Advanced statistical topics in medical research pt. A

Cross-validation
Penalised regression
(Bootstrapping)

Benjamin Skov Kaas-Hansen
PhD candidate pharmacovigilance & data science
MD, MSc epidemiology and biostatistics

18 November 2021

Code: https://github.com/epiben/course_adv_stats_A

R code available "[heading]"

Two words on terminology

Independent variables, covariates, predictors and features are used as synonyms (depending on discipline)

Statistical modelling ≠ machine learning

Regression vs. classification (purpose)

Train on: "point or aim something, typically a gun or camera, at" —so not as "træne" in Danish!

Generalised linear models

A refresher on a classic in **parametric** statistical modelling

Linear regression models

Regression equation $y_i = \beta_0 + x_1\beta_1 + x_2\beta_2 + \cdots + x_p\beta_p + \varepsilon_i$

or in matrix notation $y = \mathbf{X}\beta + \varepsilon$

Overall trend $\mathbb{E}(\mathbf{Y}) = \mathbf{X}\beta$

Generalised linear models

Extends the linear model to other outcomes and distributions. Focus on **mean effect** but for transformed data

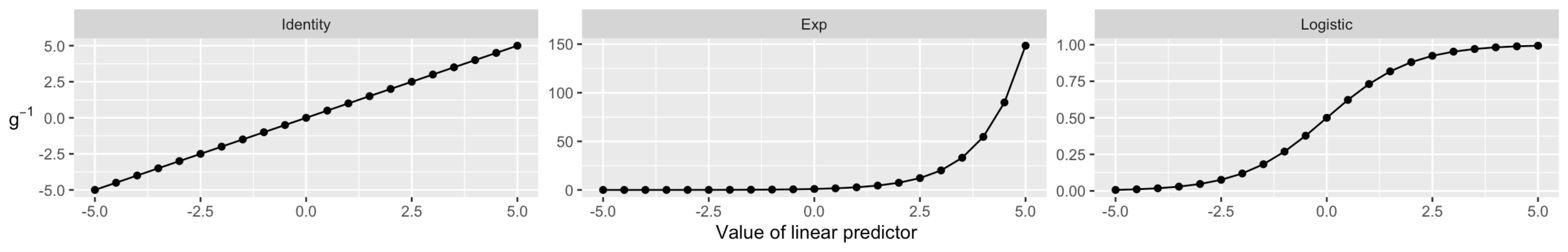
$$g(\mathbb{E}[\mathbf{Y}_i]) = \beta_0 + \mathbf{X}_{1i}\beta_1 + \cdots + \mathbf{X}_{pi}\beta_p$$

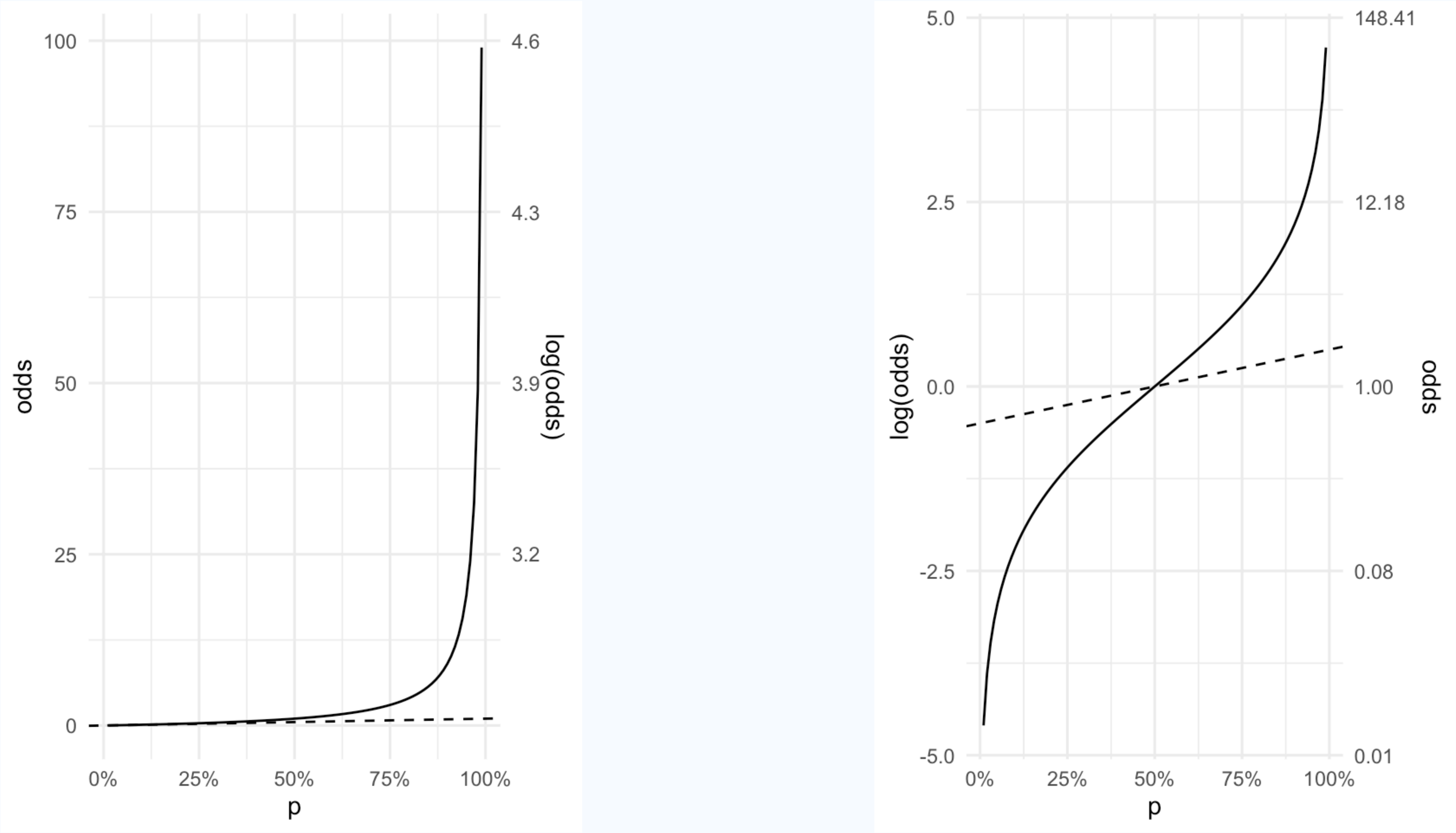
where the link function g maps the population mean into the linear predictor.

Estimator in linear regression

The **link function** g maps the population mean into the linear predictor, and g^{-1} maps the linear predictor into the scale of the population mean:

$$g(\mathbb{E}[y_i]) = \mathbf{X}_i\beta \iff \mathbb{E}[y_i] = g^{-1}(\mathbf{X}_i\beta)$$




100
75
50
25
0
odds
4.6
4.3
3.9
3.2
log(odds)
0%
25%
50%
75%
100%
p
5.0
2.5
0.0
-2.5
-5.0
log(odds)
148.41
12.18
1.00
0.08
0.01
odds
0%
25%
50%
75%
100%
p

Example: logistic regression

N = 2201 individuals on the Titanic. Who survived?

Binary outcome. Info on class (1st, 2nd, 3rd), sex, age group, and survival status.

$$\log\left(\frac{P(Y=1)}{P(Y=0)}\right) = \log\left(\frac{P(Y=1)}{1-P(Y=1)}\right) = \beta_0 + x_1\beta_1 + \cdots + x_p\beta_p$$

The ordinary least-squares estimator is

$$\hat{\beta} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{y}$$

and minimises the least-squared function (residual sum of squares):

$$(\mathbf{y} - \mathbf{X}\hat{\beta})^t(\mathbf{y} - \mathbf{X}\hat{\beta}) = ||\mathbf{y} - \mathbf{X}\beta||_2^2$$

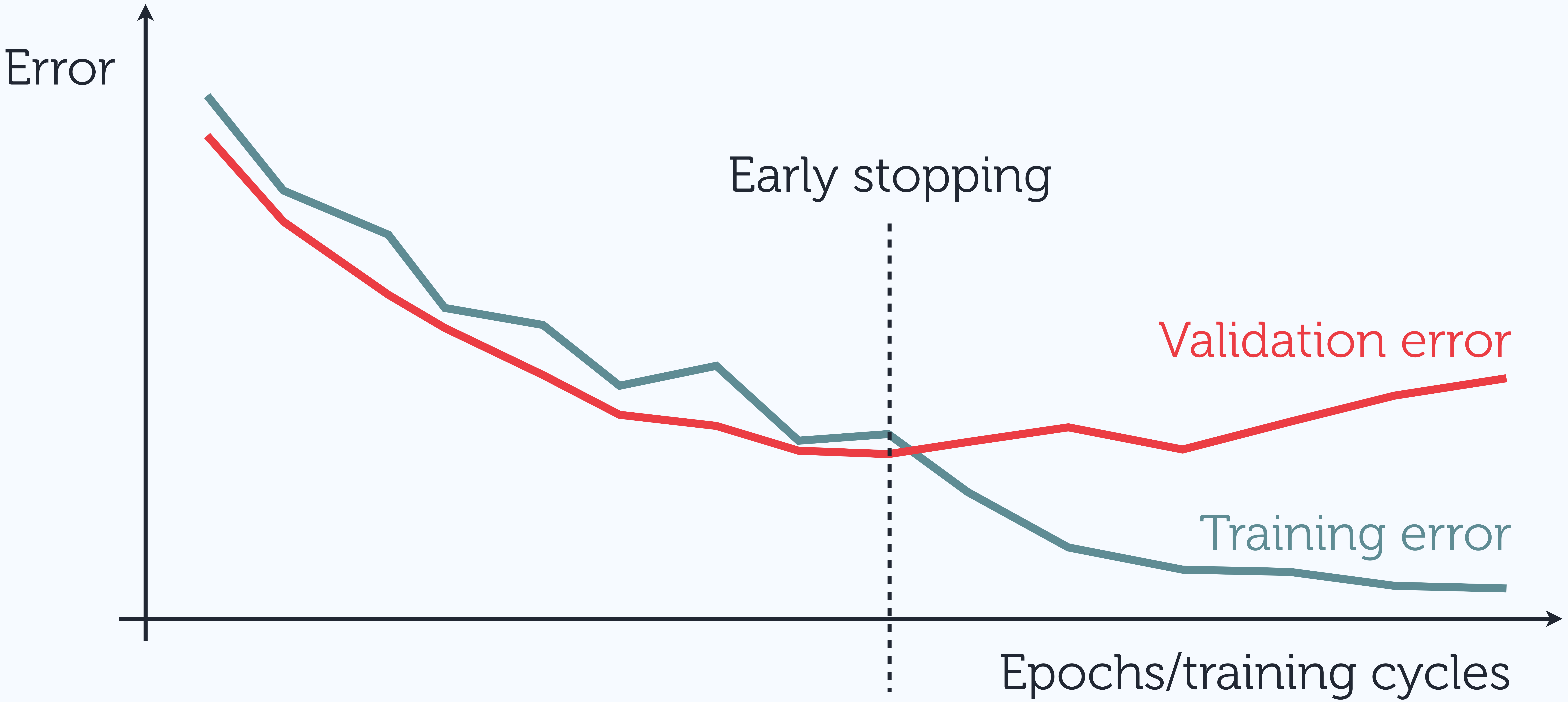
What to do when many predictors (large P)?

We'll come back to this

Cross-validation

Using the same data for fitting/training and optimisation leads to **overfitting** which hurts generalisability to other (similar) populations or future observations in the same population.

Learning curves



Prediction error

Let y_i be the observed response and $\hat{y}_i$ the prediction based on a model and predictors x_i.

Two common choices for quantifying prediction error are:

Continuous outcome: $D(y, \hat{y}) = (y - \hat{y})^2$

Binary outcome: $D(y, \hat{y}) = \begin{cases} 1 & \text{if } y \neq \hat{y} \\ 0 & \text{if } y = \hat{y} \end{cases}$

Error rates

Let y_0 be a new observation and $\hat{y}_0$ the corresponding output of a fixed prediction method, then the true error rate is

$$\mathbb{E}[D(y_0, \hat{y}_0)]$$

Because we only have the data already collected, the apparent error rate is

$$\frac{1}{N} \sum_i D(y_i, \hat{y}_i)$$

Apparent error rate is imperfect

The same data used to train and evaluate the prediction model

The apparent error rate becomes too small (over-optimistic, biased)

Evaluation on a validation set not seen during training gives unbiased estimate of the model's performance

Cross-validation obtains validation data from the original data

Leave-one-out cross-validation

Intuitive and simple - but compute time can become prohibitive

1. Drop data point (x_i, y_i) and train prediction model
2. Compute prediction error $D(y_i, \hat{y}_i)$
3. Repeat 1. and 2. for $i = 1,2,\ldots,N$
4. Compute the LOO-CV error rate:

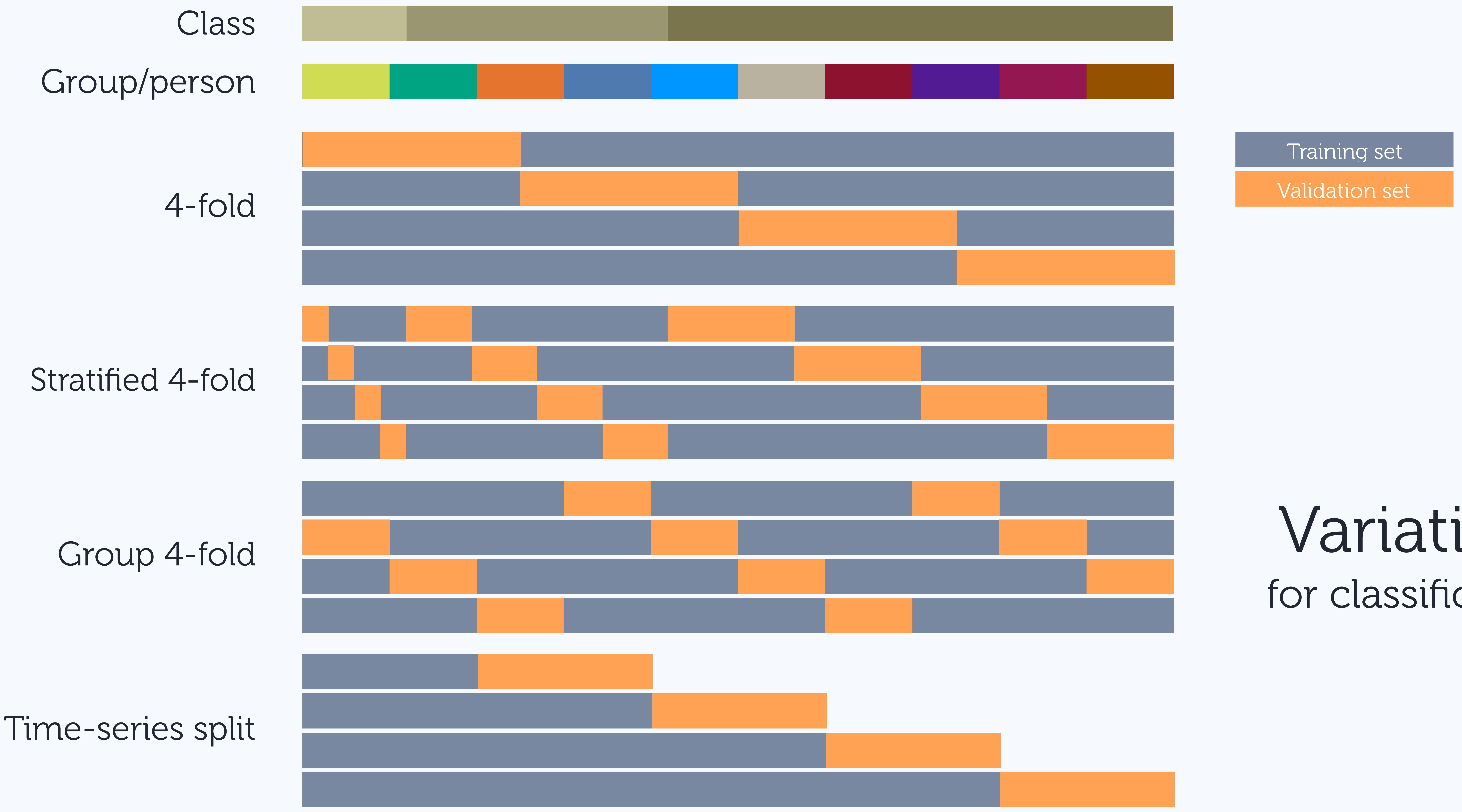
$$\text{Err}_{\text{LOOCV}} = \frac{1}{N}\sum_i D(y_i, \hat{y}_i)$$

K-fold cross-validation $K \ll N$

1. Partition data into K equal-sized folds
2. Train prediction model with all but the k'th fold
3. Compute $D(y_k, \hat{y}_k)$
4. Repeat 2. and 3. for $k = 1,2,\ldots,K$
5. Compute the CV error rate: $\mathrm{Err}_{\mathrm{CV}} = \frac{1}{K}\sum_k D(y_k, \hat{y}_k)$

Smaller values of K gives fewer models builds (shorter runtime), groups that vary more and, thus, greater variation between prediction models

Variations
for classifications



Notes of caution on CV

If you use cross-validation to optimise model settings/ hyperparameters, the validation folds become training data

You can use a split-sample validation scheme:
80% for development and 20% in hold-out test set

External validation requires new or distinct data

Lasso and ridge regression

Flexible (linear) modelling for predictor selection and to counter over-fitting. Many non-parametric methods exist: tSNE, UMAP, variational auto-encoders (DL), etc.

Wide data (large-p small-n)

E.g. SNPs or deep phenotyping

Over-parameterised

Unlikely to generalise well

Cannot learn patterns/associations

Similar problem in deep learning

y	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9
0	0	1	0	1	1	1	1	1	1
1	1	1	0	0	1	1	1	1	0
1	0	1	0	1	0	0	0	0	0
0	0	1	0	1	0	0	0	1	1

Sparse prediction model

CURB65	Patient	Points
Confusion	Yes	1
BUN > 7 mmol/L	No	0
RF ≥ 30	No	0
SBP < 90 or DBP ≥ 60	Yes	1
Age ≥ 65	Yes	1
30-day mortality	14 %	3

Bed-side clinical scoring tool that can be done by hand

Less data required, perhaps desirable w.r.t. external validation

High-dimensional propensity score models

Imagine a linear regression

$$\text{BMI} = \text{gene}_1 \cdot \beta_1 + \text{gene}_2 \cdot \beta_2$$

Lasso regression (L1 regularisation)

Assume a linear mean effect: $y = \mathbf{X}\beta + \varepsilon$

The **Lasso** estimates β by minimising the penalised least-squares function

penalty

$$Z_n(\beta) = \|(\mathbf{y} - \mathbf{X}\beta)\|_2^2 + \lambda_n \|\beta\|_1$$

l1 norm

so the lasso (= penalised) estimate $\hat{\beta}_{\text{lasso}} = \arg\min_{\beta \in \mathbb{R}^p} Z_n(\beta)$

Properties of lasso regression

"Always" useful
($P < N$, $P > N$ and $P \gg N$)

Selects **sparse** model

Yields accurate predictions

Inconsistent variable selection

Non-standard limiting distribution

No oracle property

Multiple testing problem

Example: Biopsies from Breast Cancer Patients

Biopsies of breast tumours in 699 patients up to 15 July 1992 with binary outcome: benign or malignant.

There are nine attributes (predictors), each scored between 1 to 10: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei (16 values are missing), bland chromatin, normal nucleoli, mitoses.

Ridge regression (L2 regularisation)

Assume again the linear mean effect: $y = \mathbf{X}\beta + \varepsilon$

The **Ridge** regression estimates β by minimising the penalised least-squares function

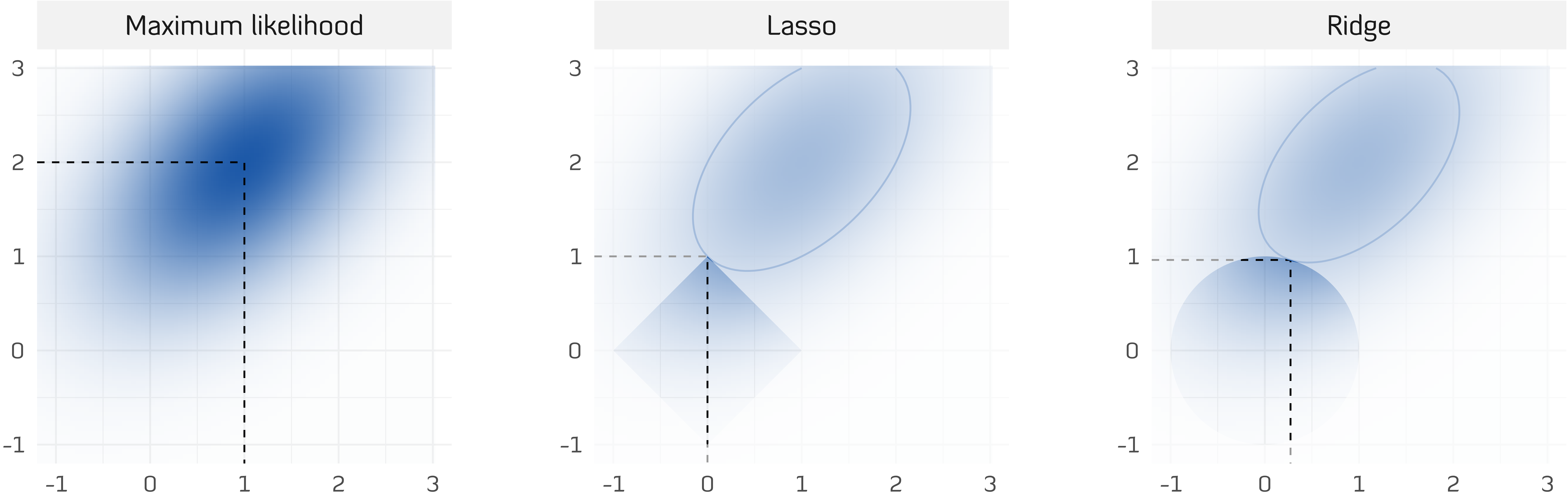
penalty

$$Z_n(\beta) = ||(\mathbf{y} - \mathbf{X}\beta)||_2^2 + \lambda_n ||\beta||_2^2$$

l2 norm (squared)

so the ridge (= penalised) estimate $\hat{\beta}_{\text{ridge}} = \arg\min_{\beta \in \mathbb{R}^P} Z_n(\beta)$

Lasso and Ridge



Elastic net (L1 and L2)

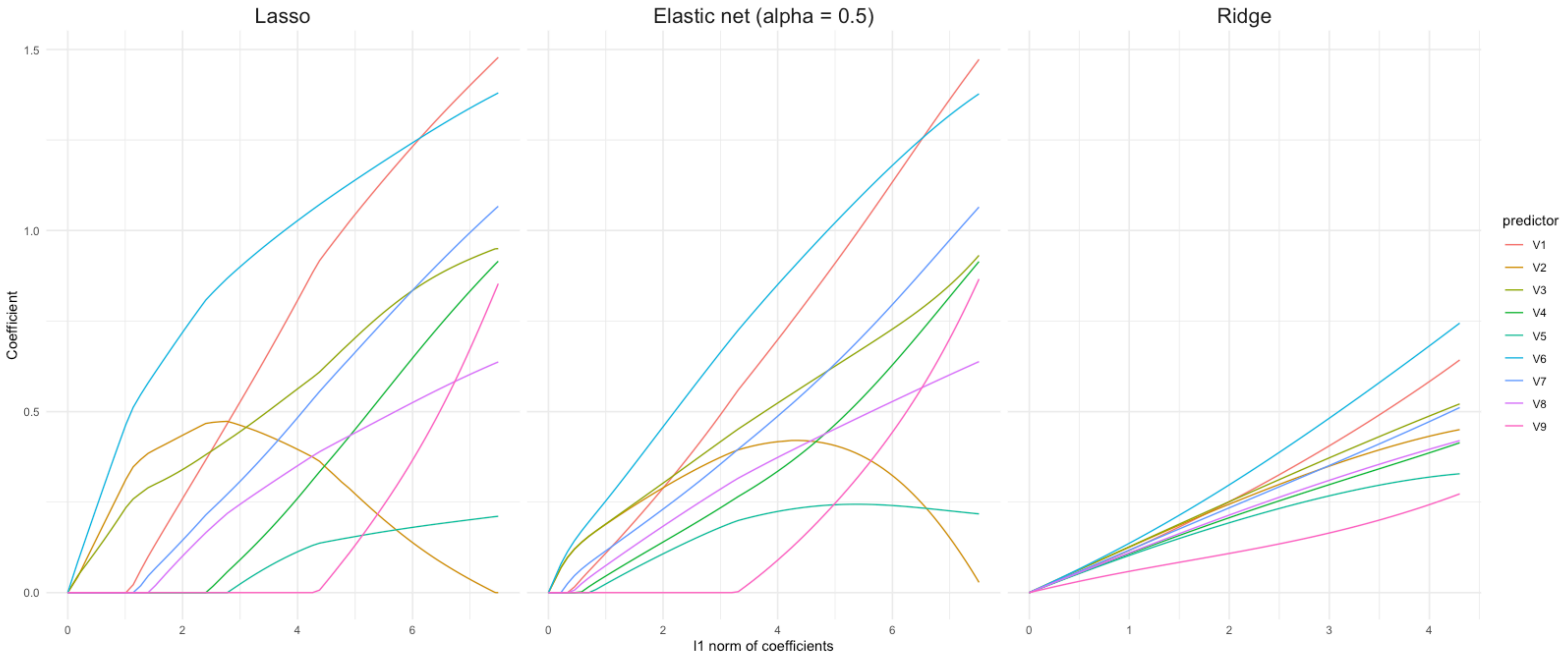
Combines the sparsity of the lasso with the flexibility of the ridge by weighting the contribution of each of them:

$$Z_n(\beta) = ||(\mathbf{y} - \mathbf{X}\beta)||_2^2 + \alpha\lambda_n||\beta||_1 + (1-\alpha)\lambda_n||\beta||_2^2$$

The elastic net handles very correlated predictors better than the lasso because it does not choose but keeps both with appropriate shrinkage

This yields two parameters to optimise over: λ_n and α

How to choose the penalty?



Delassoing and selective inference

Lasso yields a list of shrunken parameter estimates $\hat{\beta}_{(1)}, 0, \hat{\beta}_{(3)}, 0, 0, \ldots, 0, \hat{\beta}_{(k)}, 0, \ldots$

Which are actually significant? Delassoing is a way to answer this question

Pick lasso predictors, and use these in normal (G)LM

Careful! Multiple testing, selection algorithm, bias, lack of small-sample test statistic, ...

Selective inference computes p-values and CI's for the lasso estimates at fixed value of the tuning parameter λ

Perhaps better off with a (quasi)causal structure and pick predictors based on that