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

Preliminary

Assumption

The data are zero-centered variate-wise.

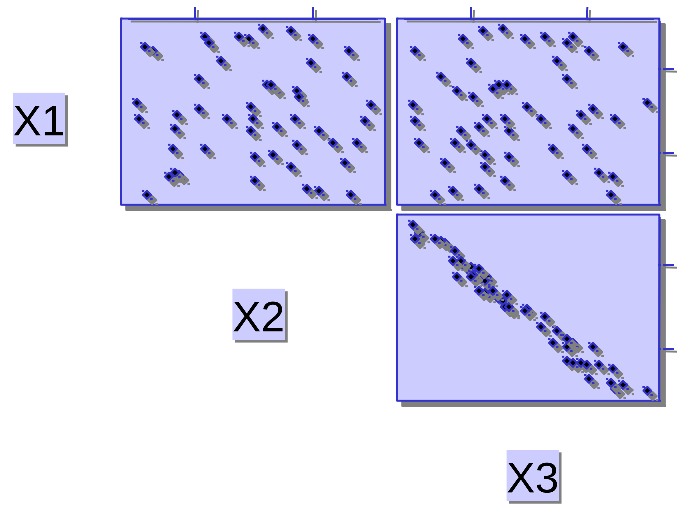
Hence, the response and the expression data of each gene is centered around zero.

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

Super-collinearity
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:
column 1 is the row-wise sum of the other two columns.

Consequence : singular $\mathbf{X}^{\mathrm{T}}\mathbf{X}$.

Problem

Super-collinearity

A square matrix that does not have an inverse is called *singular.*

A matrix $\mathbf{A}$ is singular if and only if its determinant is zero: det($\mathbf{A}$) = 0.

Example:

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

Clearly, det($\mathbf{A}$) = $a_{11}\, a_{22} - a_{12}\, a_{21} = 0$.

Hence, $\mathbf{A}$ is singular, and its inverse is undefined.

Problem

Super-collinearity

As det(**A**) is equal to the product of the eigenvalues λ_j of **A**, the matrix **A** is singular if any of the eigenvalues of **A** is zero.

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

$$\mathbf{A} = \sum_{j=1}^{p} \lambda_j \mathbf{v}_j \mathbf{v}_j^T$$

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

The inverse of **A** is then:

$$\mathbf{A}^{-1} = \sum_{j=1}^{p} \lambda_j^{-1} \mathbf{v}_j \mathbf{v}_j^T$$

Problem

Super-collinearity

A zero eigenvalue produces an undefined inverse.

Example:

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

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

$$\mathbf{A}^{-1} = \frac{1}{5}\mathbf{v}_1\mathbf{v}_1^T + \frac{1}{0}\mathbf{v}_2\mathbf{v}_2^T$$

Even **R** cannot save you now:

```
> 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

Super-collinearity

Consequence : singular $\mathbf{X}^{\mathrm{T}}\mathbf{X}$.

So?

Recall the estimator of the regression coefficients (and its variance):

$$\begin{aligned} \hat{\beta} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \\ \mathrm{Var}(\hat{\beta}) &= \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} \end{aligned}$$

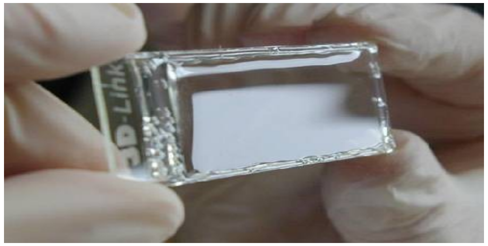
These are only defined if $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$ exits.

Hence, supercollinearity → regression coefficients cannot be estimated.

Problem

Super-collinearity occurs in a *high-dimensional situation*, that is, where the number of covariates exceeds the number of samples ($p > n$).

Microarrays measure the expression of many genes simultaneously (which genes are expressed and to what extent).



Microarray studies involve hundreds (n) samples, whose expression profiles of thousands (p) genes are generated ($p >> n$).

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

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

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:

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

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.

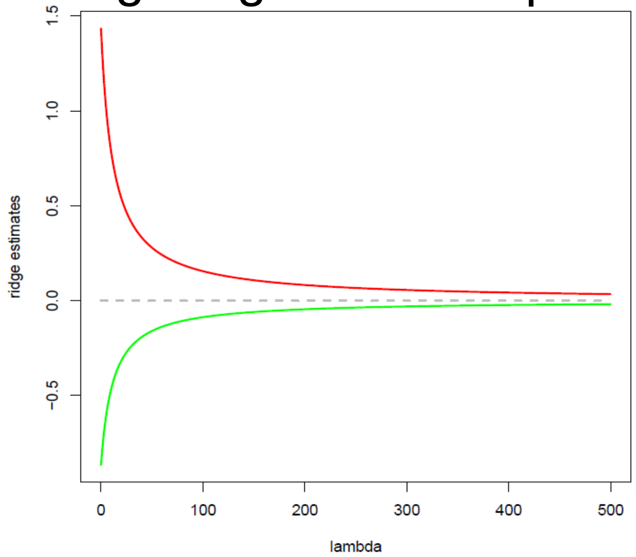
Ridge regression

Example (continued)

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

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

Ridge regularization path



E.g.: λ = 1:

```
b(1)  = (0.614, 0.548, 0.066)ᵀ.
```

E.g.: λ = 10:

```
b(10) = (0.269, 0.267, 0.002)ᵀ.
```

Question

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

Ridge regression

Ridge vs. OLS estimator

In the special case of orthonormality, there is a simple relation between the ridge estimator and the OLS estimator.

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

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

Clear, <X[,1], X[,1]> = ¼ [$(-1)^2 + (-1)^2 + 1^2 + 1^2$] = 1,

and <X[,1], X[,2]> = ¼ [-1 * -1 + -1 * 1 + 1 * -1 + 1 * 1] = 0.

Ridge regression

Ridge vs. OLS estimator

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

Check this for the example on the previous slide.

Then, the ridge estimator is proportional to the OLS estimator:

$$\begin{aligned}
\hat{\beta}(\lambda) &= (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} \\
&= (\mathbf{I} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} \\
&= (1+\lambda)^{-1}\mathbf{I}\mathbf{X}^T\mathbf{Y} \\
&= (1+\lambda)^{-1}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \\
&= (1+\lambda)^{-1}\hat{\beta}
\end{aligned}$$

Ridge regression

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

The *singular value decomposition* of a matrix **X** is:

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

where:

- $\mathbf{D}$ (n x n)-diagonal matrix with the singular values,
- $\mathbf{U}$ (n x n)-matrix with columns containing the left singular vectors, and
- $\mathbf{V}$ (p x n)-matrix with columns containing the right singular vectors.

Ridge regression

The OLS estimator can then be rewritten in terms of the SVD-matrices as:

$$
\begin{aligned}
\hat{\beta} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \\
&= (\mathbf{VDU}^T\mathbf{UDV}^T)^{-1}\mathbf{VDU}^T\mathbf{Y} \\
&= (\mathbf{VD}^2\mathbf{V}^T)^{-1}\mathbf{VDU}^T\mathbf{Y} \\
&= \mathbf{VD}^{-2}\mathbf{V}^T\mathbf{VDU}^T\mathbf{Y} \\
&= \mathbf{VD}^{-2}\mathbf{DU}^T\mathbf{Y}
\end{aligned}
$$

Role of the singular values

Ridge regression

Similarly, the ridge estimator can be rewritten in terms of the SVD-matrices as:

$$
\begin{aligned}
\hat{\boldsymbol{\beta}}(\lambda) &= (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} \\
&= (\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{V}^T + \lambda\mathbf{I})^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{Y} \\
&= (\mathbf{V}\mathbf{D}^2\mathbf{V}^T + \lambda\mathbf{V}\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{Y} \\
&= \mathbf{V}(\mathbf{D}^2 + \lambda\mathbf{I})^{-1}\mathbf{V}^T\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{Y} \\
&= \mathbf{V}\underbrace{(\mathbf{D}^2 + \lambda\mathbf{I})^{-1}\mathbf{D}}_{\text{Role of the singular values}}\mathbf{U}^T\mathbf{Y}
\end{aligned}
$$

Ridge regression

Combining the two results and writing

$$(\mathbf{D})_{jj} = d_{jj}$$

we have:

$$\underbrace{d_{jj}^{-1}}_{\text{OLS}} \geq \underbrace{d_{jj}/(d_{jj}^2 + \lambda)}_{\text{ridge}}$$

Thus, the ridge penalty shrinks the singular values.

Ridge regression

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 given by:

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

Moments of the ridge estimator

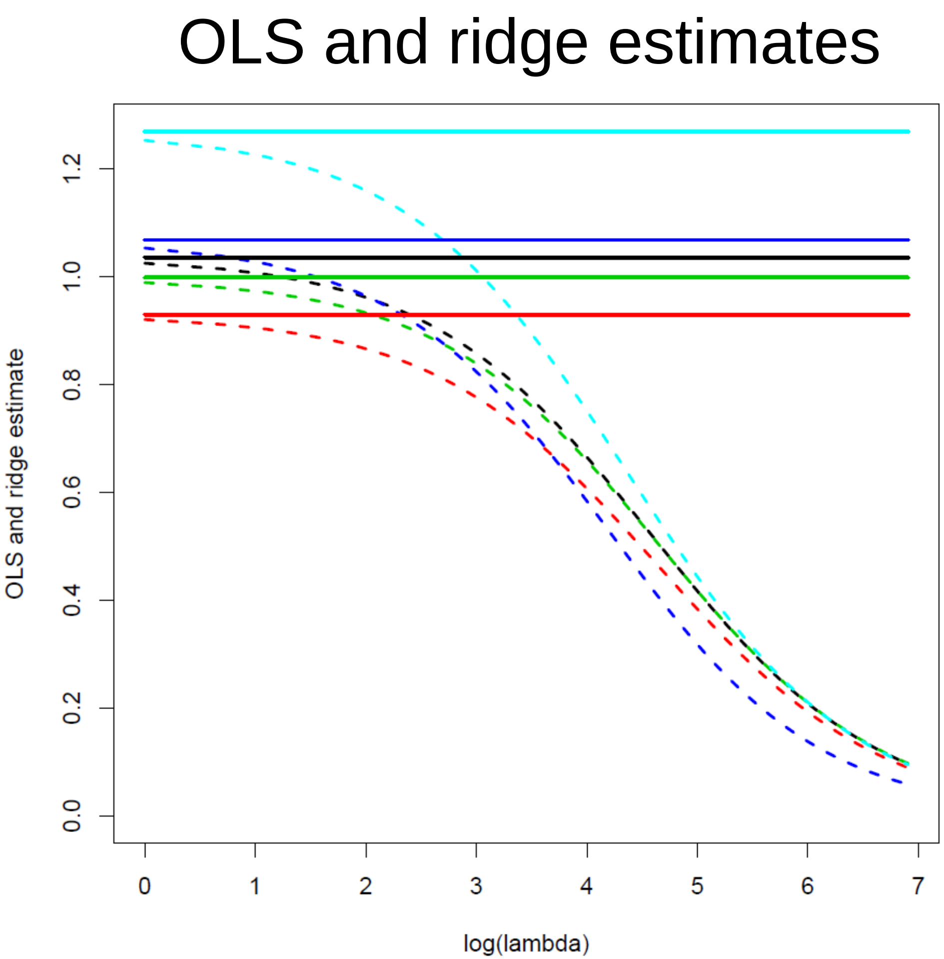
Moments

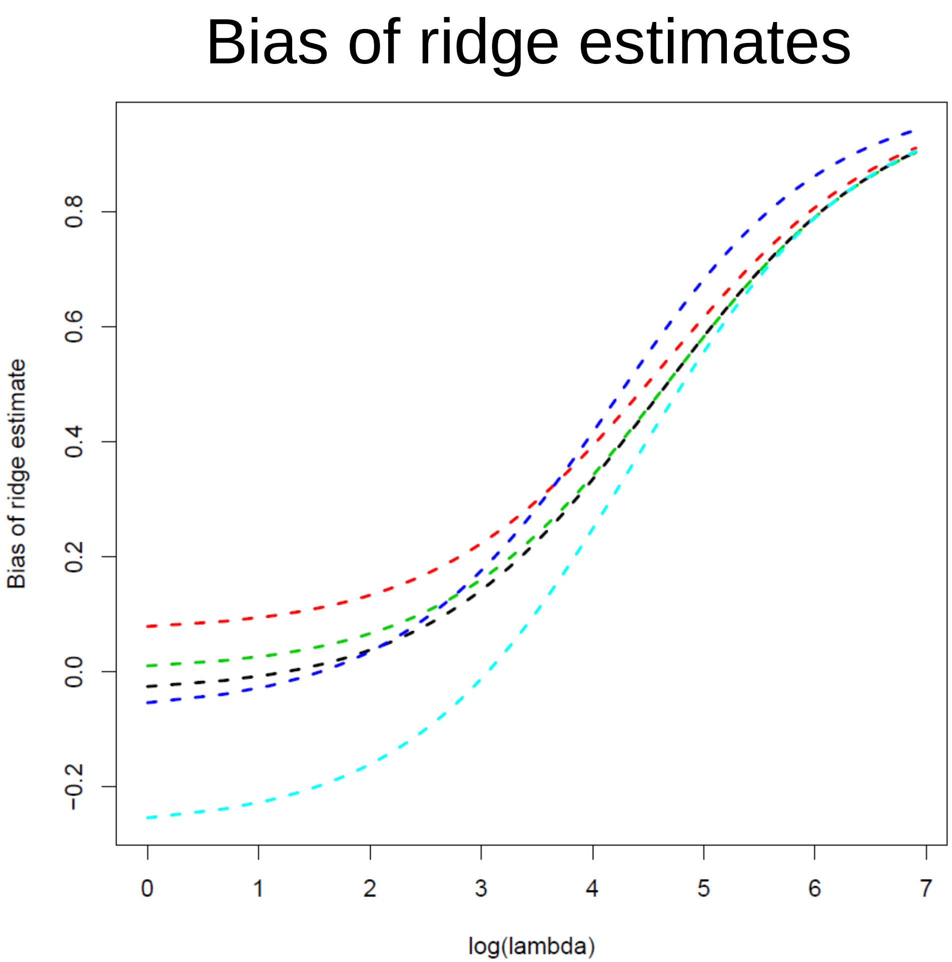
The expectation of the ridge estimator:

$$\begin{aligned}
E\left[\hat{\boldsymbol{\beta}}(\lambda)\right] &= E\left[(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}\right] \\
&= E\left\{[\mathbf{I}+\lambda\,(\mathbf{X}^T\mathbf{X})^{-1}]^{-1}(\mathbf{X}^T\mathbf{X})^{-1}\,\mathbf{X}^T\mathbf{Y}\right\} \\
&= E\left\{[\mathbf{I}+\lambda\,(\mathbf{X}^T\mathbf{X})^{-1}]^{-1}\,\hat{\boldsymbol{\beta}}\right\} \\
&= [\mathbf{I}+\lambda\,(\mathbf{X}^T\mathbf{X})^{-1}]^{-1}\,E\left(\hat{\boldsymbol{\beta}}\right) \\
&= [\mathbf{I}+\lambda\,(\mathbf{X}^T\mathbf{X})^{-1}]^{-1}\,\boldsymbol{\beta} \\
&\neq \boldsymbol{\beta}
\end{aligned}$$

Unbiased when $\lambda = 0$

Moments


OLS and ridge estimates
OLS and ridge estimate
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
1
2
3
4
5
6
7
log(lambda)


Bias of ridge estimates
Bias of ridge estimate
−0.2
0.0
0.2
0.4
0.6
0.8
0
1
2
3
4
5
6
7
log(lambda)

Moments

We now calculate the variance of the ridge estimator.

Hereto define:

$$\mathbf{W}_\lambda = [\mathbf{I} + \lambda(\mathbf{X}^T\mathbf{X})^{-1}]^{-1}$$

Then note that:

$$\begin{aligned}
\mathbf{W}_\lambda \hat{\boldsymbol{\beta}} &= [\mathbf{I} + \lambda(\mathbf{X}^T\mathbf{X})^{-1}]^{-1}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \\
&= (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \\
&= (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} \\
&= \hat{\boldsymbol{\beta}}(\lambda)
\end{aligned}$$

Moments

The variance of the ridge estimator is now straightforwardly obtained:

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

where we have used that:

$$\mathrm{Cov}(\mathbf{AX}, \mathbf{BY}) = \mathbf{A}\,\mathrm{Cov}(\mathbf{X}, \mathbf{Y})\,\mathbf{B}^T$$

Moments

The variance of the ridge estimator is thus:

$$\text{Var}[\hat{\beta}(\lambda)] \quad = \quad \sigma^2 \mathbf{W}_\lambda (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{W}_\lambda^T$$

We can now compare this to the variance of the OLS estimator. It turns out that:

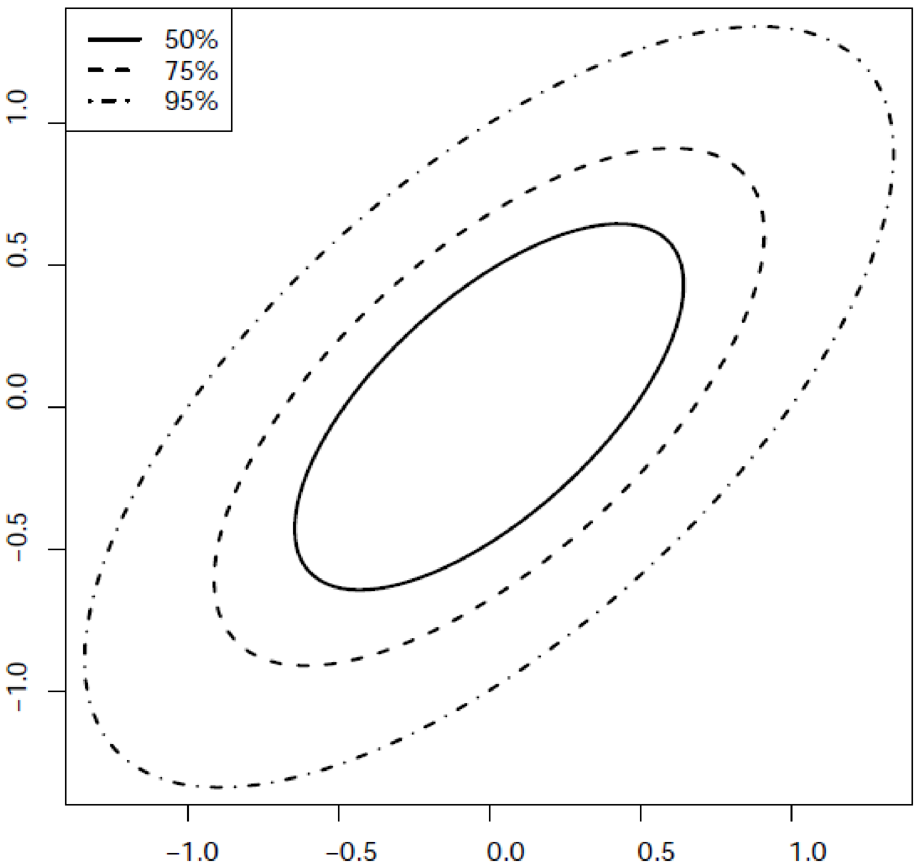
$$\text{Var}(\hat{\beta}) \quad \succeq \quad \text{Var}[\hat{\beta}(\lambda)]$$

This means that the variance of the OLS estimator is larger than that of the ridge estimator (in the sense that their difference is non-negative definite).

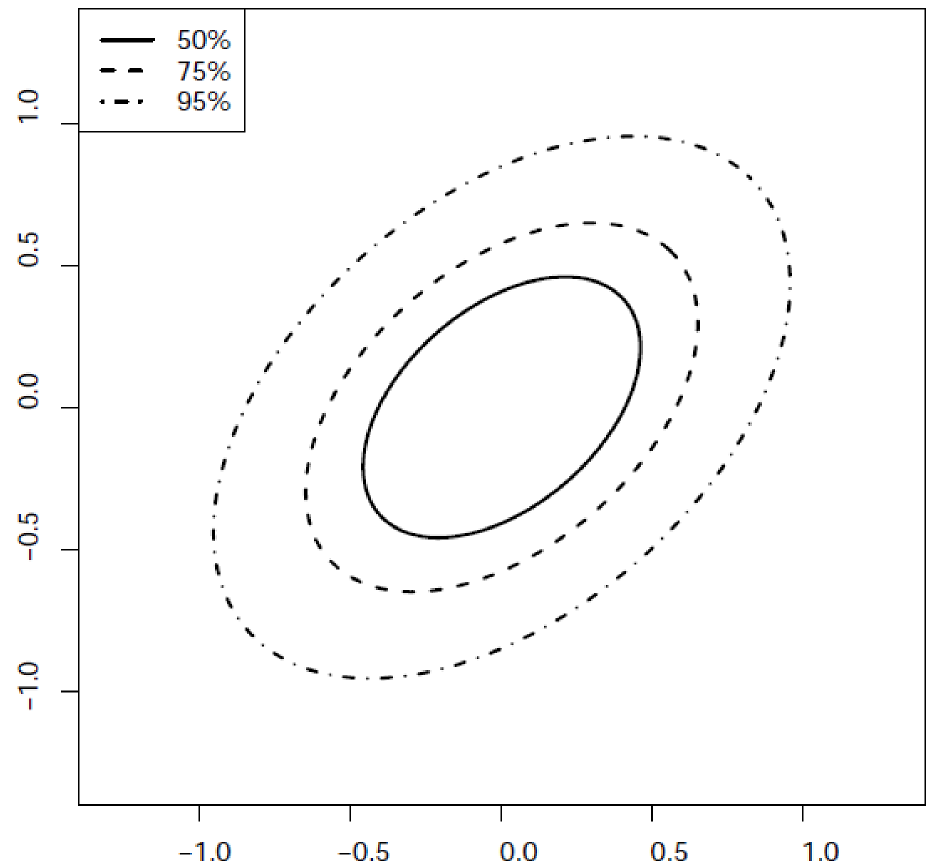
Moments

Contour plot of the variance of the ridge estimator

λ = 0



λ > 0

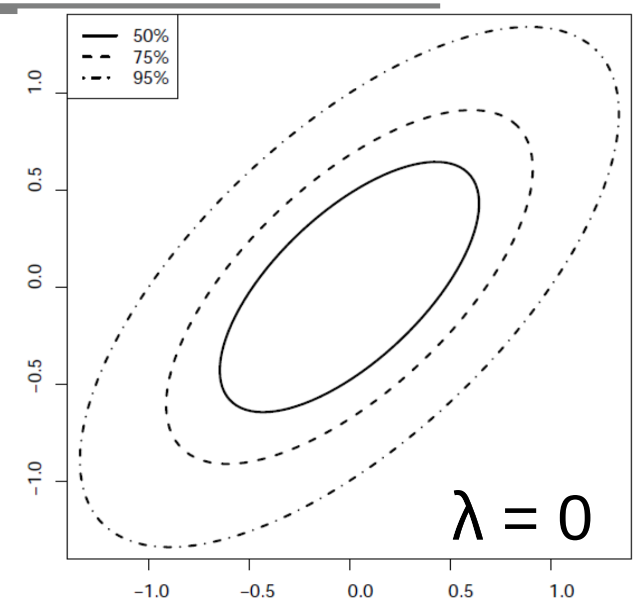


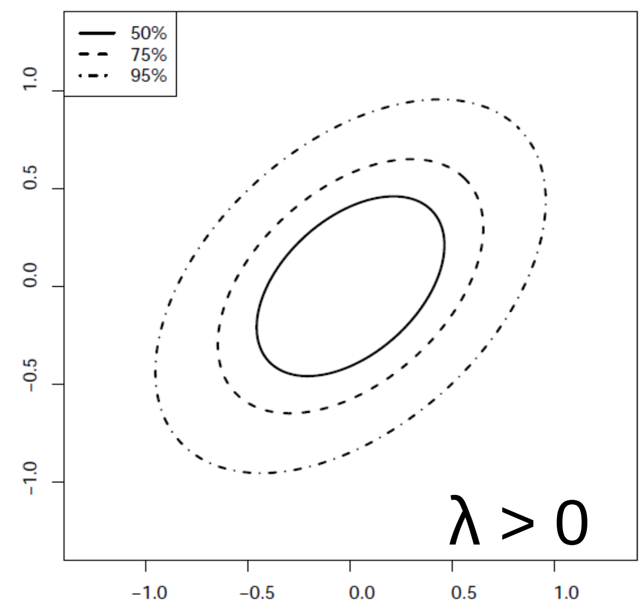
Moments

Question
Prove that the confidence ellipsoid of ridge estimator is indeed smaller than 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

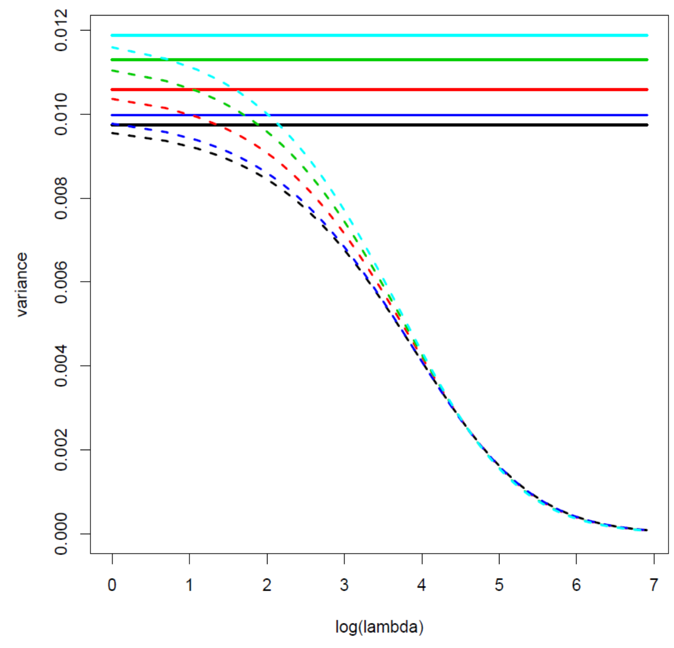
Ridge vs. OLS estimator

In the orthonormal case, we have $\mathrm{Var}(\hat{\beta}) \quad = \quad \sigma^2 \mathbf{I}$

and

$$\begin{aligned} \mathrm{Var}[\hat{\beta}(\lambda)] &= \sigma^2 \mathbf{W}_\lambda (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{W}_\lambda^T \\ &= \sigma^2 [\mathbf{I} + \lambda \mathbf{I}]^{-1} \, \mathbf{I} \, \{[\mathbf{I} + \lambda \mathbf{I}]^{-1}\}^T \\ &= \sigma^2 (1 + \lambda)^{-2} \mathbf{I} \end{aligned}$$

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



Mean squared error

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

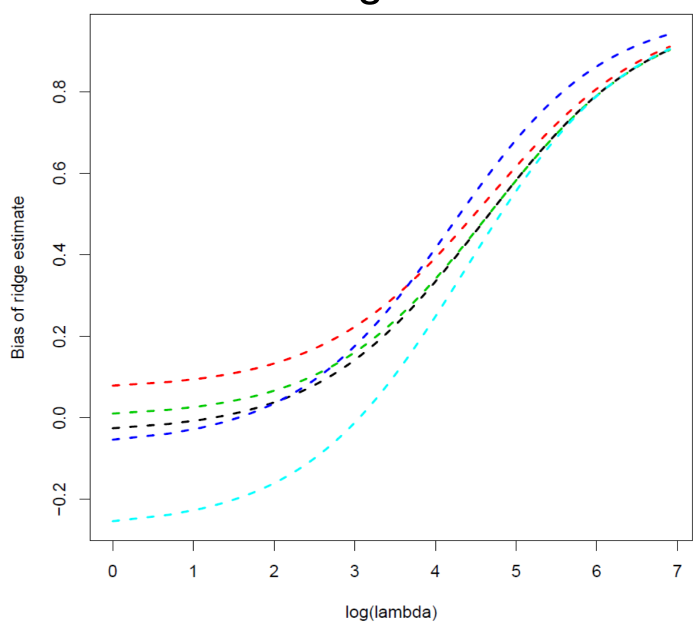
Question

So far:

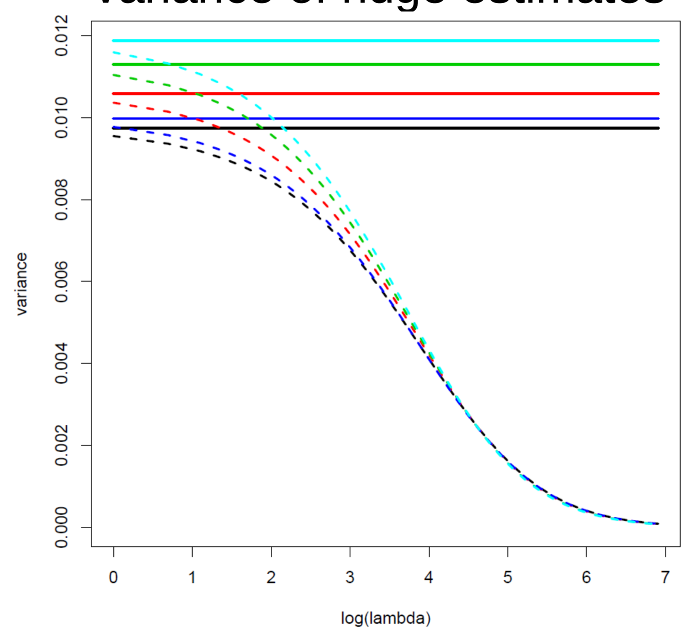
→ bias increases with λ, and

→ 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:

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

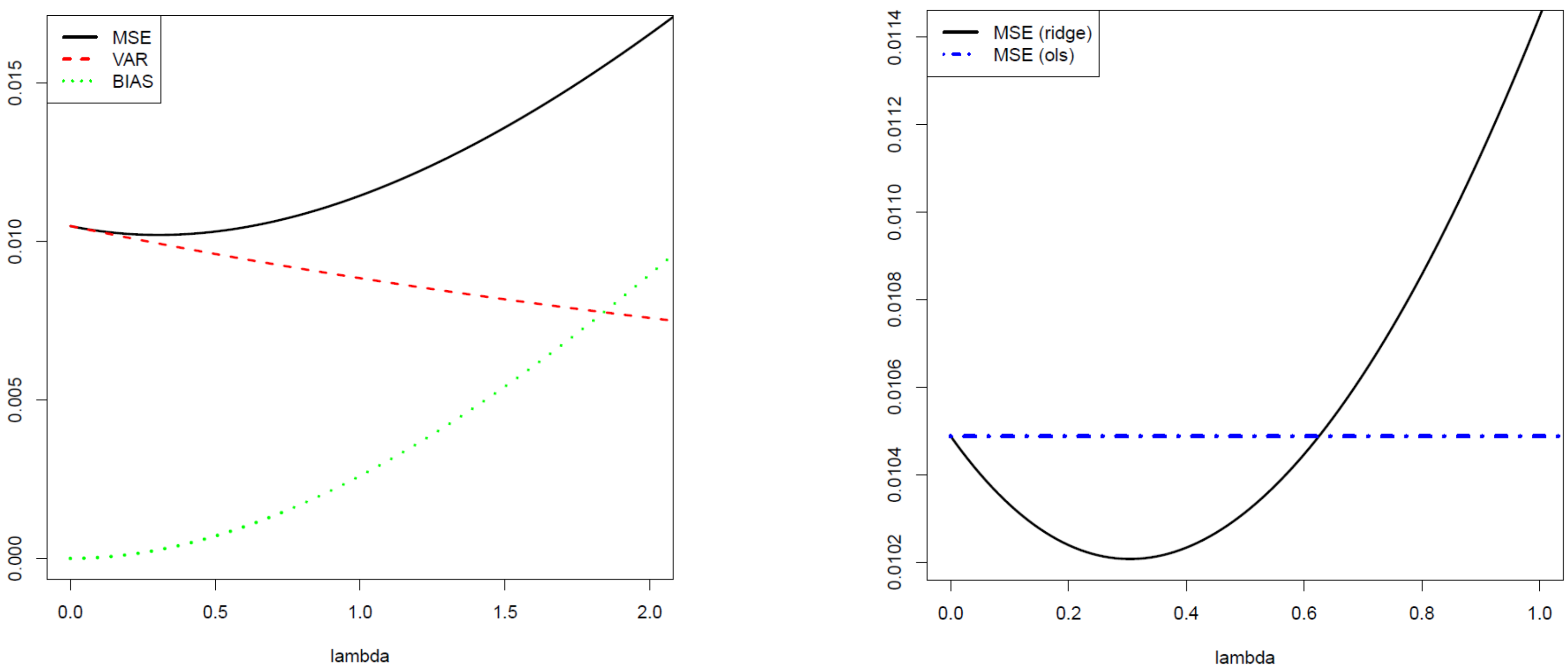
sum of variances of
the ridge estimator

"squared bias" of
the ridge estimator

Mean squared error

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

For $\lambda < 0.6$, $MSE(\lambda) < MSE(0)$ and the ridge estimator outperforms the OLS estimator.



Mean squared error

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

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

Practice
Cross-validation. The data set is split many times into a training and test set. For each split the regression parameters are estimated for all choices of λ using the training data. Estimated parameters are evaluated on the test set. The λ that on average over the test sets performs best (in some sense) is selected.

Mean squared error

Ridge vs. OLS estimator

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

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

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:

$$\lambda \quad = \quad \frac{p\,\sigma^2}{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta}}$$

the ratio between the error variance and the ‘signal’.

Constrained estimation

Constrained estimation

The ad-hoc ridge estimator minimizes the following loss function:

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\beta};\lambda) &= \|\mathbf{Y} - \mathbf{X}\,\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 \\
&= \underbrace{\sum_{i=1}^{n}(Y_i - \mathbf{X}_{i*}\,\boldsymbol{\beta})^2}_{\text{sum of squares}} + \lambda\underbrace{\sum_{j=1}^{p}\beta_j^2}_{\text{ridge penalty}}
\end{aligned}
$$

- $\lambda \geq 0$ penalty parameter
- Penalty deals with (super)-collinearity

Constrained estimation

To see this, take the derivative:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta}; \lambda) &= -2\, \mathbf{X}^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + 2\, \lambda\, \mathbf{I}\, \boldsymbol{\beta} \\ &= -2\, \mathbf{X}^T \mathbf{Y} + 2\, (\mathbf{X}^T \mathbf{X} + \lambda\, \mathbf{I}) \boldsymbol{\beta}\end{aligned}$$

where we have used some matrix calculus (beyond scope of the course).

Equate the derivative to zero and solve:

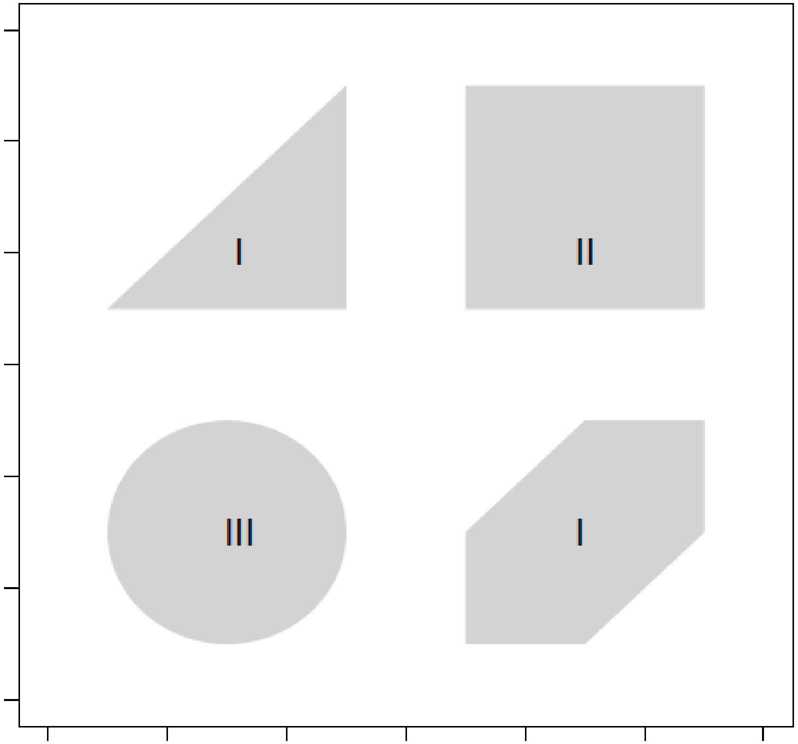
$$\hat{\boldsymbol{\beta}}(\lambda) = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{Y}$$

Constrained estimation

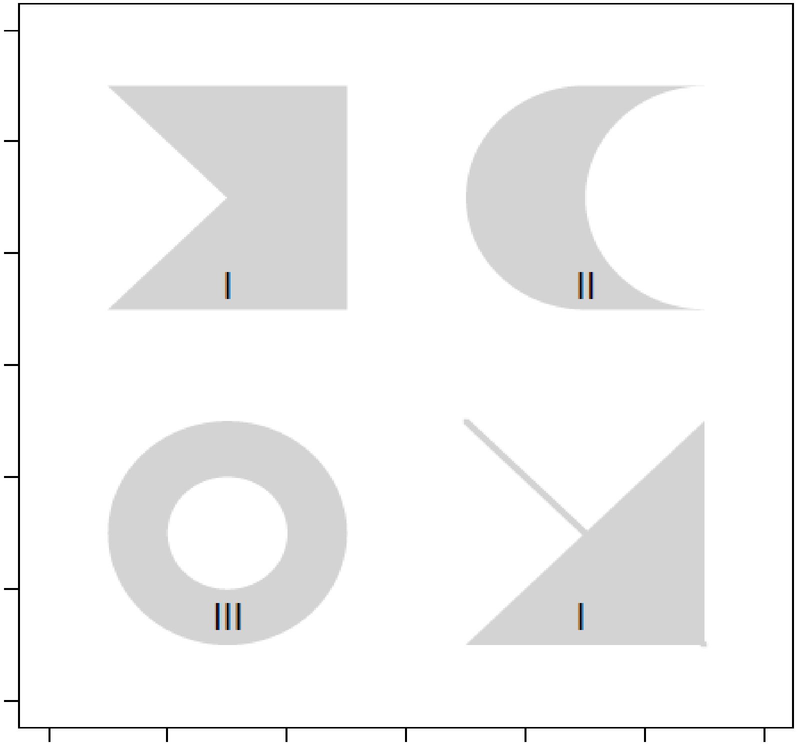
Convexity

A set *S* is *convex* if for all x, y in *S* and all t in the interval [0,1], t x + (1-t) y is also an element of *S*.

convex sets



non-convex sets



Constrained estimation

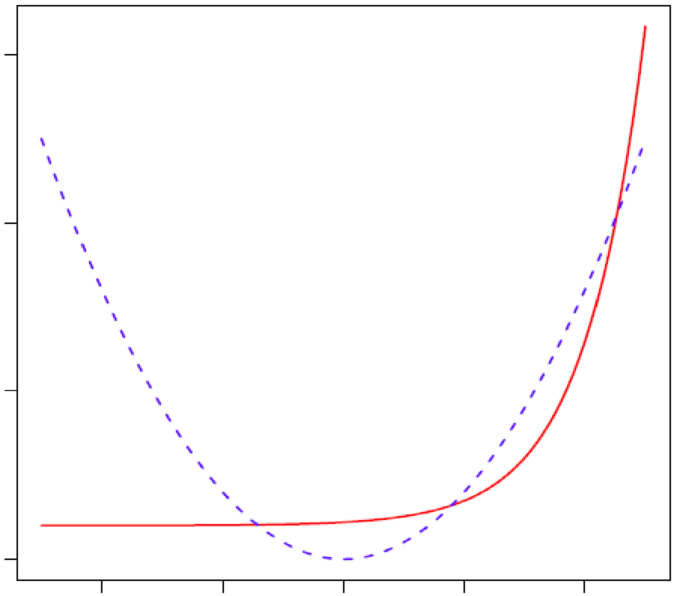
Convexity

A function f(x) defined on a convex set *S* is called *convex* if for all x, y in *S* and all t in the interval [0,1]:

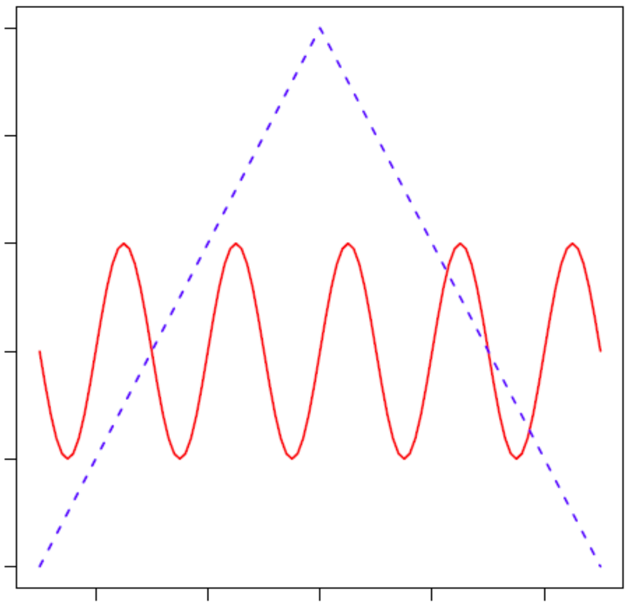
$$f(t\,x + (1-t)\,y) \leq t\,f(x) + (1-t)\,f(y).$$

A function is convex ↔ region above the curve is convex.

convex functions



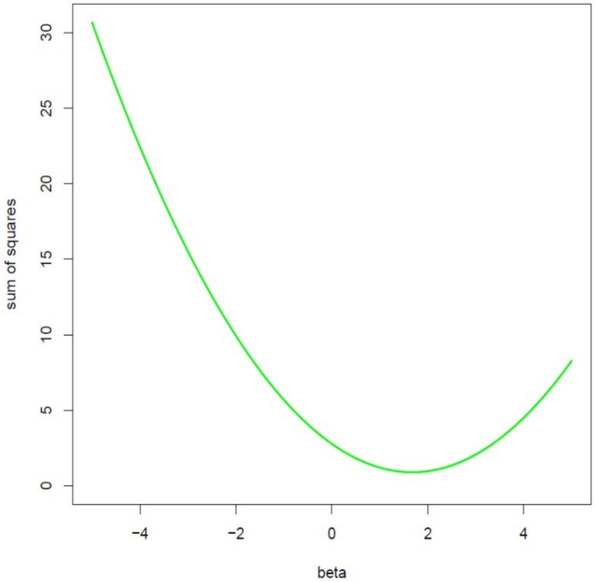
non-convex functions

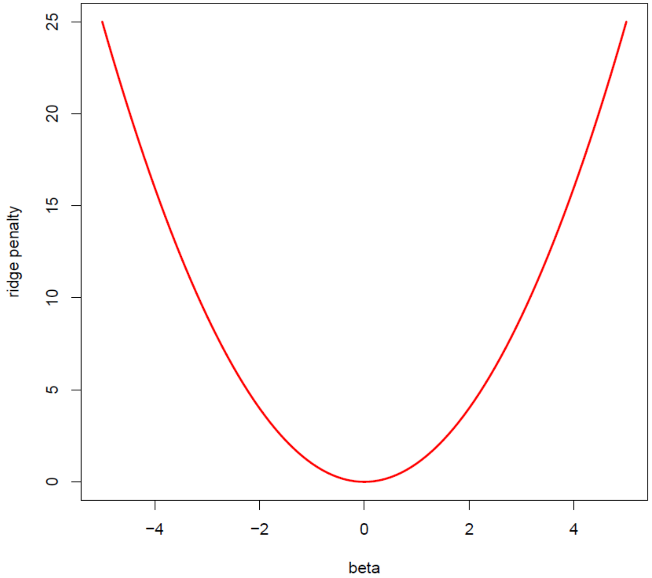


Constrained estimation

Convexity

Both the sum of squares and the penalty are convex functions in β. Consequently, so is their sum.





This ensures there is a unique β that minimizes the penalized sum of squares. Much like the “ad hoc” fix solves the singularity.

Constrained estimation

Ridge regression as constrained estimation

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

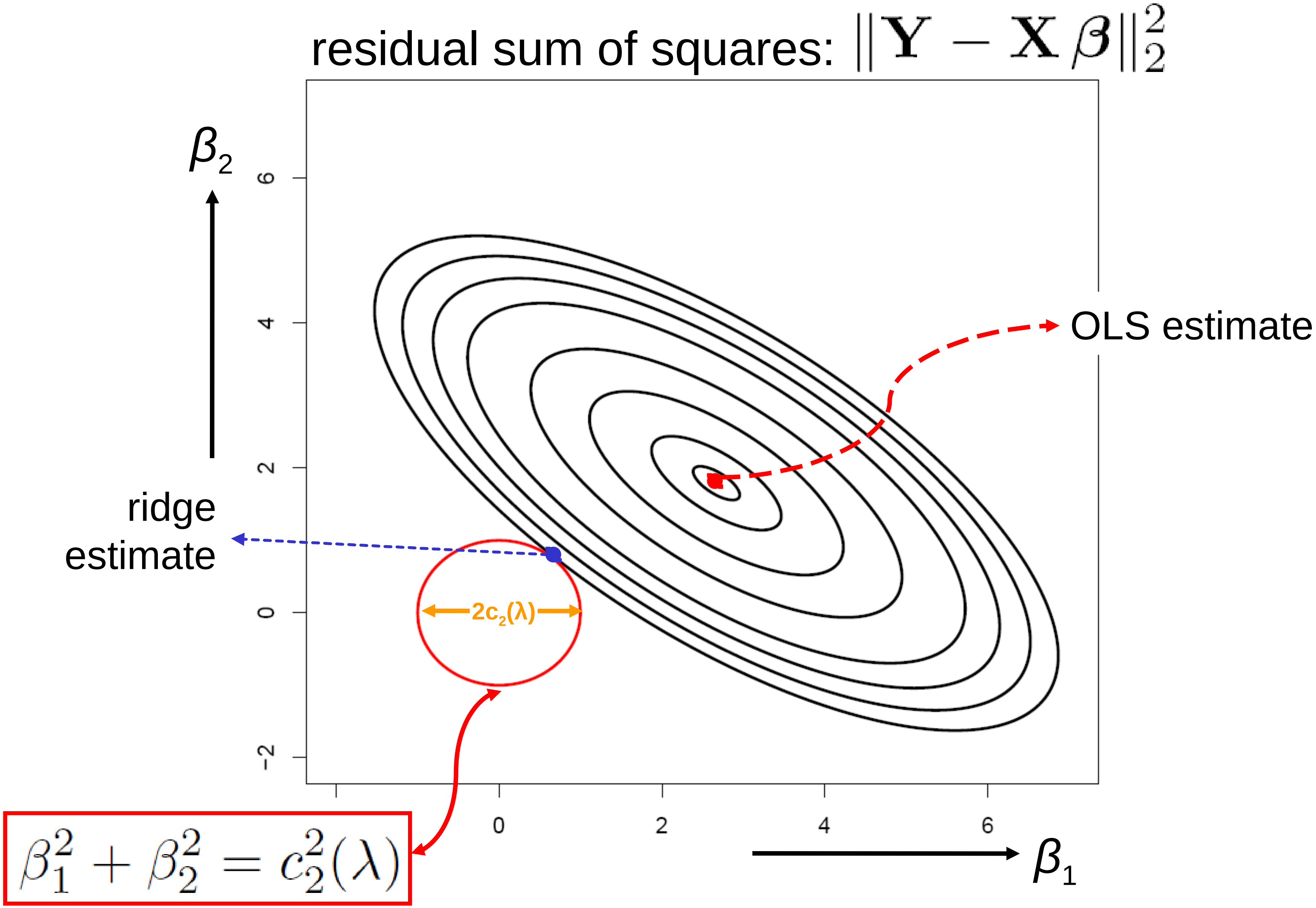
$$\min_{\boldsymbol{\beta}} \|\mathbf{Y} - \mathbf{X}\,\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2$$

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 $\theta(\lambda)$ is available.

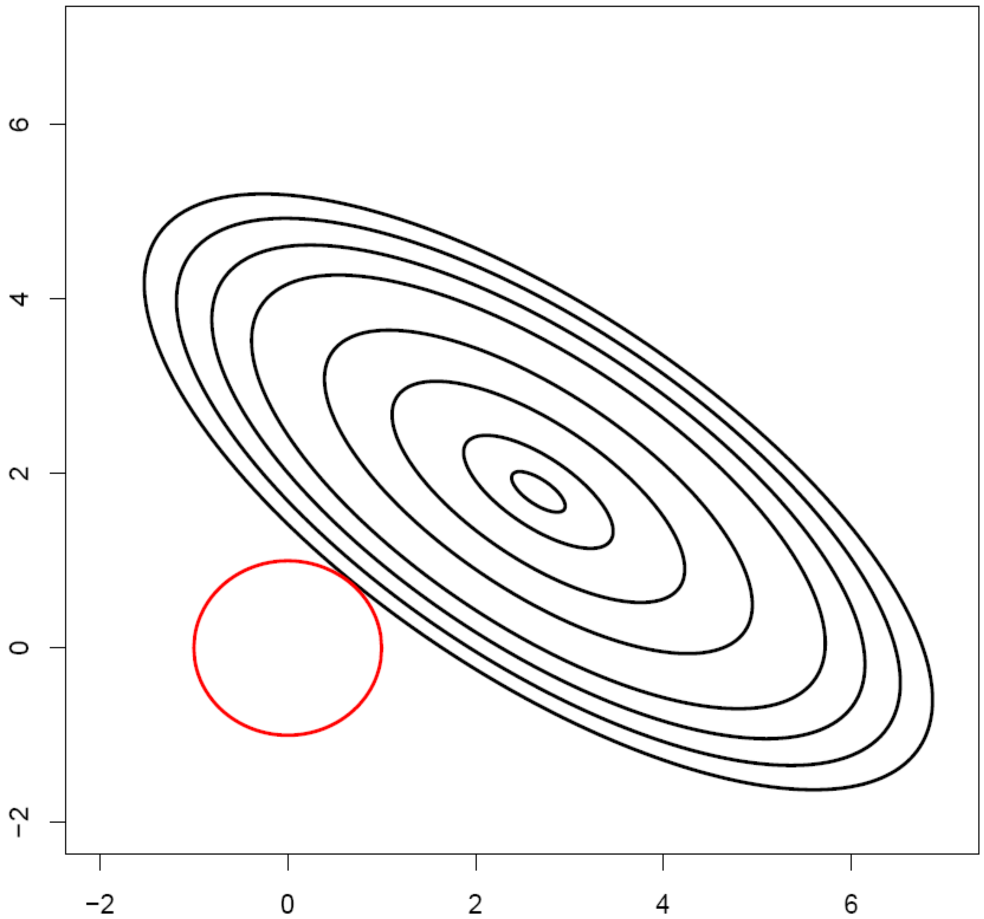
Constrained estimation



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

The following linear regression is fitted to the data:

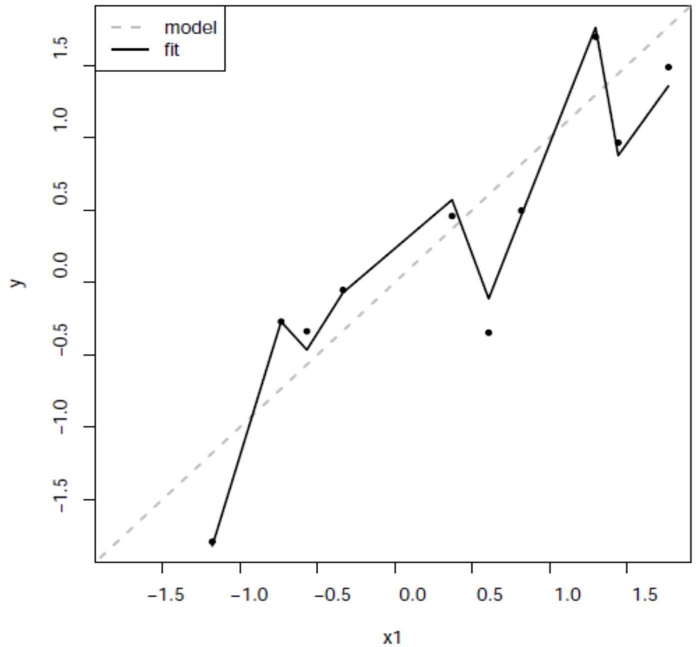
$$Y_i = \sum_{j=1}^{9} X_{i,j}\beta_j + \varepsilon_i$$

Estimate:

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

Large estimates of regression coefficients are an indication of overfitting.

Fit:



A simple remedy would constrain the parameter estimates.

A Bayesian interpretation

A Bayesian interpretation

Ridge regression has a close connection 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:

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

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

The latter denotes an inverse Gamma distribution.

A Bayesian interpretation

The posterior distribution of $\boldsymbol{\beta}$ and σ^2 is then:

$$
\begin{aligned}
& f_{\boldsymbol{\beta},\sigma^2}(\boldsymbol{\beta},\sigma^2 \,|\, \mathbf{Y},\mathbf{X}) \\
& \quad \propto \quad f_Y(\mathbf{Y}\,|\,\mathbf{X},\boldsymbol{\beta},\sigma^2) f_\beta(\boldsymbol{\beta}|\sigma^2) f_\sigma(\sigma^2) \\
& \quad \propto \quad \sigma^{-n} \exp\Big[-\frac{1}{2\sigma^2}(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta})^{\mathrm{T}}(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta})\Big] \\
& \qquad\quad \times\ \sigma^{-p} \exp\Big[-\frac{\tau}{2\sigma^2}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta}\Big] \\
& \qquad\quad \times\ [\sigma^2]^{-\alpha_0-1} \exp\Big[-\frac{\beta_0}{2\sigma^2}\Big]
\end{aligned}
$$

A Bayesian interpretation

This can be rewritten to:

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

where

$$
\begin{aligned}
& g_{\boldsymbol{\beta}}(\boldsymbol{\beta} \,|\, \sigma^2, \mathbf{Y}, \mathbf{X}) \ = \\
& \quad \sigma^{-k} \exp \Big\{ - \frac{1}{2\sigma^2} \big[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}(\lambda)\big]^{\mathrm{T}} (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda \mathbf{I}) \big[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}(\lambda)\big] \Big\}
\end{aligned}
$$

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

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

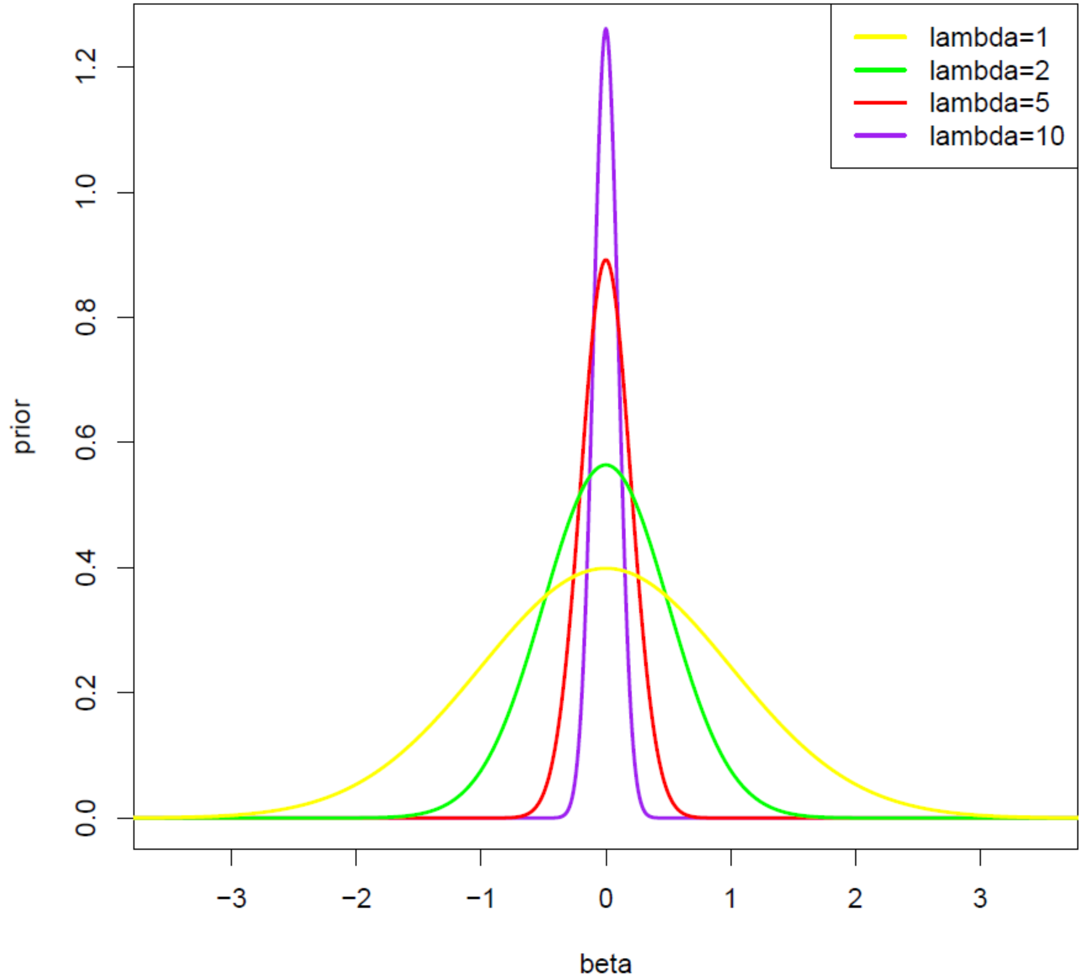
A Bayesian interpretation

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 smaller penalty corresponds to wider prior, and

→ a larger penalty to a more informative prior.



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:

$$\hat{\beta}(\lambda) = \underbrace{(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})}_{(p \times p)\text{-dim. matrix}}{}^{-1}\mathbf{X}^T\mathbf{Y}$$

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

However, there is a workaround.

Efficient computation

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

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

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:

$$\mathbf{X} = \mathbf{R}\mathbf{V}^T$$

with **R** and **V** as above.

Efficient computation

Rewrite the ridge estimator in terms of **R** and **V**:

$$\begin{aligned}
\hat{\beta}(\lambda) &= (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} \\
&= (\mathbf{V}\mathbf{R}^T\mathbf{R}\mathbf{V}^T + \lambda\mathbf{I})^{-1}\mathbf{V}\mathbf{R}^T\mathbf{Y} \\
&= (\mathbf{V}\mathbf{R}^T\mathbf{R}\mathbf{V}^T + \lambda\mathbf{V}\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{R}^T\mathbf{Y} \\
&= \mathbf{V}(\mathbf{R}^T\mathbf{R} + \lambda\mathbf{I})^{-1}\mathbf{V}^T\mathbf{V}\mathbf{R}^T\mathbf{Y} \\
&= \mathbf{V}\underbrace{(\mathbf{R}^T\mathbf{R} + \lambda\mathbf{I})}_{(n \times n)\text{-dim. matrix}}{}^{-1}\mathbf{R}^T\mathbf{Y}
\end{aligned}$$

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:

$$\begin{aligned}\hat{\mathbf{Y}} &= \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y} \\ &= \mathbf{H}\mathbf{Y}\end{aligned}$$

where **H** is the hat matrix.

The degrees of freedom of ordinary regression is then equal to $\mathrm{tr}(\mathbf{H})$.
In particular, if **X** if of full rank, i.e. rank(**X**) = *p*, then:

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

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:

$$\begin{aligned} \mathrm{tr}[\mathbf{H}(\lambda)] \quad &= \quad \mathrm{tr}[\mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}] \\ &= \quad \sum_{j=1}^{p} \frac{d_{jj}^2}{d_{jj}^2 + \lambda} \end{aligned}$$

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. The expression levels of these genes are sampled from a multivariate normal distribution, with mean zero and covariance:

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

Put differently, a diagonal covariance with:

$$(\mathbf{\Sigma})_{jj} = j/10$$

Simulation I

Effect of ridge estimation

Together they regulate a 51th gene, in accordance with the following relationship:

$$Y_i = \mathbf{X}_{i*}\boldsymbol{\beta} + \varepsilon_i$$

with

$$\varepsilon \sim \mathcal{N}(0, 1)$$

The regression coefficients are

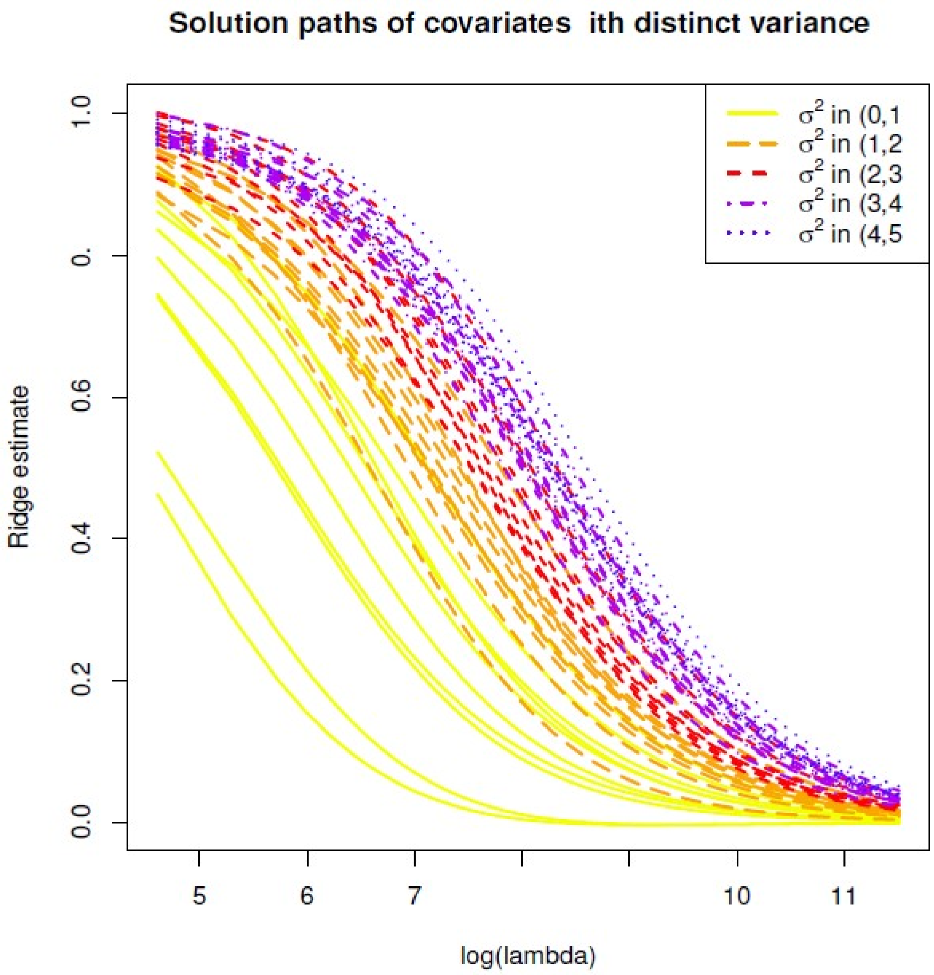
$$\boldsymbol{\beta} = \mathbf{1}_{50 \times 1}$$

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

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. The expression levels of these genes are sampled from a multivariate normal distribution, with mean zero and covariance:

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

where

$$\mathbf{\Sigma}_{jj} = \frac{j-1}{5}\mathbf{1}_{10\times 10} + \frac{6-j}{5}\mathbf{I}_{10\times 10}$$

Simulation II

Effect of ridge estimation

Together they regulate a 51th gene, in accordance with the following relationship:

$$Y_i = \mathbf{X}_{i*}\boldsymbol{\beta} + \varepsilon_i$$

with

$$\varepsilon \sim \mathcal{N}(0, 1)$$

The regression coefficients are

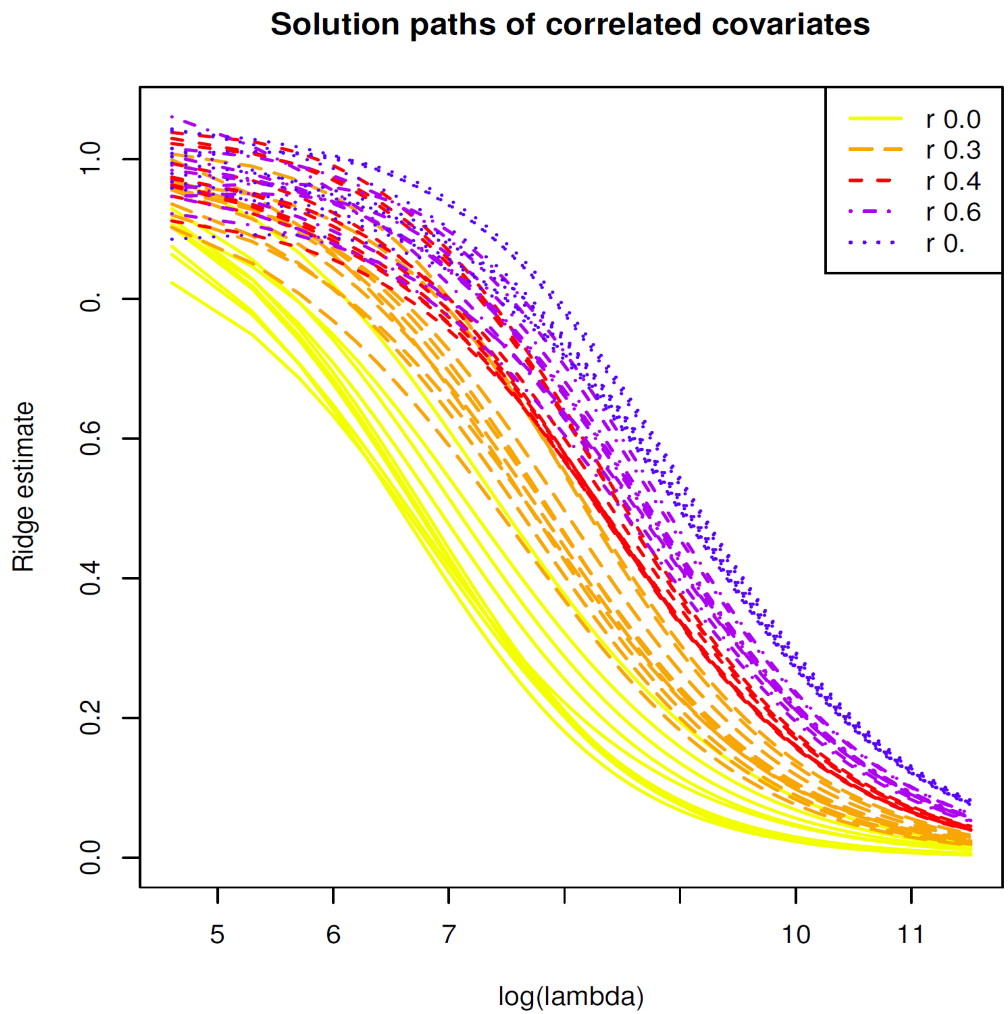
$$\boldsymbol{\beta} = \mathbf{1}_{50\times 1}$$

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:

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

For large λ:

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

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

Cross-validation

Cross-validation

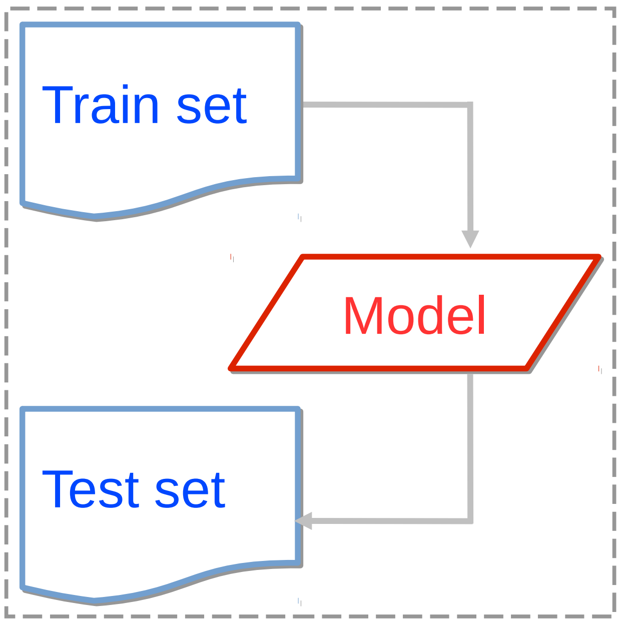
Methods for choosing penalty parameter	
1.	Cross-validation
2.	Information criteria

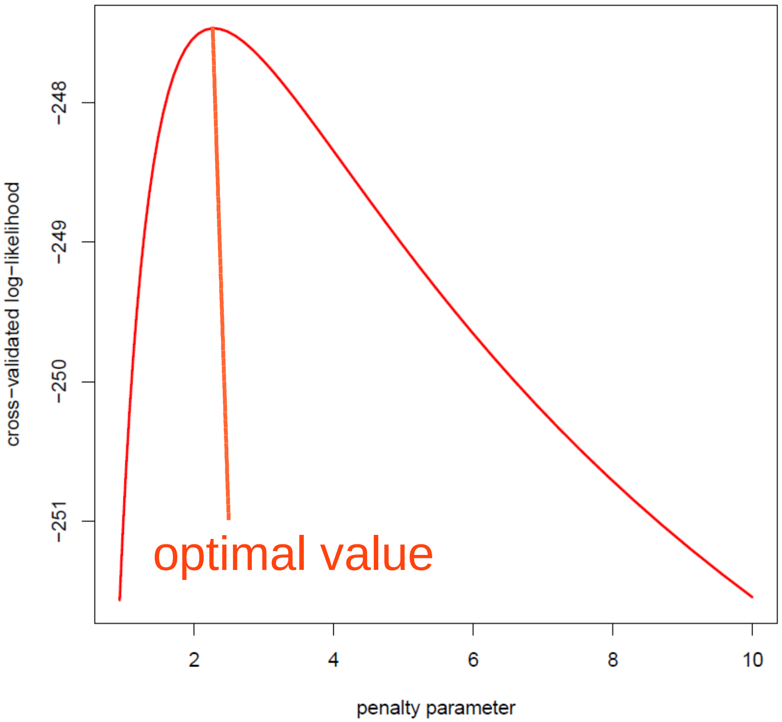
Cross-validation

- 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

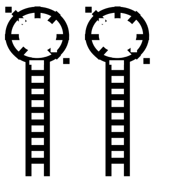
Cross validation

- K-fold cross-validation divides the learning set Λ randomly into K equal (or almost equal) sized subsets $\Lambda_1, \ldots, \Lambda_K$.
- Models C_k are built on training $\Lambda - \Lambda_k$.
- Models C_k are applied to the training or validation 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 C_k.

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:

$$\texttt{mrna expr.} = \texttt{f(mir expression)}$$

$$= \beta_0 + \beta_1 * \texttt{mir}_1 + \beta_2 * \texttt{mir}_2 + \ldots + \beta_p * \texttt{mir}_p + \texttt{error}$$

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:

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

Finally, we obtain the predicted survival:

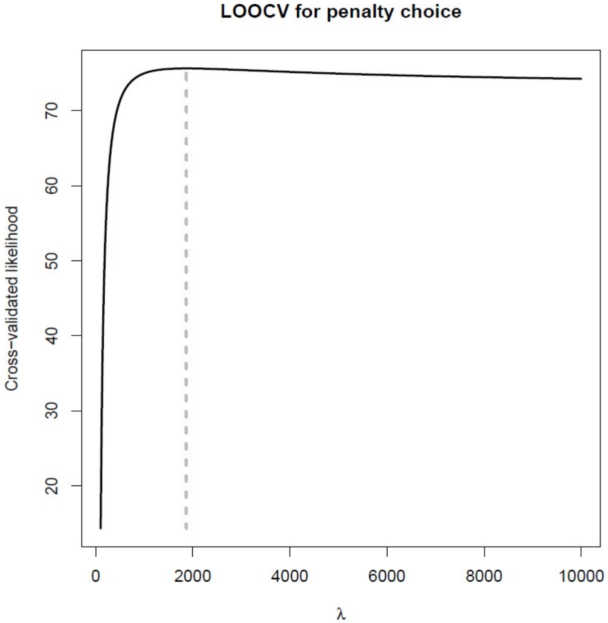
$$\texttt{pred. mrna expr.} = \texttt{f(linear predictor)}$$

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

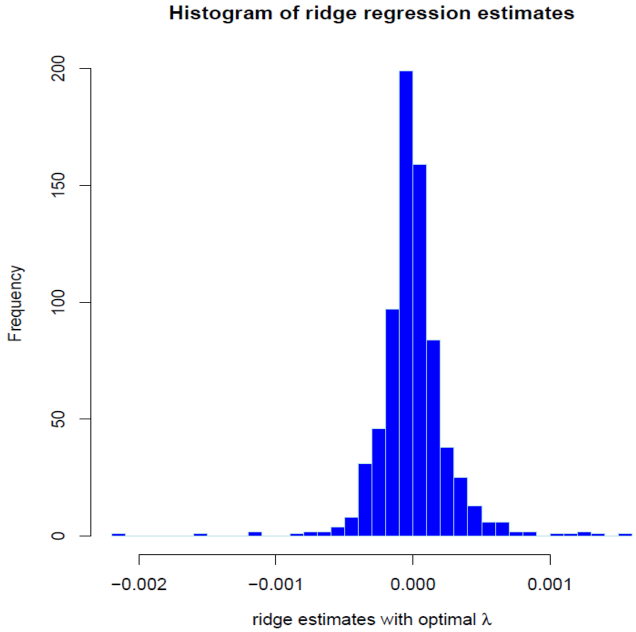
Compare observed and predicted mRNA expression.

Example: microRNA-mRNA regulation

Penalty parameter choice

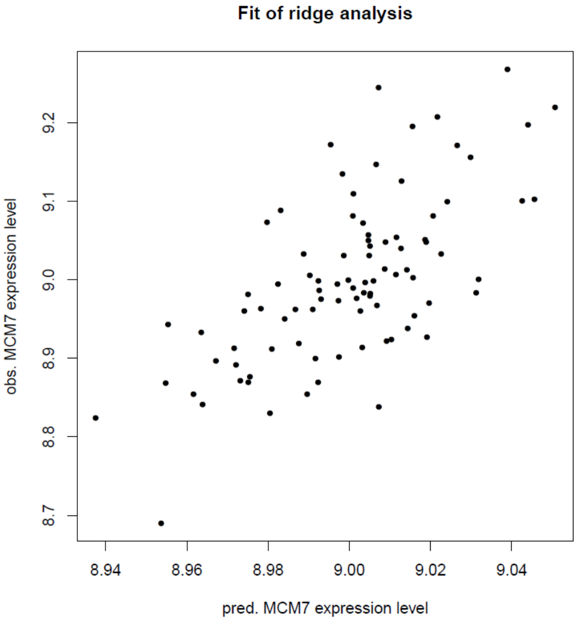


Beta hat distribution



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

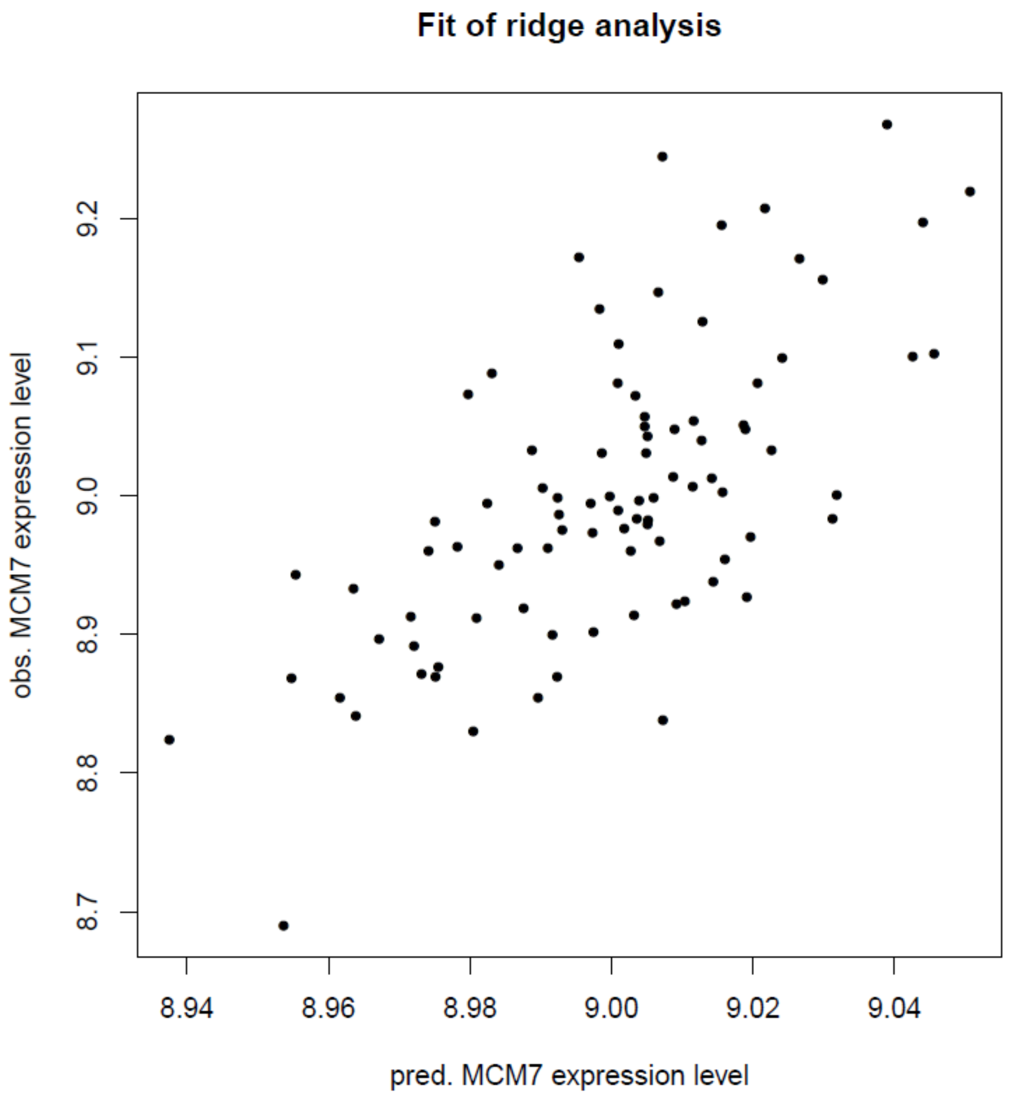
Obs. vs. pred. mRNA expression



$\rho_{sp} = 0.629$
$R^2 = 0.449$

Example: microRNA-mRNA regulation

Question: explain the difference in scale.



Example: microRNA-mRNA regulation

Biological dogma

MicroRNAs down-regulate mRNA levels.

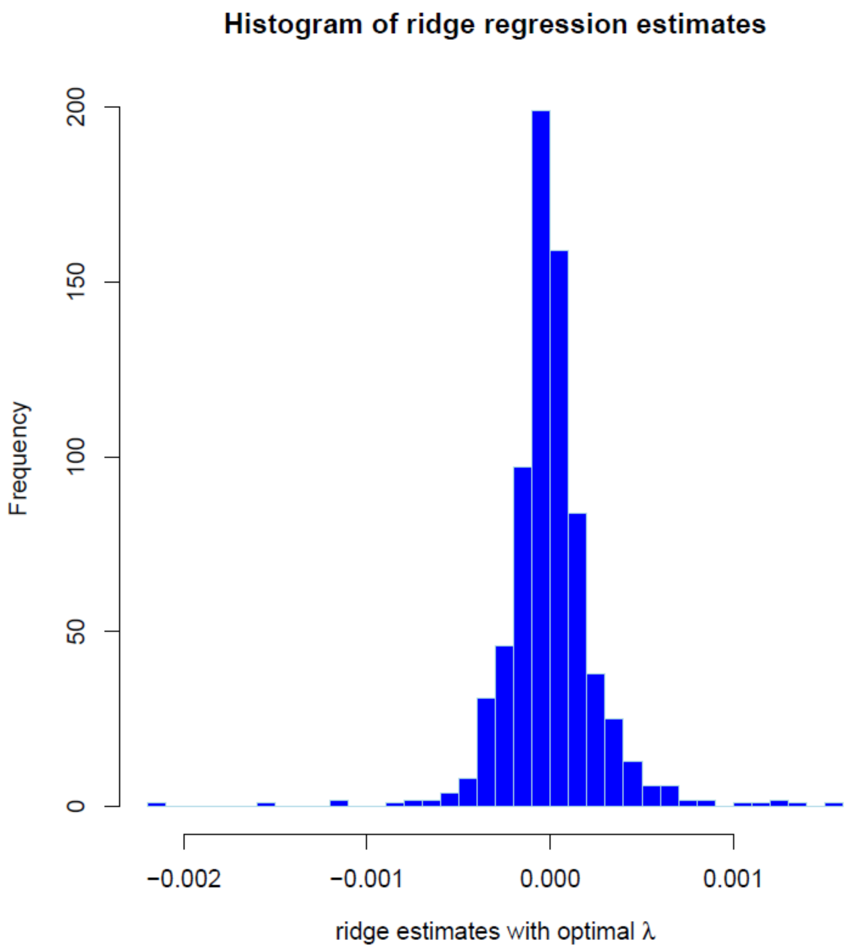
The dogma suggests that negative regression coefficients prevail.

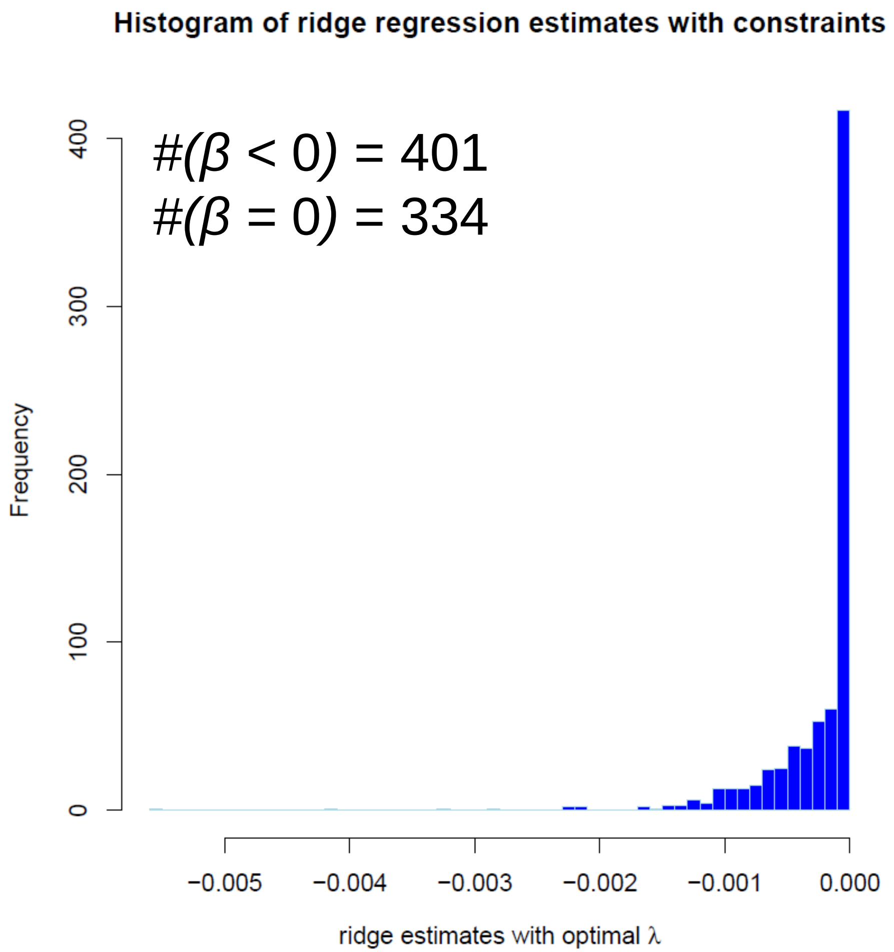
The **`penalized`** package allows for the specification of the sign of the regression parameters. No explicit expression for ridge estimator: numeric optimization of the loss function.

Re-analysis of the data with negative constraints.

Example: microRNA-mRNA regulation

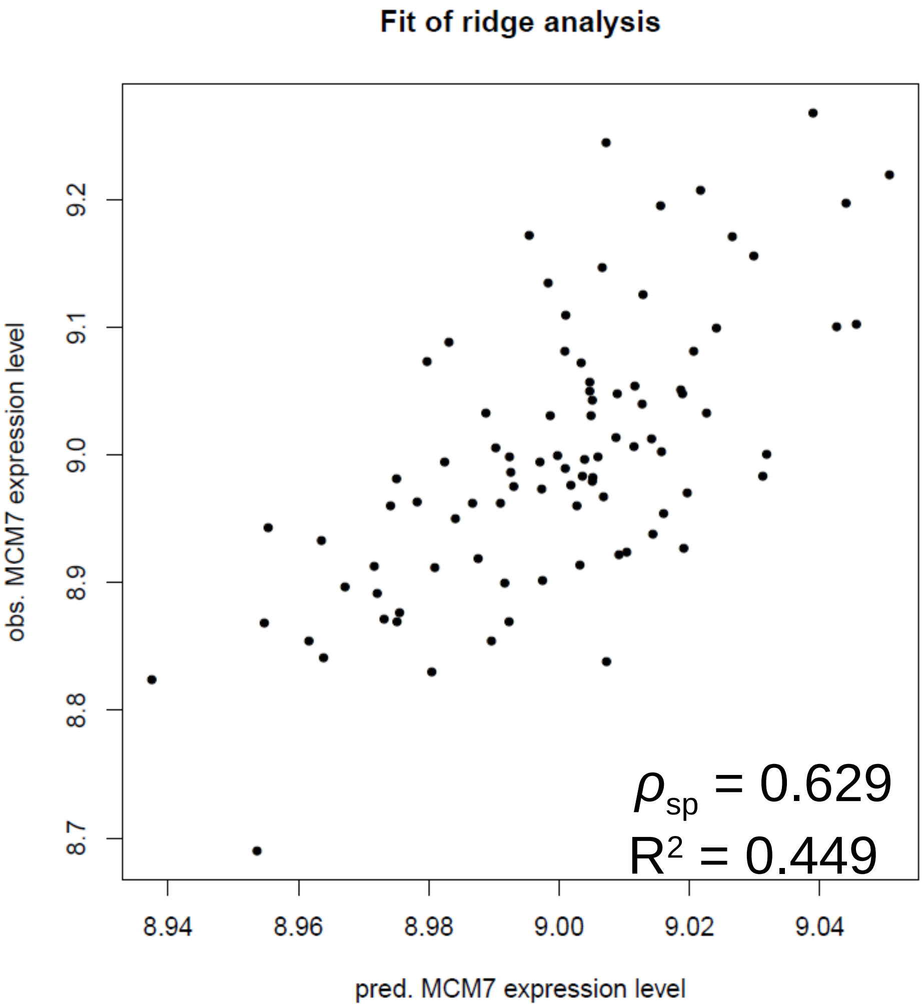
Histograms of ridge estimates of both analyses.

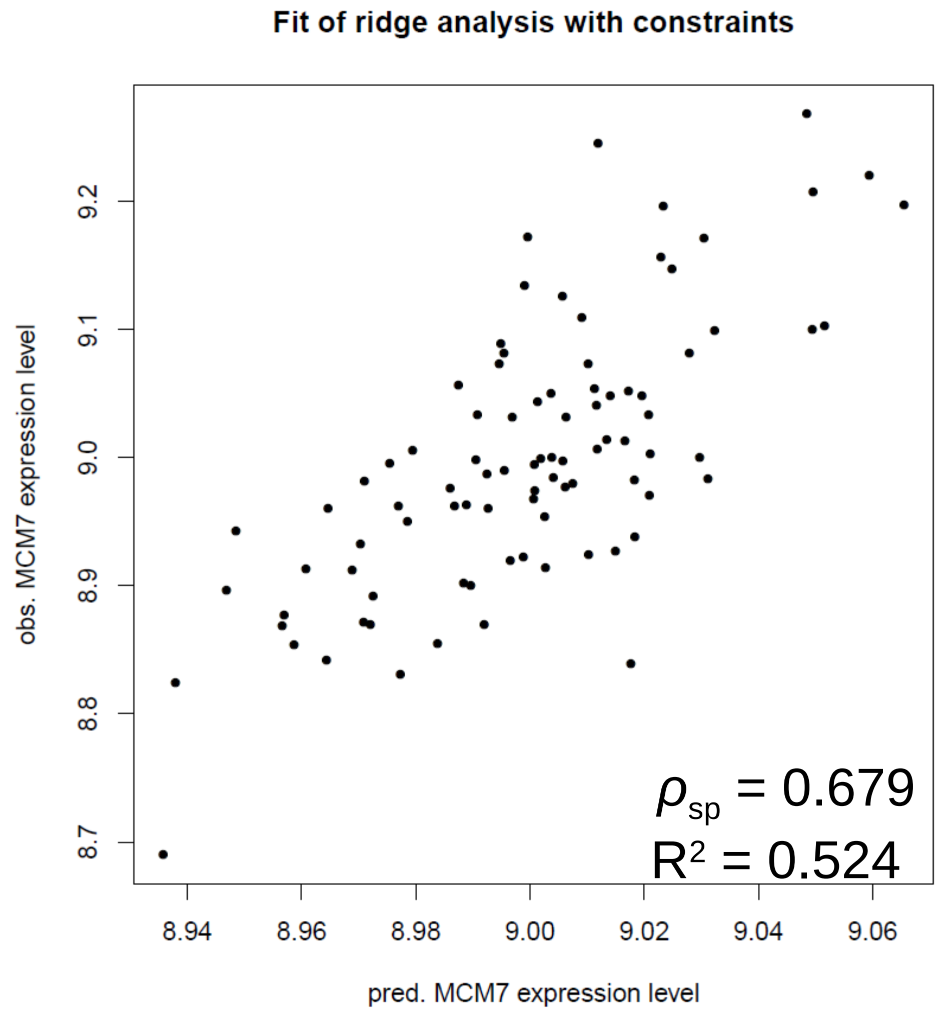




Example: microRNA-mRNA regulation

Observed vs. predicted mRNA expression for both analyses.





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
```

References & further reading

References & further reading

Banerjee, O., El Ghaoui, L., d'Aspremont, A. (2008), "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data", *Journal of Machine Learning Research*, 9, 485-516.

Bickel, P.J., Doksum, K.A. (2001), *Mathematical Statistics, Volume I*, New York: Prentice Hall.

Friedman, J., Hastie, T., Tibshirani, R. (2008), "Sparse inverse covariance estimation with the graphical lasso", *Biostatistics*, 9(3), 432-441.

Goeman, J.J. (2010), "..", Biometrical Journal, ..

Harville, D.A. (2008), *Matrix Algebra From a Statistician's Perspective*, New York: Springer.

Margolin, A.A., Califano, A. (2007), "Theory and limitations of genetic network inference from microarray data", *Annals of the New York Academy of Sciences,* 1115, 51-72.

Markowetz, F., Spang, R. (2007), "Inferring cellular networks : a review", *BMC Bioinformatics,* 8(Supple 6):S5.

Meinshausen, N., Buhlmann, P. (2010), "Stability selection", *Journal of the Royal Statistical Society*, 74(4), 417-473.

Rao, C.R. (1973), *Linear Statistical Inference and Its Applications*, New York: John Wiley.

Shafer, J., Strimmer, K. (2005), "A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics", *Statistical Applications in Genetics and Molecular Biology , 4, Article 32.*

Whittaker, J. (1991), *Graphical models in applied multivariate statistics*, John Wiley.