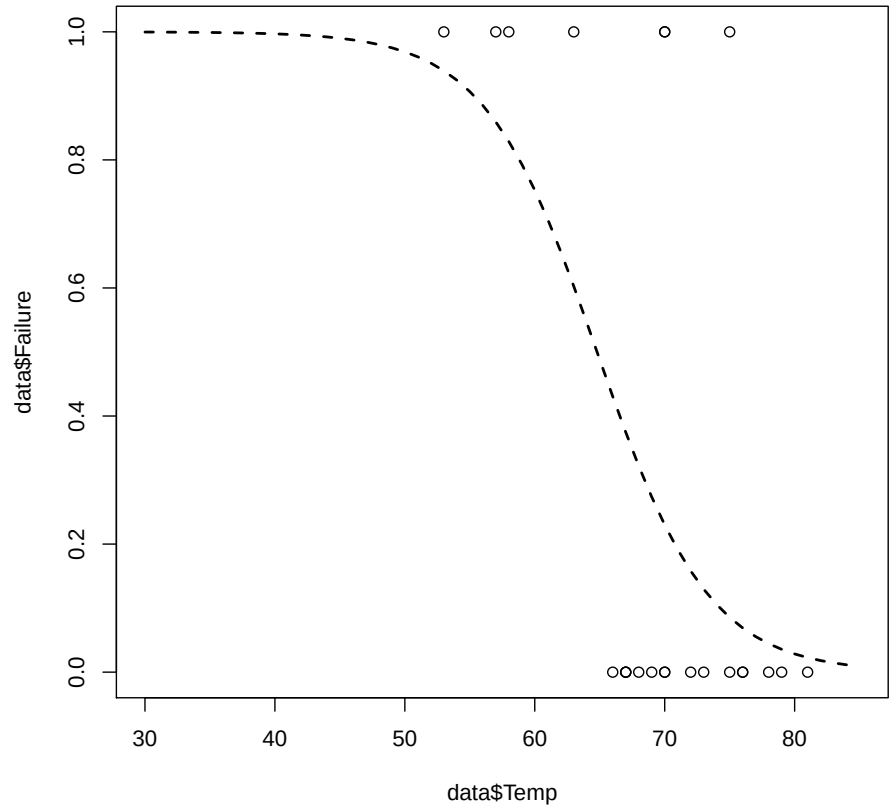
```
library(gdata,quietly=TRUE,verbose=FALSE, warn.conflicts=FALSE)  # for reading xls
data = read.xls("ChallengerData.xls") # Taken From Casella & Berger (2002)
plot (data$Temp, data$Failure,xlim=c(30,85))

summary(out.int <- glm(Failure ~ Temp, family=binomial , data = data))

a= out.int$coefficients[1]
b= out.int$coefficients[2]
x=seq(30,85,by=1)
lines(x,exp(a+b*x)/(1+exp(a+b*x)))

x=31; exp(a+b*x)/(1+exp(a+b*x))
```

The estimated probability
for a failure at 31° is 0.9996088.

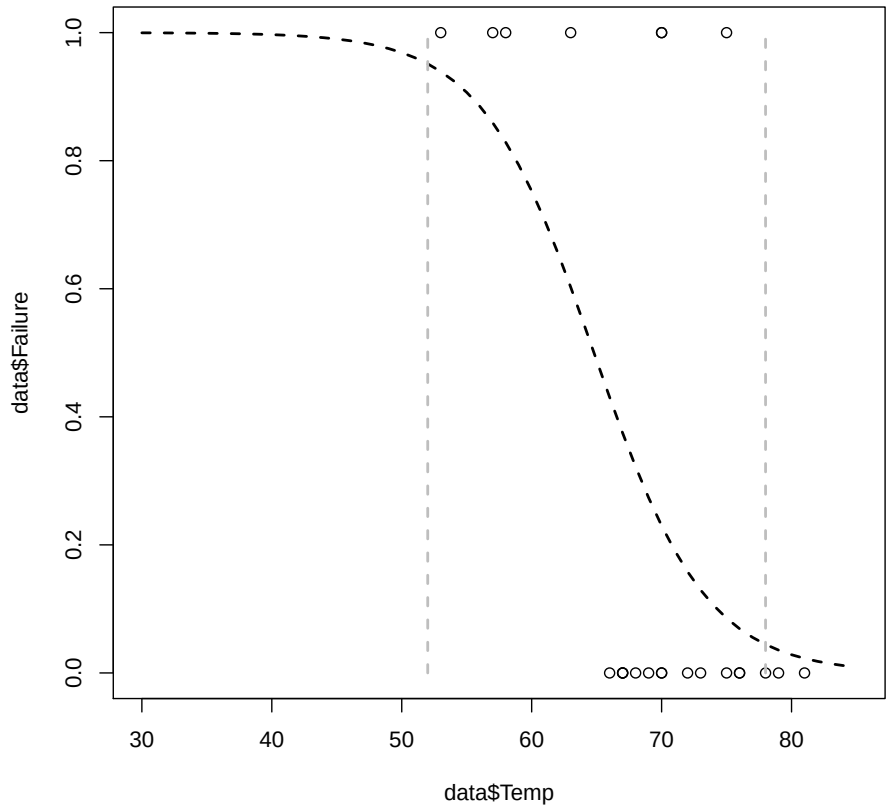


- Logistic regression naturally classifies the data into two fields: the ones with probability above 0.5, where we would optimally decide for outcome one and the ones with probability below 0.5, where we would decide for outcome 0.
- Hence, we obtain a **decision boundary**, given by the hyperplane

$$\alpha + \boldsymbol{\beta}\boldsymbol{x} = 0.$$

- If the decision boundary separates the two groups, then the data is called **linearly separable.** Note that this is can not be achieved in the Challenger dataset.
- Note that the logistic regression also provides probabilities of false decisions: at the boundary this is 50/50, but further out the probability of a false decision decrease. **Significant** decisions requires the probability of a false decision to be below a significance level, e.g. $\alpha = 0.05$ or $\alpha = 0.01$.

With significance level $\alpha = 0.05$ obtained decision boundaries.



Load the R example[2] from the homepage and revisit the above steps. Try your own examples.

[2] Called LogisticRegression.R

- The likelihood-function has to be maximized numerically.
- A first-order iterative scheme is the **gradient-descent** algorithm. Look this algorithm up and recall its properties and functionality.

Support vector machines

- The first example of a tool we visit but which is not typical for classical statistics is **Support Vector Machines**.
- For the introduction we mainly follow Hastie et. al. (2009) and Steinwart & Covel[3].

[3] I. Steinwart und C. Scovel (2007). „Fast rates for support vector machines using Gaussian kernels“. In: **Ann. Statist.** 35.2, S. 575–607.

- We start by formally introducing the **statistical classification problem**.
- We have a finite training set

$$T = ((x_1, y_1), \dots, (x_n, y_n)) \in (X \times Y)^n,$$

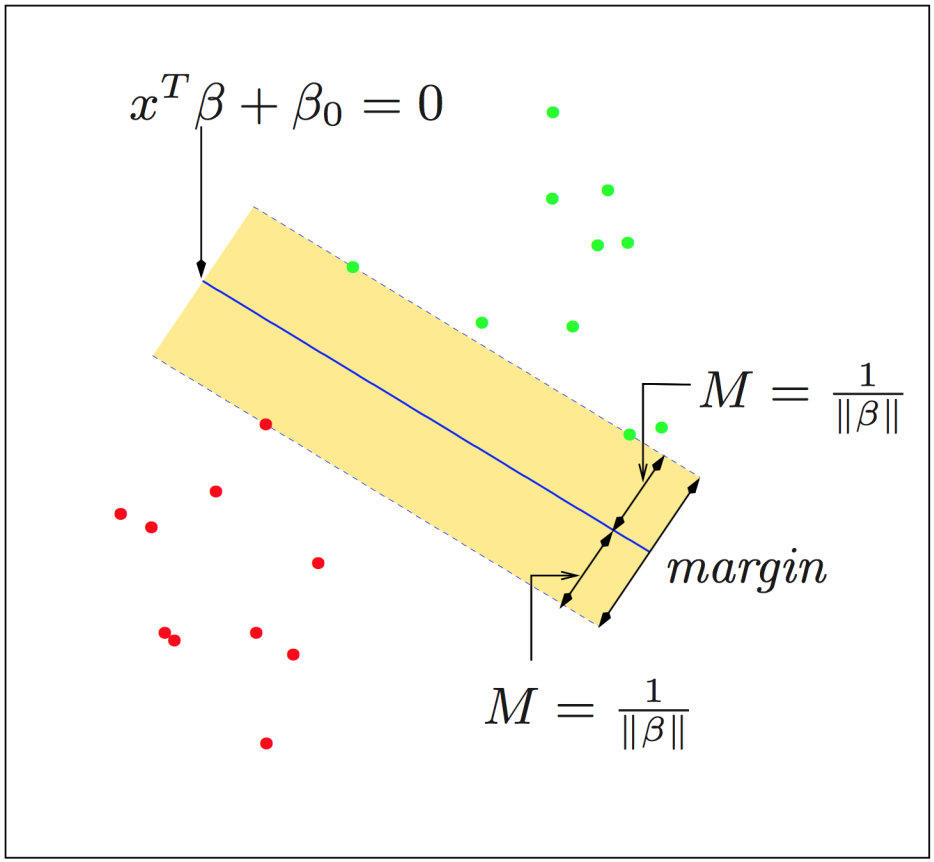
  where $X \subset \mathbb{R}^d$ and $Y = \{-1, 1\}$.
- The standard **batch model** assumes that the samples $(x_i, y_i)_{1 \le i \le n}$ are i.i.d. according to an unknown probability measure P on $X \times Y$. Furthermore, a new sample (x, y) is drawn from P independently of T.
- A **classifier** $\mathscr{C}$ assigns to every T a measurable function $f = f_T : X \to \mathbb{R}$.
- The prediction of $\mathscr{C}$ for y is

$$\operatorname{sign} f(x)$$

  with the convention $\operatorname{sign}(0) := 1$.
- We measure the quality of the classification f by the **classification risk**

$$\mathscr{R}(f) := P(\{(x, y) : \operatorname{sign} f(x) \neq y\}).$$

- Clearly, it is the goal to achieve the smallest possible risk, the so-called **Bayes risk**
$$\mathscr{R}_P := \inf\{\mathscr{R}(f) | f : X \to \mathbb{R} \text{ measurable}\}.$$
- A function which attains this level is called a **Bayes decision function**.
- Let us start with an illustrative introduction to SVM (Pictures taken from Hastie et.al. (2009))



Consider the hyperplane described by $x^\top \beta + \beta_0 = 0$, with $\| \beta \| = 1$ and the classification $\mathcal{G}$:

$$\text{sign}(x^\top \beta + \beta_0).$$

The maximal margin is obtained by the following optimization problem

$$\max_{\beta, \beta_0 : \|\beta\| = 1} M$$

subject to $y_i(x_i^\top \beta + \beta_0) \geq M, \ i = 1, \ldots, n$

- Such classifiers, computing a linear combination of the input and returning the sign were called **perceptrons** in the late 1950s (Rosenblatt, 1958) and set the foundations for later models of neural networks in the 80s and the 90s.
- We reformulate this criterion as follows: first, we get rid of $\| \beta \| = 1$ by considering

$$\frac{1}{\| \beta' \|} y_i (x_i^\top \beta' + \beta_0') \geq M$$

or, equivalently

$$y_i (x_i^\top \beta' + \beta_0') \geq M \| \beta' \|.$$

- With β', β_0' satisfying these equations, any (positive) multiple will also satisfy these, we rescale to $\| \beta' \| = M^{-1}$ and arrive at

$$\begin{aligned} &\min_{\beta, \beta_0} \frac{1}{2} \| \beta \|^2 \\ &\text{subject to } y_i (x_i^\top \beta + \beta_0) \geq 1, \ i = 1, \dots, n. \end{aligned} \tag{1}$$

- This is a convex optimization problem and can be solved via the classical Karush-Kuhn-Tucker conditions.
- It should be noted that the solution does only depend on a small amount of the data, and hence has a certain kind of robustness. On the other side, it will possibly not be optimal under additional information on the underlying distribution.

Non-linearly separable data

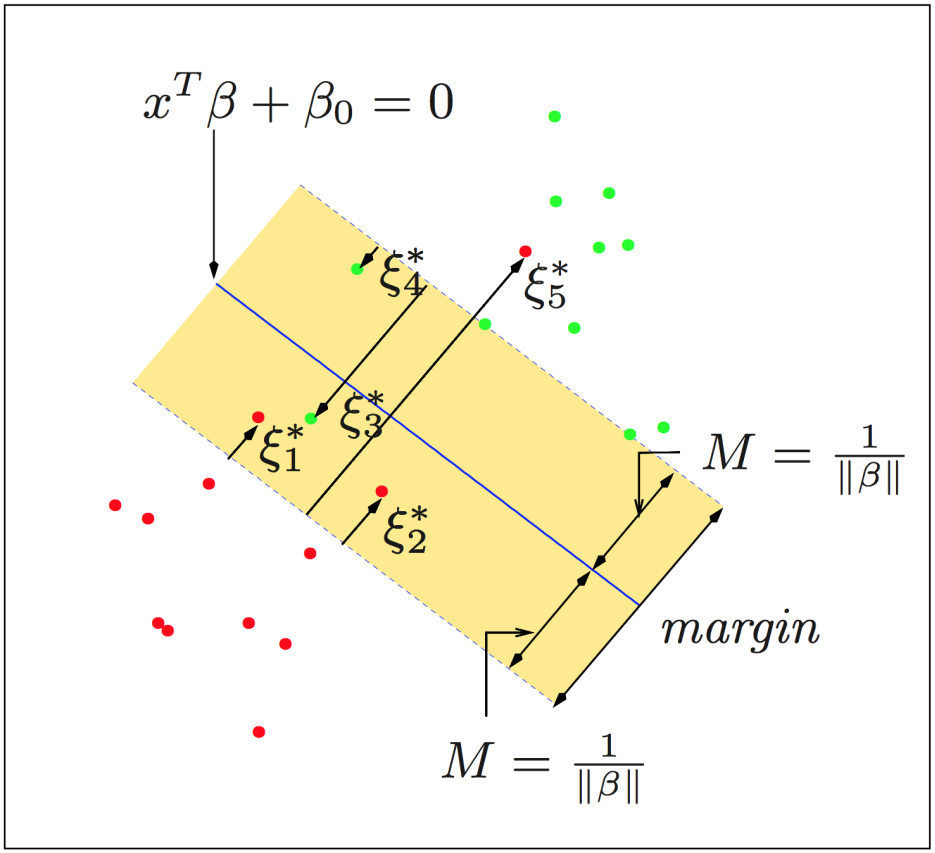
- When the data does not separate fully we will allow some points to be on the wrong side.
- In this regard, define the **slack variables** $\xi_1, \dots, \xi_n$ and consider

$$y_i(x_i^\top \beta + \beta_0) \geq M - \xi_i \tag{2}$$

or

$$y_i(x_i^\top \beta + \beta_0) \geq M(1 - \xi_i) \tag{3}$$

  with $\xi_i \geq 0$, $\sum \xi_i \leq K$ with a constant K.
- The first conditions seems more natural, while the second choice measures the overlap in relative distance, which chances with the width of the margin, M. However, (2) leads to a non-convex optimization problem. The second problem is convex and is the ”standard” support vector classifier.



The case for data which is not fully linearly separable.
Misclassification occurs when $\xi_i > 1$.

- Summarizing we arrive at the following minimization problem (again choosing $\| \beta \| = M^{-1}$) for the **support vector classifier**.

$$\min \| \beta \| \text{ subject to } \begin{cases} y_i(x_i^\top \beta + \beta_0) \geq (1 - \xi_i), \forall i \\ \xi_i \geq 0, \sum \xi_i \leq K. \end{cases}$$

- For a detailed description how to solve this convex optimization problem see Section 12.2.1 in Hastie et. al. (2009).

The kernel trick

- It seems quite restrictive to consider only linear classification rules. We have already seen that in linear regression we were able to overcome this problem by a suitable transformation of the data. This can also be achieved here and is often called **the kernel trick**.
- We first give a rather informative introduction and thereafter discuss the mathematical properties.