Ridge regression

Wessel van Wieringen
w.n.van.wieringen@vu.nl

Department of Epidemiology and Biostatistics, VUmc
& Department of Mathematics, VU University
Amsterdam, The Netherlands

vrije Universiteit

VU medisch centrum

Preliminary

Scribed lecture

Van Wieringen, W.N. (2018), *Lecture notes on ridge regression,* arXiv:1509.09169.

Assumption

The data are zero-centered variate-wise.

Hence, the response and the expression data of each gene is centered around zero.

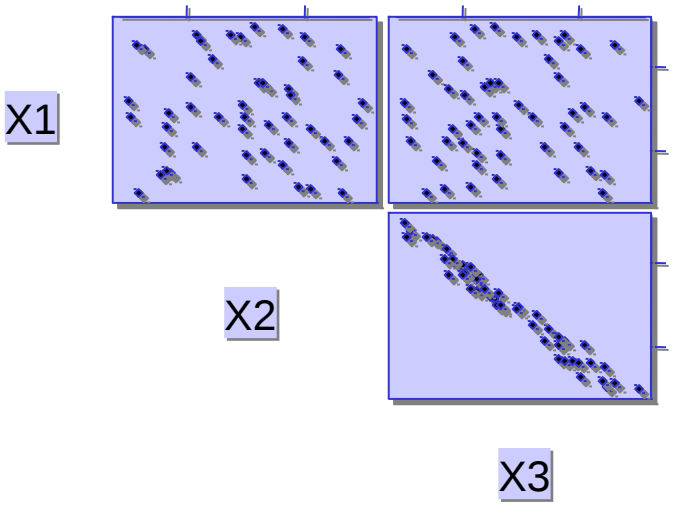
That is, X_{ij} replaced by $X_{ij} - \hat{\mu}_j$ where

$$\hat{\mu}_j = \frac{1}{n}\sum_{i=1}^{n} X_{ij}$$

Problem

Collinearity
Two (or multiple) covariates are highly linearly related.

Consequence
High standard error of estimates.




```
The regression equation is
Y = 0.126 + 0.437 X1 + 1.09 X2 + 0.937 X3

Predictor         Coef      SE Coef           T         P
Constant        0.1257       0.4565        0.28     0.784
X1             0.43731      0.05550        7.88     0.000
X2              1.0871       0.3399        3.20     0.003
X3              0.9373       0.6865        1.37     0.179
```

Problem

Supercollinearity

Two (or multiple) covariates are fully linearly dependent.

Example:

$$\mathbf{X} = \begin{pmatrix} 1 & -1 & 2 \\ 1 & 0 & 1 \\ 1 & 2 & -1 \\ 1 & 1 & 0 \end{pmatrix}$$

The columns are dependent: C1 = C2 + C3.

Consequence : singular $\mathbf{X}^\top \mathbf{X}$.

A square matrix with no inverse is called *singular.*
A matrix $\mathbf{A}$ is singular iff det($\mathbf{A}$) = 0.

Problem

Supercollinearity

Example:

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$$

As det($\mathbf{A}$) = 0, $\mathbf{A}$ is singular and its inverse is undefined.

Det(**A**) equals the product of the eigenvalues θ_j of **A**:
the matrix **A** is singular if any eigenvalue of **A** is zero.

To see this, consider the spectral decomposition of **A**:

$$\mathbf{A} = \sum_{j=1}^{p} \theta_j \mathbf{v}_j \mathbf{v}_j^\top$$

where $\mathbf{v}_j$ is the eigenvector belonging to θ_j.

Problem

Supercollinearity

The inverse of **A** is then:

$$\mathbf{A}^{-1} = \sum_{j=1}^{p} \theta_j^{-1} \mathbf{v}_j \mathbf{v}_j^{\top}$$

A has eigenvalues 5 and 0. The inverse of **A** via the spectral decomposition is then undefined:

$$\mathbf{A}^{-1} = \frac{1}{5} \mathbf{v}_1 \mathbf{v}_1^{\top} + \frac{1}{0} \mathbf{v}_1 \mathbf{v}_1^{\top}$$

Even `R` says no:

```
> A <- matrix(c(1,2,2,4), ncol=2)
> Ainv <- solve(A)
Error in solve.default(A) :
  Lapack routine dgesv: system is exactly singular
```

Problem

Supercollinearity

Consequence : singular $\mathbf{X}^\top\mathbf{X}$.

So?

Recall the ML regression estimator (and its variance):

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$$

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$$

These are only defined if $(\mathbf{X}^\top\mathbf{X})^{-1}$ exists.

Supercollinearity → ML regression estimator undefined.

Supercollinearity occurs *high-dimensionally*, i.e. when the number of covariates exceeds the number of samples ($p > n$).

Ridge regression

Ridge regression

Problem

In case of singular $\mathbf{X}^T\mathbf{X}$ its inverse $(\mathbf{X}^T\mathbf{X})^{-1}$ is not defined. Consequently, the OLS estimator

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$$

does not exist. This happens in high-dimensional data.

Solution

An *ad-hoc* solution adds $\lambda\mathbf{I}$ to $\mathbf{X}^T\mathbf{X}$, leading to:

$$\hat{\boldsymbol{\beta}}(\lambda) = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{X}^\top\mathbf{Y}$$

This is called the *ridge estimator*.

Ridge regression

Example

Let:

$$\mathbf{X} = \begin{pmatrix} 1 & -1 & 2 \\ 1 & 0 & 1 \\ 1 & 2 & -1 \\ 1 & 1 & 0 \end{pmatrix} \quad \text{then} \quad \mathbf{X}^T\mathbf{X} = \begin{pmatrix} 4 & 2 & 2 \\ 2 & 6 & -4 \\ 2 & -4 & 6 \end{pmatrix}$$

which has eigenvalues equal to 10, 6 and 0.

With the “ridge-fix”, we get e.g.:

$$\mathbf{X}^T\mathbf{X} + \mathbf{I} = \begin{pmatrix} 5 & 2 & 2 \\ 2 & 7 & -4 \\ 2 & -4 & 7 \end{pmatrix}$$

which has eigenvalues equal to 11, 7 and 1.

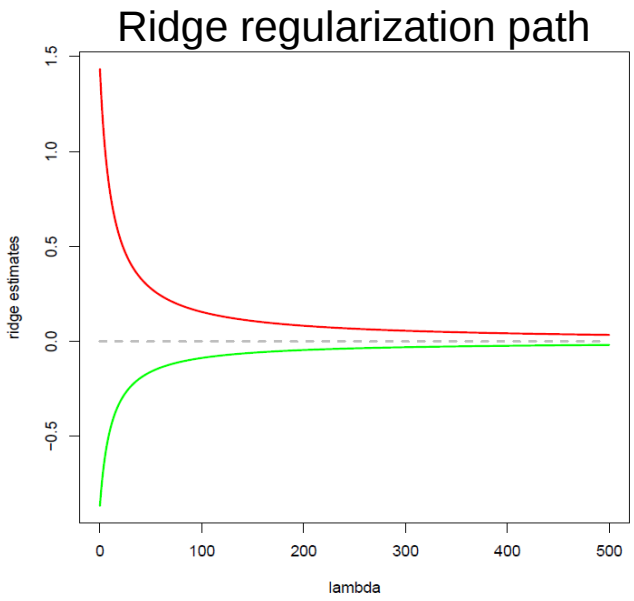
Ridge regression

Example (continued)

Suppose now that $\mathbf{Y} = (1.3, -0.5, 2.6, 0.9)^\top$.

For every choice of λ, we have a ridge estimate of the coefficients of the regression equation: $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$.

$$\lambda = 1:$$
$$\hat{\boldsymbol{\beta}}(\lambda) = (0.614, 0.548, 0.066)^\top$$
$$\lambda = 10:$$
$$\hat{\boldsymbol{\beta}}(\lambda) = (0.269, 0.267, 0.002)^\top$$



Question

Does ridge estimate always tend to zero as λ tends to infinity?

Ridge regression

Ridge vs. OLS estimator

The columns of the matrix $\mathbf{X}$ are *orthonormal* if the columns are orthogonal and have a unit length.

Orthonormality of the design matrix implies:

$$\mathbf{X}^T\mathbf{X} = \mathbf{I} = (\mathbf{X}^T\mathbf{X})^{-1}$$

Then, there is a simple relation between the ridge estimator and the OLS estimator:

$$\hat{\boldsymbol{\beta}}(\lambda) = (1+\lambda)^{-1}\hat{\boldsymbol{\beta}}$$

Ridge regression

Why does the ad hoc fix work?
Study its effect from the perspective of singular values.

Use the *singular value decomposition* of matrix **X**:

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

to rewrite:

$$\hat{\boldsymbol{\beta}}(\lambda) = \mathbf{V}(\mathbf{D}^2 + \lambda \mathbf{I}_{pp})^{-1}\mathbf{D}\mathbf{U}^\top\mathbf{Y}$$

and:

role of singular values

$$\hat{\boldsymbol{\beta}} = \mathbf{V}\ \mathbf{D}^{-2}\ \mathbf{D}\mathbf{U}^\top\mathbf{Y}$$

Ridge regression

Why does the ad hoc fix work?

Combine the two results and write $(\mathbf{D})_{jj} = d_{jj}$ to obtain:

$$d_{jj}^{-1} \geq d_{jj}/(d_{jj}^2 + \lambda)$$

d_{jj}^{-1}: OLS

$d_{jj}/(d_{jj}^2 + \lambda)$: ridge

Thus, the ridge estimator shrinks the singular values of **X**.

Return to the problem of super-collinearity: $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is singular but $\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I}$ is not. Its inverse is:

$$(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I})^{-1} = \sum_{j=1}^{p} (d_{jj}^2 + \lambda)^{-1} \mathbf{v}_j \mathbf{v}_j^{\mathrm{T}}$$

$(d_{jj}^2 + \lambda)^{-1}$: non-zero

Ridge regression

Contrast to principal component regression

Let $\mathbf{Z}_k = \mathbf{X}\mathbf{V}_k$ contain the 1st *k* principal components.

PC regression then fits: $\mathbf{Y} = \mathbf{Z}_k\boldsymbol{\gamma} + \boldsymbol{\varepsilon}$

The least squares estimate gives:

$$\hat{\boldsymbol{\gamma}} = (\mathbf{Z}_k^\top \mathbf{Z}_k)^{-1}\mathbf{Z}_k^\top \mathbf{Y}$$

Translated to the linear regression model:

$$\hat{\boldsymbol{\beta}}_{\mathrm{pcr}} = \mathbf{V}_k\hat{\boldsymbol{\gamma}}$$

this gives:

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{\mathrm{pcr}} &= \mathbf{V}_x(\mathbf{I}_{nk}\mathbf{D}_x\mathbf{I}_{kn})^{-1}\mathbf{U}_x^\top\mathbf{Y},\\ \hat{\boldsymbol{\beta}}(\lambda) &= \mathbf{V}_x(\mathbf{D}_x^2 + \lambda\mathbf{I}_{nn})^{-1}\mathbf{D}_x\mathbf{U}_x^\top\mathbf{Y}.\end{aligned}$$

I.e. thresholding vs. shrinkage.

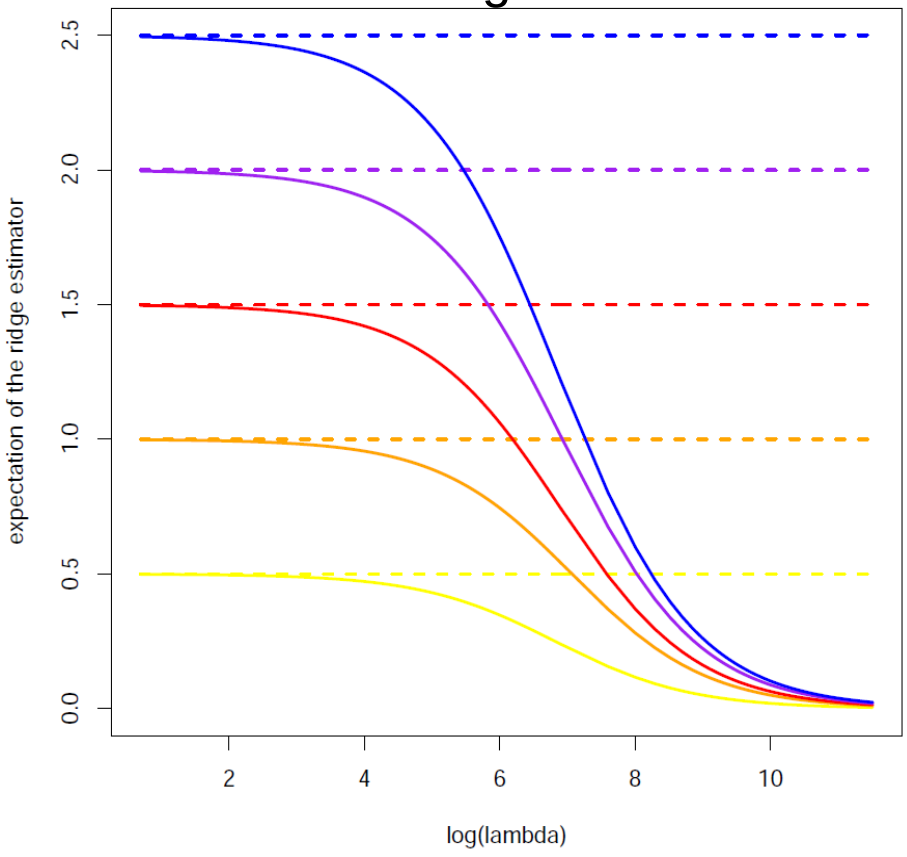
Moments of the ridge estimator

Moments

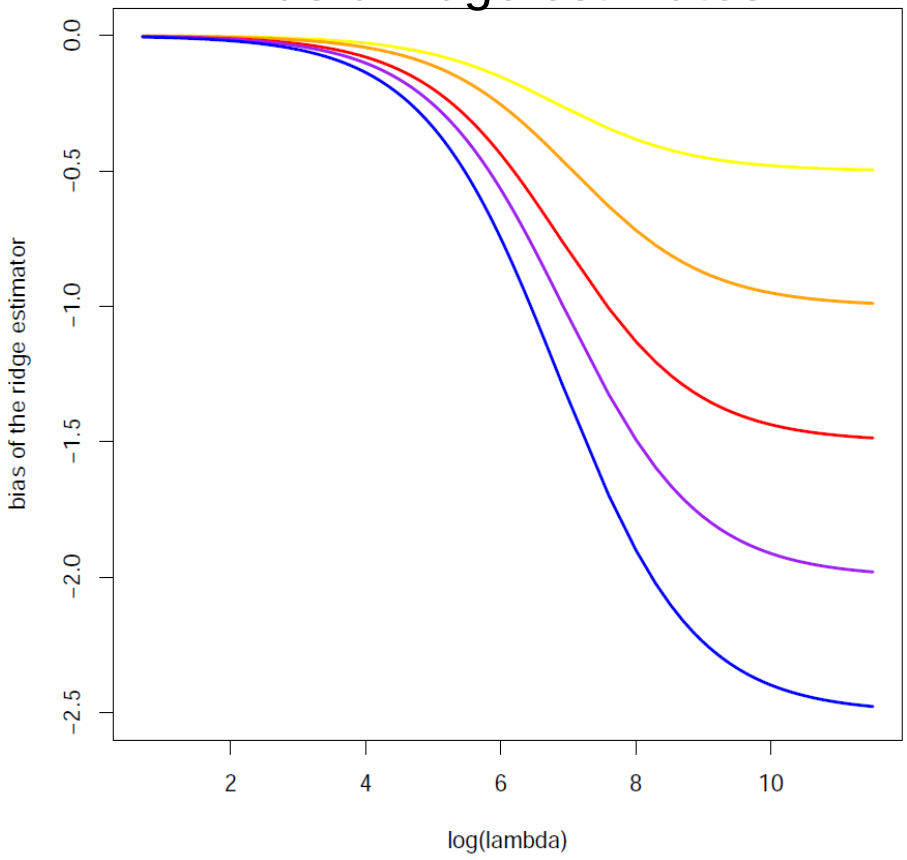
The expectation of the ridge estimator:

$$\mathbb{E}[\hat{\boldsymbol{\beta}}(\lambda)] = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} \neq \boldsymbol{\beta}$$

OLS and ridge estimates



Bias of ridge estimates

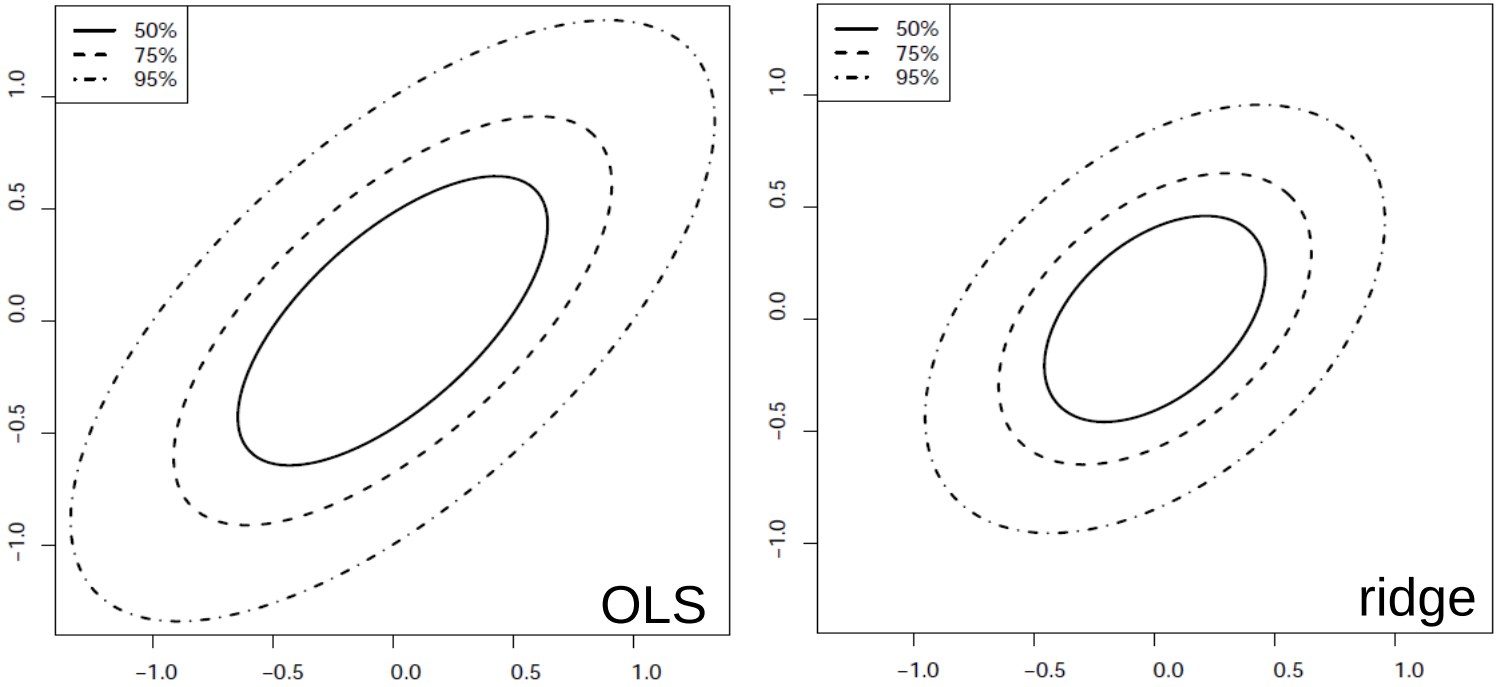


Moments

Define: $\mathbf{W}_\lambda = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{X}^\top\mathbf{X}$. Then: $\hat{\boldsymbol{\beta}}(\lambda) = \mathbf{W}_\lambda\hat{\boldsymbol{\beta}}$ and variance of the ridge estimator becomes:

$$\mathrm{Var}[\hat{\boldsymbol{\beta}}(\lambda)] = \sigma^2\mathbf{W}_\lambda(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{W}_\lambda^\top$$

Consequence: $\mathrm{Var}[\hat{\boldsymbol{\beta}}] \succeq \mathrm{Var}[\hat{\boldsymbol{\beta}}(\lambda)]$. Translated to the levels of the distribution of both estimators:



Moments

Question

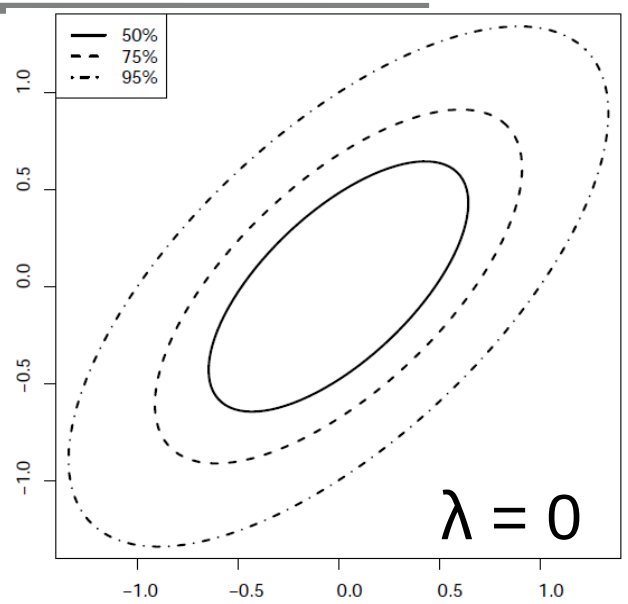
Prove that the ellipsoid level sets of the distribution of the ridge estimator are indeed smaller than that of the OLS.

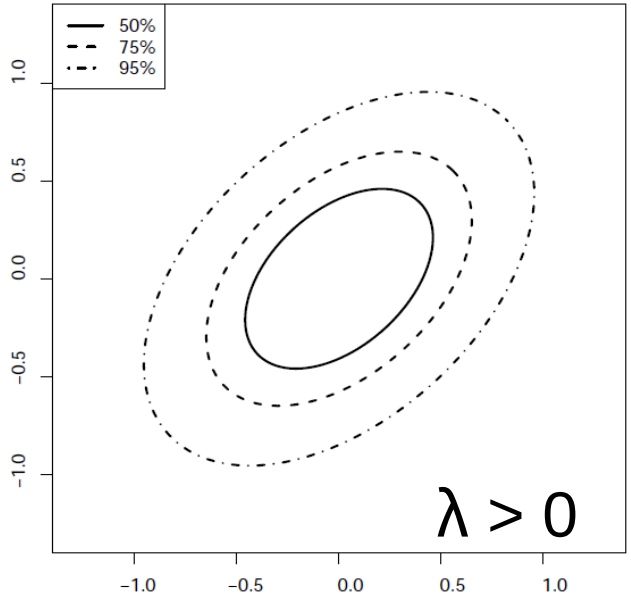
Hints

→ Express determinant in terms of eigenvalues.

→ Write:

$$\mathbf{X}^\top \mathbf{X} = \mathbf{V}_x \mathbf{D}_x^2 \mathbf{V}_x^\top$$





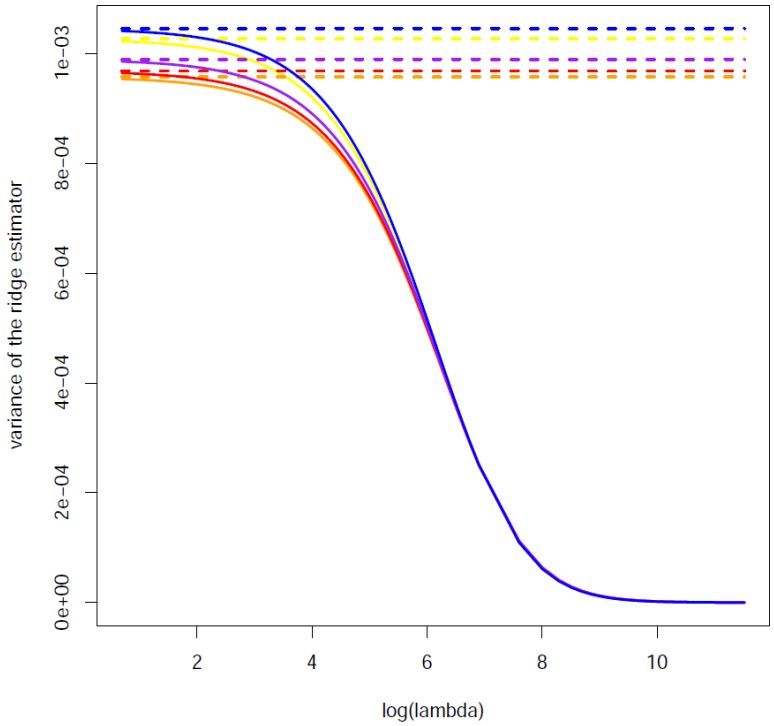
Moments

Ridge vs. OLS estimator

In the orthonormal case:

$$\mathrm{Var}[\hat{\boldsymbol{\beta}} \quad] = \sigma^2 \qquad\qquad \mathbf{I}_{pp},$$
$$\mathrm{Var}[\hat{\boldsymbol{\beta}}(\lambda)] = \sigma^2(1+\lambda)^{-1}\mathbf{I}_{pp}.$$

As the penalty parameter is non-negative, the former exceeds the latter.



Moments

Distribution

The distribution of the ridge estimator is:

$$\hat{\boldsymbol{\beta}}(\lambda) \sim \mathcal{N}[(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{W}_\lambda(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{W}_\lambda^\top]$$

Question

Why is the estimator normally distributed?

Question

Why can we not use this distribution for testing:

$$H_0 : \beta_j = 0$$

Mean squared error

Previous motivation for the ridge estimator:

→ Ad hoc solution to collinearity.

An alternative motivation: comes from studying the *Mean Squared Error (MSE)* of the ridge regression estimator.

In general, for any estimator of a parameter $\boldsymbol{\mu}$:

$$\begin{aligned} \mathrm{MSE}(\hat{\boldsymbol{\mu}}) &= E[(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})^2] \\ &= \mathrm{Var}(\hat{\boldsymbol{\mu}}) + [\mathrm{Bias}(\hat{\boldsymbol{\mu}})]^2 \end{aligned}$$

Hence, the MSE is a measure of the quality of the estimator.

Mean squared error

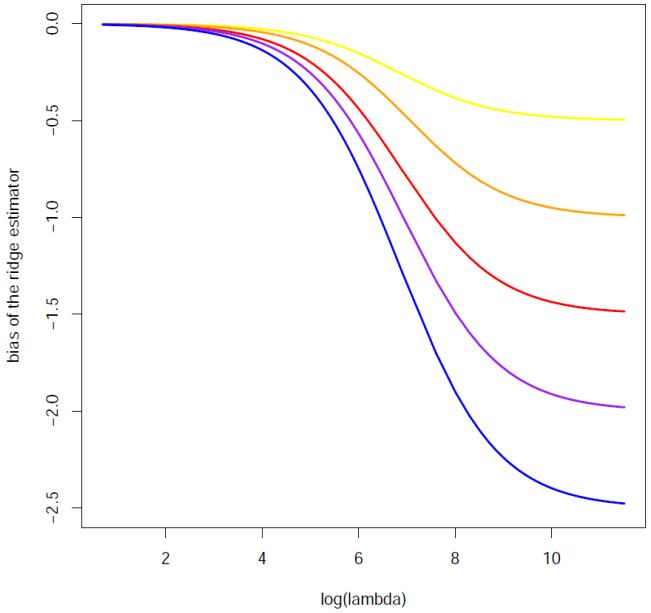
Question

So far:

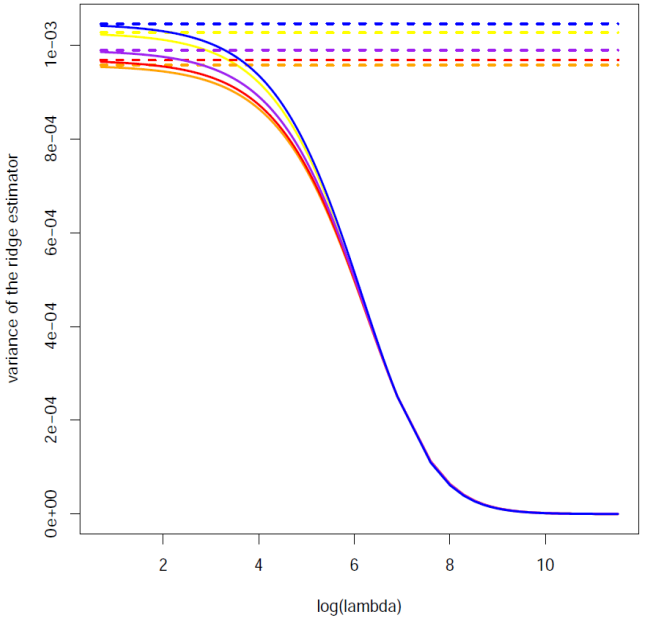
→ bias increases with λ, and

→ variance decreases with λ.

bias of ridge estimates



variance of ridge estimates



What happens to the MSE when λ increase?

Mean squared error

The mean squared error of the ridge estimator is then:

$$\begin{aligned} MSE(\lambda) &= E\left\{(\mathbf{W}_\lambda \hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T (\mathbf{W}_\lambda \hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\right\} \\ &= \sigma^2 \operatorname{tr}\left\{\mathbf{W}_\lambda (\mathbf{X}^T\mathbf{X})^{-1} \mathbf{W}_\lambda^T\right\} \\ &\quad + \boldsymbol{\beta}^T (\mathbf{W}_\lambda - \mathbf{I})^T (\mathbf{W}_\lambda - \mathbf{I})\, \boldsymbol{\beta} \end{aligned}$$

sum of variances of the ridge estimator

"squared bias" of the ridge estimator

Mean squared error

Ridge vs. OLS estimator

In the orthonormal case, i.e. $\mathbf{X}^T\mathbf{X} = \mathbf{I} = (\mathbf{X}^T\mathbf{X})^{-1}$:

$$\mathrm{MSE}[\hat{\boldsymbol{\beta}}] \quad = \quad p\,\sigma^2$$

and

$$\mathrm{MSE}[\hat{\boldsymbol{\beta}}(\lambda)] \quad = \quad \frac{p\,\sigma^2}{(1+\lambda)^2} + \frac{\lambda^2}{(1+\lambda)^2}\boldsymbol{\beta}^T\boldsymbol{\beta}$$

The latter achieves its minimum at: $\lambda = p\sigma^2(\boldsymbol{\beta}^\top\boldsymbol{\beta})^{-1}$
the ratio between the error variance and the 'signal'.

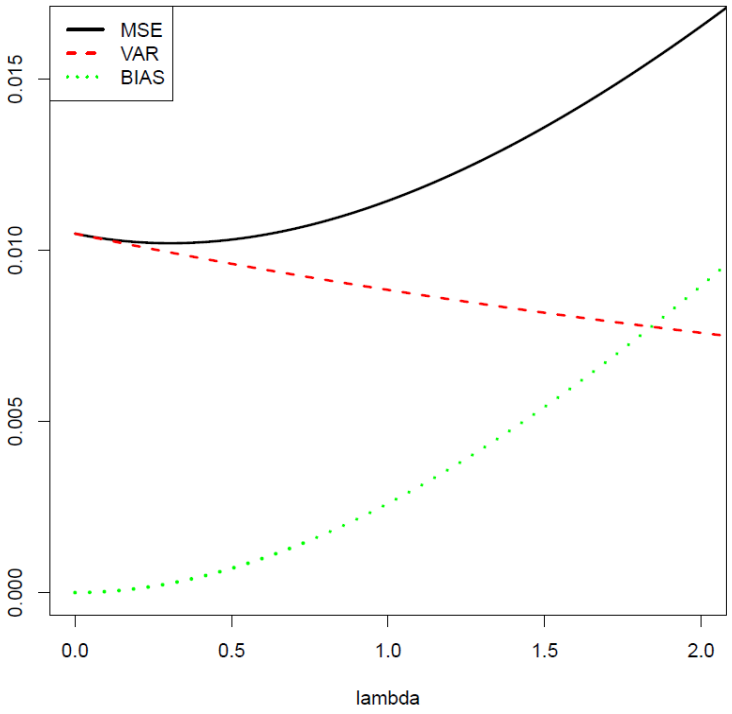
Question

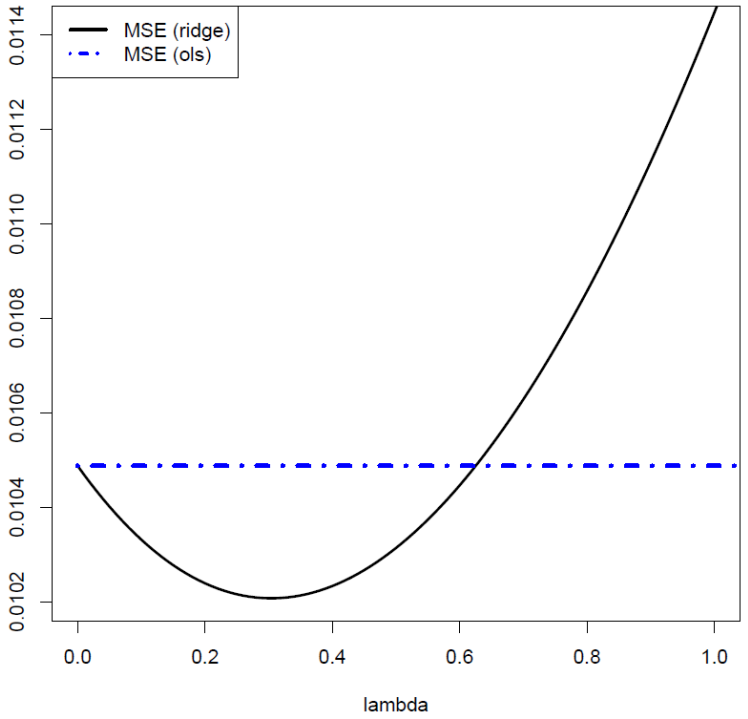
What is the practical relevance of this result?

Mean squared error

For small/large λ, variance/bias dominates the MSE.

For λ < 0.6, MSE(λ) < MSE(0) and the ridge estimator outperforms the OLS estimator.





Mean squared error

Theorem

There exists $\lambda > 0$ such that $\mathrm{MSE}(\lambda) < \mathrm{MSE}(0)$.

Problem

The optimal λ depends on unknown quantities $\boldsymbol{\beta}$ and σ^2.

Practice

Choose in data-driven manner by:

→ cross-validation,

→ information criterion,

→ empirical Bayes.

Constrained estimation

Constrained estimation

The ad-hoc ridge estimator minimizes the loss function:

$$\mathcal{L}(\boldsymbol{\beta};\lambda) \quad = \quad \underbrace{\|\mathbf{Y}-\mathbf{X}\,\boldsymbol{\beta}\|_2^2}_{\text{sum of squares}} + \underbrace{\lambda\|\boldsymbol{\beta}\|_2^2}_{\text{ridge penalty}}$$

with $\lambda \geq 0$ penalty parameter

Take the derivative:

$$\frac{\partial}{\partial\boldsymbol{\beta}}\mathcal{L}(\boldsymbol{\beta};\lambda) \quad = \quad -2\mathbf{X}^\top\mathbf{Y} + 2(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I}_{pp})\boldsymbol{\beta}.$$

Equate the derivative to zero and solve:

$$\hat{\boldsymbol{\beta}}(\lambda) = (\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I}_{pp})^{-1}\mathbf{X}^\top\mathbf{Y}$$

Constrained estimation

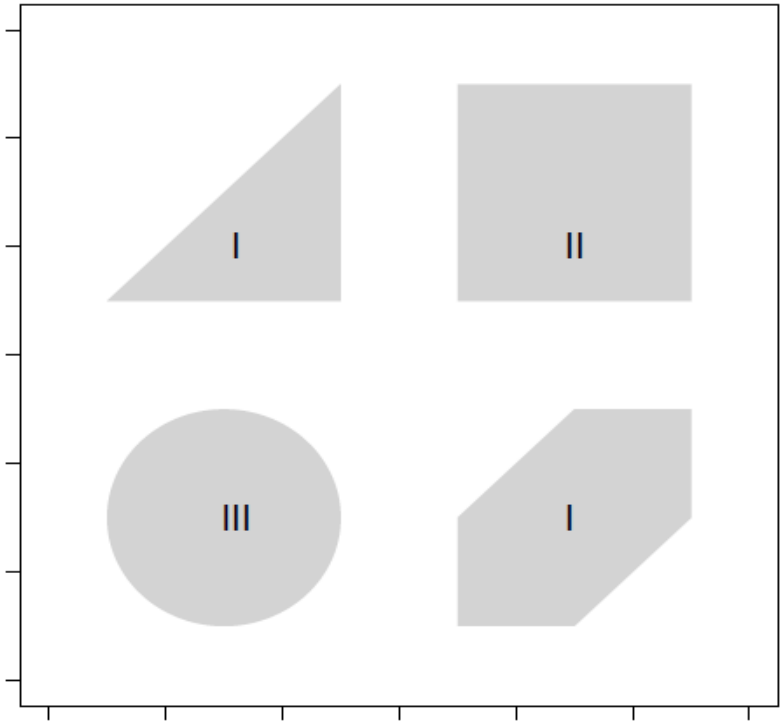
Convexity

A set $\mathcal{S}$ is *convex* if for all $x, y \in \mathcal{S}$, $\theta \in [0, 1]$:

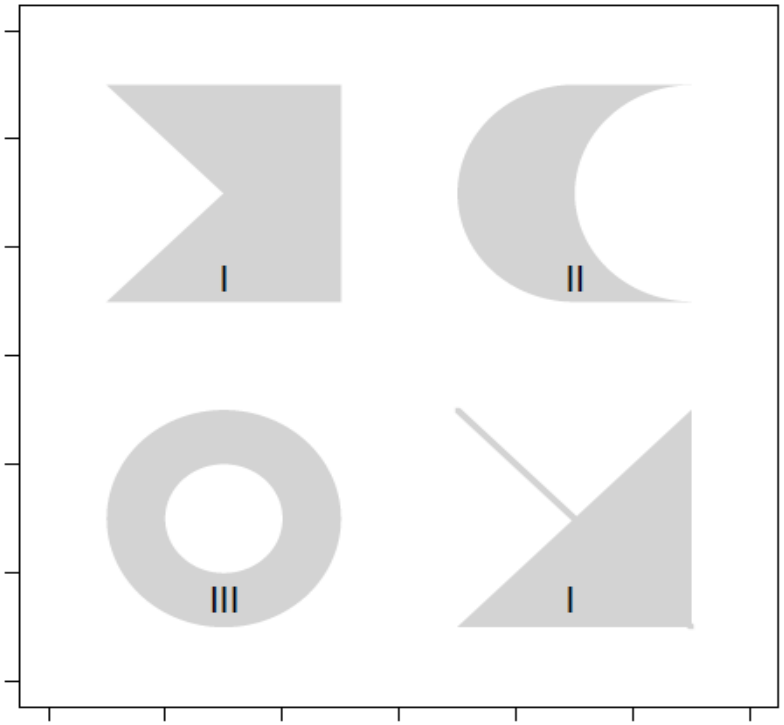
$$(1 - \theta)x + \theta y \in \mathcal{S}\,.$$

It is strict *convex* if $(1 - \theta)x + \theta y \in \mathcal{S} \setminus \partial\mathcal{S}$ for all $\theta \in (0, 1)$.

convex sets



non-convex sets



Constrained estimation

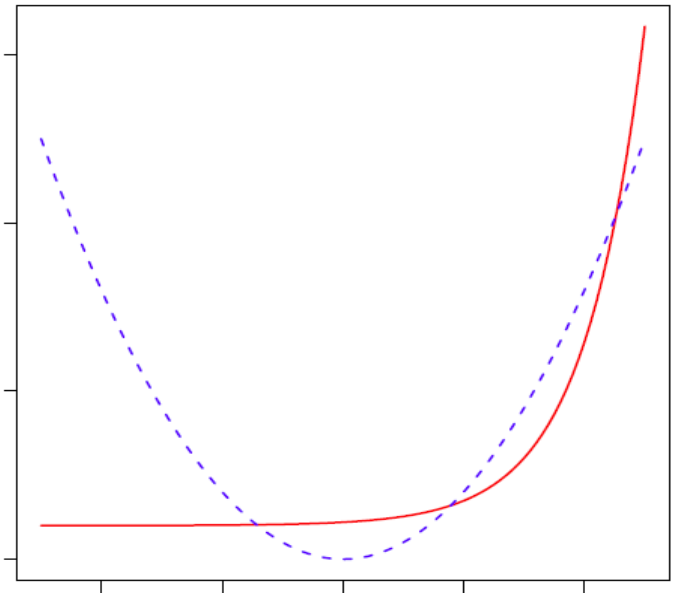
Convexity

A map $f : \mathcal{S} \mapsto \mathbb{R}$ is *convex* if for all $x, y \in \mathcal{S}$, $\mathcal{S}$ convex, and $\theta \in [0, 1]$: $\quad f[(1-\theta)x + \theta y] \leq (1-\theta)f(x) + \theta f(y)$.

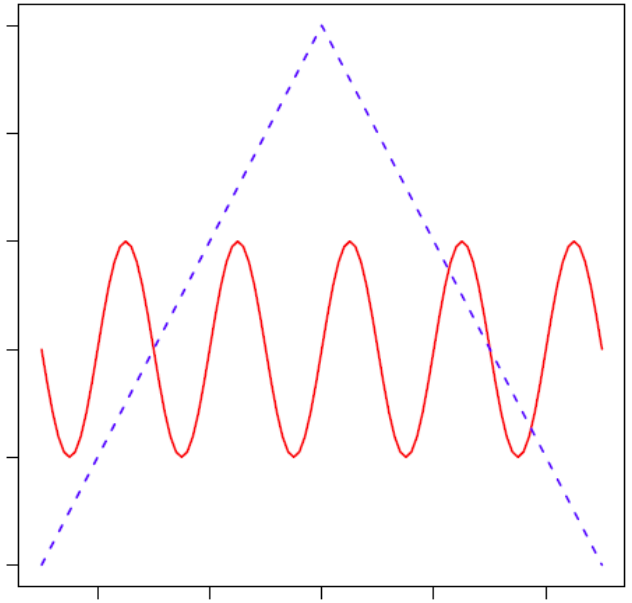
A function is convex $\leftrightarrow$ region above the function is convex.

Strict convex: strict inequality for all $\theta \in (0, 1)$.

convex functions



non-convex functions

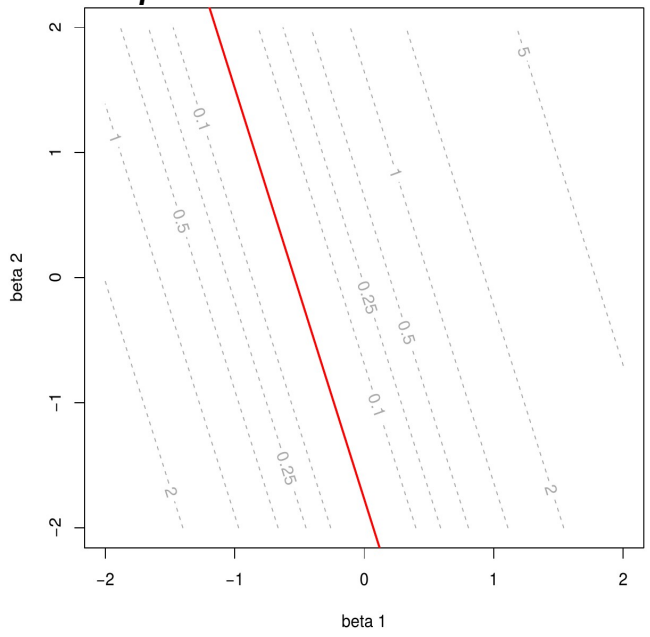


Constrained estimation

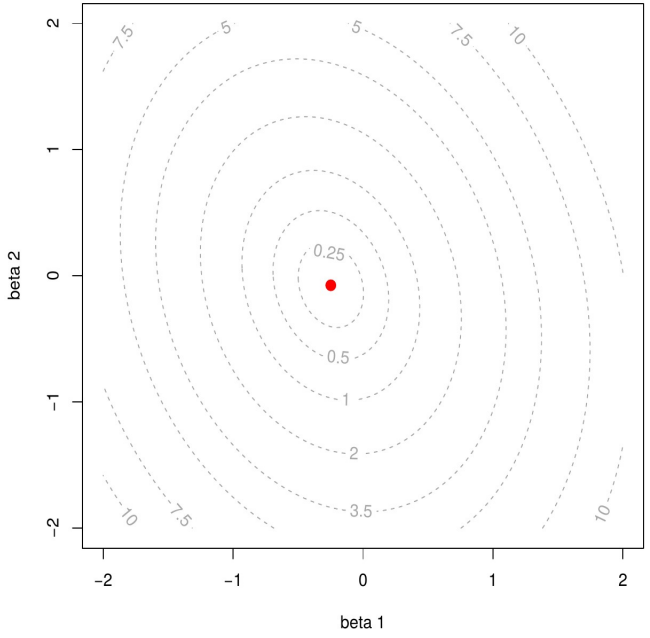
Convexity

Sum of squares, $\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$, is convex in $\boldsymbol{\beta}$. Penalty, $\lambda\|\boldsymbol{\beta}\|_2^2$, is strict convex in $\boldsymbol{\beta}$. Consequently, their sum is strict convex.

unpenalized loss



penalized loss



The red line / dot represents the optimum (minimum) of the loss function.

Strict convexity ensures the existence of a unique minimizer of the penalized sum of squares.

Constrained estimation

Ridge regression as constrained estimation

The method of Lagrange multipliers enables the reformulation of the penalized least squares problem:

$$\min_{\boldsymbol{\beta}} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2$$

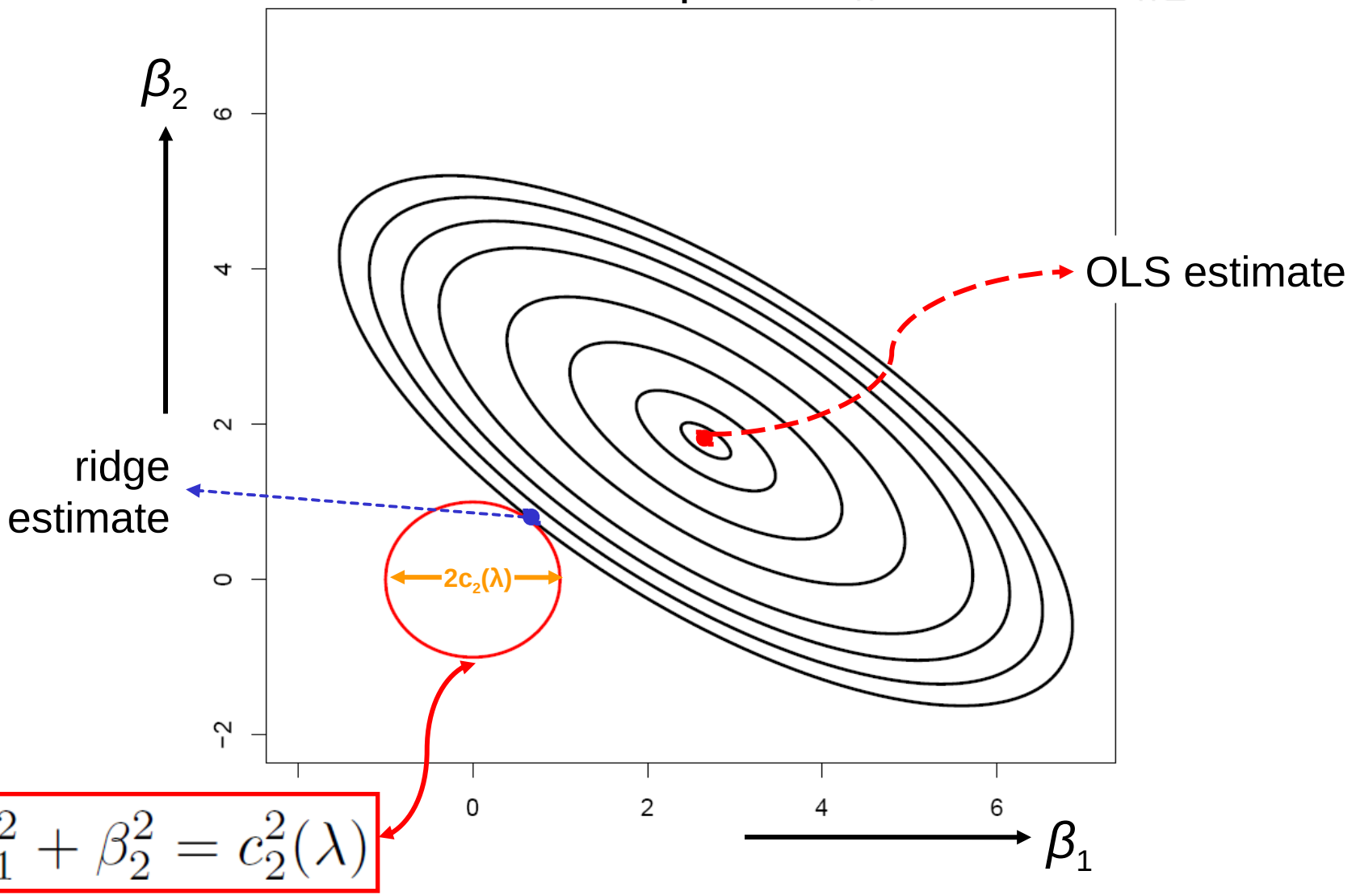
into a constrained estimation problem:

$$\min_{\|\boldsymbol{\beta}\|_2^2 \leq \theta(\lambda)} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$$

An explicit expression of θ(λ) is available.

Constrained estimation

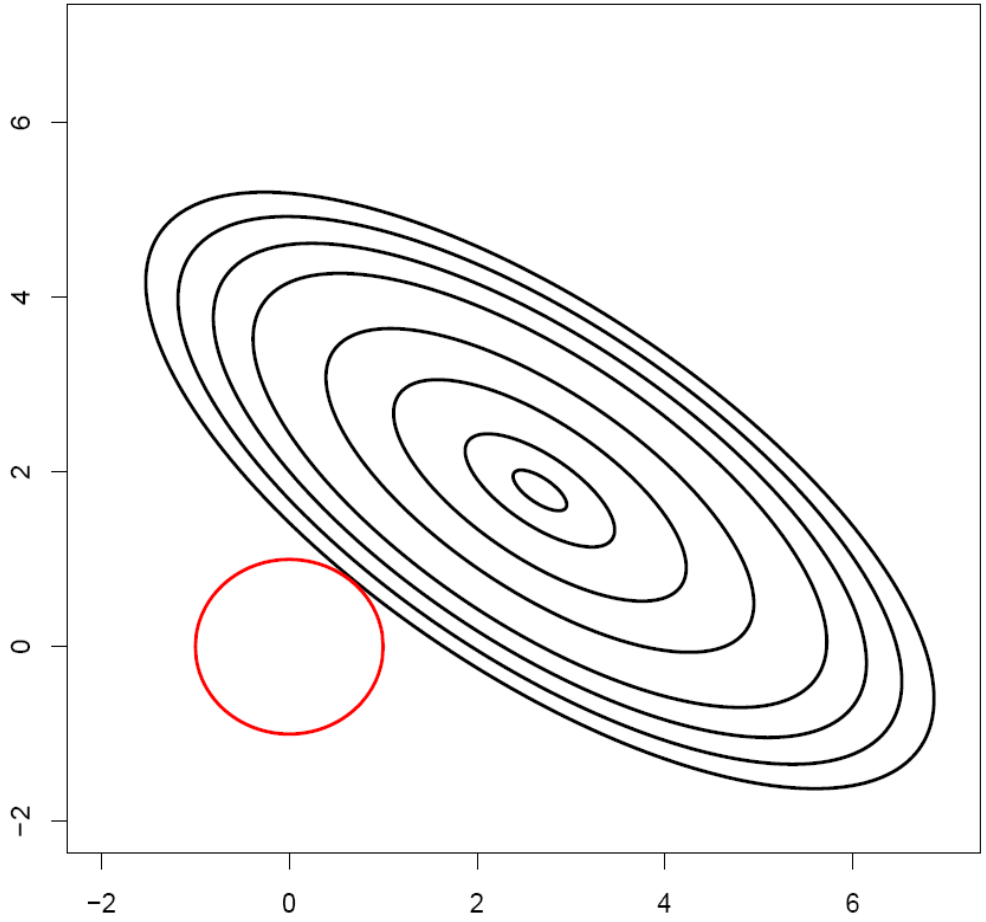
residual sum of squares: $\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$



Constrained estimation

Question

How does the parameter constraint domain fare with λ?



Over-fitting

Simple example

Consider 9 covariates with data drawn from the standard normal distribution: $X_{i,j} \sim \mathcal{N}(0,1)$

A response links to the covariates by the following linear regression model:

$$Y_i = X_{i,1} + \varepsilon_i$$

where $\varepsilon_i \sim \mathcal{N}(0, 1/4)$.

Only ten observations are drawn from model.
Hence, *n*=10 and *p*=9.

Over-fitting

Simple example

Fit the following linear regression model to the data:

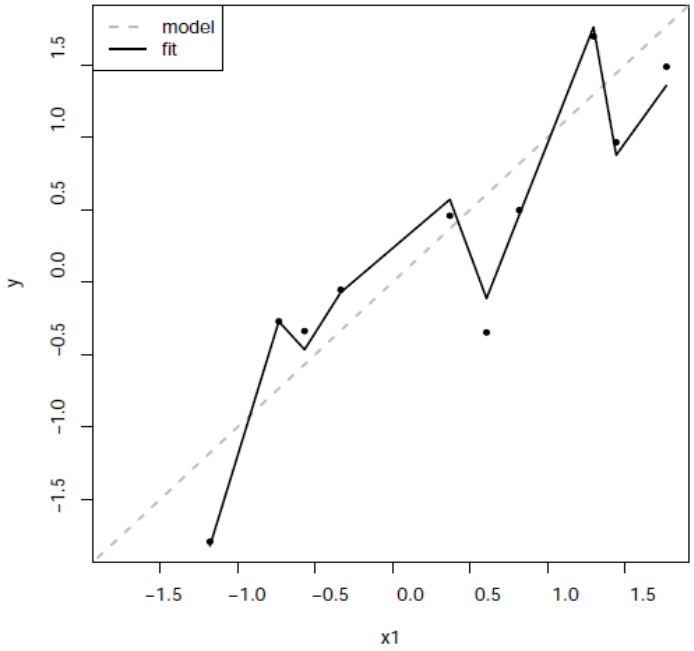
$$Y_i = \sum_{j=1}^{9} X_{ij}\beta_j + \varepsilon_i$$

Estimate:

```
b = (0.049, -2.386,
     -5.528,  6.243,
     -4.819,  0.760,
     -3.345, -4.748,
      2.136)
```

Large estimate values
→ indication of overfitting.

Fit:



A simple remedy: constrains the parameter estimator.
Another motivation for the ridge estimator!

A Bayesian interpretation

A Bayesian interpretation

Ridge regression is closely related to Bayesian linear regression.

Bayesian linear regression assumes the parameters β and σ^2 to be the random variables.

The conjugate priors for the parameters are:

$$\boldsymbol{\beta} \mid \sigma^2 \sim \mathcal{N}(\mathbf{0}, \frac{\sigma^2}{\lambda}\mathbf{I})$$

$$\sigma^2 \sim \mathcal{IG}(\alpha_0, \beta_0)$$

The latter denotes an inverse Gamma distribution.

A Bayesian interpretation

The posterior distribution of $\boldsymbol{\beta}$ and σ^2 can then be written as:

$$f_{\boldsymbol{\beta},\sigma^2}(\boldsymbol{\beta}, \sigma^2 \,|\, \mathbf{Y}, \mathbf{X})$$
$$\propto \quad g_{\boldsymbol{\beta}}(\boldsymbol{\beta} \,|\, \sigma^2, \mathbf{Y}, \mathbf{X})\, g_{\sigma^2}(\sigma^2 \,|\, \mathbf{Y}, \mathbf{X})$$

where

$$g_{\boldsymbol{\beta}}(\boldsymbol{\beta} \,|\, \sigma^2, \mathbf{Y}, \mathbf{X}) \;=$$
$$\sigma^{-k} \exp\left\{ -\frac{1}{2\sigma^2} \left[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}(\lambda)\right]^{\mathrm{T}} (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda \mathbf{I}) \left[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}(\lambda)\right] \right\}$$

Then, clearly the posterior mean of $\boldsymbol{\beta}$ is:

$$E(\boldsymbol{\beta}) \quad = \quad \hat{\boldsymbol{\beta}}(\lambda)$$

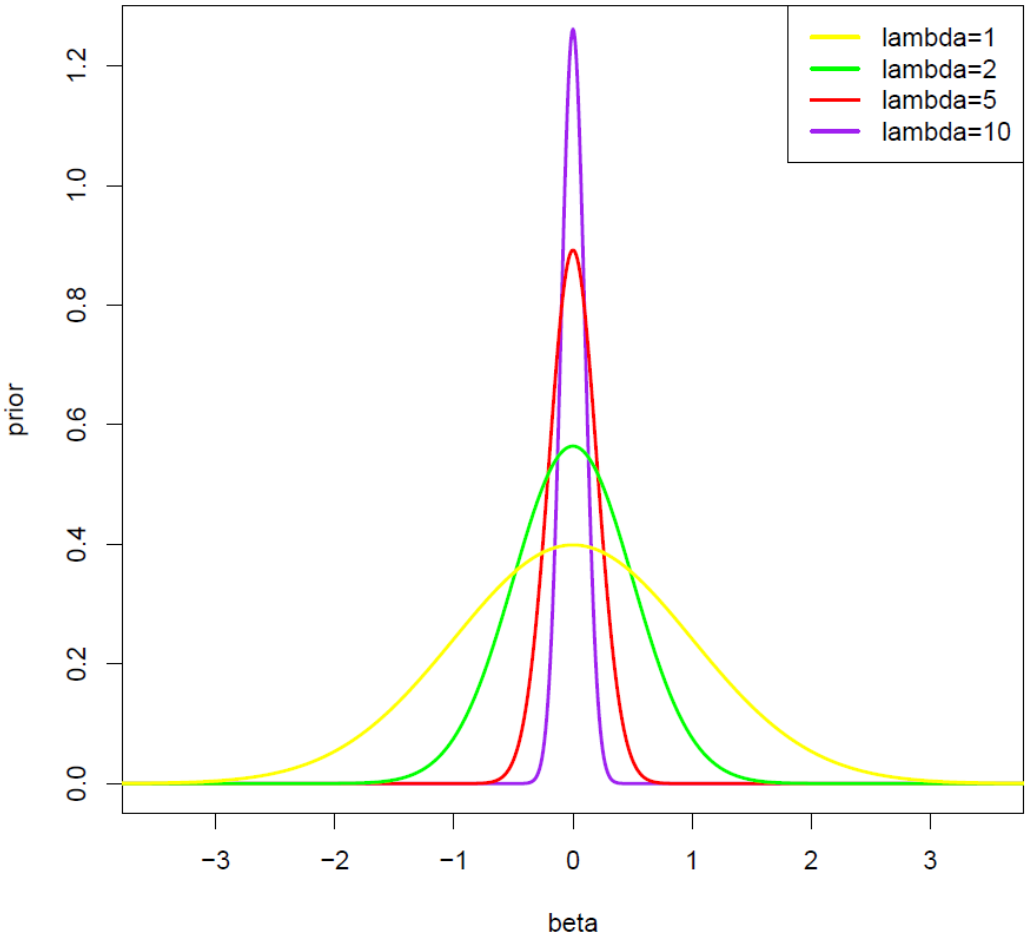
A Bayesian interpretation

Hence, the ridge regression estimator can be viewed as a Bayesian estimate of β when imposing a Gaussian prior.

The penalty parameter relates to the prior:

→ a small λ corresponds to wide/vague prior,

→ a large λ yields a narrow/informative one.



Efficient computation

Efficient computation

In the high-dimensional setting the number of covariates *p* is large compared to the number of samples *n*. In a microarray experiment *p* = 40000 and *n* = 100 is not uncommon.

If we wish to perform ridge regression in this context, we need to evaluate the expression:

$$\hat{\boldsymbol{\beta}}(\lambda) \quad = \quad (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$$

(p x *p)*-dim. matrix

For *p* = 40000 this is unfeasible on most computers.

However, there is a workaround.

Efficient computation

Revisit the singular value decomposition of **X**:

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$$

and write $\mathbf{R} = \mathbf{U}\mathbf{D}$.

As both **U** and **D** are (*n* x *n)*-dimensional matrices, so is **R**.

Consequently, **X** is now decomposed as: $\mathbf{X} = \mathbf{R}\mathbf{V}^\top$.

The ridge estimator can now be rewritten as:

$$\hat{\boldsymbol{\beta}}(\lambda) = \mathbf{V}(\mathbf{R}^\top\mathbf{R} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{R}^\top\mathbf{Y}$$

(n x *n)*-dim. matrix

Efficient computation

Hence, the reformulated ridge estimator involves the inversion of a (n x n)-dimensional matrix. With n = 100, this feasible on any standard computer.

Tibshirani and Hastie (2004) point out that the number of computation operations reduces from $O(p^3)$ to $O(pn^2)$.

In addition, they point out that this computation short-cut can be used in combition with other loss functions (GLM).

Degrees of freedom

Degrees of freedom

The degrees of freedom of ridge regression is calculated.

Recall from ordinary regression that:

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y} = \mathbf{H}\mathbf{Y}$$

where **H** is the hat matrix.

The degrees of freedom of ordinary regression: $\operatorname{tr}(\mathbf{H})$.

In particular, if **X** if of full rank, i.e. rank(**X**) = *p*, then:

$$\operatorname{tr}(\mathbf{H}) = p$$

Degrees of freedom

By analogy, the ridge-version of the hat matrix is:

$$\mathbf{H}(\lambda) \quad = \quad \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}$$

Continuing this analogy, the degrees of freedom of ridge regression is given by the trace of the hat matrix:

$$\mathrm{tr}[\mathbf{H}(\lambda)] = \sum_{j=1}^{p} d_j^2(d_j^2 + \lambda)^{-1}.$$

The d.o.f. is monotone decreasing in λ. In particular:

$$\lim_{\lambda \to \infty} \mathrm{tr}[\mathbf{H}(\lambda)] \quad = \quad 0$$

Simulation I

Variance of covariates

Simulation I

Effect of ridge estimation

Consider a set of 50 genes. Their expression levels follow a multivariate normal law with mean zero and covariance:

$$\boldsymbol{\Sigma} = \frac{1}{10}\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 2 & 0 & \dots & 0 \\ 0 & 0 & 3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 50 \end{pmatrix}$$

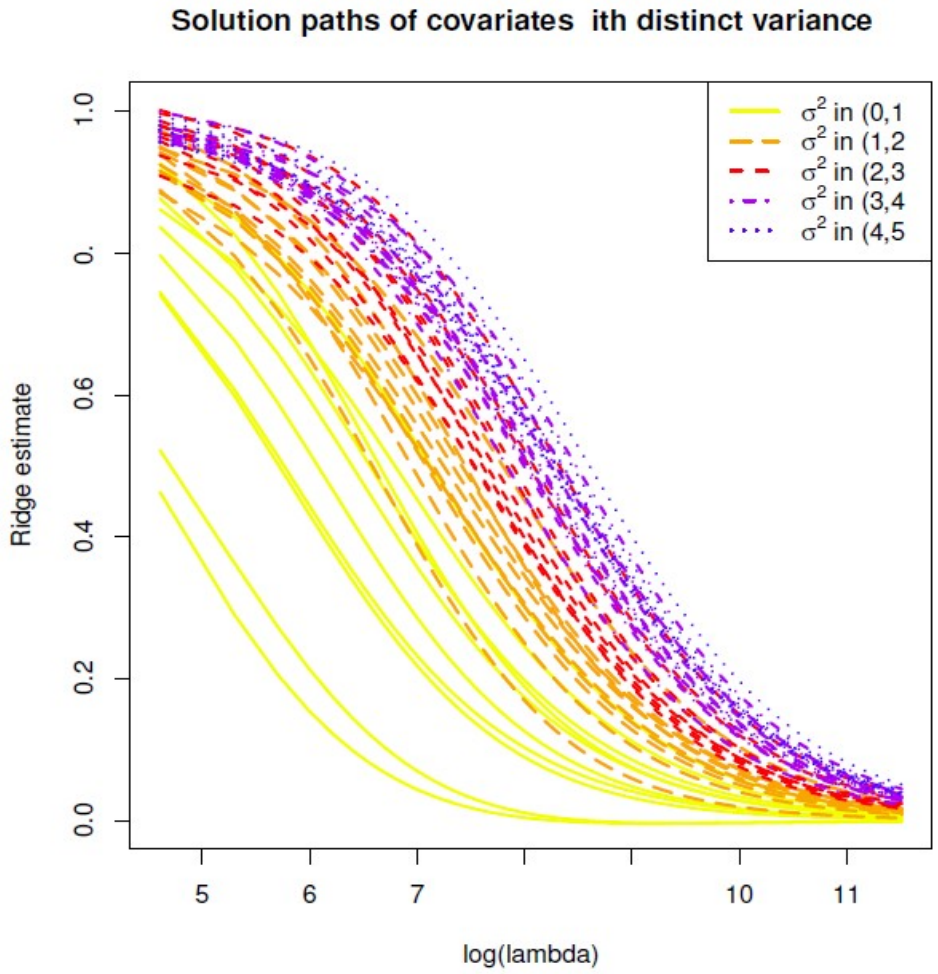
Put differently, a diagonal covariance with: $(\boldsymbol{\Sigma})_{jj} = j/10$

Together they regulate a 51th gene through: $Y_i = \mathbf{X}_{i*}\boldsymbol{\beta} + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0,1)$ and regression coefficients $\boldsymbol{\beta} = \mathbf{1}_{50}$.
Hence, the 50 genes contribute equally.

Simulation I

Effect of ridge estimation

Ridge regularization paths for coefficients of the 50 genes.



Ridge regression prefers (i.e. shrinks less) coefficient estimates of covariates with larger variance.

Simulation I

Some intuition

Rewrite the ridge regression estimator:

$$\begin{aligned}\boldsymbol{\beta}(\lambda) &= [\mathrm{Var}(\mathbf{X}) + \lambda \mathbf{I}_{50\times 50}]^{-1}\mathrm{Cov}(\mathbf{X}, \mathbf{Y}) \\ &= (\boldsymbol{\Sigma} + \lambda \mathbf{I}_{50\times 50})^{-1}\boldsymbol{\Sigma}[\mathrm{Var}(\mathbf{X})]^{-1}\mathrm{Cov}(\mathbf{X}, \mathbf{Y}) \\ &= (\boldsymbol{\Sigma} + \lambda \mathbf{I}_{50\times 50})^{-1}\boldsymbol{\Sigma}\boldsymbol{\beta}.\end{aligned}$$

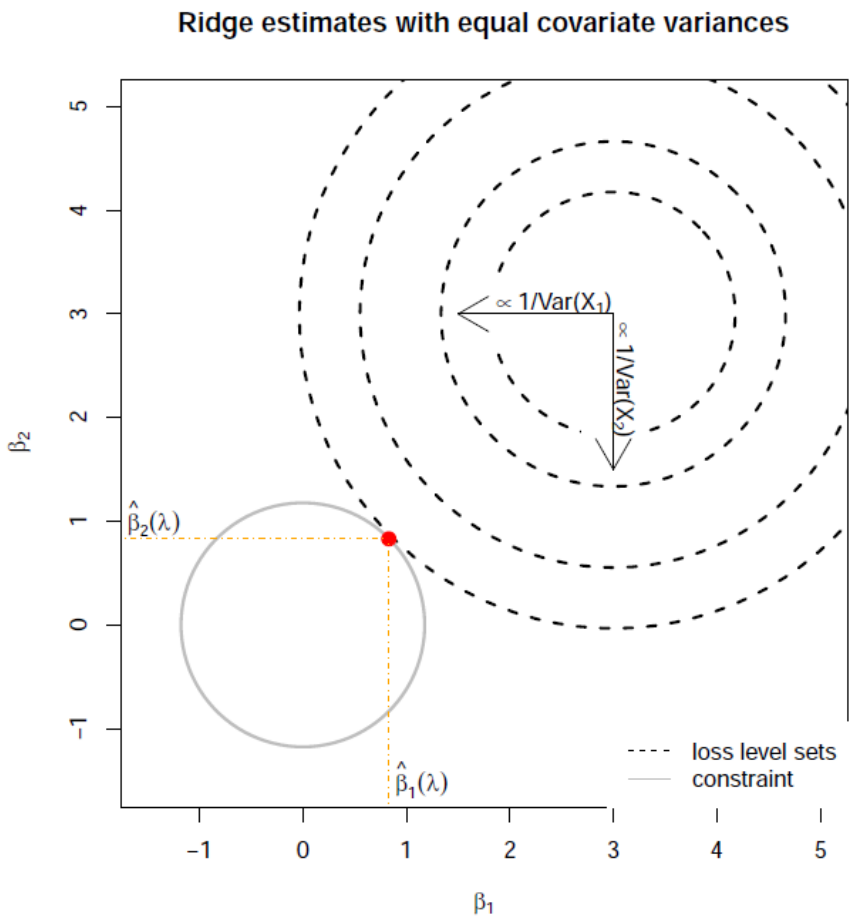
Plug in the employed covariance matrix:

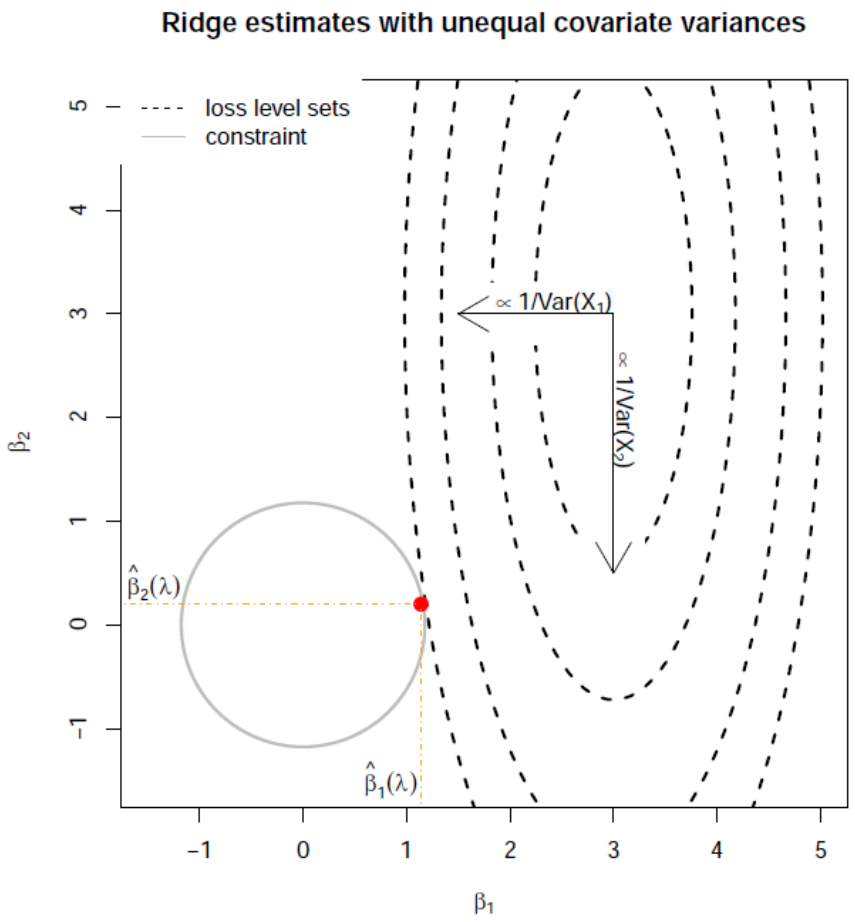
$$[\boldsymbol{\beta}(\lambda)]_j = \frac{j}{j + 50\lambda}(\boldsymbol{\beta})_j$$

Hence, larger variances = slower shrinkage.

Simulation I

Geometrically





Simulation I

Consider the ridge penalty:

$$\lambda \sum_{j=1}^{p} \beta_j^2$$

Each regression coefficient is penalized in the same way.

Considerations:

→ Some form of standardization seems reasonable, at least to ensure things are penalized comparably.

→ After preprocessing expression data of genes are often assumed to have a comparable scale.

→ Standardization affects the estimates.

Simulation II

Effect of collinearity

Simulation II

Effect of ridge estimation

Consider a set of 50 genes. Their expression levels followa multivariate normal law with mean zero and covariance:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{22} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{33} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{44} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{55} \end{pmatrix}$$

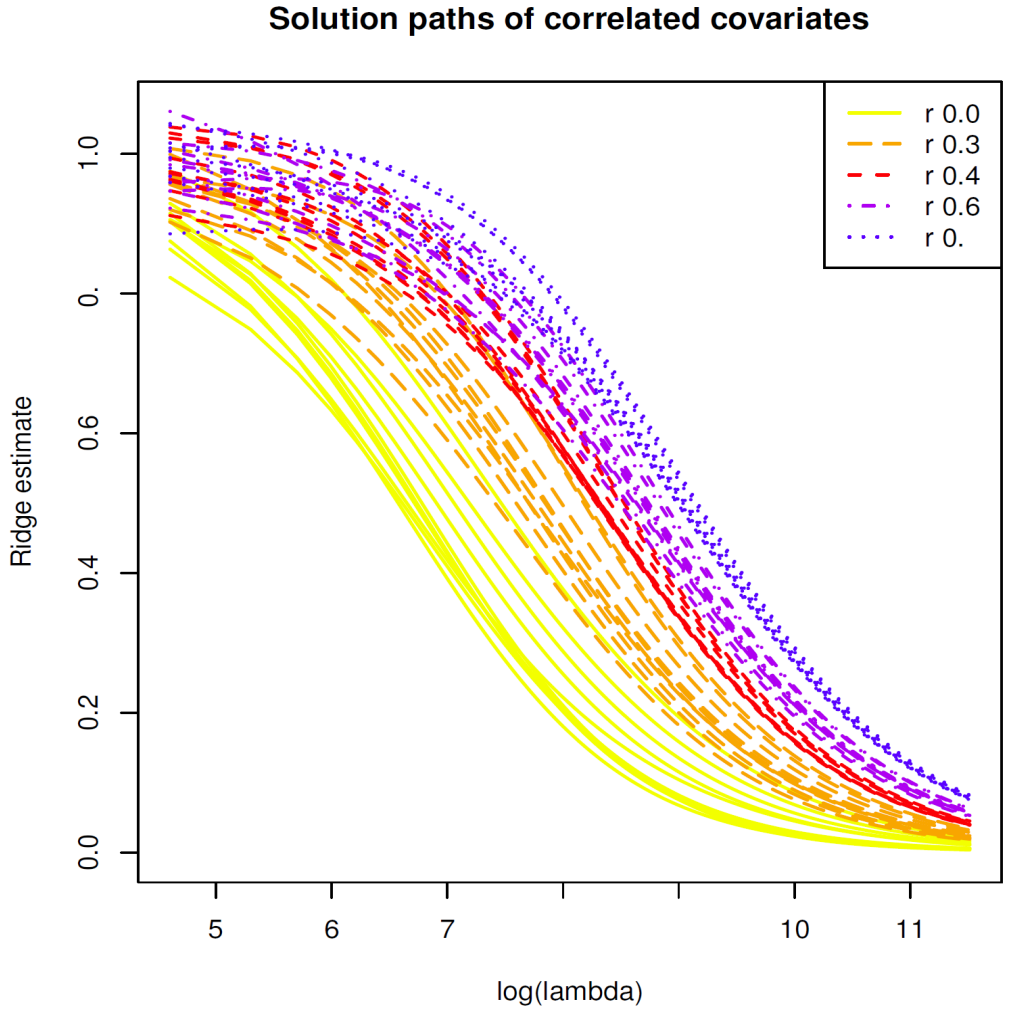
where $\boldsymbol{\Sigma}_{bb} = \frac{b-1}{5}\mathbf{1}_{10\times 10} + \frac{6-b}{5}\mathbf{I}_{10\times 10}$.

Together they regulate a 51th gene through: $Y_i = \mathbf{X}_{i*}\boldsymbol{\beta} + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0,1)$ and regression coefficients $\boldsymbol{\beta} = \mathbf{1}_{50}$.
Hence, the 50 genes contribute equally.

Simulation II

Effect of ridge estimation

Ridge regularization paths for coefficients of the 50 genes.



Ridge regression prefers (i.e. shrinks less) coefficient estimates of strongly positively correlated covariates.

Simulation II

Some intuition

Let p=2 and write U=X_1+X_2 and V=X_1-X_2. Then:

$$Y = (\beta_1 + \beta_2)U + (\beta_1 - \beta_2)V + \varepsilon$$

Write γ_a=β_1+β_2 and γ_b=β_1-β_2. Its ridge estimator is:

$$\boldsymbol{\gamma}(\lambda) = \begin{pmatrix} \mathrm{Var}(U) + \lambda & 0 \\ 0 & \mathrm{Var}(V) + \lambda \end{pmatrix}^{-1} \begin{pmatrix} \mathrm{Cov}(U, Y) \\ \mathrm{Cov}(V, Y) \end{pmatrix}$$

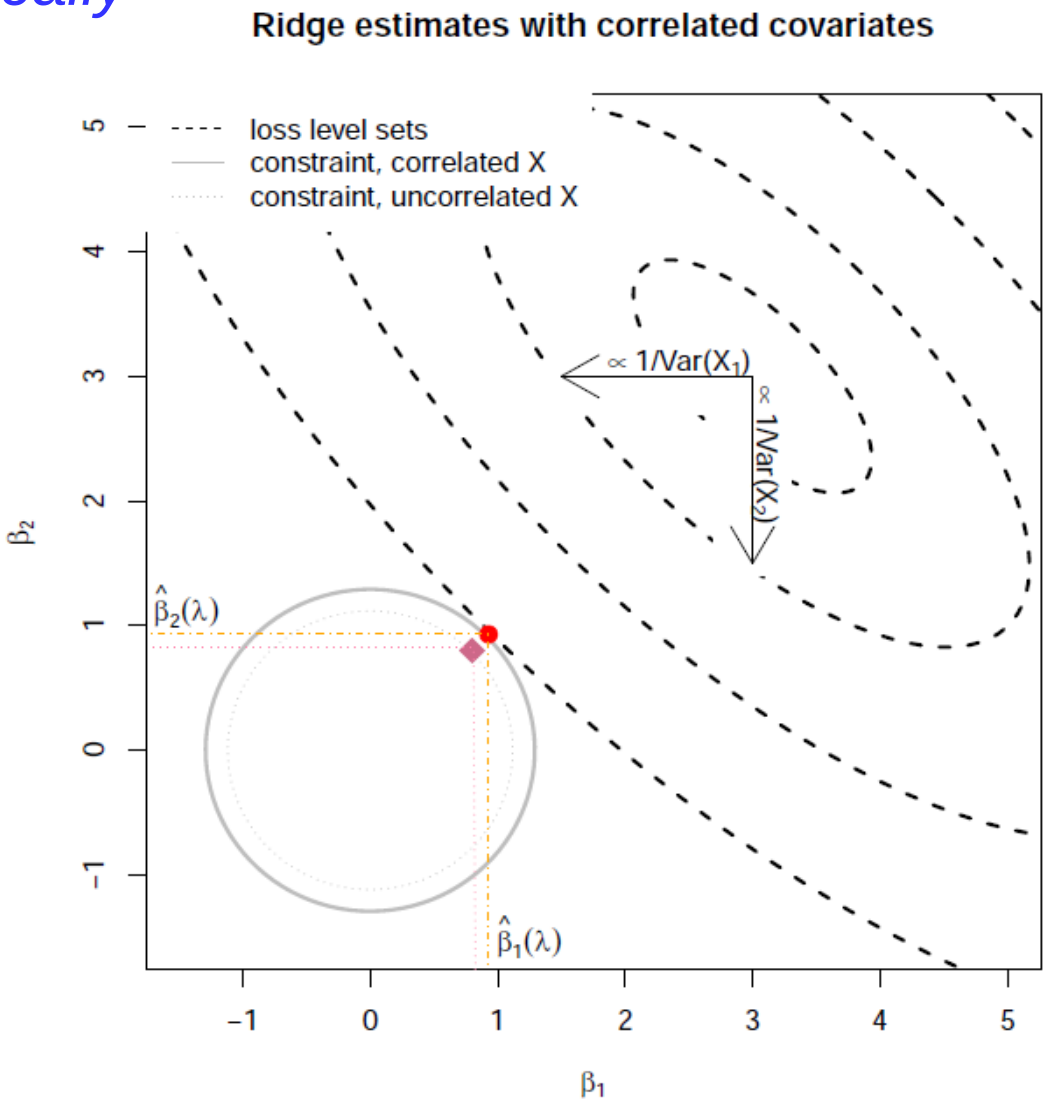
For large λ:

$$\boldsymbol{\gamma}(\lambda) \approx \frac{1}{\lambda} \begin{pmatrix} \mathrm{Var}(U) & 0 \\ 0 & \mathrm{Var}(V) \end{pmatrix} \begin{pmatrix} \beta_1 + \beta_2 \\ \beta_1 - \beta_2 \end{pmatrix}$$

Now use Var(U) >> Var(V) due to strong collinearity.

Simulation II

Geometrically



Cross-validation

Cross-validation

Methods for choosing penalty parameter

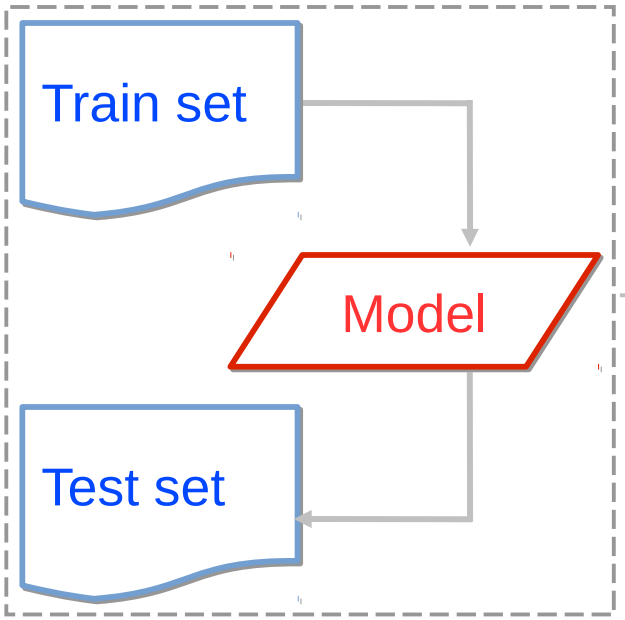
1.	Cross-validation
2.	Information criteria

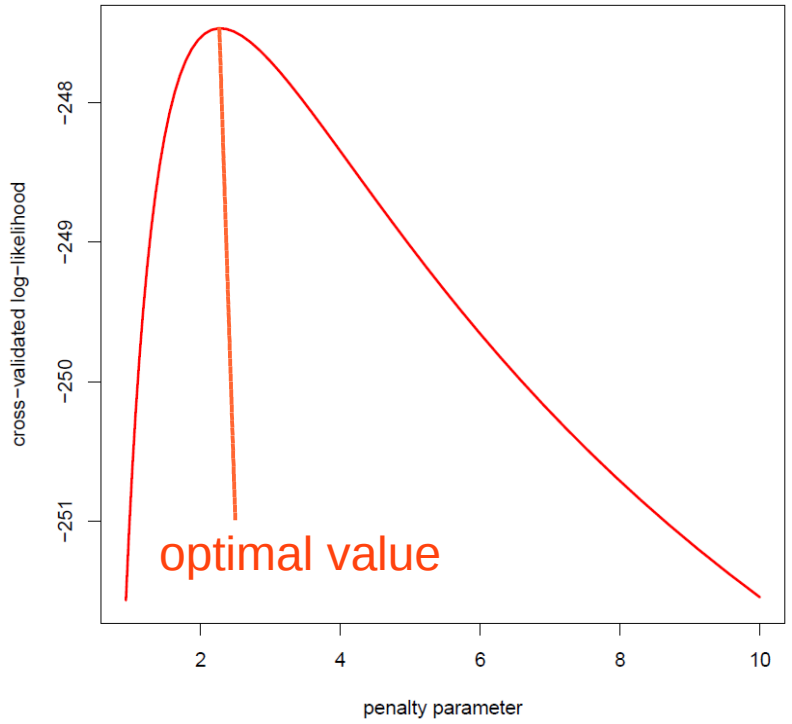
Cross-validation

- Estimation of the performance of a model, which is reflected in the error (often operationalized as log-likelihood or MSE).
- The data used to construct the model is also used to estimate the error.

Cross-validation

Penalty selection





→ *K*-fold
→ LOOCV

Cross-validation

Cross validation

- K-fold cross-validation divides the data set Λ randomly into K equal (or almost equal) sized subsets Λ_1, …, Λ_K.
- Model built on training set $\Lambda - \Lambda_k$.
- Model applied to the test set Λ_k to estimate the error.
- The average of these error estimates the error rate of the original classifier.
- n-fold cross-validation or leave-one-out cross-validation sets K = n, using Λ but one sample to built the models.

Cross-validation

LOOCV

The LOOCV loss can be calculated without resampling:

$$\begin{aligned}\lambda_{\text{opt}} &= \arg\min_{\lambda} \frac{1}{n}\sum_{i=1}^{n}[Y_i - \mathbf{X}_{i,*}\hat{\boldsymbol{\beta}}_{-i}(\lambda)]^2 \\ &= \arg\min_{\lambda} \frac{1}{n}\|\mathbf{B}(\lambda)[\mathbf{I}_{nn} - \mathbf{H}(\lambda)]\mathbf{Y}\|_F^2,\end{aligned}$$

where $\mathbf{B}(\lambda)$ diagonal and $[\mathbf{B}(\lambda)]_{ii} = [1 - \mathbf{H}_{ii}(\lambda)]^{-1}$.

Hence, instead of n evaluations of a *p*x*p* dimensional inverse only a single one is needed: a considerable gain.

Cross-validation

Generalized cross-validation

Diagonal elements of the hat matrix may assume value close or equal to one. Consequently, the LOOCV loss may become unstable.

This is resolved in the generalized cross-validation criterion:

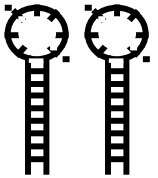
$$GCV(\lambda) = \frac{1}{n}\{\mathrm{tr}[\mathbf{I}_{nn} - \mathbf{H}(\lambda)]/n\}^{-2} \times \left\|[\mathbf{I}_{nn} - \mathbf{H}(\lambda)]\mathbf{Y}\right\|_2^2$$

The GCV too avoids the re-evaluation of the regression parameter estimate for each training set.

Example

Regulation of mRNA by microRNA

Example: microRNA-mRNA regulation



microRNAs

Recently, a new class of RNA was discovered: MicroRNA (mir). Mirs are non-coding RNAs of approx. 22 nucleotides. Like mRNAs, mirs are encoded in and transcribed from the DNA.

Mirs down-regulate gene expression by either of two post-transcriptional mechanisms: mRNA cleavage or transcriptional repression. Both depend on the degree of complementarity between the mir and the target.

A single mir can bind to and regulate many different mRNA targets and, conversely, several mirs can bind to and cooperatively control a single mRNA target.

Example: mir-mRNA regulation

Aim

Model microRNA regulation of mRNA expression levels.

Data

→ 90 prostate cancers
→ expression of 735 mirs
→ mRNA expression of the MCM7 gene

Motivation

→ MCM7 involved in prostate cancer.
→ mRNA levels of MCM7 reportedly affected by mirs.

Not part of the objective: *feature selection* ≈ understanding the basis of this prediction by identifying features (mirs) that characterize the mRNA expression.

Example: microRNA-mRNA regulation

Analysis

Find:

$$\text{mrna expr.} = f(\text{mir expression})$$
$$= \beta_0 + \beta_1 * \text{mir}_1 + \beta_2 * \text{mir}_2 + \ldots + \beta_p * \text{mir}_p + \text{error}$$

However, $p > n$: ridge regression. Having found the optimal λ, we obtain the ridge estimates for the coefficients: $b_j(\lambda)$.

With these estimates we calculate the linear predictor:

$$b_0 + b_1(\lambda) * \text{mir}_1 + \ldots + b_p(\lambda) * \text{mir}_p$$

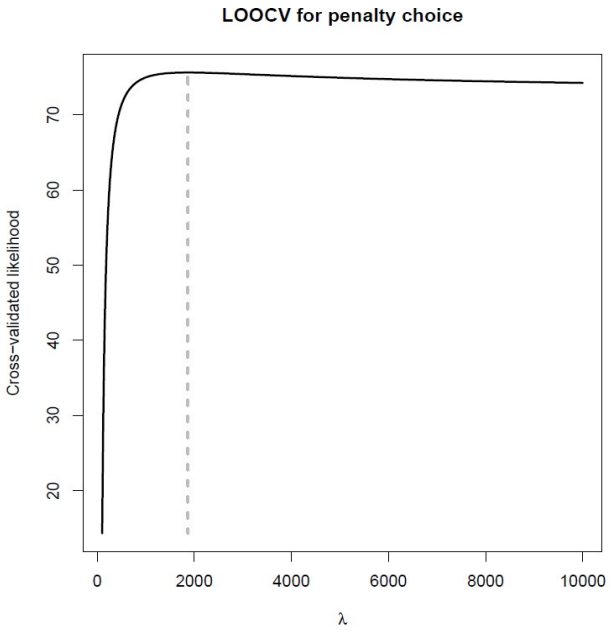
Finally, we obtain the predicted survival:

$$\text{pred. mrna expr.} = f(\text{linear predictor})$$
$$= b_0 + b_1(\lambda) * \text{mir}_1 + \ldots + b_p(\lambda) * \text{mir}_p$$

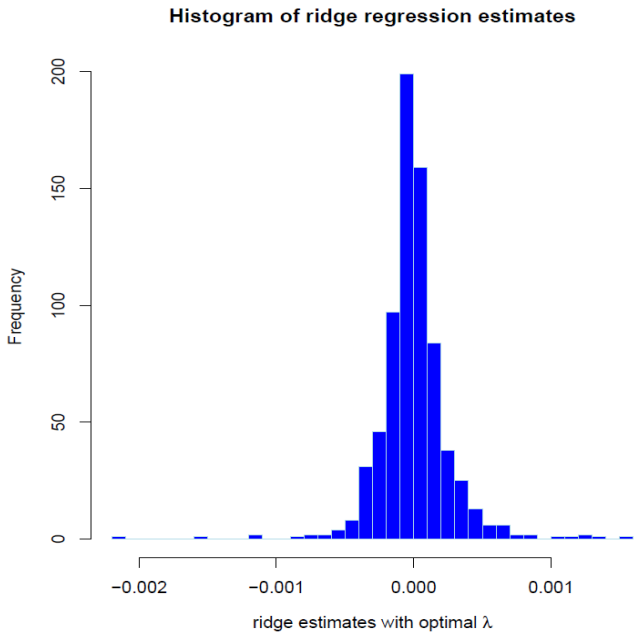
Compare observed and predicted mRNA expression.

Example: microRNA-mRNA regulation

Penalty parameter choice

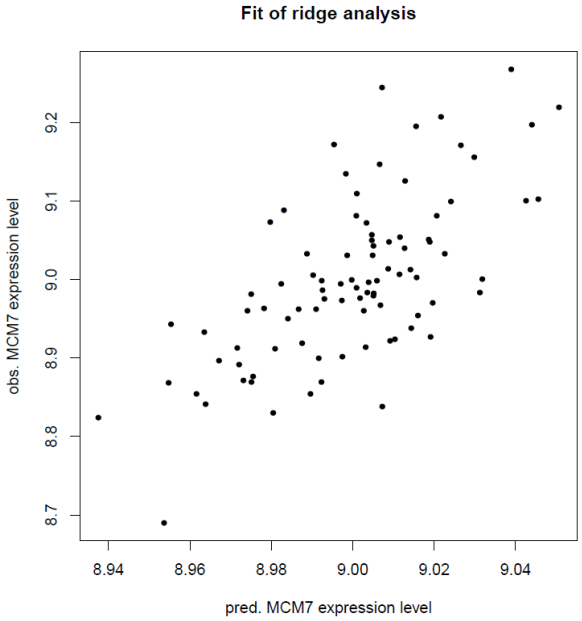


Beta hat distribution



$\#(\beta < 0) = 394$ (out of 735)

Obs. vs. pred. mRNA expression



$\rho_{sp} = 0.629$

$R^2 = 0.449$

Question: explain axes' scale difference in the RHS plot.

Example: microRNA-mRNA regulation

Biological dogma

MicroRNAs down-regulate mRNA levels.

The dogma suggests that negative regression coefficients prevail.

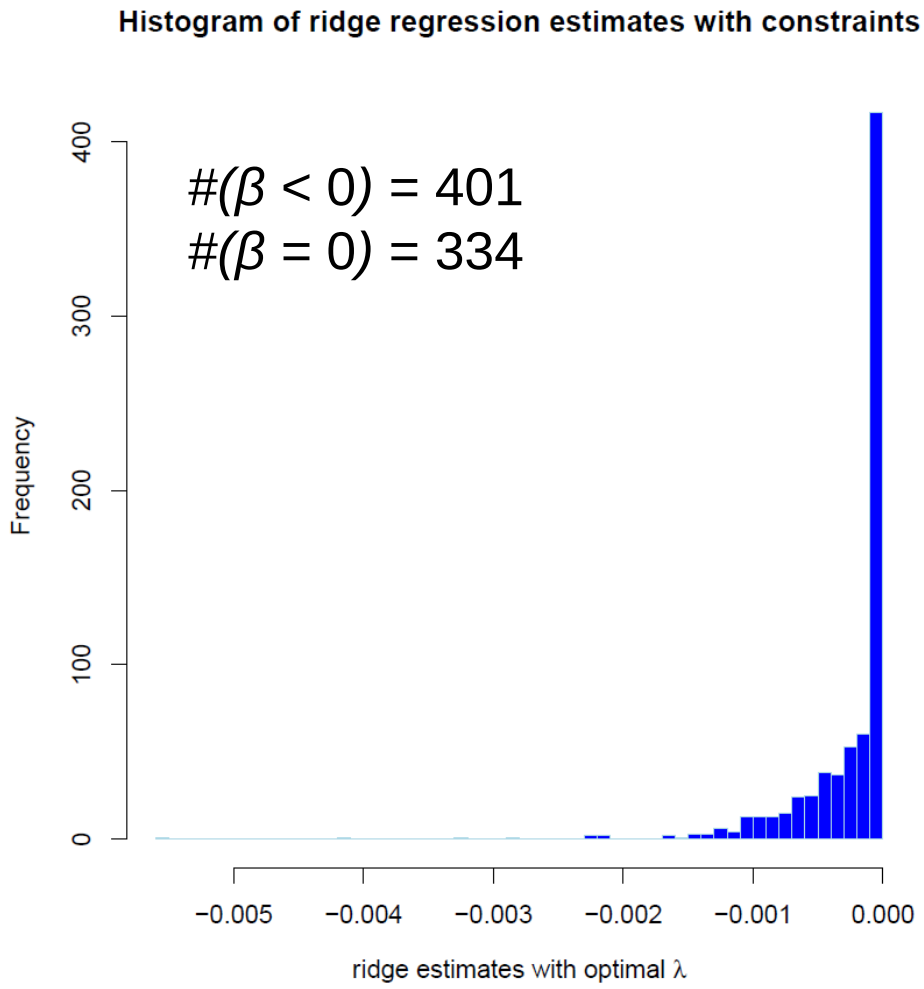
The **`penalized`** package allows for the specification of the sign of the regression parameters. No explicit expression for ridge estimator: numeric optimization of the loss function.

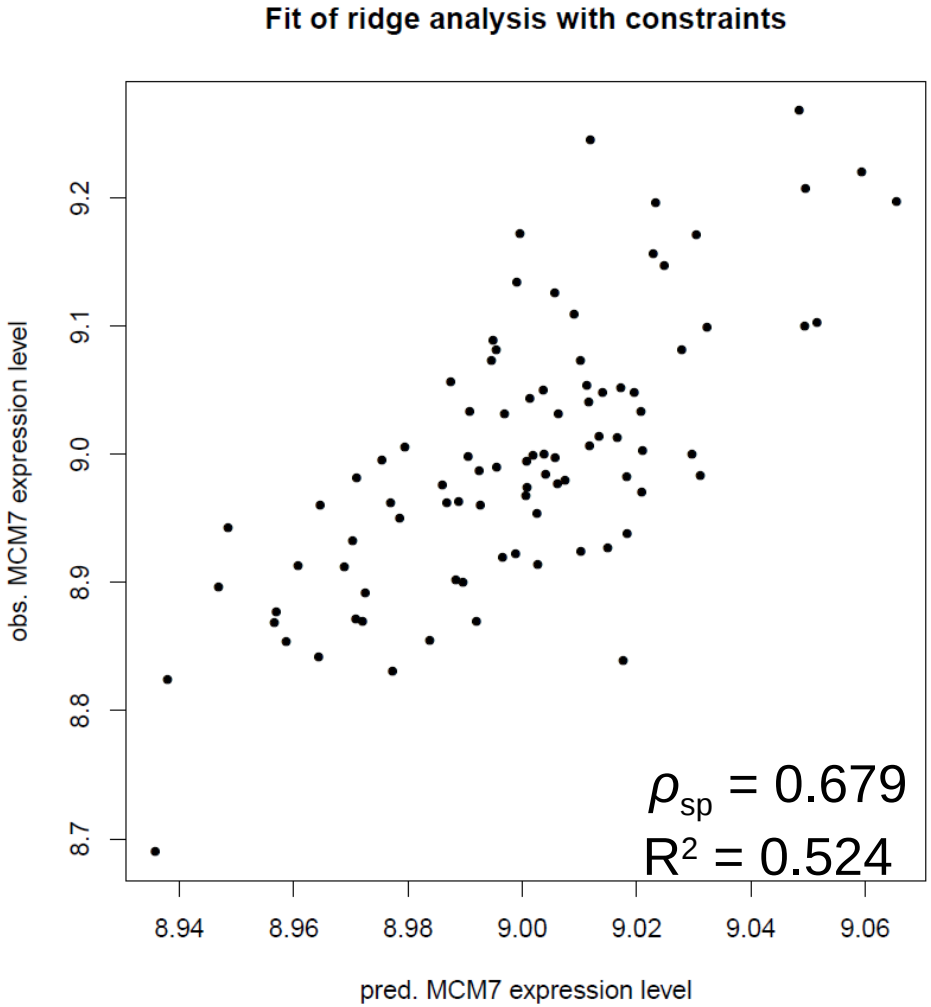
Re-analysis of the data with negative constraints.

Example: microRNA-mRNA regulation

Histograms of ridge estimates.

Observed vs. predicted mRNA expression.





Example: microRNA-mRNA regulation

The parameter constraint implies feature selection. Are the microRNAs identified to down-regulate MCM7 expression levels also reported by prediction tools?

Contingency table

	prediction tool	
ridge regression	no-mir2MCM7	mir2MCM7
β = 0	323	11
β < 0	390	11

Chi-square test

```
Pearson's Chi-squared test with Yates' continuity correction

data:  table(nonzeroBetas, nonzeroPred)
X-squared = 0.0478, df = 1, p-value = 0.827
```

Generalized ridge regression

Generalized ridge regression

A generalized ridge regression estimator minimizes a weighted least squares criterion augmented with a generalized ridge penalty:

$$(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top \mathbf{W} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top \boldsymbol{\Delta} (\boldsymbol{\beta} - \boldsymbol{\beta}_0),$$

with:

→ weight matrix $\mathbf{W}$,

→ penalty parameter $\boldsymbol{\Delta}$,

→ non-random target $\boldsymbol{\beta}_0$.

Set:

→ $\mathbf{W} = \mathbf{I}_{nn}$

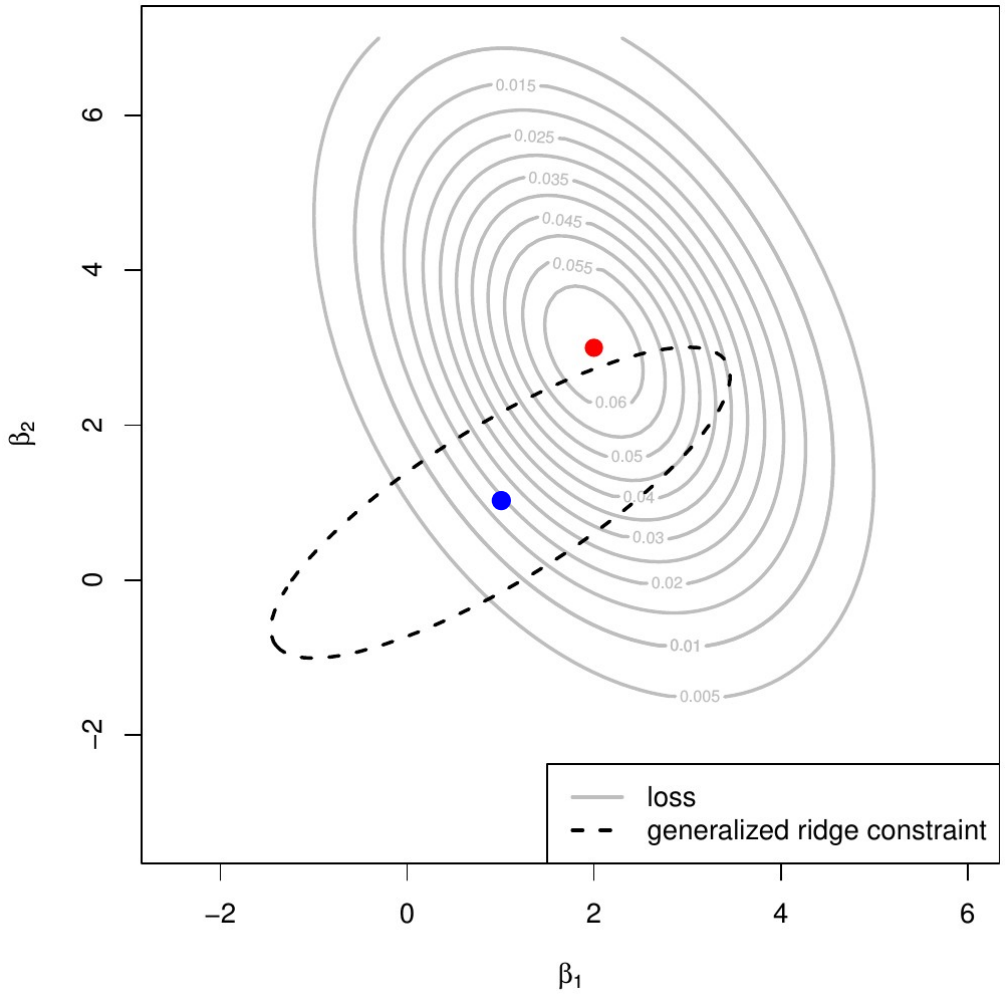
→ $\boldsymbol{\Delta} = \lambda \mathbf{I}_{pp}$

→ $\boldsymbol{\beta}_0 = \mathbf{0}_p$

to obtain the original ridge regression estimator.

Generalized ridge regression

The generalized penalty is a quadratic form. It implies a non-zero centered, ellipsoid parameter constraint.



Generalized ridge regression

Differentiate the loss criterion w.r.t. $\boldsymbol{\beta}$, equate to zero, and obtain the estimating equation:

$$2\mathbf{X}^\top\mathbf{W}\mathbf{Y} - 2\mathbf{X}^\top\mathbf{W}\mathbf{X}\boldsymbol{\beta} - 2\boldsymbol{\Delta}\boldsymbol{\beta} + 2\boldsymbol{\Delta}\boldsymbol{\beta}_0 \quad = \quad \mathbf{0}_p.$$

Solving this for $\boldsymbol{\beta}$ yields the generalized estimator:

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\Delta}) = (\mathbf{X}^\top\mathbf{W}\mathbf{X} + \boldsymbol{\Delta})^{-1}(\mathbf{X}^\top\mathbf{W}\mathbf{Y} + \boldsymbol{\Delta}\boldsymbol{\beta}_0).$$

Clearly, the generalized estimator reduces to the regular ridge regression estimator when simultaneously:

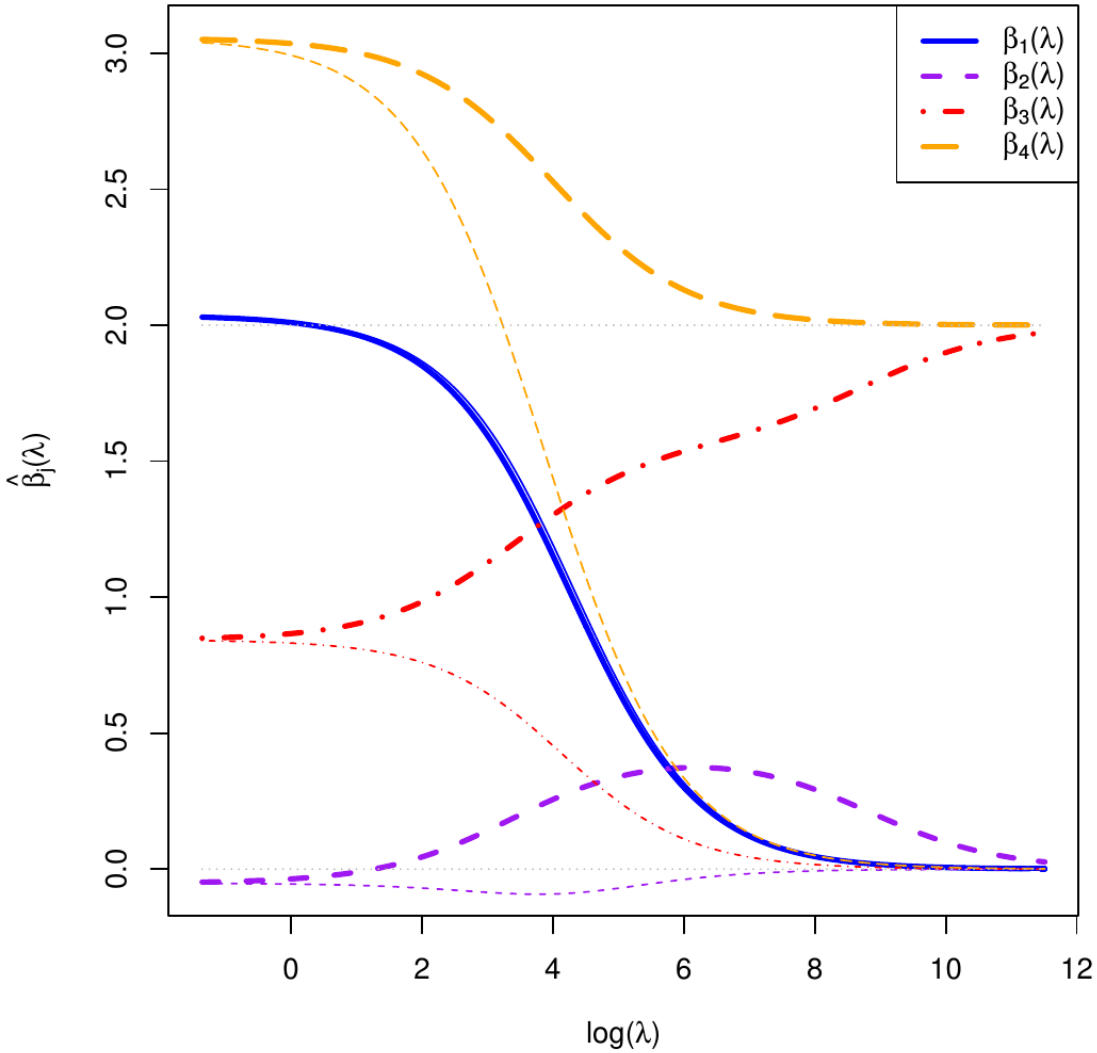
$\rightarrow \mathbf{W} = \mathbf{I}_{nn}$

$\rightarrow \boldsymbol{\Delta} = \lambda\mathbf{I}_{pp}$

$\rightarrow \boldsymbol{\beta}_0 = \mathbf{0}_p$

Generalized ridge regression

Regular and generalized regularization paths



Of note:

→ the limits of the 3rd and 4th regression coefficient.

→ more subtly, regularization paths of the 2nd and 3rd regression coefficient temporarily convergence.

Generalized ridge regression

The 1st and 2nd order moments of the generalized ridge regression estimator are:

$$\mathbb{E}[\hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})] = (\mathbf{X}^\top \mathbf{W}\mathbf{X} + \boldsymbol{\Delta})^{-1}(\mathbf{X}^\top \mathbf{W}\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\Delta}\boldsymbol{\beta}_0),$$

$$\mathrm{Var}[\hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})] = (\mathbf{X}^\top \mathbf{W}\mathbf{X} + \boldsymbol{\Delta})^{-1}\mathbf{X}^\top \mathbf{X}(\mathbf{X}^\top \mathbf{W}\mathbf{X} + \boldsymbol{\Delta})^{-1}.$$

Clearly, the generalized estimator is biased.

The generalized estimator has limiting behaviour:

$$\lim_{\boldsymbol{\Delta}\to\infty} \mathbb{E}[\hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})] = \boldsymbol{\beta}_0$$

$$\lim_{\boldsymbol{\Delta}\to\infty} \mathrm{Var}[\hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})] = \mathbf{0}_{pp}$$

Question

What is the effect of $\boldsymbol{\beta}_0$ on the MSE of the estimator?

Generalized ridge regression

Bayes

The generalized ridge estimator too has a Bayesian interpretation. Set $\mathbf{W} = \mathbf{I}_{nn}$ and replace the prior on the regression coefficients by: $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, \sigma^2 \boldsymbol{\Delta}^{-1})$.

The joint posterior then is:

$$f_{\boldsymbol{\beta},\sigma^2}(\boldsymbol{\beta}, \sigma^2 \,|\, \mathbf{Y}, \mathbf{X}) \quad \propto \quad g_{\boldsymbol{\beta}}(\boldsymbol{\beta} \,|\, \sigma^2, \mathbf{Y}, \mathbf{X})\, g_{\sigma^2}(\sigma^2 \,|\, \mathbf{Y}, \mathbf{X})$$

$$\exp\Big\{ -\frac{1}{2\sigma^2} \big[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})\big]^\top (\mathbf{X}^\top \mathbf{X} + \boldsymbol{\Delta}) \big[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})\big] \Big\}.$$

This implies: $\mathbb{E}(\boldsymbol{\beta} \,|\, \sigma^2, \mathbf{Y}, \mathbf{X}) = \hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})$.

Generalized ridge regression

Example

Consider the linear regression model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with coefficients $\beta_j = \phi_{0,1}(z_j)$, $z_j = -30 + \frac{6}{50}j$ for *j*=1, …, 500 and a standard normal error.

Estimate $\boldsymbol{\beta}$ by minimization of the fused ridge loss function:

$$\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{j=2}^{p} \|\beta_j - \beta_{j-1}\|_2^2.$$

where

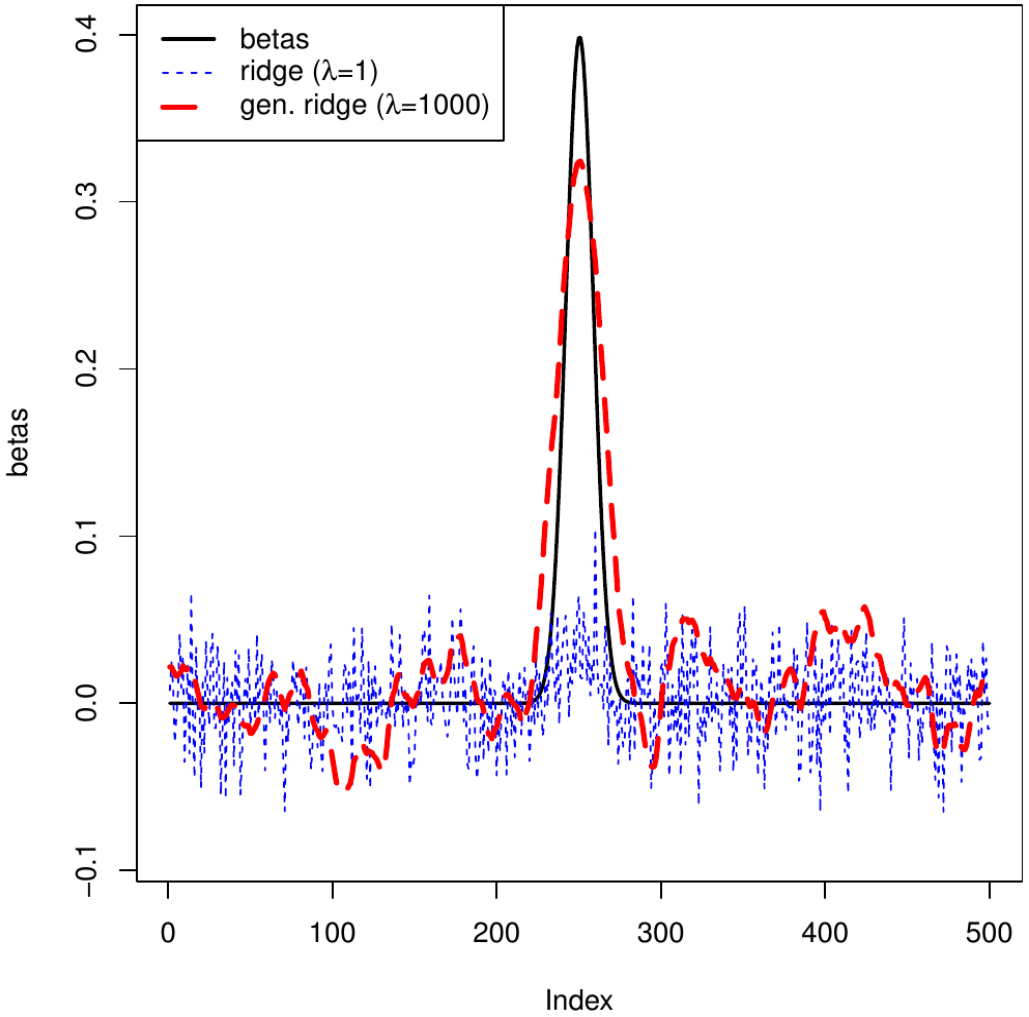
$$\lambda \sum_{j=2}^{p} \|\beta_j - \beta_{j-1}\|_2^2 = \boldsymbol{\beta}^\top \begin{pmatrix} \lambda & -\lambda & 0 & \dots & \dots & 0 \\ -\lambda & 2\lambda & -\lambda & \ddots & & \vdots \\ 0 & -\lambda & 2\lambda & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & -\lambda \\ 0 & \dots & \dots & 0 & -\lambda & \lambda \end{pmatrix} \boldsymbol{\beta}.$$

Cf., e.g., Goeman, 2008.

Generalized ridge regression

Example

Regular vs. fused ridge estimates



Generalized ridge regression

Example

DNA copy number : # gene copies encoded in the DNA.

→ 2 : most genes on the autosomal chromosomes,

→ 1 : genes on the X or Y chromosomes in males,

→ 0 : genes on the Y chromosome in females,

→ anything goes in cancer.

The *cis*-dogma: more copies, more transcription.

Q: Does a *trans*-effect exist? Does one gene's high copy number lead to elevated transcription levels of another?

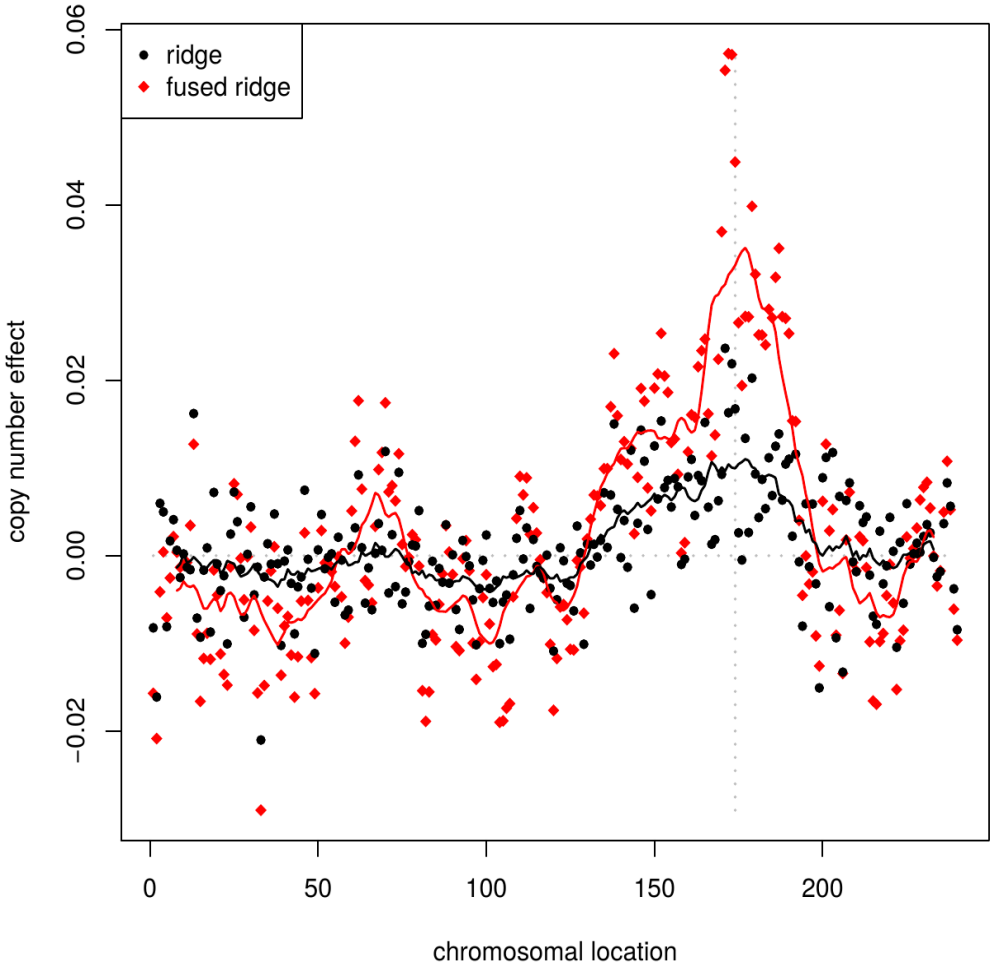
Regress a gene's expression on copy number of all genes.

→ *trans*-effect: large coefficients away from response gene.

Generalized ridge regression

Example

Ridge vs. fused ridge estimates: local copy number effect.



Generalized ridge regression

What is generally referred as the generalized ridge uses:

$\rightarrow \mathbf{W} = \mathbf{I}_{nn}$

$\rightarrow \boldsymbol{\beta}_0 = \mathbf{0}_p$

$\rightarrow \boldsymbol{\Delta} = \mathbf{V}_x \boldsymbol{\Lambda} \mathbf{V}_x^\top$ with $\mathbf{V}_x$ from the SVD $\mathbf{X} = \mathbf{U}_x \mathbf{D}_x \mathbf{V}_x^\top$, with positive definite diagonal matrix $\boldsymbol{\Lambda}$.

The generalized ridge estimator then is:

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\Lambda}) = \mathbf{V}_x (\mathbf{D}_x^2 + \boldsymbol{\Lambda})^{-1} \mathbf{D}_x \mathbf{U}_x \mathbf{Y}$$

Question: verify this expression.

Eigenvalues are shrunken individually rather than jointly.

Hoerl, Kennard, 1970; Hemmerle, 1975.

Generalized ridge regression

Rewrite the linear regression model to simplify notation:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} = \mathbf{X}\mathbf{V}_x\mathbf{V}_x^\top\boldsymbol{\beta} + \boldsymbol{\varepsilon} = \tilde{\mathbf{X}}\boldsymbol{\alpha} + \boldsymbol{\varepsilon},$$

where $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{V}_x = \mathbf{U}_x\mathbf{D}_x$ and $\boldsymbol{\alpha} = \mathbf{V}_x^\top\boldsymbol{\beta}$.

The loss function then becomes:

$$(\mathbf{Y} - \tilde{\mathbf{X}}\boldsymbol{\alpha})^\top(\mathbf{Y} - \tilde{\mathbf{X}}\boldsymbol{\alpha}) + \boldsymbol{\alpha}^\top\boldsymbol{\Lambda}\boldsymbol{\alpha}$$

which is optimized by:

$$\hat{\boldsymbol{\alpha}}(\boldsymbol{\Lambda}) = (\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}} + \boldsymbol{\Lambda})^{-1}\tilde{\mathbf{X}}^\top\mathbf{Y} = (\mathbf{D}_x^2 + \boldsymbol{\Lambda})^{-1}\tilde{\mathbf{X}}^\top\mathbf{Y},$$

In the original notation this results in:

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\Lambda}) = \mathbf{V}_x\,\hat{\boldsymbol{\alpha}}(\boldsymbol{\Lambda})$$

Generalized ridge regression

Use the 1st and 2nd moments of this estimator,

$$\mathbb{E}[\hat{\boldsymbol{\alpha}}(\boldsymbol{\Lambda})] = (\mathbf{D}_x^2 + \boldsymbol{\Lambda})^{-1}\mathbf{D}_x^2\boldsymbol{\alpha}$$

$$\mathrm{Var}[\hat{\boldsymbol{\alpha}}(\boldsymbol{\Lambda})] = \sigma^2(\mathbf{D}_x^2 + \boldsymbol{\Lambda})^{-1}\mathbf{D}_x^2(\mathbf{D}_x^2 + \boldsymbol{\Lambda})^{-1}$$

to obtain its MSE:

$$\mathrm{MSE}[\hat{\boldsymbol{\alpha}}(\boldsymbol{\Lambda})] = \sum_{j=1}^{p}(\sigma^2 d_{x,j}^2 + \alpha_j^2\lambda_j^2)(d_{x,j}^2 + \lambda_j)^{-2},$$

where $d_{x,j} = (\mathbf{D}_x)_{jj}$ and $\lambda_j = (\boldsymbol{\Lambda})_{jj}$.

The MSE of $\hat{\boldsymbol{\alpha}}(\boldsymbol{\Lambda})$ is minimized when $\lambda_j = \sigma^2/\alpha_j^2$ for all *j*.
Both quantities are unknown but may be estimated.
Estimates, however, need not yield the desired MSE.

Hemmerle, 1975; Lawless, 1981.

Generalized ridge regression

Questions

What is the effect of $\boldsymbol{\beta}_0$ on the MSE of the estimator:

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\Delta}) = (\mathbf{X}^\top \mathbf{W} \mathbf{X} + \boldsymbol{\Delta})^{-1} (\mathbf{X}^\top \mathbf{W} \mathbf{Y} + \boldsymbol{\Delta} \boldsymbol{\beta}_0).$$

Does the MSE of the generalized ridge estimator:

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\Lambda}) = \mathbf{V}_x (\mathbf{D}_x^2 + \boldsymbol{\Lambda})^{-1} \mathbf{D}_x \mathbf{U}_x \mathbf{Y}$$

outperform that of the regular ridge estimator:

$$\begin{aligned} \hat{\boldsymbol{\beta}}(\lambda \mathbf{I}_{pp}) &= \mathbf{V}_x (\mathbf{D}_x^2 + \lambda \mathbf{I}_{pp})^{-1} \mathbf{D}_x \mathbf{U}_x \mathbf{Y} \\ &= (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} \mathbf{X}^\top \mathbf{Y} \end{aligned}$$

The mixed model

Mixed model

The mixed or random effect model is:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon},$$

where **Z** is a (*n*x*q*)-dimensional design matrix, and:

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_n, \sigma^2_\varepsilon \mathbf{I}_{nn}),$$
$$\boldsymbol{\gamma} \sim \mathcal{N}(\mathbf{0}_q, \mathbf{R}_{\boldsymbol{\theta}}) \text{ with } \mathbf{R}_{\boldsymbol{\theta}} \in \mathcal{S}_{++},$$
$$\boldsymbol{\varepsilon} \perp\!\!\!\perp \boldsymbol{\gamma}.$$

The covariance matrix of the random effect, $\mathbf{R}_{\boldsymbol{\theta}}$, is parametrized by a low-dimensional parameter $\boldsymbol{\theta}$.

Reformulated:

$$\mathbf{Y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \mathbf{Z}\mathbf{R}_\theta\mathbf{Z}^\top + \sigma^2_\varepsilon \mathbf{I}_{nn}).$$

Mixed model

Example (growth rate of cells)

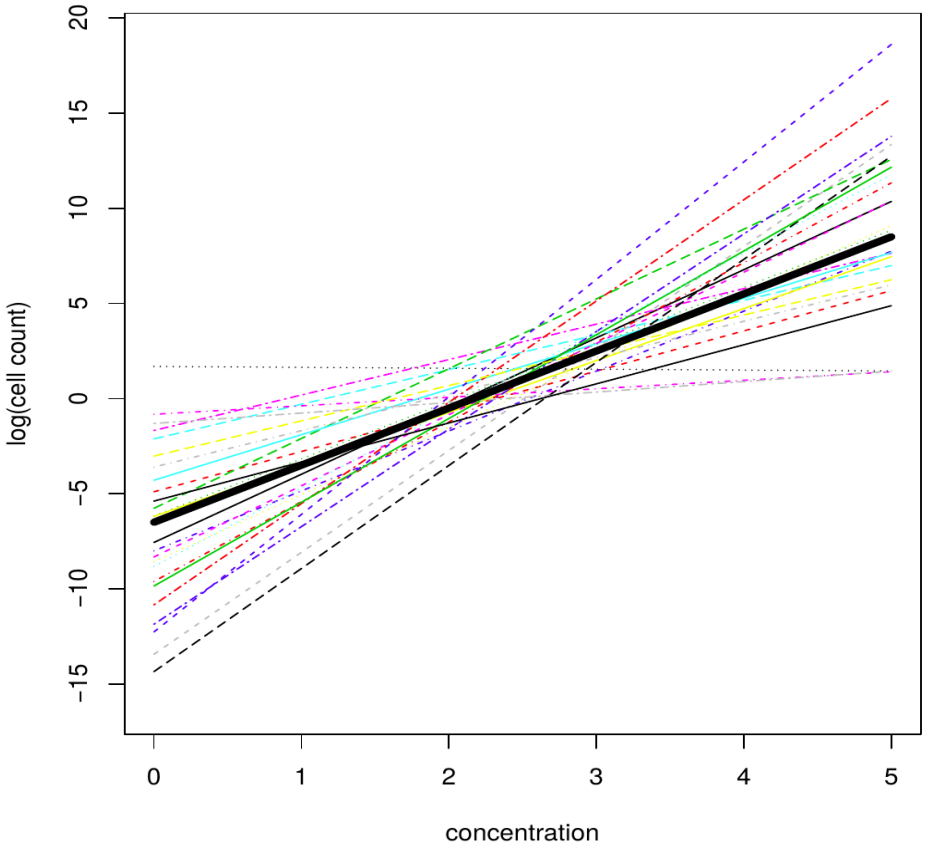
→ Longitudinal study

→ Y_{it} = (log) cell count in petri dish *i* at time *t*

→ X_i = concentration of growth medium in petri dish i.

→ Mixed model:

$$Y_{it} = \beta_0 + X_i\beta_1 + \mathbf{Z}_i\boldsymbol{\gamma} + \varepsilon_{it}$$



Mixed model

Estimation

Parameters are estimated by maximization of the likelihood:

$$L(\mathbf{Y}) = \int_{\mathbb{R}^q} L(\mathbf{Y} \,|\, \boldsymbol{\gamma} = \mathbf{g})\, f_{\boldsymbol{\gamma}}(\mathbf{g})\, d\mathbf{g}$$

with conditional likelihood:

$$L(\mathbf{Y} \,|\, \boldsymbol{\gamma} = \mathbf{g}) \;=\; (2\pi\sigma_\varepsilon^2)^{-n/2} \exp(-\tfrac{1}{2}\sigma_\varepsilon^{-2} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{g}\|_2^2).$$

Or, by restricted likelihood maximization (REML):

$$\hat{\boldsymbol{\theta}}, \hat{\sigma}_\varepsilon^2 = \arg\max_{\boldsymbol{\theta}, \sigma_\varepsilon^2} \int_{\mathbb{R}^p} L(\mathbf{Y}) d\boldsymbol{\beta}$$

Mixed model

Link to ridge regression

In absence of fixed effects the mixed model becomes:

$$\mathbf{Y} = \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}$$

where $\boldsymbol{\gamma} \sim \mathcal{N}(\mathbf{0}_q, \sigma_\gamma^2 \mathbf{I}_{qq})$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_n, \sigma_\varepsilon^2 \mathbf{I}_{nn})$.

Temporarily consider the random effect as fixed. Then:

$$\hat{\boldsymbol{\gamma}} = \arg\min_{\boldsymbol{\gamma} \in \mathbb{R}^q} \|\mathbf{Y} - \mathbf{Z}\boldsymbol{\gamma}\|_2^2 + \sigma_\gamma^{-2} \boldsymbol{\gamma}^\top \boldsymbol{\gamma},$$

which can be solved explicitly:

$$\hat{\boldsymbol{\gamma}} = (\mathbf{Z}^\top \mathbf{Z} + \sigma_\gamma^{-2} \mathbf{I}_{qq})^{-1} \mathbf{Z}^\top \mathbf{Y}.$$

A shrinkage estimator that allows for $q > n$!

Mixed model

Theorem

Assume $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}_p, \sigma_\beta^2 \mathbf{I}_{pp})$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_n, \sigma_\varepsilon^2 \mathbf{I}_{nn})$. The expected generalized cross-validation error of the ridge estimator $\mathbb{E}_{\boldsymbol{\beta}}\{\mathbb{E}_{\varepsilon}[GCV(\lambda)]\}$ is minimized for $\lambda = \sigma_\varepsilon^2 / \sigma_\beta^2$.

A familiar ratio, confer the MSE of ridge estimator with orthonormal design.

Practical take-away

Handy for penalty parameter selection:

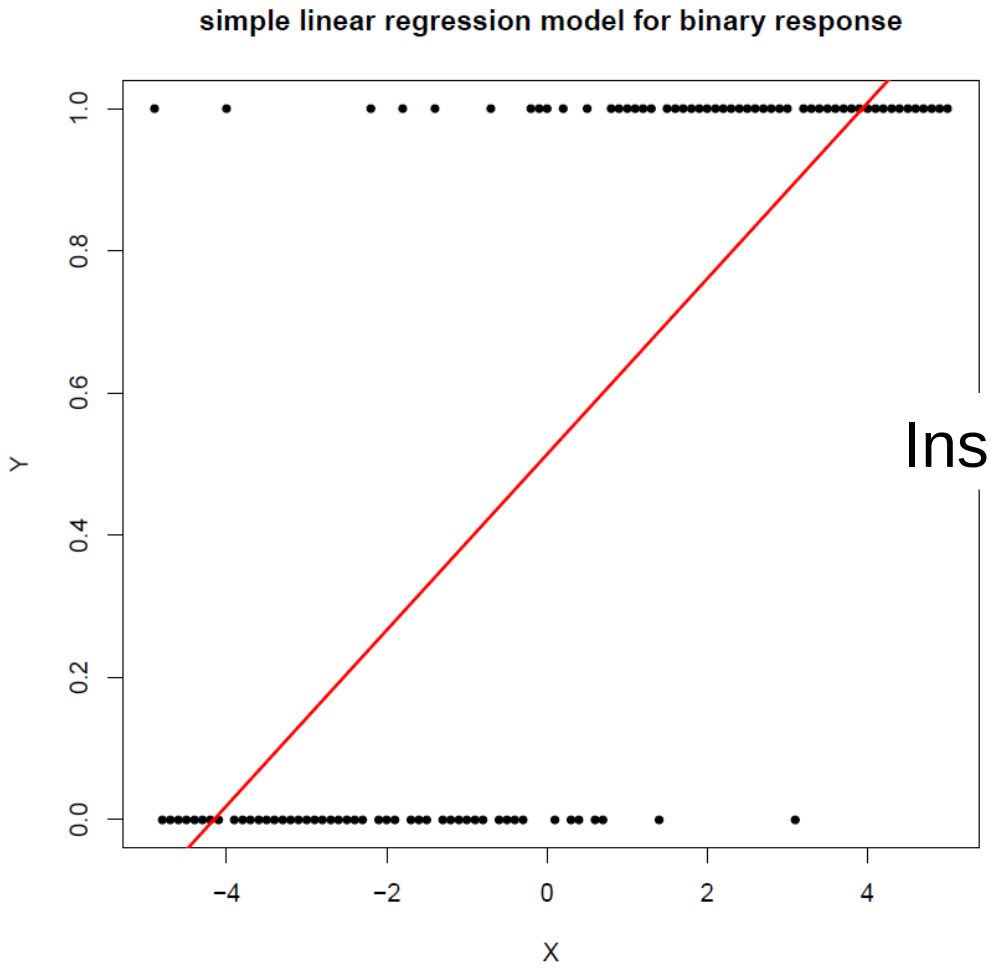
→ use REML to estimate σ_ε^2 and σ_β^2.

→ set $\lambda = \hat{\sigma}_\varepsilon^2 / \hat{\sigma}_\beta^2$.

Logistic regression (recap)

The logistic regression model

Linear regression relates a continuous response to explanatory variables. What if the response were binary?



Insightful?

The logistic regression model

Do not model $Y_i \in \{0, 1\}$ directly, but rather:

$$p_i = P(Y_i = 1 \mid X_{i1}, \ldots, X_{ip})$$

What would then be an appropriate model?

i) $$p_i = \beta_0 + \beta_1 X_i$$

LHS may yield values outside [0,1].

ii) $$p_i/(1 - p_i) = \beta_0 + \beta_1 X_i$$

LHS may yield negative values.

iii) $$\log[p_i/(1 - p_i)] = \beta_0 + \beta_1 X_i$$

Both RHS and LHS cover the real line!

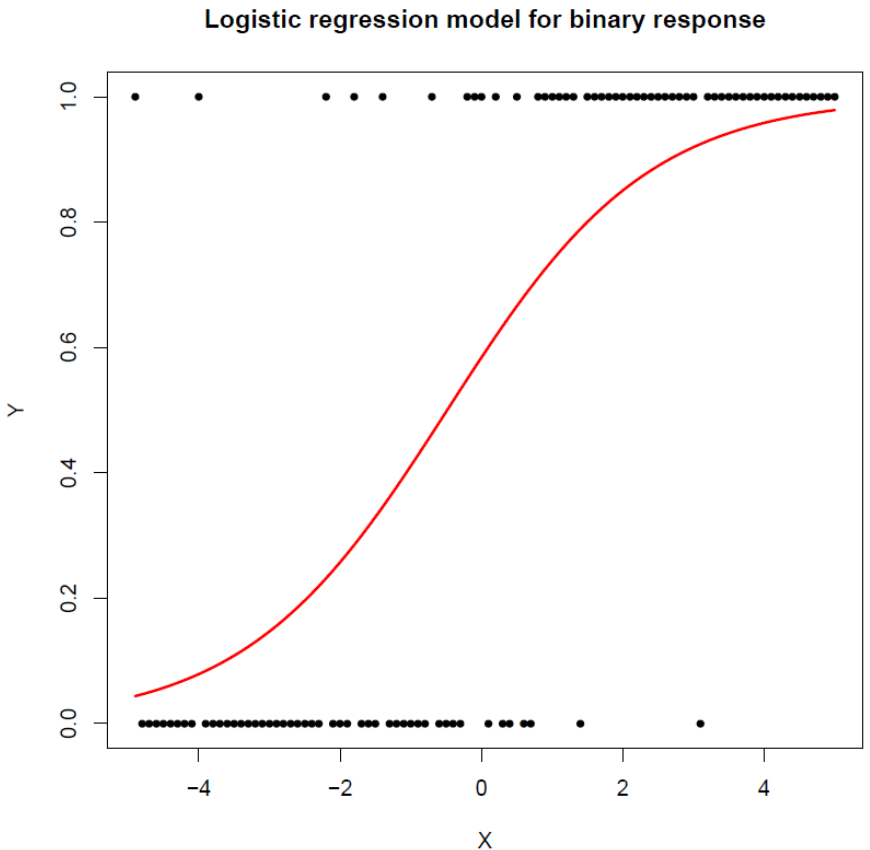
The logistic regression model

The model may be rewritten as:

$$p_i = f(X_i; \beta_0, \beta_1) = \frac{\exp(\beta_0 + \beta_1 X_i)}{1 + \exp(\beta_0 + \beta_1 X_i)}$$

The function $f^{-1}(\cdot;\cdot)$ is called the *link function*. It links the response to the explanatory variables.

The one above is called the *logistic* link function. Or short, *logit*.

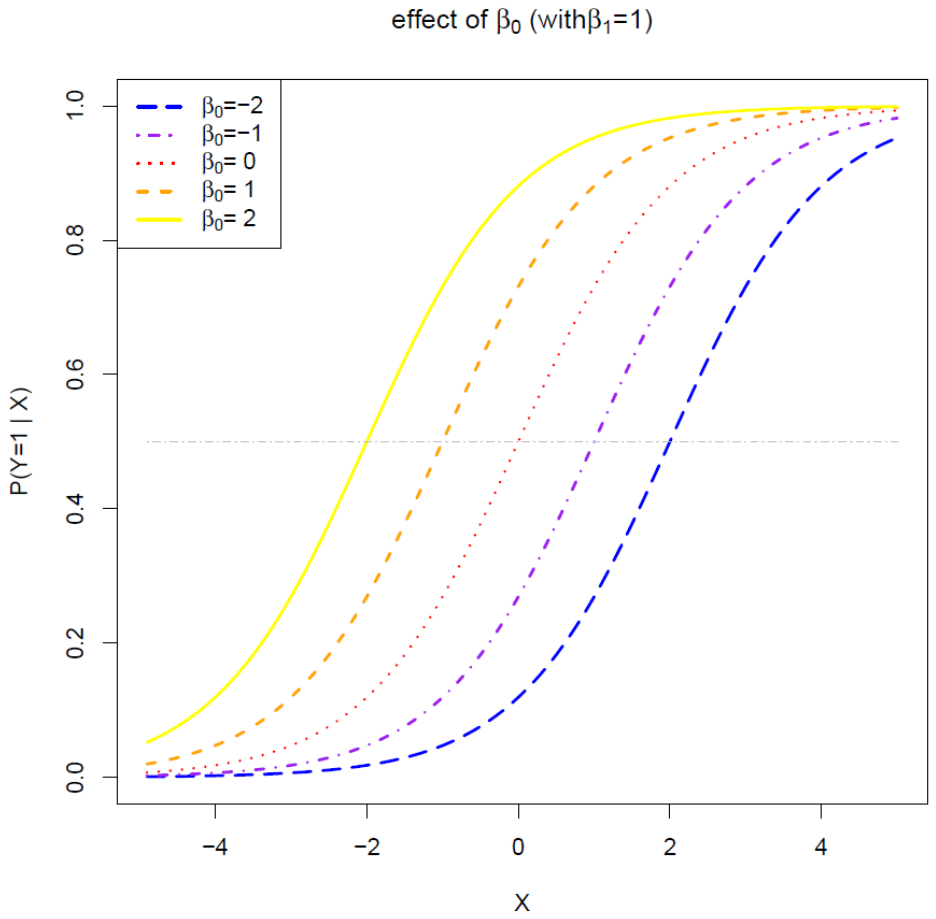


The logistic regression model

The β_0 parameter determines where (on the x-axis)

$$P(Y_i = 1 \mid X_i) = 0.5$$

The logistic link function for several values of β_0.

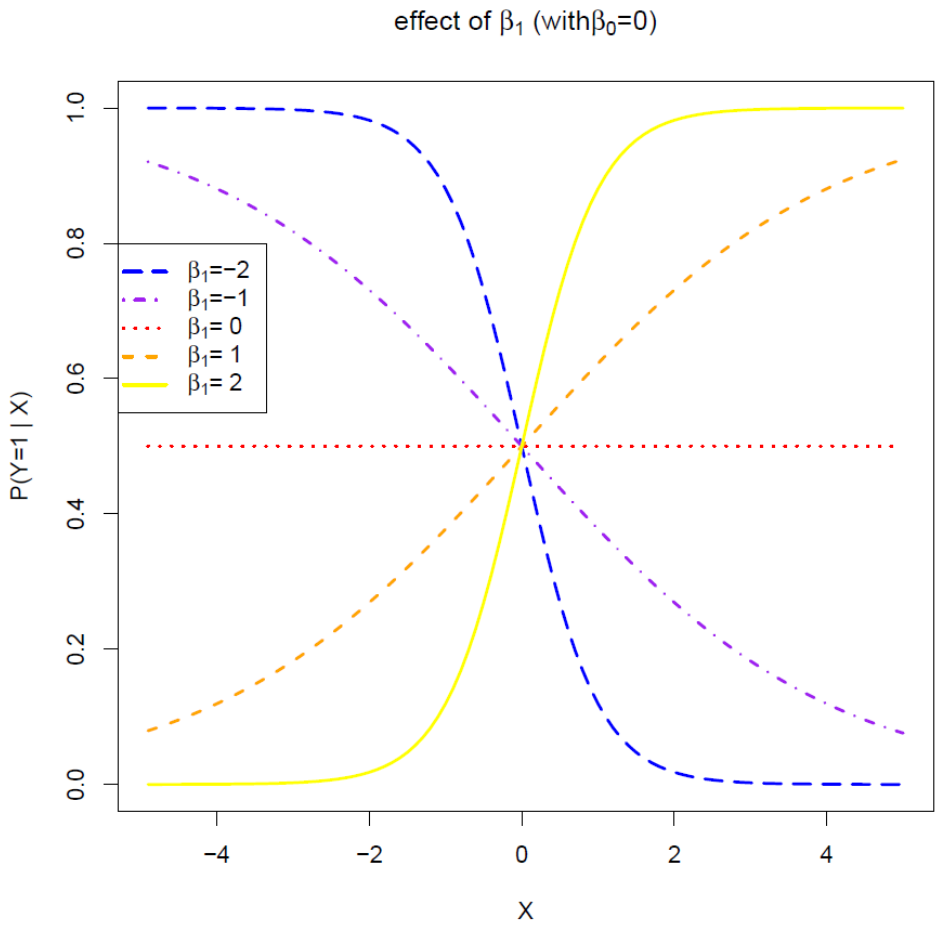


The logistic regression model

The β_1 parameter determines slope at the point where

$$P(Y_i = 1 \mid X_i) = 0.5$$

The logistic link function for several values of β_1.



The logistic regression model

The *odds* is the ratio between the probability of an event (success) and the probability that this event will not happen (failure).

$$odds = \frac{P(\text{succes})}{P(\text{failure})} = \frac{p_i}{1-p_i}$$

The *odds ratio* is the relative increase in the odds as the explanatory variable increases with 1 unit.

$$odds\ ratio = \frac{\text{odds}(x+1)}{\text{odds}(x)} = \exp(\beta_1)$$

Question

What if the confidence interval of the odds ratio contains 1?

The logistic regression model

Many other link functions for binary data exist, e.g.:

i) *Probit*:

$$p_i = \Phi_{0,1}(\mathbf{X}_i\boldsymbol{\beta})$$

ii) *Cloglog*:

$$p_i = \exp[-\exp(\mathbf{X}_i\boldsymbol{\beta})]$$

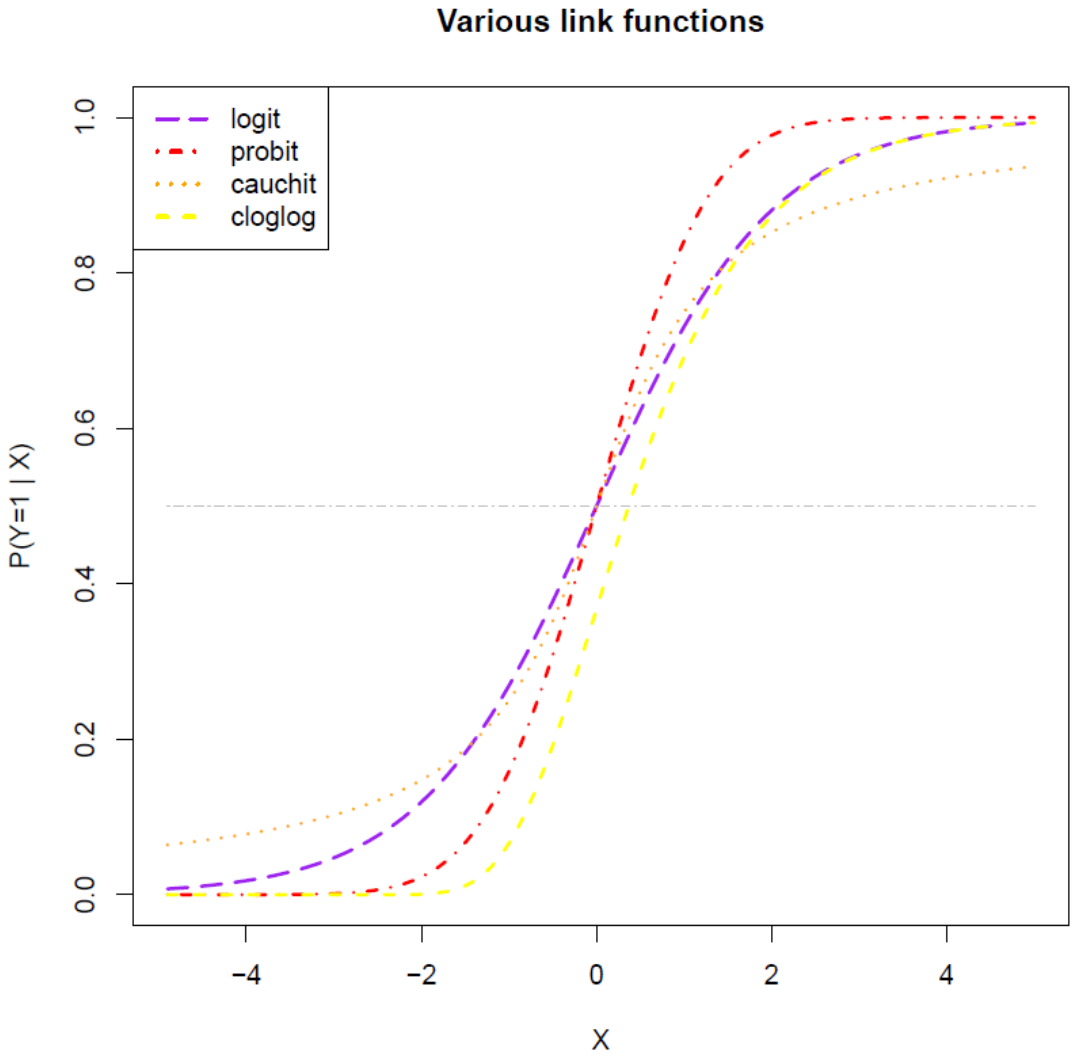
iii) *Cauchit*:

$$p_i = \frac{1}{\pi}\arctan(\mathbf{X}_i\boldsymbol{\beta}) + \frac{1}{2}$$

All these link functions are invertible.

The logistic regression model

Comparison of link function for binary data.



Estimation

Consider an experiment with Yi in {0, 1} for i=1, …, n and with each sample Xi available

The likelihood of the experiment is then:

$$\prod_{i=1}^{n} \left[P(Y_i = 1 \mid \mathbf{X}_i)\right]^{y_i} \left[P(Y_i = 0 \mid \mathbf{X}_i)\right]^{1-y_i}$$

After taking the logarithm and some ready algebra, the log-likelihood is found to be:

$$\mathcal{L}(\mathbf{Y} = \mathbf{y}, \mathbf{X}; \boldsymbol{\beta}) = \sum_{i=1}^{n} y_i \left\{\mathbf{X}_i\boldsymbol{\beta} - \log[1 + \exp(\mathbf{X}_i\boldsymbol{\beta})]\right\}$$

Estimation

Differentiate the log-likelihood w.r.t. $\underline{\beta}$, equate it zero, and obtain the estimating equation for $\underline{\beta}$.

This derivative is:

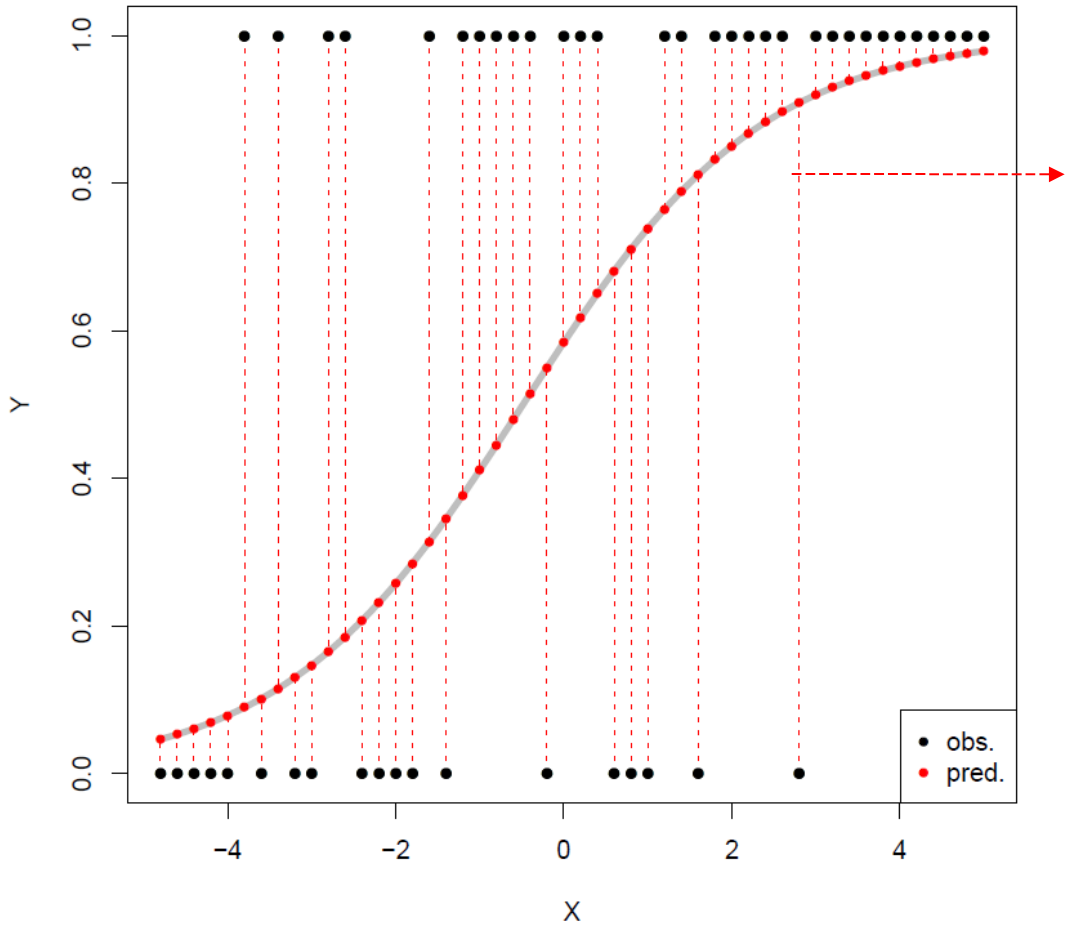
$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \left[y_i - \frac{\exp(\mathbf{X}_i \boldsymbol{\beta})}{1 + \exp(\mathbf{X}_i \boldsymbol{\beta})} \right] \mathbf{X}_i$$

difference between
observation and model

Hence, a curve is fit through data by minimizing the distance between them: at the ML estimate of $\underline{\beta}$, a weighted average of the deviations is zero.

Estimation

Interpretation of the ML estimation:



$$y_i - \frac{\exp(\mathbf{X}_i\boldsymbol{\beta})}{1+\exp(\mathbf{X}_i\boldsymbol{\beta})}$$

ML estimation considers a weighted average of these distance.

Estimation

Newton-Raphson algorithm iteratively finds the zeros of a function f(•).

Let x_0 denote an initial guess of the zero. Then, approximate f(•) around x_0 by means of a Taylor series:

$$f(x) \approx x_0 + (x - x_0) \left. \frac{df}{dx} \right|_{x=x_0}$$

Solve this for x:

$$x = x_0 - \left(\left. \frac{df}{dx} \right|_{x=x_0} \right)^{-1} f(x_0)$$

Let x_1 the solution for x, use this as the new guess and repeat the above until convergence.

Estimation

When the function f(•) has multiple arguments and is vector-valued, the Taylor approximation becomes:

$$\vec{\mathbf{f}}(\mathbf{x}) \approx \mathbf{x}_0 + J\vec{\mathbf{f}}\Big|_{\mathbf{x}=\mathbf{x}_0}(\mathbf{x}-\mathbf{x}_0)$$

with

$$J\vec{\mathbf{f}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_p} \\ \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_q}{\partial x_1} & \frac{\partial f_q}{\partial x_2} & \cdots & \frac{\partial f_q}{\partial x_p} \end{pmatrix}$$

the Jacobi matrix.

Estimation

When applied here to the estimation of $\underline{\beta}$, the Newton-Raphson update is:

$$\hat{\boldsymbol{\beta}}^{\text{new}} = \hat{\boldsymbol{\beta}}^{\text{old}} - \left(\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} \right)^{-1} \Bigg|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^{\text{old}}} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} \Bigg|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^{\text{old}}}$$

where the Hessian of the log-likelihood equals:

$$\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} = -\sum_{i=1}^{n} \frac{\exp(\mathbf{X}_i \boldsymbol{\beta})}{[1 + \exp(\mathbf{X}_i \boldsymbol{\beta})]^2} \mathbf{X}_i \mathbf{X}_i^\top$$

Iterative application of this updating formula converges to the ML estimate of $\underline{\beta}$.

Estimation

The Newton-Raphson algorithm is often reformulated to an *iteratively re-weighted least squares* algorithm.

First write the gradient and Hessian in matrix notation:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} = \mathbf{X}^\top [\mathbf{y} - g^{-1}(\mathbf{X}; \boldsymbol{\beta})]$$

$$\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} = -\mathbf{X}^\top \mathbf{W} \mathbf{X}$$

with $g^{-1}(\bullet) = \exp(\bullet) / [1 + \exp(\bullet)]$ and **W** diagonal with:

$$(\mathbf{W})_{ii} = \exp(\mathbf{X}\hat{\boldsymbol{\beta}}^{\text{old}})[1 + \exp(\mathbf{X}\hat{\boldsymbol{\beta}}^{\text{old}})]^{-2}$$

Estimation

The updating formula of the estimate then becomes:

$$
\begin{aligned}
&\hat{\boldsymbol{\beta}}^{\text{new}} \\
&\quad = \hat{\boldsymbol{\beta}}^{\text{old}} + (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top [\mathbf{y} - g^{-1}(\mathbf{X}; \boldsymbol{\beta}^{\text{old}})] \\
&\quad = (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W} \\
&\qquad\quad \times \{\mathbf{X} \hat{\boldsymbol{\beta}}^{\text{old}} + \mathbf{W}^{-1} [\mathbf{y} - g^{-1}(\mathbf{X}; \boldsymbol{\beta}^{\text{old}})]\} \\
&\quad = (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W} \mathbf{z}
\end{aligned}
$$

where

$$
\mathbf{z} = \{\mathbf{X} \hat{\boldsymbol{\beta}}^{\text{old}} + \mathbf{W}^{-1} [\mathbf{y} - g^{-1}(\mathbf{X}; \boldsymbol{\beta}^{\text{old}})]\}
$$

Estimation

The Newton-Raphson update is thus the solution to the following weighted least squares problem:

$$\hat{\boldsymbol{\beta}}^{\text{new}} = \arg\min_{\boldsymbol{\beta}} (\mathbf{z} - \mathbf{X}\boldsymbol{\beta})^{\top} \mathbf{W} (\mathbf{z} - \mathbf{X}\boldsymbol{\beta})$$

Effectively, at each iteration the *adjusted response* **z** is regressed on the covariates that comprise **X**.

Ridge logistic regression

Le Cessie, van Houwelingen (1992)

Ridge logistic regression

Problem

The logistic model parameters cannot be estimated with maximum likelihood from high-dimensional data.

Solution

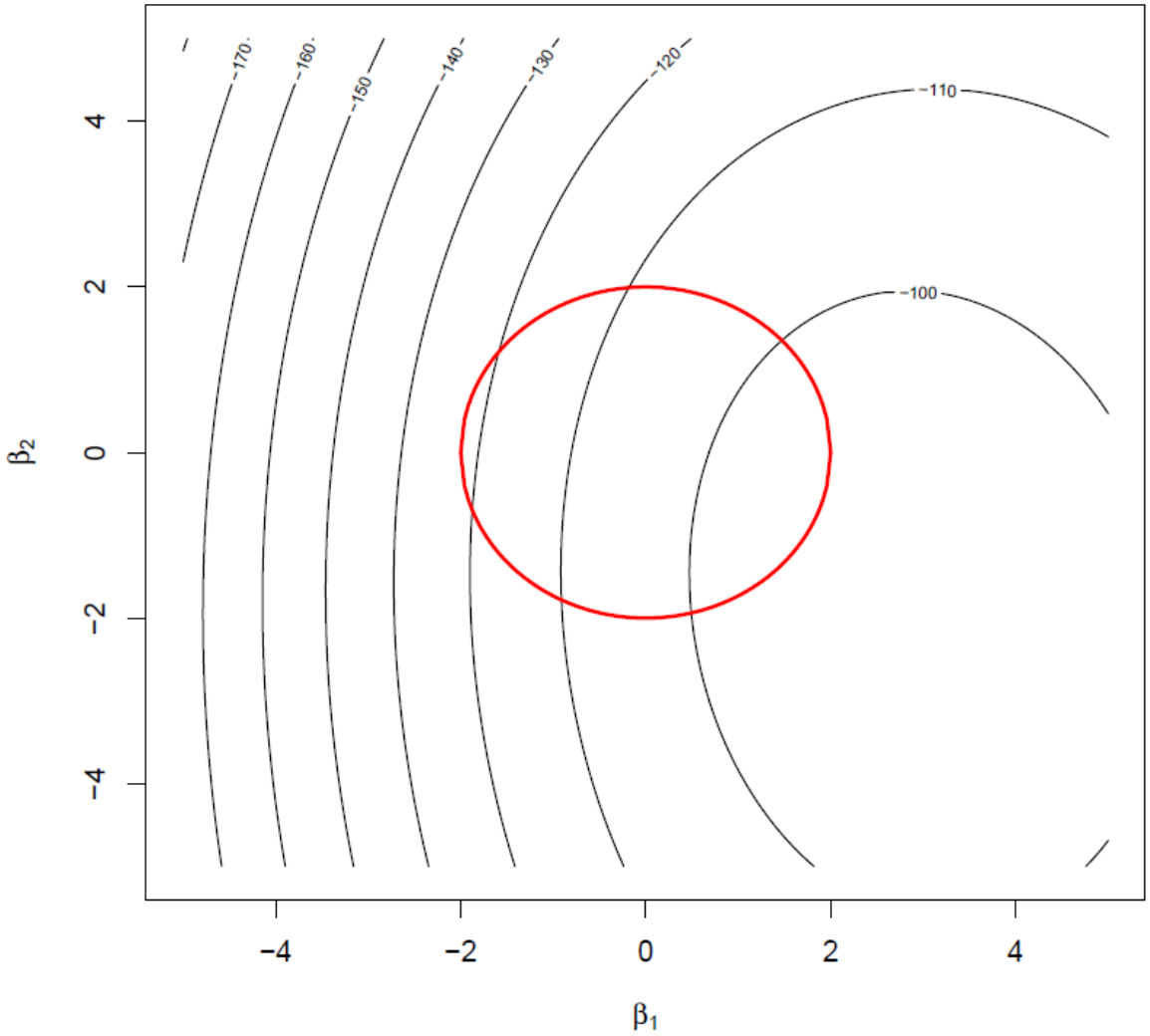
Augmentation of the loglikelihood with a ridge penalty:

$$\begin{aligned}
&\mathcal{L}^{\text{pen}}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}, \lambda) \\
&\quad = \mathcal{L}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}, \lambda) - \lambda \|\boldsymbol{\beta}\|_2^2 \\
&\quad = \sum_{i=1}^{n} Y_i \{\mathbf{X}_i \boldsymbol{\beta} - \log[1 + \exp(\mathbf{X}_i \boldsymbol{\beta})]\} - \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta}
\end{aligned}$$

This does not change the model, only the estimates!

Ridge logistic regression

Penalized log-likelihood contour + ridge constraint.



Ridge estimation

Ridge ML estimates of the logistic model parameters are found by the maximization of the penalized loglikelihood.

Again, use the Newton-Raphson algorithm for solving the (penalized) estimating equation. The gradient is now:

$$\frac{\partial \mathcal{L}^{\text{pen}}}{\partial \boldsymbol{\beta}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} - 2\lambda\boldsymbol{\beta}$$

and the Hessian:

$$\frac{\partial^2 \mathcal{L}^{\text{pen}}}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}^\top} = \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}^\top} - 2\lambda\mathbf{I}_{p\times p}$$

The rest stays the same.

Ridge estimation

Again, the Newton-Raphson algorithm is reformulated as an iteratively re-weighted least squares algorithm with the updating step modified accordingly:

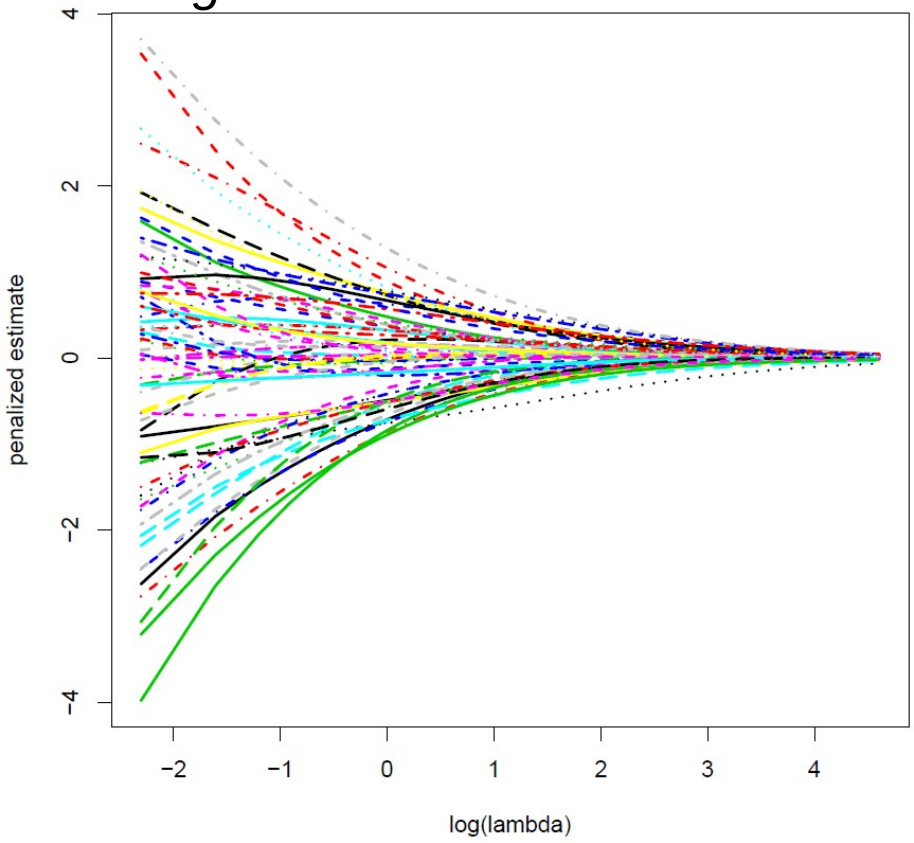
$$
\begin{aligned}
&\hat{\boldsymbol{\beta}}^{\text{new}} \\
&\quad = \hat{\boldsymbol{\beta}}^{\text{old}} + \mathbf{V}^{-1}\{\mathbf{X}^{\top}[\mathbf{y} - g^{-1}(\mathbf{X}; \boldsymbol{\beta}^{\text{old}})] - 2\lambda\boldsymbol{\beta}^{\text{old}}\} \\
&\quad = \mathbf{V}^{-1}\mathbf{V}\hat{\boldsymbol{\beta}}^{\text{old}} - 2\lambda\mathbf{V}^{-1}\hat{\boldsymbol{\beta}}^{\text{old}} \\
&\qquad\quad + \mathbf{V}^{-1}\mathbf{X}^{\top}\mathbf{W}\mathbf{W}^{-1}[\mathbf{y} - g^{-1}(\mathbf{X}; \boldsymbol{\beta}^{\text{old}})] \\
&\quad = \mathbf{V}^{-1}\mathbf{X}^{\top}\mathbf{W}\{\mathbf{X}\hat{\boldsymbol{\beta}}^{\text{old}} + \mathbf{W}^{-1}[\mathbf{y} - g^{-1}(\mathbf{X}; \boldsymbol{\beta}^{\text{old}})]\} \\
&\quad = [\mathbf{X}^{\top}\mathbf{W}\mathbf{X} + 2\lambda\mathbf{I}_{p\times p}]^{-1}\mathbf{X}^{\top}\mathbf{W}\mathbf{z}
\end{aligned}
$$

where $\mathbf{V} = \mathbf{X}^{\top}\mathbf{W}\mathbf{X} + 2\lambda\mathbf{I}_{p\times p}$ and **W** and **z** as before.

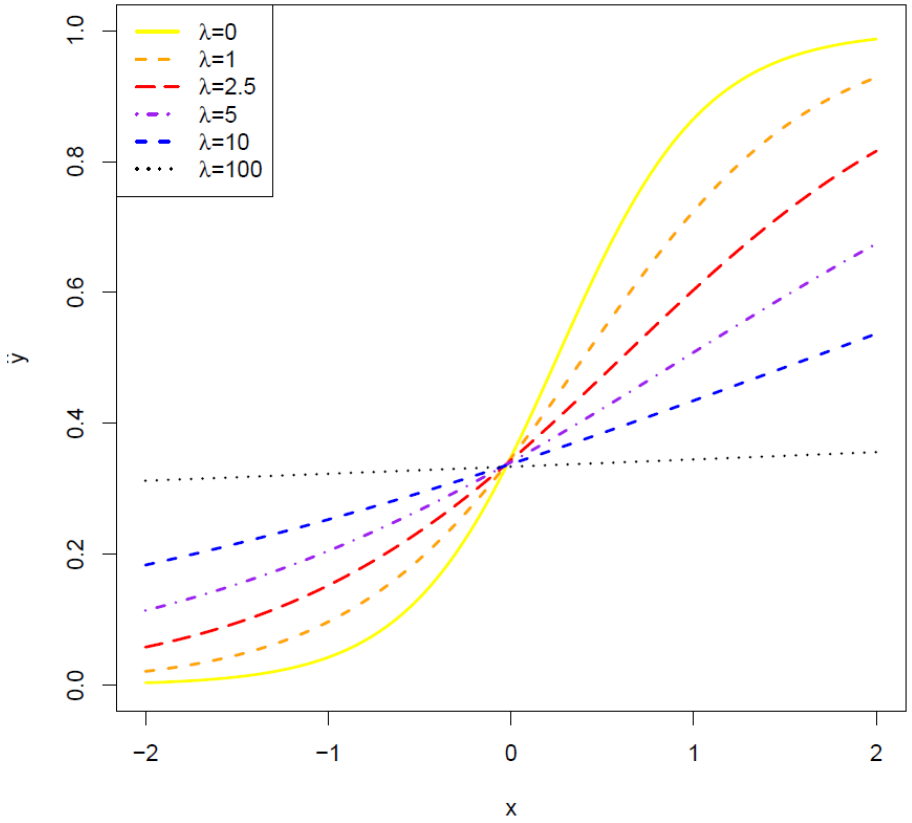
Ridge estimation

Effect of ridge penalization

Regression estimate vs λ



Y-hat vs x for various λ



Ridge estimation

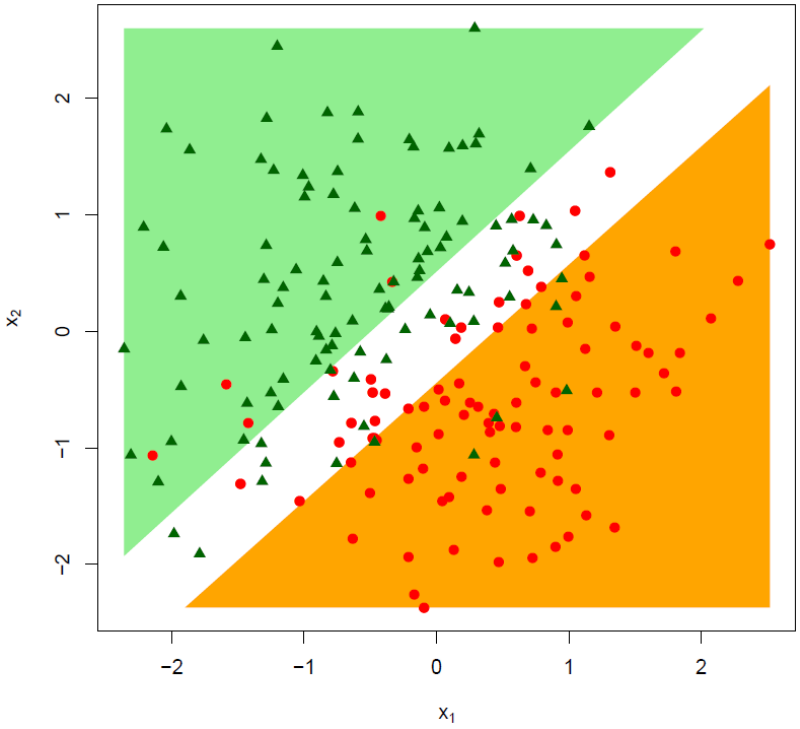
To illustrate this further, consider the resulting classification. Define the *red* and *green* domain through:

$$\{(x_1, x_2) : P(\mathbf{Y = 0} \,|\, X_1 = x_1, X_2 = x_2, \hat{\boldsymbol{\beta}}(\lambda)) > 0.75\}$$

$$\{(x_1, x_2) : P(\mathbf{Y = 1} \,|\, X_1 = x_1, X_2 = x_2, \hat{\boldsymbol{\beta}}(\lambda)) > 0.75\}$$

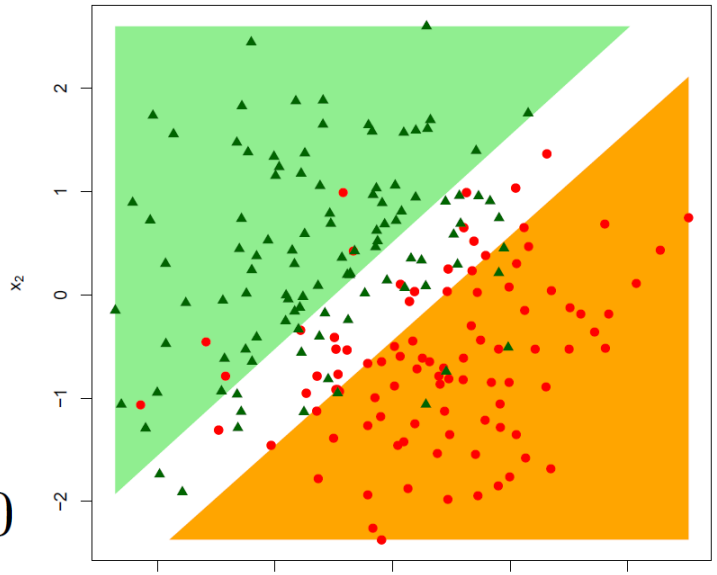
Separate design space in red, green domain.

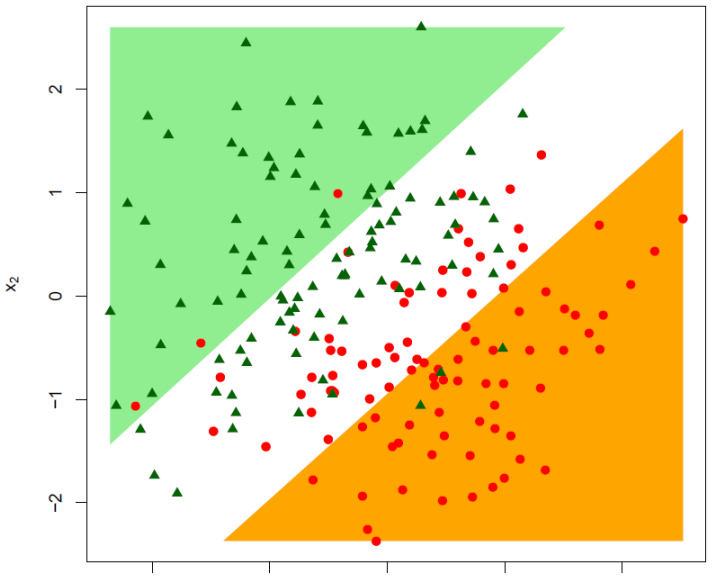
The white bar between them is the domain where samples cannot be classified with high certainty.



Ridge estimation

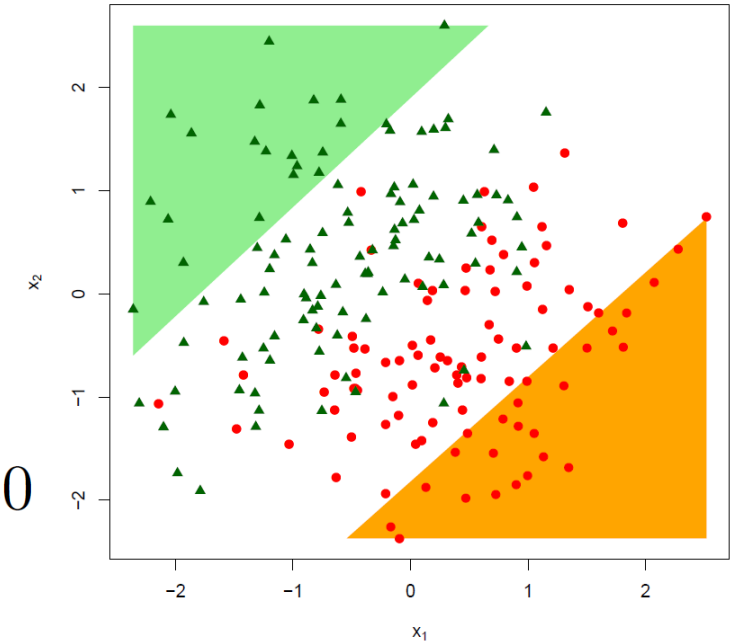
$\lambda = 0$

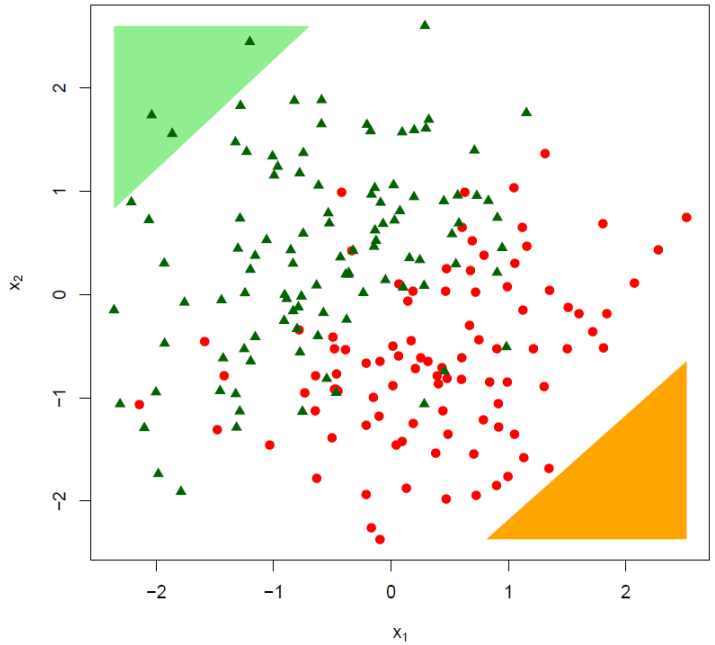




$\lambda = 10$

$\lambda = 40$





$\lambda = 100$

Moments

From the final Newton-Raphson step, we can approximate the 1st and 2nd order moment:

$$\mathbb{E}\left(\hat{\boldsymbol{\beta}}^{\text{new}}\right) = \left[\mathbf{X}^{\top}\mathbf{W}\mathbf{X} + 2\lambda\mathbf{I}_{p\times p}\right]^{-1}\mathbf{X}^{\top}\mathbf{W}\mathbb{E}(\mathbf{z})$$

$$\begin{aligned}\mathrm{Var}\left(\hat{\boldsymbol{\beta}}^{\text{new}}\right) = &\left[\mathbf{X}^{\top}\mathbf{W}\mathbf{X} + 2\lambda\mathbf{I}_{p\times p}\right]^{-1}\mathbf{X}^{\top}\mathbf{W}\left[\mathrm{Var}(\mathbf{z})\right] \\ &\times\mathbf{W}\mathbf{X}\left[\mathbf{X}^{\top}\mathbf{W}\mathbf{X} + 2\lambda\mathbf{I}_{p\times p}\right]^{-1}\end{aligned}$$

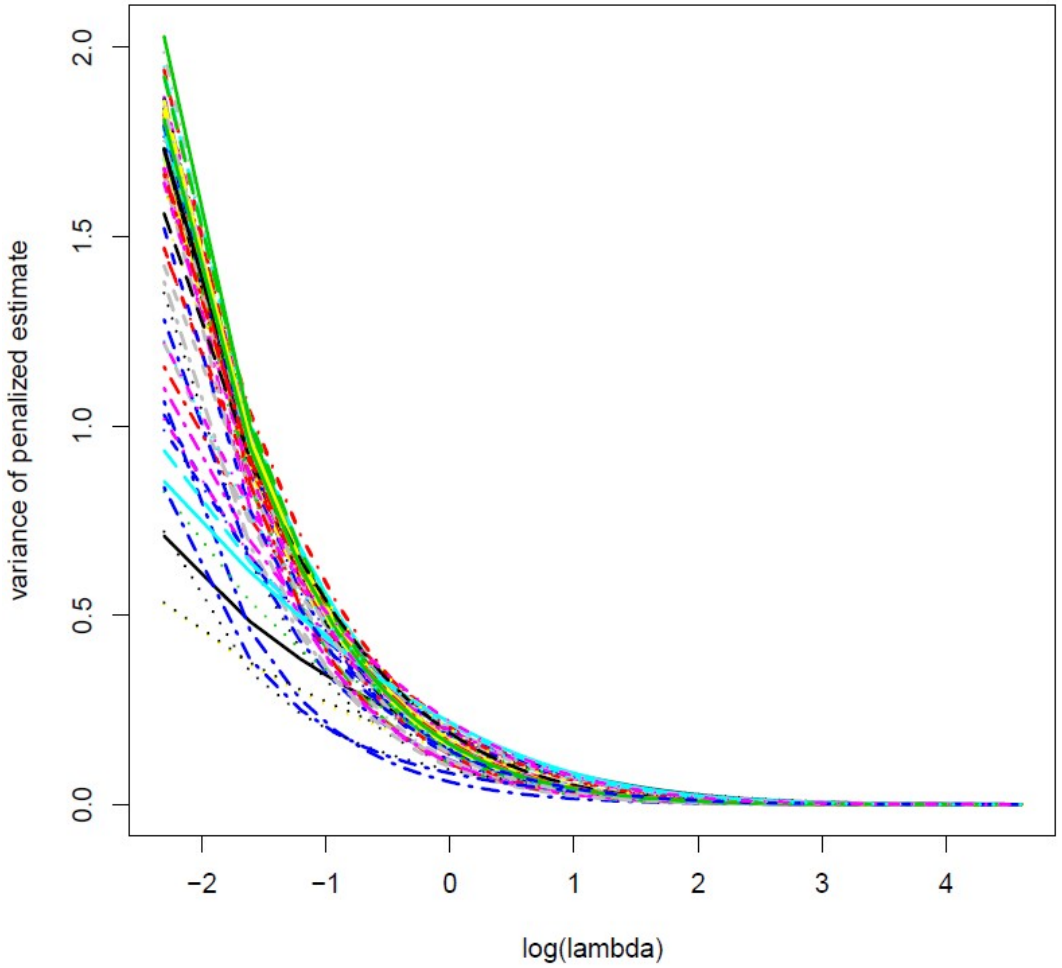
with:

$$\mathbb{E}(\mathbf{z}) = \left\{\mathbf{X}\hat{\boldsymbol{\beta}}^{\text{old}} + \mathbf{W}^{-1}\left[\mathbb{E}(\mathbf{y}) - g^{-1}(\mathbf{X};\boldsymbol{\beta}^{\text{old}})\right]\right\}$$

$$\mathrm{Var}(\mathbf{z}) = \mathbf{W}^{-1}\mathrm{Var}(\mathbf{y})\mathbf{W}^{-1} = \mathbf{W}^{-1}$$

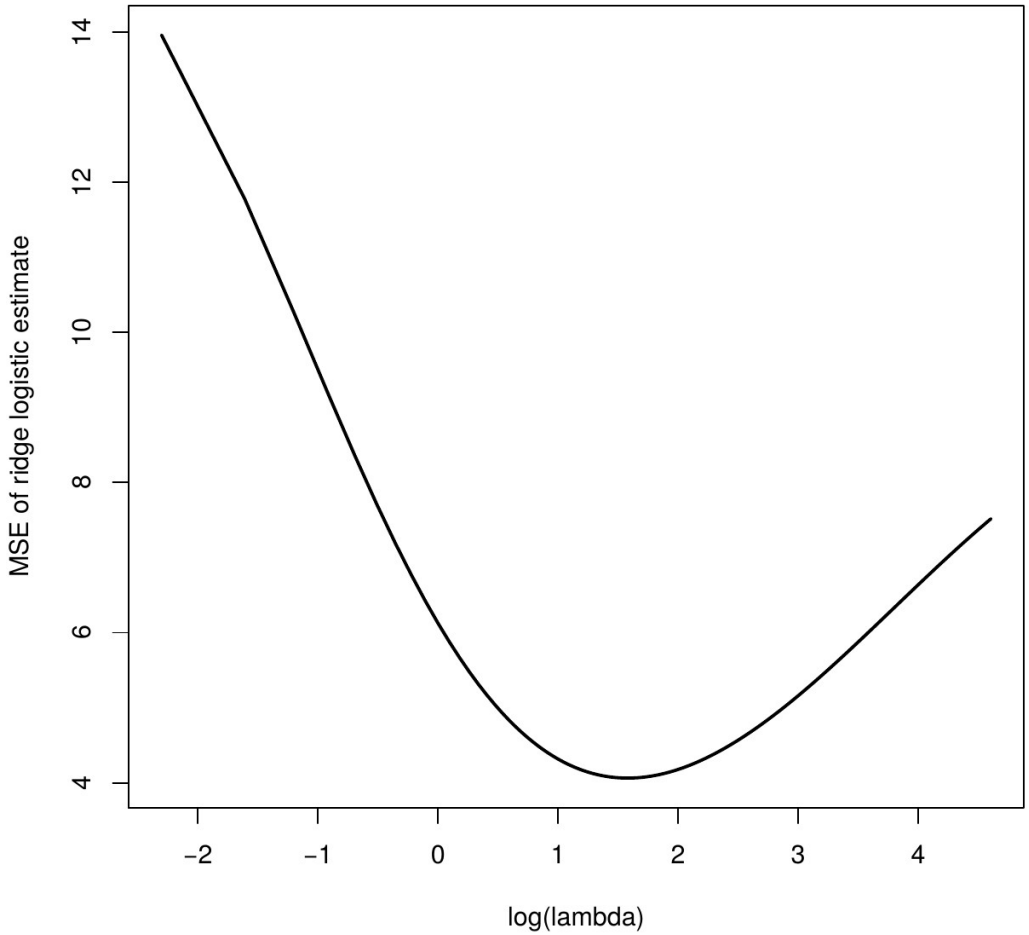
Moments

As with ridge regression, the variances of the parameter estimates vanish as the penalty parameter increases.



Moments

What about the MSE?



Question: can this be understood (from the moments)?

Bayes

Assume $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}_p, \boldsymbol{\Delta}^{-1})$.

The posterior $f_{\boldsymbol{\beta}}(\boldsymbol{\beta} \,|\, \mathbf{Y}, \mathbf{X})$ then is proportional to:

$$\prod_{i=1}^{n} \left[P(Y_i = 1 \,|\, \mathbf{X}_i)\right]^{y_i} \left[P(Y_i = 0 \,|\, \mathbf{X}_i)\right]^{1-y_i} \exp(-\tfrac{1}{2}\boldsymbol{\beta}\boldsymbol{\Delta}\boldsymbol{\beta}).$$

This is not a familiar distribution, but asymptotically normal.

Laplace's method approximates the posterior by a Gaussian
→ centered at the posterior mode, and
→ covariance equal to the curvature at the posterior mode.

The posterior mode, denoted $\hat{\boldsymbol{\beta}}_{\text{MAP}}$, coincides with the ridge logistic regression estimator.

Bishop (2006).

Bayes

For the posterior covariance approximate the logarithm of the posterior by a 2nd order Taylor series around the mode:

$$\log[f_{\boldsymbol{\beta}}(\boldsymbol{\beta} \mid \mathbf{Y}, \mathbf{X})]|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}_{\text{MAP}}}$$

$$+\tfrac{1}{2}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_{\text{MAP}})^{\top} \left.\frac{\partial^2}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^{\top}} \log[f_{\boldsymbol{\beta}}(\boldsymbol{\beta} \mid \mathbf{Y}, \mathbf{X})]\right|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}_{\text{MAP}}} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}_{\text{MAP}})^{\top}$$

Question: where is the 1st order term?

Take exponential and match arguments to that of a Gaussian to arrive at the normal approximation of the posterior:

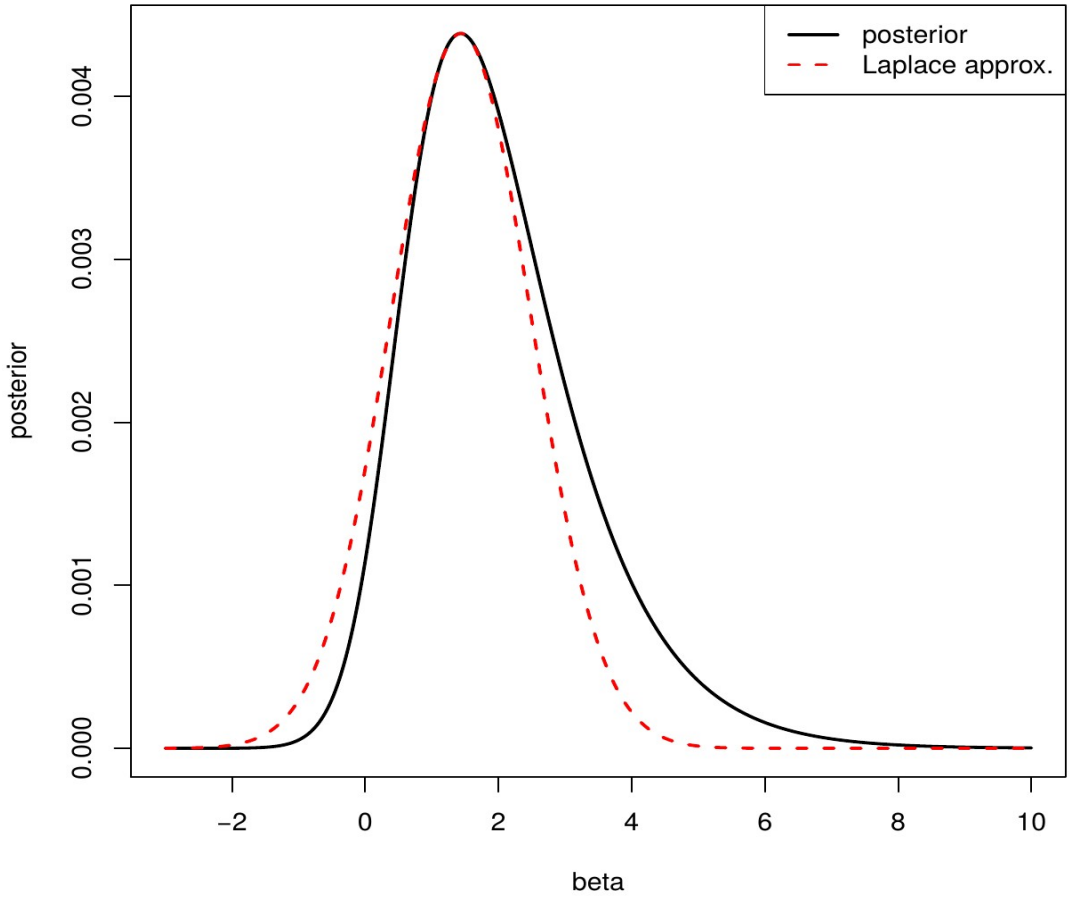
$$\boldsymbol{\beta}|\mathbf{Y}, \mathbf{X} \sim \mathcal{N}\left(\hat{\boldsymbol{\beta}}_{\text{MAP}}, \left\{\boldsymbol{\Delta} + \sum_{i=1}^{n} \frac{\exp(\mathbf{X}_i\boldsymbol{\beta})}{[1+\exp(\mathbf{X}_i\boldsymbol{\beta})]^2}\mathbf{X}_i\mathbf{X}_i^{\top}\right\}^{-1}\right).$$

Hessian

Bishop (2006).

Bayes

A Gaussian approximation is convenient, but is it any good?



The Bernstein-von Mises theorem warrants that (under smoothness conditions) the difference between posterior and its normal approximation vanishes (in probability).

van der Vaart (2007).

Penalty parameter selection

Cross-validation

Again, the LOOCV loss may be evaluated computationally efficient. Use the approximate leave-one-out estimator:

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{-i}(\lambda) &\approx \hat{\boldsymbol{\beta}}(\lambda) - \left(\frac{\partial^2 \mathcal{L}_{-i}^{\text{pen}}}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} \bigg|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}(\lambda)} \right)^{-1} \frac{\partial \mathcal{L}_{-i}^{\text{pen}}}{\partial \boldsymbol{\beta}} \bigg|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}(\lambda)} \\ &= \hat{\boldsymbol{\beta}}(\lambda) - (\mathbf{X}^\top \mathbf{W} \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} \mathbf{X}_{i,*}^\top \\ &\qquad [1 - \mathbf{H}_{ii}(\lambda)]^{-1} [Y_i - g^{-1}(\mathbf{X}_{i,*}; \hat{\boldsymbol{\beta}}(\lambda))].\end{aligned}$$

Substitute these approximations in $\sum_{i=1}^{n} \mathcal{L}[Y_i \mid \mathbf{X}_{i,*}, \hat{\boldsymbol{\beta}}_{-i}(\lambda)]$

This 'approximate' LOOCV loss often yields an optimal penalty parameter close to that produces by LOOCV loss.

Meijer, Goeman (2013).

Application

Aim

Model / predict ovarian cancer survival status.

Data

→ 295 ovarian cancer patients,
→ status (dead/alive) at end of study,
→ expression of 19990 transcript counts at study onset,
→ count transformed.

Analysis

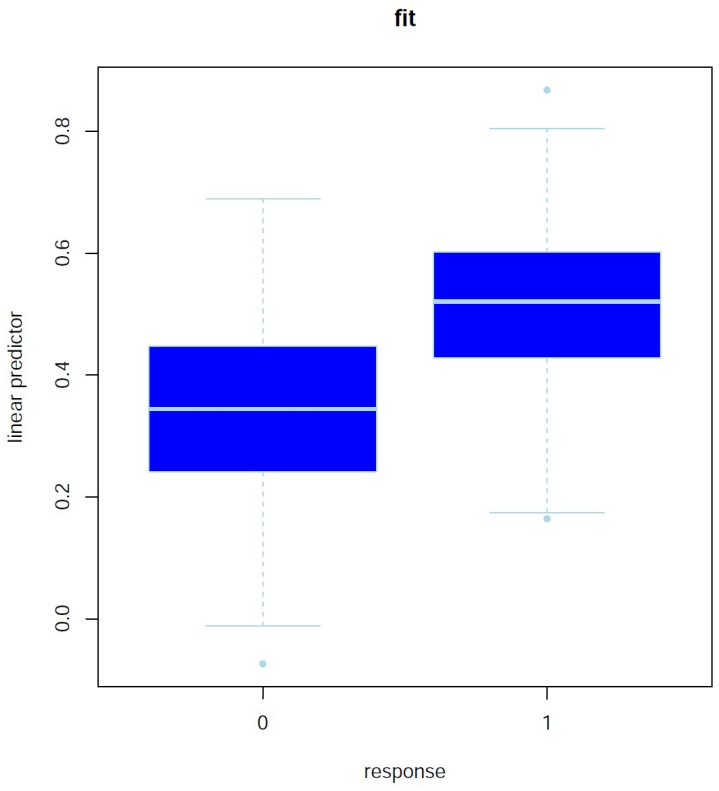
→ ridge logistic regression.
→ model: λ chosen by LOOCV.
→ prediction: double CV loop, both λ and prediction by LOOCV

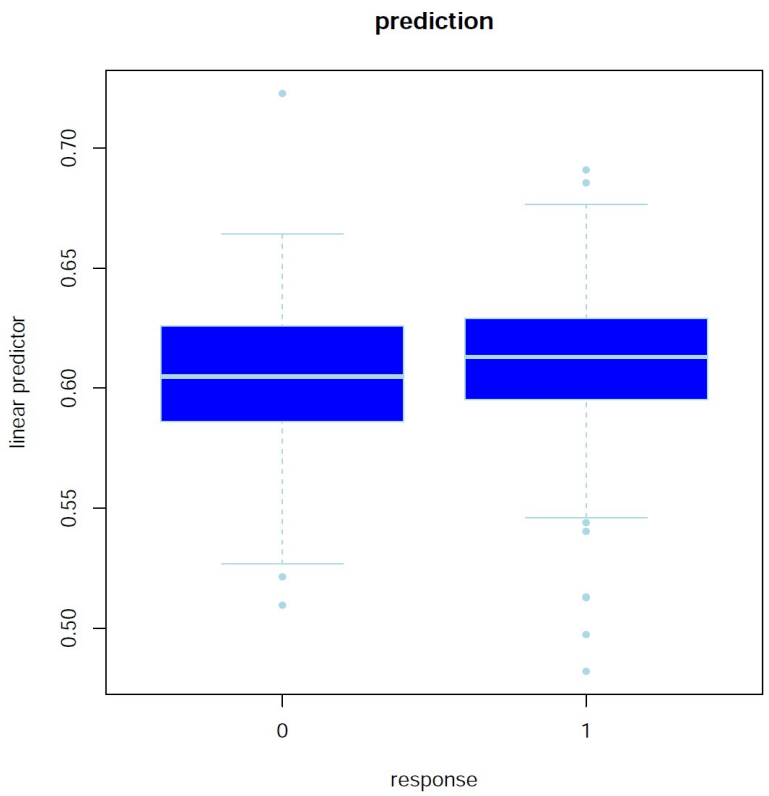
Application

Evaluation

→ Fitted model gives reasonable description for data.

→ It extrapolates poorly to new samples.





References & further reading

References & further reading

Le Cessie, S., & Van Houwelingen, J. C. (1992), “Ridge estimators in logistic regression”, *Applied Statistics*, 191-201.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Goeman, J. J. (2008), “Autocorrelated logistic ridge regression for prediction based on proteomics spectra”, *Statistical Applications in Genetics and Molecular Biology*, 7(2).

Hemmerle, W. J. (1975), “An explicit solution for generalized ridge regression”, *Technometrics*, 17(3), 309-314.

Hoerl, A. E., & Kennard, R. W. (1970), “Ridge regression: Biased estimation for nonorthogonal problems”, *Technometrics*, 12(1), 55-67.

Hosmer Jr, D.W., Lemeshow, S., Sturdivant, R.X. (2013), *Applied Logistic Regression*. John Wiley & Sons.

Lawless, J. F. (1981). Mean squared error properties of generalized ridge estimators. Journal of the American Statistical Association, 76(374), 462-466.

Meijer, R. J. and Goeman, J. J. (2013). “Efficient approximate k-fold and leave-one-out cross-validation for ridge regression”. *Biometrical Journal* , 55(2), 141–155.

Van der Vaart, A.W. (2007), *Asymptotic Statistics,* Cambridge University Press.

Van Wieringen, W.N. (2018), *Lecture notes on ridge regression,* arXiv:1509.09169.