# Lecture 10: Regularized/penalized regression (cont'd)

Felix Held, Mathematical Sciences

**MSA220/MVE440** Statistical Learning for Big Data

2nd May 2019

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

# A short recap

# Goals of modelling

1. **Predictive strength:** How well can we reconstruct the observed data? Has been most important so far.
2. **Model/variable selection:** Which variables are **part of the true model**? This is about uncovering structure to allow for mechanistic understanding.

# Feature selection

**Feature selection** can be addressed in multiple ways

- **Filtering:** Remove variables before the actual model for the data is built
  - Often crude but fast
  - Typically only pays attention to one or two features at a time (e.g. F-Score, MIC) or does not take the outcome variable into consideration (e.g. PCA)
- **Wrapping:** Consider the selected features as an additional hyper-parameter
  - computationally very heavy
  - most approximations are greedy algorithms
- **Embedding:** Include feature selection into parameter estimation through penalisation of the model coefficients
  - Naive form is equally computationally heavy as wrapping
  - **Soft-constraints** create biased but useful approximations

# Penalised regression

The optimization problem

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_q^q \leq t$$

for $q > 0$ is equivalent to

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_q^q$$

when $q \geq 1$.

- For $q = 2$ known as **ridge regression** for $q = 1$ known as the **lasso**
- Constraints are convex for all $q \geq 1$ but not differentiable in $\boldsymbol{\beta} = \mathbf{0}$ for $q = 1$

# Intuition for the penalties (I)

Assume the OLS solution $\boldsymbol{\beta}_{\mathrm{OLS}}$ exists and set

$$\mathbf{r} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}_{\mathrm{OLS}}$$

it follows for the **residual sum of squares (RSS)** that

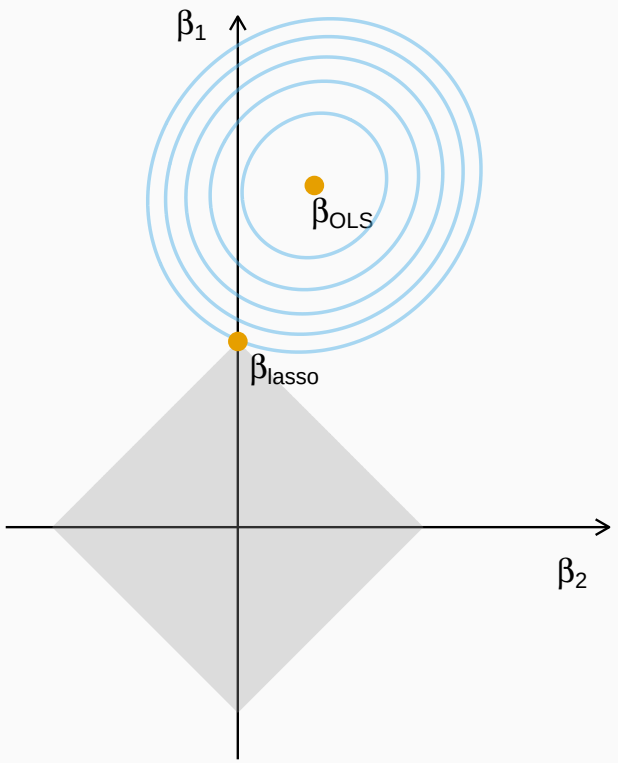
$$\begin{aligned}
\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 &= \|(\mathbf{X}\boldsymbol{\beta}_{\mathrm{OLS}} + \mathbf{r}) - \mathbf{X}\boldsymbol{\beta}\|_2^2 \\
&= \|(\mathbf{X}(\boldsymbol{\beta} - \boldsymbol{\beta}_{\mathrm{OLS}}) - \mathbf{r}\|_2^2 \\
&= (\boldsymbol{\beta} - \boldsymbol{\beta}_{\mathrm{OLS}})^T \mathbf{X}^T \mathbf{X}(\boldsymbol{\beta} - \boldsymbol{\beta}_{\mathrm{OLS}}) - 2\mathbf{r}^T \mathbf{X}(\boldsymbol{\beta} - \boldsymbol{\beta}_{\mathrm{OLS}}) + \mathbf{r}^T \mathbf{r}
\end{aligned}$$

which is an **ellipse** (at least in 2D) centred on $\boldsymbol{\beta}_{\mathrm{OLS}}$.

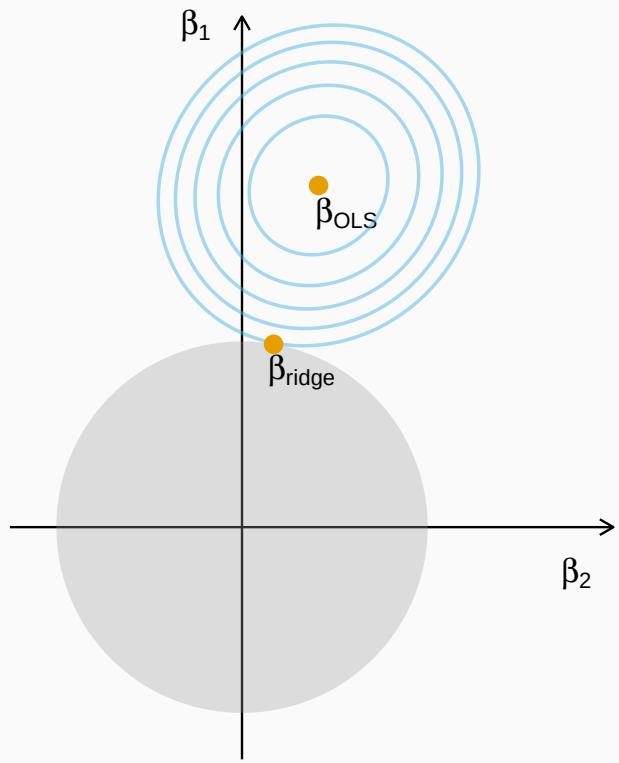
# Intuition for the penalties (II)

The least squares RSS is minimized for $\boldsymbol{\beta}_{\mathrm{OLS}}$. If a constraint is added ($||\boldsymbol{\beta}||_q^q \leq t$) then the RSS is minimized by the closest $\boldsymbol{\beta}$ possible that fulfills the constraint.

Lasso




Ridge



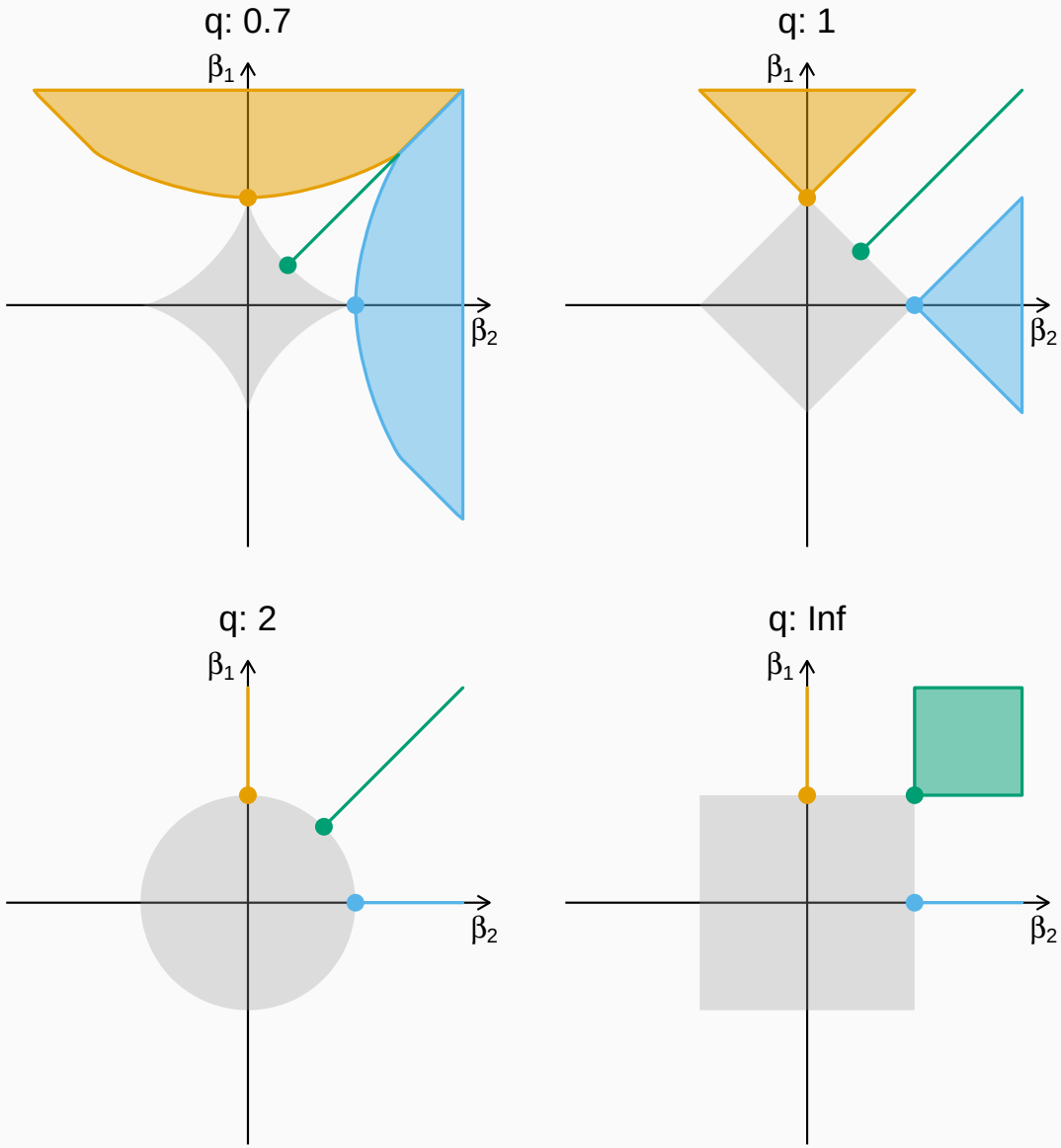

The blue lines are the contour lines for the RSS.

# Intuition for the penalties (III)

Depending on $q$ the different constraints lead to different solutions. If $\boldsymbol{\beta}_{\mathrm{OLS}}$ is in one of the coloured areas or on a line, the constrained solution will be at the corresponding dot.

**Sparsity** only for $q \leq 1$

**Convexity** only for $q \geq 1$



# Shrinkage and effective degrees of freedom

When $\lambda$ is fixed, the **shrinkage** of the lasso estimate $\boldsymbol{\beta}_{\text{lasso}}(\lambda)$ compared to the OLS estimate $\boldsymbol{\beta}_{\text{OLS}}$ is defined as

$$s(\lambda) = \frac{\|\boldsymbol{\beta}_{\text{lasso}}(\lambda)\|_1}{\|\boldsymbol{\beta}_{\text{OLS}}\|_1}$$

**Note:** $s(\lambda) \in [0, 1]$ with $s(\lambda) \to 0$ for increasing $\lambda$ and $s(\lambda) = 1$ if $\lambda = 0$

For ridge regression define

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

and

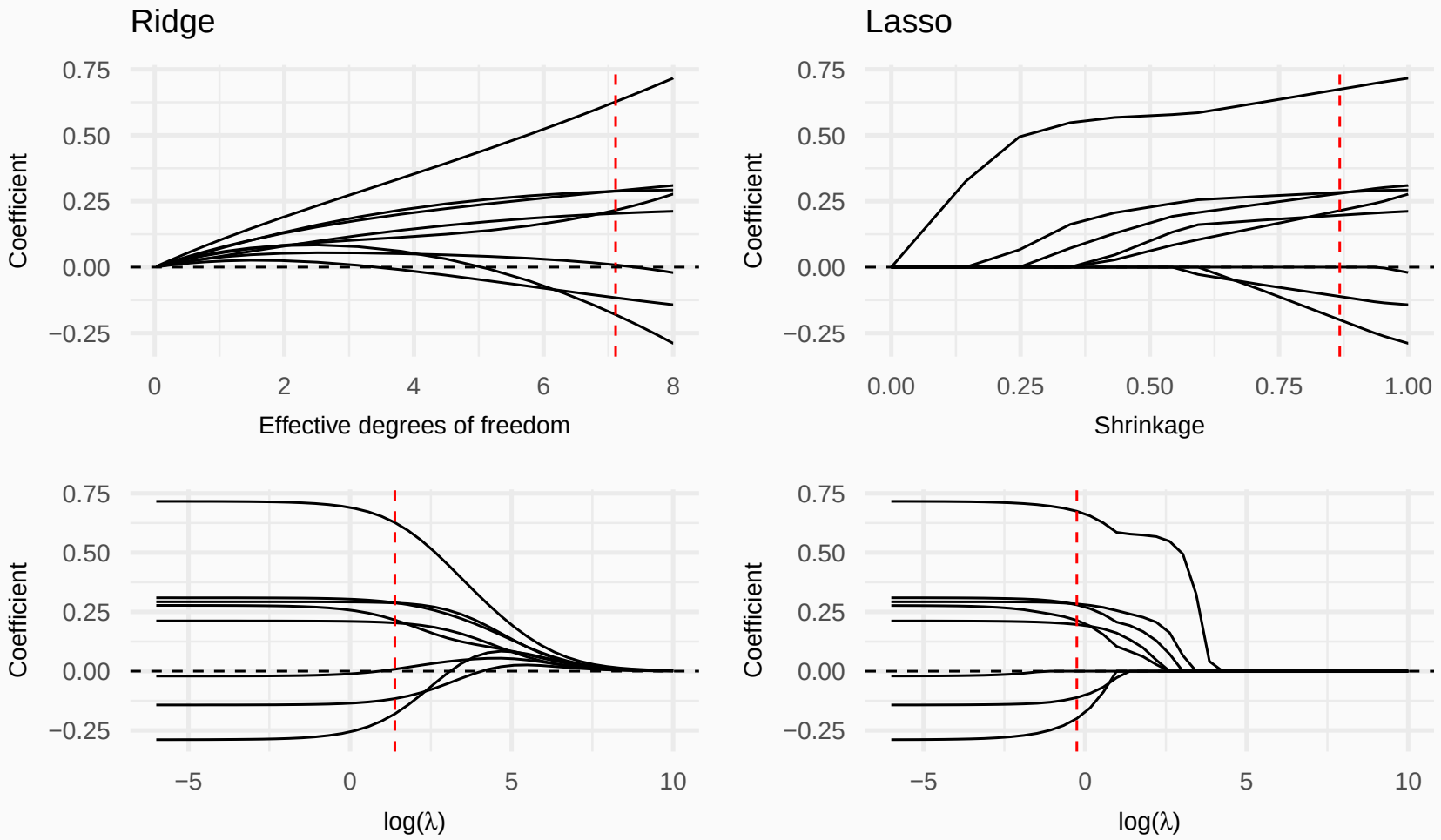
$$\text{df}(\lambda) := \text{tr}(\mathbf{H}(\lambda)) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda},$$

the **effective degrees of freedom**.

# A regularisation path

**Prostate cancer dataset** $(n = 67, p = 8)$

Red dashed lines indicate the $\lambda$ selected by cross-validation



# Connection to classification

# Recall: Regularised Discriminant Analysis (RDA)

Given training samples $(i_l, \mathbf{x}_l)$, quadratic DA models

$$p(\mathbf{x}|i) = N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad \text{and} \quad p(i) = \pi_i$$

Estimates $\hat{\boldsymbol{\mu}}_i$, $\hat{\boldsymbol{\Sigma}}_i$ and $\hat{\pi}_i$ are straight-forward to find,...

...but evaluating the normal density requires inversion of $\hat{\boldsymbol{\Sigma}}_i$. If it is (near-)singular, this can lead to **numerical instability**.

**Penalisation can help here:**

- Use $\hat{\boldsymbol{\Sigma}}_i = \hat{\boldsymbol{\Sigma}}_i^{\mathrm{QDA}} + \lambda \hat{\boldsymbol{\Sigma}}^{\mathrm{LDA}}$ for $\lambda > 0$
- Use LDA (i.e. $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}$) and $\hat{\boldsymbol{\Sigma}} = \hat{\boldsymbol{\Sigma}}^{\mathrm{LDA}} + \lambda \boldsymbol{\Delta}$ for $\lambda > 0$ and a diagonal matrix $\boldsymbol{\Delta}$

# Recall: Naive Bayes LDA

Naive Bayes LDA means that we assume that $\widehat{\mathbf{\Sigma}} = \widehat{\mathbf{\Delta}}$ for a diagonal matrix $\widehat{\mathbf{\Delta}}$. The diagonal elements are estimated as

$$\widehat{\Delta}_{jj}^2 = \frac{1}{n-K} \sum_{i=1}^{K} \sum_{i_l=i} (x_{lj} - \widehat{\boldsymbol{\mu}}_{i,j})^2$$

which is the **pooled within-class variance**.

Classification is performed by evaluating the **discriminant functions**

$$\delta_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \widehat{\boldsymbol{\mu}}_i)^T \widehat{\mathbf{\Delta}}^{-1} (\mathbf{x} - \widehat{\boldsymbol{\mu}}_i) + \log(\widehat{\pi}_i)$$

and by choosing

$$c(\mathbf{x}) = \arg\max_i \delta_i(\mathbf{x})$$

as the predicted class.

# Shrunken centroids (I)

In high-dimensional problems, centroids will

- contain noise
- be hard to interpret when all variables are active

As in regression, we would like to perform **variable selection** and **reduce noise**.

**Note:** The class centroids solve

$$\widehat{\boldsymbol{\mu}}_i = \arg\min_{\mathbf{v}} \frac{1}{2} \sum_{i_l = i} \|\mathbf{x}_l - \mathbf{v}\|_2^2$$

**Nearest shrunken centroids** performs variable selection and stabilises centroid estimates by solving

$$\widehat{\boldsymbol{\mu}}_i^s = \arg\min_{\mathbf{v}} \frac{1}{2} \sum_{i_l = i} \|(\widehat{\boldsymbol{\Delta}} + s_0 \mathbf{I}_p)^{-1/2} (\mathbf{x}_l - \mathbf{v})\|_2^2 + \lambda \sqrt{\frac{(n - n_i) n_i}{n}} \|\mathbf{v} - \widehat{\boldsymbol{\mu}}_T\|_1$$

# Shrunken centroids (II)

## Nearest shrunken centroids

$$\widehat{\boldsymbol{\mu}}_i^s = \arg\min_{\mathbf{v}} \frac{1}{2} \sum_{i_l = i} \|(\widehat{\boldsymbol{\Delta}} + s_0 \mathbf{I}_p)^{-1/2}(\mathbf{x}_l - \mathbf{v})\|_2^2 + \lambda \sqrt{\frac{(n - n_i)n_i}{n}} \|\mathbf{v} - \widehat{\boldsymbol{\mu}}_T\|_1$$

- Penalises distance of class centroid to the overall centroid $\boldsymbol{\mu}_T$
- $\widehat{\boldsymbol{\Delta}} + s_0 \mathbf{I}_p$ is the diagonal regularised within-class covariance matrix. Leads to greater weights for variables that are less variable across samples (**interpretability**)
- $\sqrt{(n - n_i)n_i/n}$ is only there for technical reasons
- If the predictors are centred ($\widehat{\boldsymbol{\mu}}_T = 0$) this is a scaled lasso problem

# Shrunken centroids (III)

The solution for component $j$ can be derived using subdifferentials as

$$\widehat{\boldsymbol{\mu}}_{i,j}^{S} = \widehat{\boldsymbol{\mu}}_{T,j} + m_i(\Delta_{jj} + s_0)\,\mathrm{ST}\left(t_{i,j}, \lambda\right) \quad \text{where} \quad t_{i,j} = \frac{\widehat{\boldsymbol{\mu}}_{i,j} - \widehat{\boldsymbol{\mu}}_{T,j}}{m_i(\Delta_{jj} + s_0)}$$

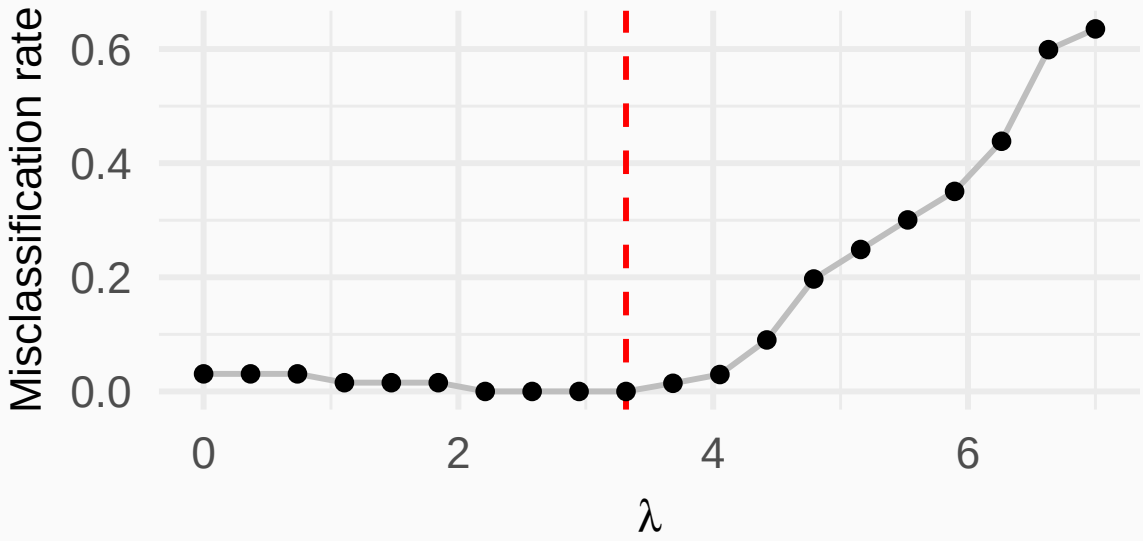
and $m_i = \sqrt{\frac{1}{n_i} - \frac{1}{n}}$.

**Note:** $\lambda$ is a tuning parameter and has to be determined through e.g. cross-validation.

- Typically, misclassification rate improves first with increasing $\lambda$ and declines for too high values
- The larger $\lambda$ the more components will be equal to the respective component of the overall centroid.

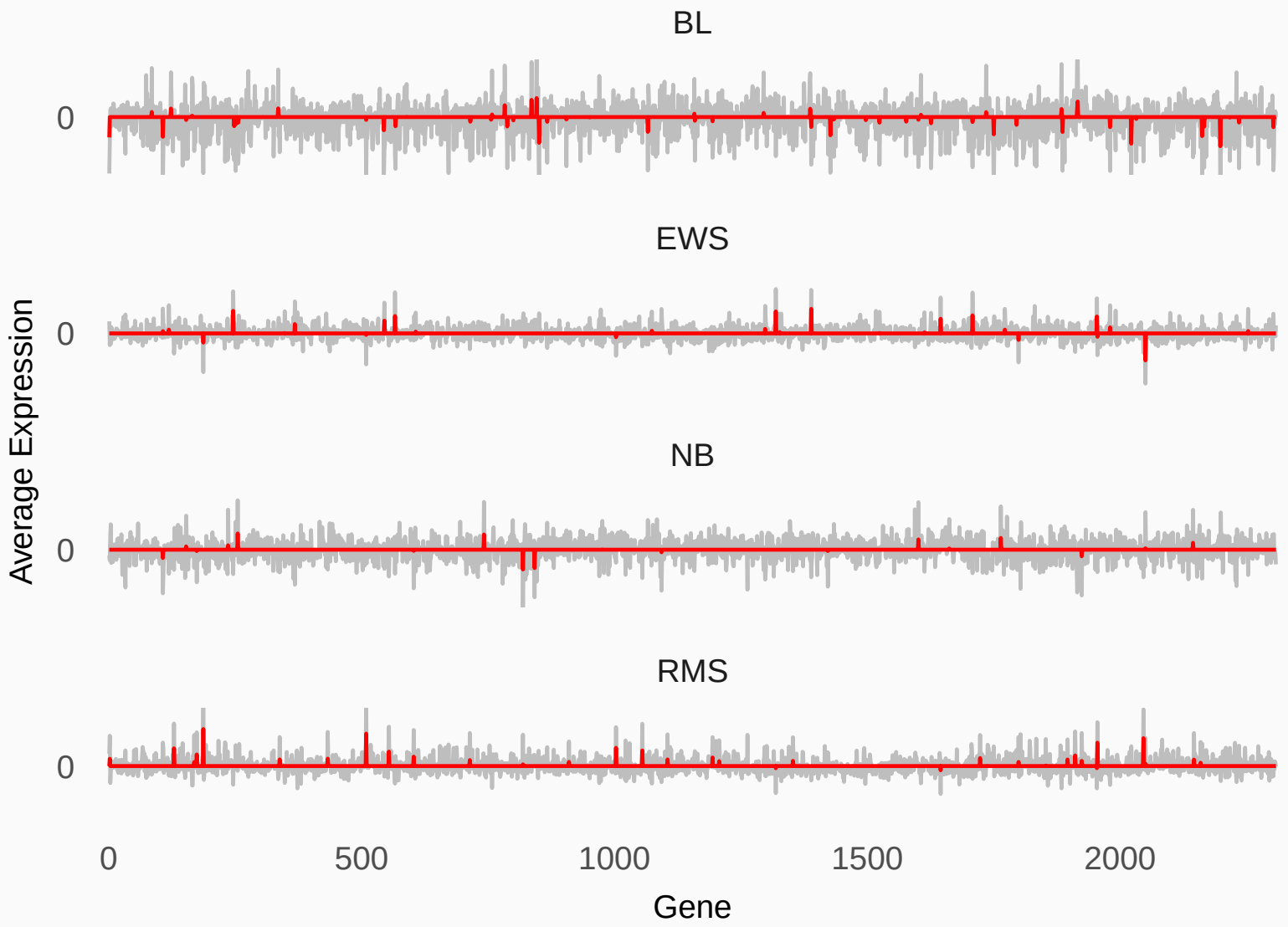
# Application of nearest shrunken centroids (I)

A gene expression data set with $n = 63$ and $p = 2308$. There are four classes (cancer subtypes) with $n_{\text{BL}} = 8$, $n_{\text{EWS}} = 23$, $n_{\text{NB}} = 12$, and $n_{\text{RMS}} = 20$.




5-fold cross-validation curve and largest $\lambda$ that leads to minimal misclassification rate

# Application of nearest shrunken centroids (II)




Grey lines show the original centroids and red lines show the shrunken centroids

# General calculation of the lasso estimates

# Calculation of the lasso estimate

**Last lecture:** When $\mathbf{X}^T\mathbf{X} = \mathbf{I}_p$ and $\boldsymbol{\beta}_{\text{OLS}}$ are the OLS estimates then

$$\widehat{\beta}_{\text{lasso},j}(\lambda) = \text{sign}(\beta_{\text{OLS},j})(|\beta_{\text{OLS},j}| - \lambda)_+ = \text{ST}(\beta_{\text{OLS},j}, \lambda)$$

where $x_+ = \max(x, 0)$ and the **soft-thresholding operator** $\text{ST}$.

**What about the general case?**

**Coordinate Descent:** The lasso problem

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1$$

can be written in coordinates (omitting terms not dependent on any $\beta_i$)

$$\arg\min_{\beta_1,\ldots,\beta_p} \frac{1}{2}\sum_{i,j=1}^{p} \mathbf{x}_i^T\mathbf{x}_j\beta_i\beta_j - \sum_{l=1}^{n}\sum_{i=1}^{p} y_l x_{li}\beta_i + \lambda\sum_{i=1}^{p}|\beta_i|$$

# Subderivative and subdifferential

Let $f : I \to \mathbb{R}$ be a convex function in an open interval $I$ and $x_0 \in I$. A $c \in \mathbb{R}$ is called a **subderivative** of $f$ at $x_0$ if

$$f(x) - f(x_0) \geq c(x - x_0)$$

It can be shown that for

$$a = \lim_{x \to x_0^-} \frac{f(x) - f(x_0)}{x - x_0} \quad \text{and} \quad b = \lim_{x \to x_0^+} \frac{f(x) - f(x_0)}{x - x_0}$$

all $c \in [a, b]$ are subderivatives. Call $\delta f(x_0) := [a, b]$ the **subdifferential** of $f$ at $x_0$.

**Example:** Let $f(x) = |x|$, then

$$\delta f(x_0) = \begin{cases} \{-1\} & x_0 < 0 \\ [-1, 1] & x_0 = 0 \\ \{+1\} & x_0 > 0 \end{cases}$$

# Properties of subdifferentials

1. A convex function is differentiable at $x_0$, if and only if its subdifferential at $x_0$ contains only one point. This point is the derivative.
2. **Moreau-Rockafellar theorem:** If $f, g$ are convex with subdifferentials $\delta f$ and $\delta g$, then

$$\delta(f + g) = \delta f + \delta g$$

   where $\delta f + \delta g = \{v_1 + v_2 : v_1 \in \delta f, v_2 \in \delta g\}$
3. **Stationarity condition:** A point $x_0$ is a global minimum of a convex function, if and only if $0$ is contained in the subdifferential at $x_0$

# Coordinate Descent (I)

**Idea:** Use subdifferentials to find the global minimum $\widehat{\beta}_k$ for a single coefficient $\beta_k$ given the other coefficients $\boldsymbol{\beta}_{-k}$, i.e.

$$\widehat{\beta}_k = \arg\min_{\beta_k} J(\beta_k) = \arg\min_{\beta_k} \frac{1}{2} \sum_{i,j=1}^{p} \mathbf{x}_i^T \mathbf{x}_j \beta_i \beta_j - \sum_{l=1}^{n} \sum_{i=1}^{p} y_l x_{li} \beta_i + \lambda \sum_{i=1}^{p} |\beta_i|$$

Taking the subdifferential of $J$ at $\beta_k$ leads to

$$\delta J(\beta_k) = -\mathbf{x}_k^T \Big( \overbrace{\mathbf{y} - \sum_{\substack{i=1 \\ i \neq k}} \beta_i \mathbf{x}_i}^{=: \mathbf{r}_k} - \beta_k \mathbf{x}_k \Big) + \lambda \begin{cases} \{-1\} & \beta_k < 0 \\ [-1, 1] & \beta_k = 0 \\ \{+1\} & \beta_k > 0 \end{cases} =$$

$$= \begin{cases} \{-\mathbf{x}_k^T \mathbf{r}_k + \beta_k \mathbf{x}_k^T \mathbf{x}_k - \lambda\} & \beta_k < 0 \\ [-\mathbf{x}_k^T \mathbf{r}_k - \lambda, -\mathbf{x}_k^T \mathbf{r}_k + \lambda] & \beta_k = 0 \\ \{-\mathbf{x}_k^T \mathbf{r}_k + \beta_k \mathbf{x}_k^T \mathbf{x}_k + \lambda\} & \beta_k > 0 \end{cases}$$

# Coordinate Descent (II)

Subdifferential of $J$ at $\beta_k$

$$\delta J(\beta_k) = \begin{cases} \{-\mathbf{x}_k^T \mathbf{r}_k + \beta_k \mathbf{x}_k^T \mathbf{x}_k - \lambda\} & \beta_k < 0 \\ [-\mathbf{x}_k^T \mathbf{r}_k - \lambda, -\mathbf{x}_k^T \mathbf{r}_k + \lambda] & \beta_k = 0 \\ \{-\mathbf{x}_k^T \mathbf{r}_k + \beta_k \mathbf{x}_k^T \mathbf{x}_k + \lambda\} & \beta_k > 0 \end{cases}$$

**Two cases:**

1. Standard derivative for $\beta_k \neq 0$, i.e.

$$\frac{\partial J}{\partial \beta_k} = 0 \Leftrightarrow \beta_k = \begin{cases} \frac{\mathbf{x}_k^T \mathbf{r}_k + \lambda}{\mathbf{x}_k^T \mathbf{x}_k} & \mathbf{x}_k^T \mathbf{r}_k < -\lambda \\ \frac{\mathbf{x}_k^T \mathbf{r}_k - \lambda}{\mathbf{x}_k^T \mathbf{x}_k} & \mathbf{x}_k^T \mathbf{r}_k > +\lambda \end{cases}$$

# Coordinate Descent (III)

Subdifferential of $J$ at $\beta_k$

$$\delta J(\beta_k) = \begin{cases} \{-\mathbf{x}_k^T \mathbf{r}_k + \beta_k \mathbf{x}_k^T \mathbf{x}_k - \lambda\} & \beta_k < 0 \\ [-\mathbf{x}_k^T \mathbf{r}_k - \lambda, -\mathbf{x}_k^T \mathbf{r}_k + \lambda] & \beta_k = 0 \\ \{-\mathbf{x}_k^T \mathbf{r}_k + \beta_k \mathbf{x}_k^T \mathbf{x}_k + \lambda\} & \beta_k > 0 \end{cases}$$

**Two cases:**

2. Stationarity condition for $\beta_k = 0$, i.e.

$$0 \in \delta J(0) \Leftrightarrow -\lambda \leq \mathbf{x}_k^T \mathbf{r}_k \leq \lambda$$

# Coordinate Descent (IV)

In total we get the solution

$$\widehat{\beta}_k(\boldsymbol{\beta}_{-k}) = \left\{ \begin{array}{ll} \frac{\mathbf{x}_k^T \mathbf{r}_k + \lambda}{\mathbf{x}_k^T \mathbf{x}_k} & \mathbf{x}_k^T \mathbf{r}_k < -\lambda \\ 0 & -\lambda \leq \mathbf{x}_k^T \mathbf{r}_k \leq \lambda \\ \frac{\mathbf{x}_k^T \mathbf{r}_k - \lambda}{\mathbf{x}_k^T \mathbf{x}_k} & \mathbf{x}_k^T \mathbf{r}_k > \lambda \end{array} \right\} = \frac{\mathrm{ST}(\mathbf{x}_k^T \mathbf{r}_k, \lambda)}{\mathbf{x}_k^T \mathbf{x}_k}$$

the unique minimizer when all coefficients but $\beta_k$ are fixed.

Multiple options for updating order

- **Cyclic coordinate descent:** Update one coordinate at a time in a fixed order. Once every coordinate has been updated, start over.
- Choose coordinate that leads to best decrease in total target function value

**Note:** Coordinate descent is not guaranteed to converge if the target function's level curves are not smooth

# Another algorithmic approach

- Cyclic coordinate descent is a popular approach (e.g. R package `glmnet`) but is hard to parallelise due to its sequential order
- **Augmented Directions Method of Multipliers (ADMM)** re-formulates the lasso problem

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1 \quad \text{as}$$

$$\arg\min_{\boldsymbol{\beta},\theta} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\theta||_1 \quad \text{such that} \quad \boldsymbol{\theta} = \boldsymbol{\beta}$$

and (approximately) minimizes the **augmented Lagragian**

$$\arg\min_{\boldsymbol{\beta},\theta} \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\theta||_1 + \mathbf{y}^T(\boldsymbol{\beta} - \theta) + \frac{\rho}{2}||\boldsymbol{\beta} - \theta||_2^2$$

Iteratively solves for $\boldsymbol{\beta}$ and $\theta$, then updates $\mathbf{y}$

# Extensions of the lasso

# The lasso and groups of highly correlated variables

- The lasso does not handle groups of highly correlated variables well.
- **Example:** Two groups of highly correlated variables, e.g.

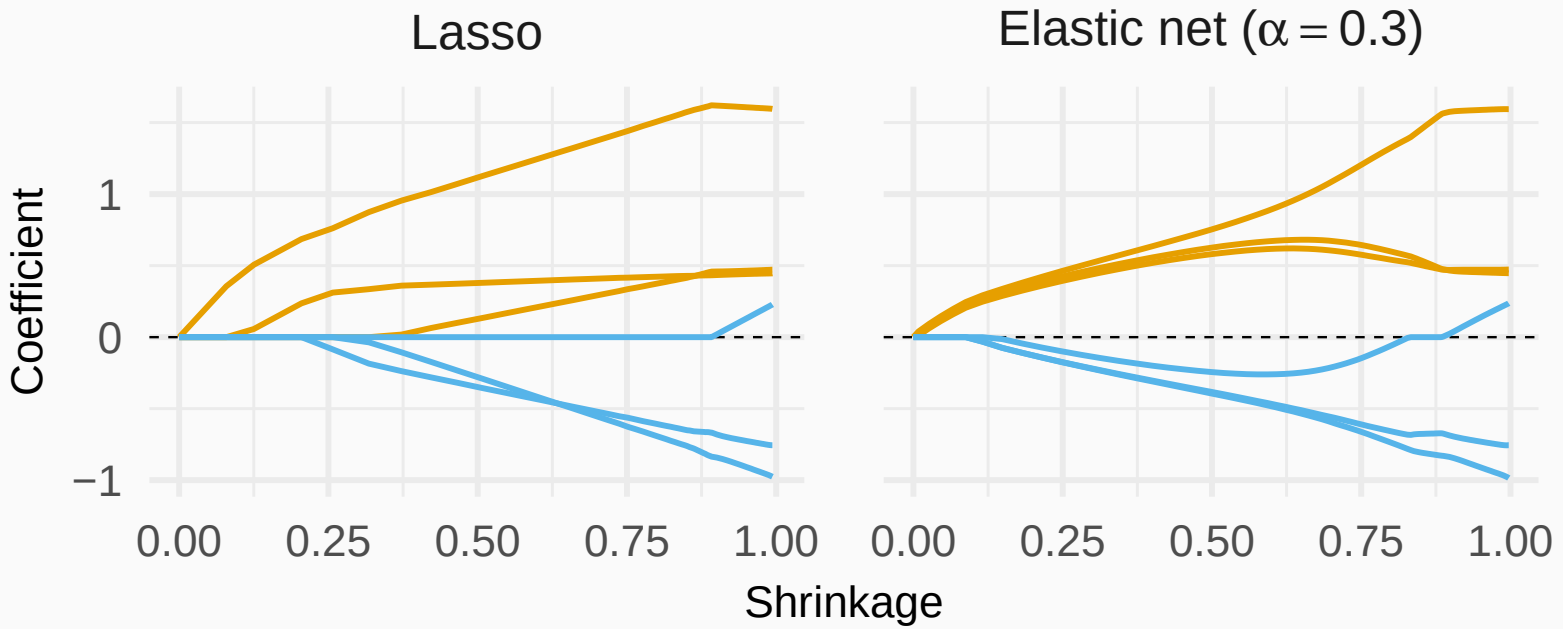
$$\mathbf{X} \sim N(\mathbf{0}, \boldsymbol{\Sigma}) \quad \text{where} \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_1 \end{pmatrix}, \boldsymbol{\Sigma}_1 = \begin{pmatrix} 1.04 & 1 & 1 \\ 1 & 1.04 & 1 \\ 1 & 1 & 1.04 \end{pmatrix}$$

and

$$\mathbf{y} = 3\mathbf{x}_1 - 1.5\mathbf{x}_5 + \boldsymbol{\varepsilon} \quad \text{where} \quad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, 4\mathbf{I}_n)$$

- The (very theoretical) **irrepresentable condition** from last lecture tells us that the lasso will not be able to recover the true model.
- What happens in practice?

# The lasso and groups of highly correlated variables in practice



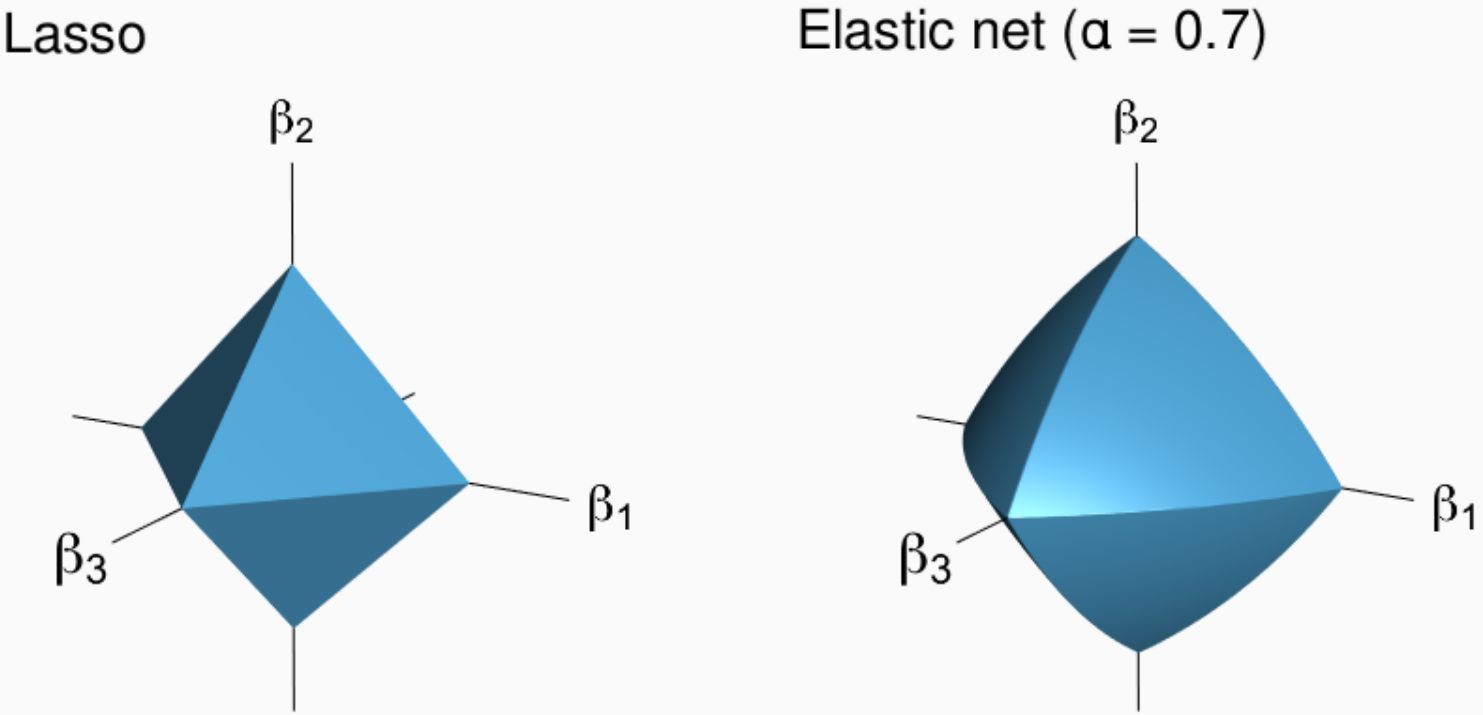

- The lasso typically selects one variable from a group of highly correlated variables, more or less randomly, instead of distributing the coefficients evenly.
- The **elastic net** is an extension of the lasso, which "finds" correlated groups of variables
- Note that the elastic net is not given explicit knowledge of the groups of variables.

# The elastic net (I)

The **elastic net** solves the problem

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\left(\frac{1-\alpha}{2}\|\boldsymbol{\beta}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_1\right)$$

striking a balance between lasso (**variable selection**) and ridge regression (**grouping of variables**)



# The elastic net (II)

The solution can be found through **cyclic coordinate descent** and the coefficient updates are

$$\widehat{\beta}_k(\boldsymbol{\beta}_{-k}) = \frac{\mathrm{ST}(\mathbf{x}_k^T \mathbf{r}_k, \lambda\alpha)}{\mathbf{x}_k^T \mathbf{x}_k + \lambda(1-\alpha)}$$

- **Note:** $\alpha$ is an additional tuning parameter that should be determined by cross-validation
- The lasso and ridge regression are special cases of the elastic net ($\alpha = 1$ and $\alpha = 0$, respectively). The R package `glmnet` is a popular implementation of the elastic net.

# The lasso and groups of variables

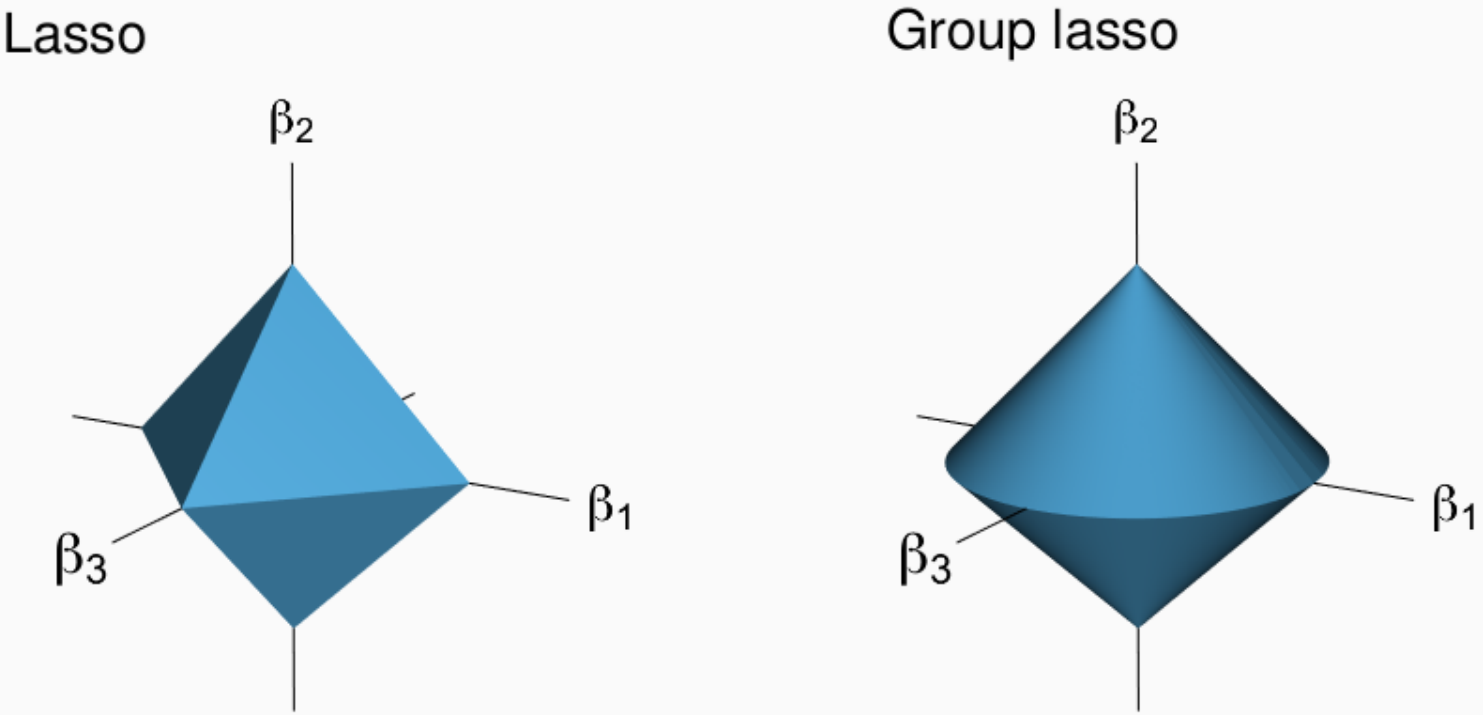
- The lasso in it's original formulation considers each variable separately
- Groups in data can form through e.g.
  - Correlation
  - Categorical variables in dummy encoding
  - Domain-knowledge (e.g. genes in the same signal pathway, signals that only appear in groups in a compressed sensing problem,...)
- Ideally the whole group is either present or not
- The elastic net can find groups, but only does so for highly correlated variables and without external influence. Sometimes more control is necessary.

# The group lasso (I)

The **group lasso** solves the problem

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{k=1}^{K} \|\mathbf{B}_k\|_2$$

where $\mathbf{B}_k$ is a vector of coefficients $\beta_i$ for the $k$-th group. Note that $\|\beta_i\|_2 = |\beta_i|$ for singleton groups.



# The group lasso (II)

The solution can be similarly derived as for the lasso or the elastic net, but on group level. This leads to the coefficient update

$$\mathbf{B}_k^{(i+1)} = \left( \mathbf{X}_k^T \mathbf{X}_k + \frac{\lambda}{\left\| \mathbf{B}_k^{(i)} \right\|_2} \mathbf{I} \right)^{-1} \mathbf{X}_k \mathbf{r_k} \quad \text{when} \quad \|\mathbf{X}_k \mathbf{r}_k\|_2 > 0$$

and 0 otherwise. Note that $\mathbf{X}_k$ contains all predictors belonging to group $k$ and

$$\mathbf{r}_k = \mathbf{y} - \sum_{\substack{j=1 \\ j \neq k}} \mathbf{X}_j \mathbf{B}_j$$

# Take-home message

- Penalisation methods are not only restricted to regression, also applicable to classification
- Sparsity is a very important concept when interpretability of models is important
- Many extensions to the lasso exist, which make it more suitable for a variety of different situations