Lecture 7:
Linear models & Regularization

Olexandr Isayev

Department of Chemistry, CMU

olexandr@cmu.edu

In class project

Please submit information about your projects:

- Title
- Names of team members
- Briefly describe the problem you are trying to solve and rough size of your dataset.

Deadline: Sunday, October 8

Linear Regression

Linear models can be used to model the dependence of a regression target y on some features x. The learned relationships are linear and can be written for a single instance i as follows:

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \epsilon$$

The predicted outcome of an instance is a weighted sum of its p features. The betas (βj) represent the learned feature weights or coefficients. The first weight in the sum (β0) is called the intercept and is not multiplied with a feature. The epsilon (ϵ) is the error we still make, i.e. the difference between the prediction and the actual outcome.

These errors are assumed to follow a Gaussian distribution, which means that we make errors in both negative and positive directions and make many small errors and few large errors.

Model Optimization

The ordinary least squares method is usually used to find the weights that minimize the squared differences between the actual and the estimated outcomes:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta_0,\ldots,\beta_p} \sum_{i=1}^{n} \left(y^{(i)} - \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j^{(i)} \right) \right)^2$$

Advantage

The biggest advantage of linear regression models is linearity: It makes the estimation procedure simple and, most importantly, these linear equations have an easy to understand interpretation by individual weights

Linear model and all similar models are so widespread in academic fields such as medicine, sociology, psychology, and many other quantitative research fields.

For example, in the medical field, it is not only important to predict the clinical outcome of a patient, but also to quantify the influence of the drug and at the same time take sex, age, and other features into account in an interpretable way.

Estimated weights come with confidence intervals.

Assumptions

Linearity

The linear regression model forces the prediction to be a linear combination of features, which is both its greatest strength and its greatest limitation.

Linearity leads to interpretable models.

Linear effects are easy to quantify and describe. They are additive, so it is easy to separate the effects.

If you suspect feature interactions or a nonlinear association of a feature with the target value, you can add interaction terms or use regression splines.

Assumptions

Normality
It is assumed that the target outcome given the features follows a normal distribution. If this assumption is violated, the estimated confidence intervals of the feature weights are invalid.

Homoscedasticity (constant variance)
The variance of the error terms is assumed to be constant over the entire feature space.

This assumption is often violated in reality.

Assumptions

Independence
It is assumed that each instance is independent of any other instance. If you perform repeated measurements, such as multiple blood tests per patient, the data points are not independent. For dependent data you need special linear regression models

Fixed features
The input features are considered “fixed”. Fixed means that they are treated as “given constants” and not as statistical variables. This implies that they are free of measurement errors.

This is a rather unrealistic assumption.

Assumptions

Absence of multicollinearity
You do not want strongly correlated features, because this messes up the estimation of the weights.

In a situation where two features are strongly correlated, it becomes problematic to estimate the weights because the feature effects are additive and it becomes indeterminable to which of the correlated features to attribute the effects.

Interpretation

- Numerical feature: Increasing the numerical feature by one unit changes the estimated outcome by its weight..
- Binary feature: Changing the feature from the reference category to the other category changes the estimated outcome by the feature's weight.
- Categorical feature with multiple categories: One-hot encoding. For a categorical feature with L categories, you only need L-1 columns, because the L-th column would have redundant information (e.g. when columns 1 to L-1 all have value 0 for one instance, we know that the categorical feature of this instance takes on category L). The interpretation for each category is then the same as the interpretation for binary features.
- Intercept $\beta 0$

Feature Importance

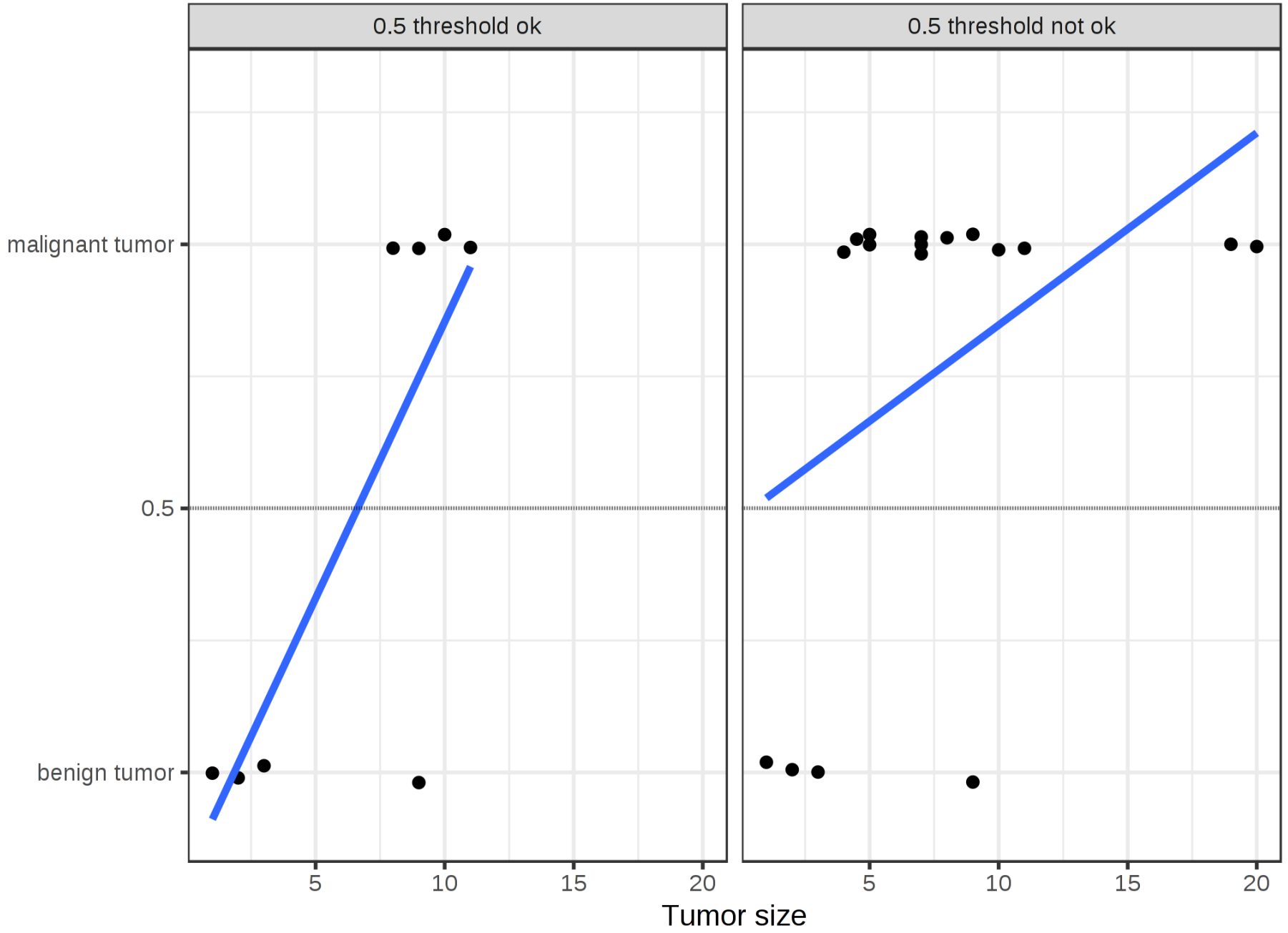
The importance of a feature in a linear regression model can be measured by the absolute value of its t-statistic.

The t-statistic is the estimated weight scaled with its standard error.

$$t_{\hat{\beta}_j} = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}$$

Logistic Regression

- Logistic regression models the probabilities for classification problems with two possible outcomes.
- It’s an extension of the linear regression model for classification problems.
- A linear model does not output probabilities, but it treats the classes as numbers (0 and 1) and fits the best hyperplane that minimizes the distances between the points and the hyperplane.
- So it simply interpolates between the points, and you cannot interpret it as probabilities.
- A linear model also extrapolates and gives you values below zero and above one.

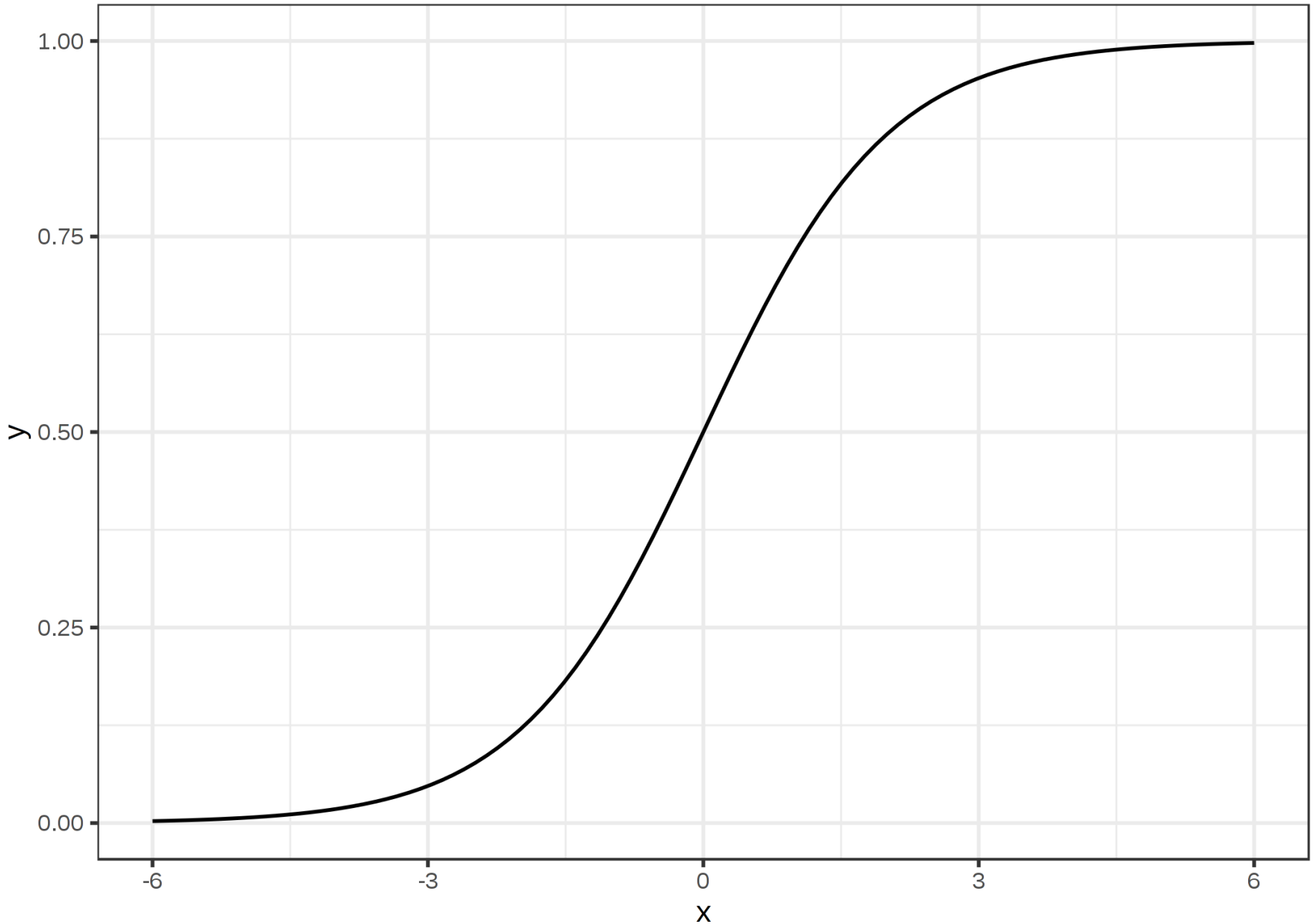

0.5 threshold ok
0.5 threshold not ok
malignant tumor
0.5
benign tumor
5
10
15
20
Tumor size

Solution

A solution for classification is logistic regression.

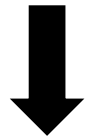
Instead of fitting a straight line or hyperplane, the logistic regression model uses the logistic function to squeeze the output of a linear equation between 0 and 1.

$$\text{logistic}(\eta) = \frac{1}{1 + exp(-\eta)}$$

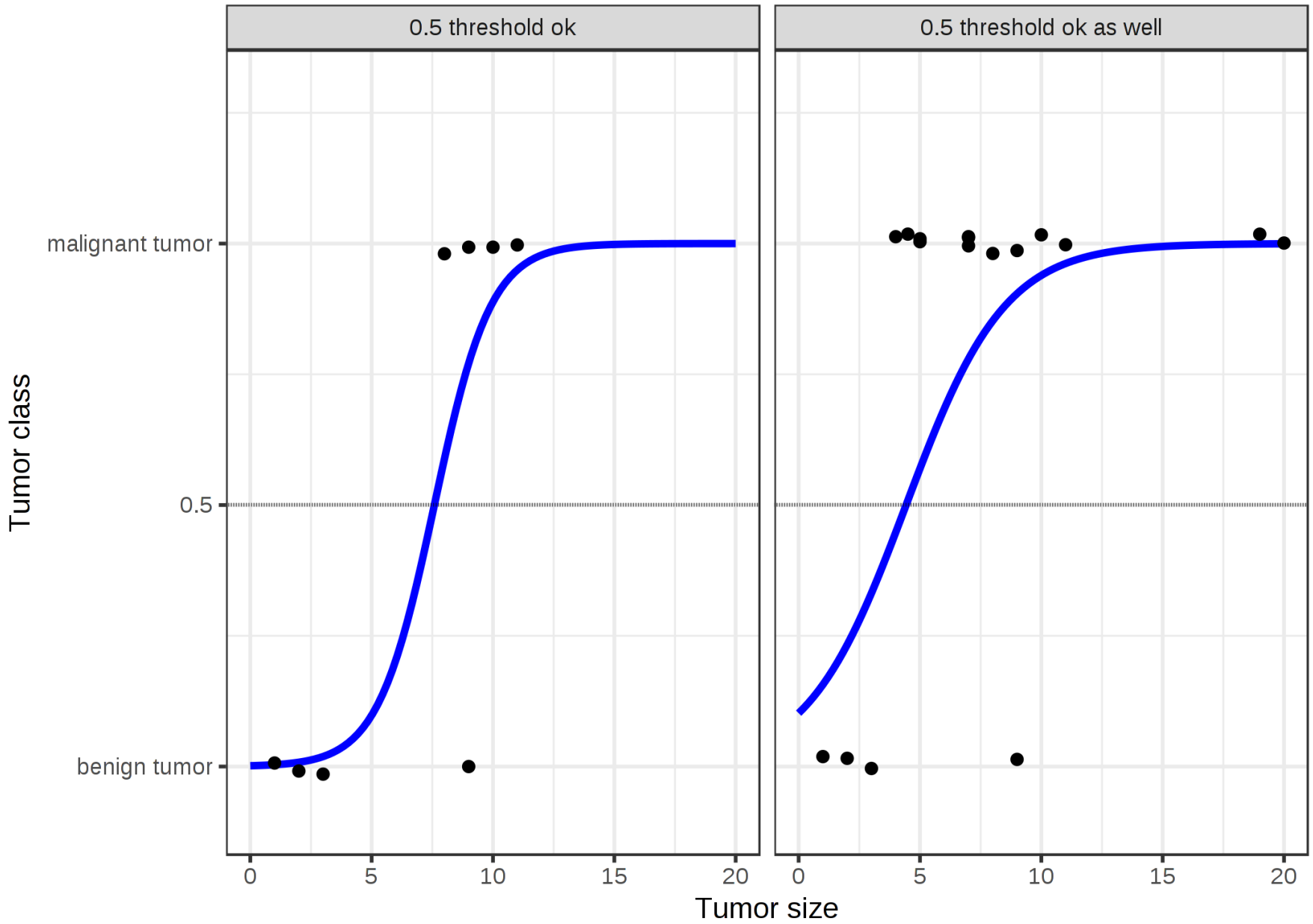

1.00
0.75
0.50
0.25
0.00
y
-6
-3
0
3
6
x

Logistic Regression

$$\hat{y}^{(i)} = \beta_0 + \beta_1 x_1^{(i)} + \ldots + \beta_p x_p^{(i)}$$



$$P(y^{(i)} = 1) = \frac{1}{1 + exp(-(\beta_0 + \beta_1 x_1^{(i)} + \ldots + \beta_p x_p^{(i)}))}$$


0.5 threshold ok
0.5 threshold ok as well
malignant tumor
0.5
benign tumor
Tumor class
0
5
10
15
20
Tumor size

Interpretation

The interpretation of the weights in logistic regression differs from the interpretation of the weights in linear regression, since the outcome in logistic regression is a probability between 0 and 1.

The weights do not influence the probability linearly any longer.

log() function “odds”:

$$\frac{P(y=1)}{1-P(y=1)} = odds = exp\left(\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p\right)$$

Interpretation

$$\frac{odds_{x_j+1}}{odds} = exp\left(\beta_j(x_j+1) - \beta_j x_j\right) = exp\left(\beta_j\right)$$

A change in xj by one unit increases the log odds ratio by the value of the corresponding weight.

Most people interpret the odds ratio because thinking about the log() of something is known to be hard on the brain.

Interpretation

- Numerical feature: If you increase the value of feature xj by one unit, the estimated odds change by a factor of exp(βj)

- Binary categorical feature: Changing the feature xj from the reference category to the other category changes the estimated odds by a factor of exp(βj).
- Categorical feature with more than two categories: One solution to deal with multiple categories is one-hot-encoding, meaning that each category has its own column. You only need L-1 columns for a categorical feature with L categories, otherwise it is over-parameterized. The L-th category is then the reference category. You can use any other encoding that can be used in linear regression. The interpretation for each category then is equivalent to the interpretation of binary features.
- Intercept β0: The interpretation of the intercept weight is usually not relevant.

Disadvantages

- LR struggles with its restrictive expressiveness (e.g. interactions must be added manually) and other models may have better predictive performance

- Interpretation is more difficult because the interpretation of the weights is multiplicative and not additive.

- Logistic regression can suffer from **complete separation**. If there is a feature that would perfectly separate the two classes, the logistic regression model can no longer be trained. This is because the weight for that feature would not converge, because the optimal weight would be infinite.

LR and Regularization

- *If there is noise in the training data, then the estimated coefficients won't generalize well to the future data. This is where regularization comes in and shrinks or regularizes these learned estimates towards zero.*

- **Regularization** is a form of regression, that constrains/ regularizes or shrinks the coefficient estimates towards zero.

- In other words, ***this technique discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.***

Stepwise Selection

- Forward stepwise selection:
  - First, we approximate the response variable y with a constant (i.e., an intercept-only regression model).
  - Then we gradually add one more variable at a time (or add main effects first, then interactions).
  - Every time we always choose from the rest of the variables the one that yields the best accuracy in prediction when added to the pool of already selected variables.

  - For example, if we have 10 predictor variables, first we would approximate *y* with a constant, and then use one variable out of the 10 (I would perform 10 regressions, each time using a different predictor variable; for every regression I have a residual sum of squares; the variable that yields the minimum residual sum of squares is chosen and put in the pool of selected variables). We then proceed to choose the next variable from the 9 left, etc.
- Backward stepwise selection: This is similar to forward stepwise selection, except that we start with the full model using all the predictors and gradually delete variables one at a time.

Shrinkage

Shrinkage refers to a class of regularization methods that involve fitting a regression model using all *p* predictors, under some constraint on the size of their estimated coefficients

Among the most important features of this regularization approach, we highlight that shrinking

- tends to reduce the variability of the estimates, hence improving the model's stability
- I can go as far as to set some of the coefficients to zero, thus also allowing for variable selection.

Today we discuss the **Ridge**, **LASSO**, and **elastic net** approaches

The Ridge regression

The solution to the ordinary least squares fitting procedure is the vector $\hat{\beta} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)$ that minimizes the Residual Sum of Squares

$$RSS = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

Ridge regression, similarly, seeks the vector $\hat{\beta}^{\text{Ridge}}$ that minimizes the *penalized* or *regularized* RSS

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

where $\lambda \geq 0$ is a **complexity parameter**.

The shrinkage effect

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

called a **shrinkage penalty** and is small when $\beta_1, \ldots, \beta_p$ are close to zero, hence it has the effect of shrinking the coefficients towards zero.

The tuning parameter λ acts as a regulator of the amount of shrinkage on the regression estimates:

$$\text{if } \lambda = 0, \text{then } \hat{\beta}^{\text{Ridge}} \equiv \hat{\beta}$$

$$\text{if } \lambda = \infty, \text{then } \hat{\beta}^{\text{Ridge}} = 0$$

Note: unlike least squares, $\hat{\beta}^{\text{Ridge}}$ is not unique, rather a function of λ. Selecting good values for λ is critical, and is usually done numerically via cross-validation

Standardization of X

Since the Ridge solutions, unlike the standard OLS estimates, are not equivariant under scaling, it is common to standardize the inputs before estimation.

Moreover, penalization of the model intercept would make the procedure depend on the origin chosen for the dependent variable.

Hence β_0 is estimated separately.

- The remaining parameters are then estimated by a Ridge regression without intercept, using the standardized covariates. Let **X** denote the $p \times p$ (standardized) data matrix. It is easy to show that the solutions take the form:

$$\hat{\beta}^{\text{ridge}} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$$

- where **I** is the $p \times p$ identity matrix.

Ridge coefficients profiles

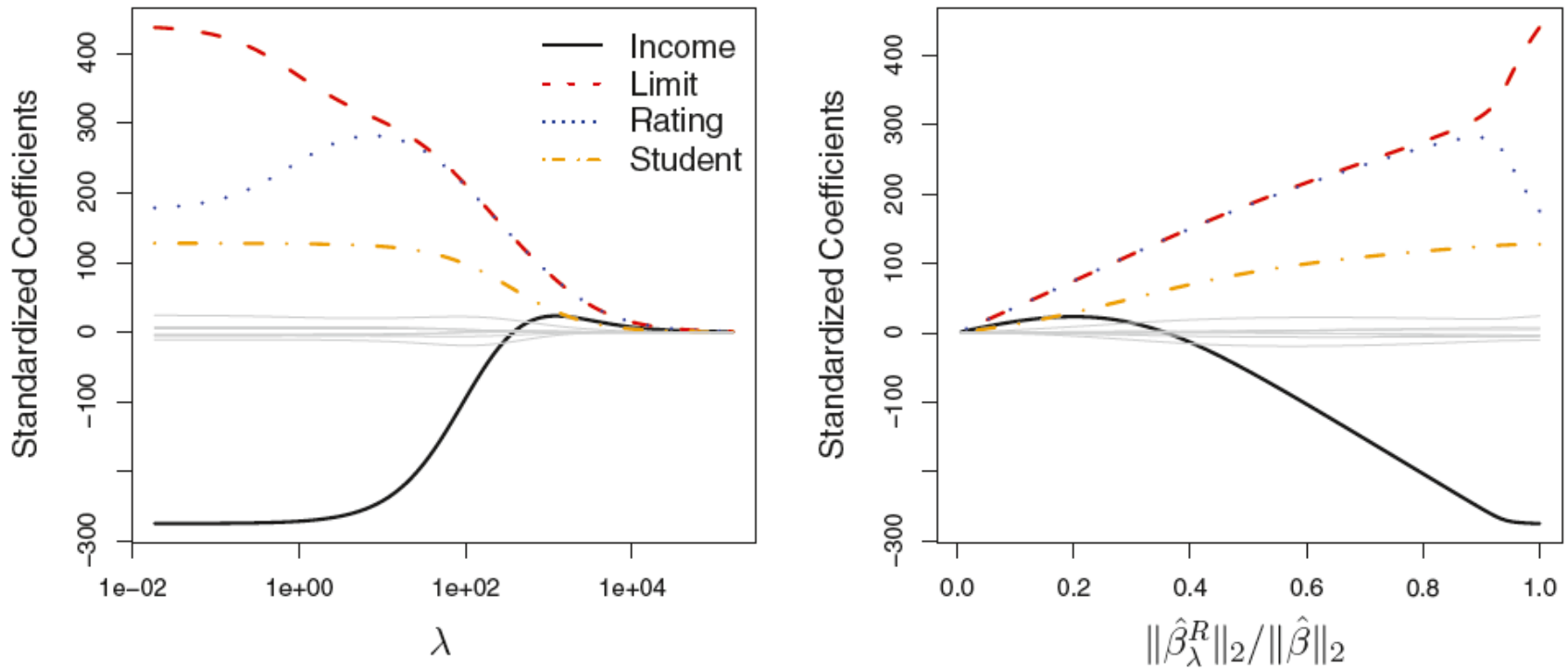


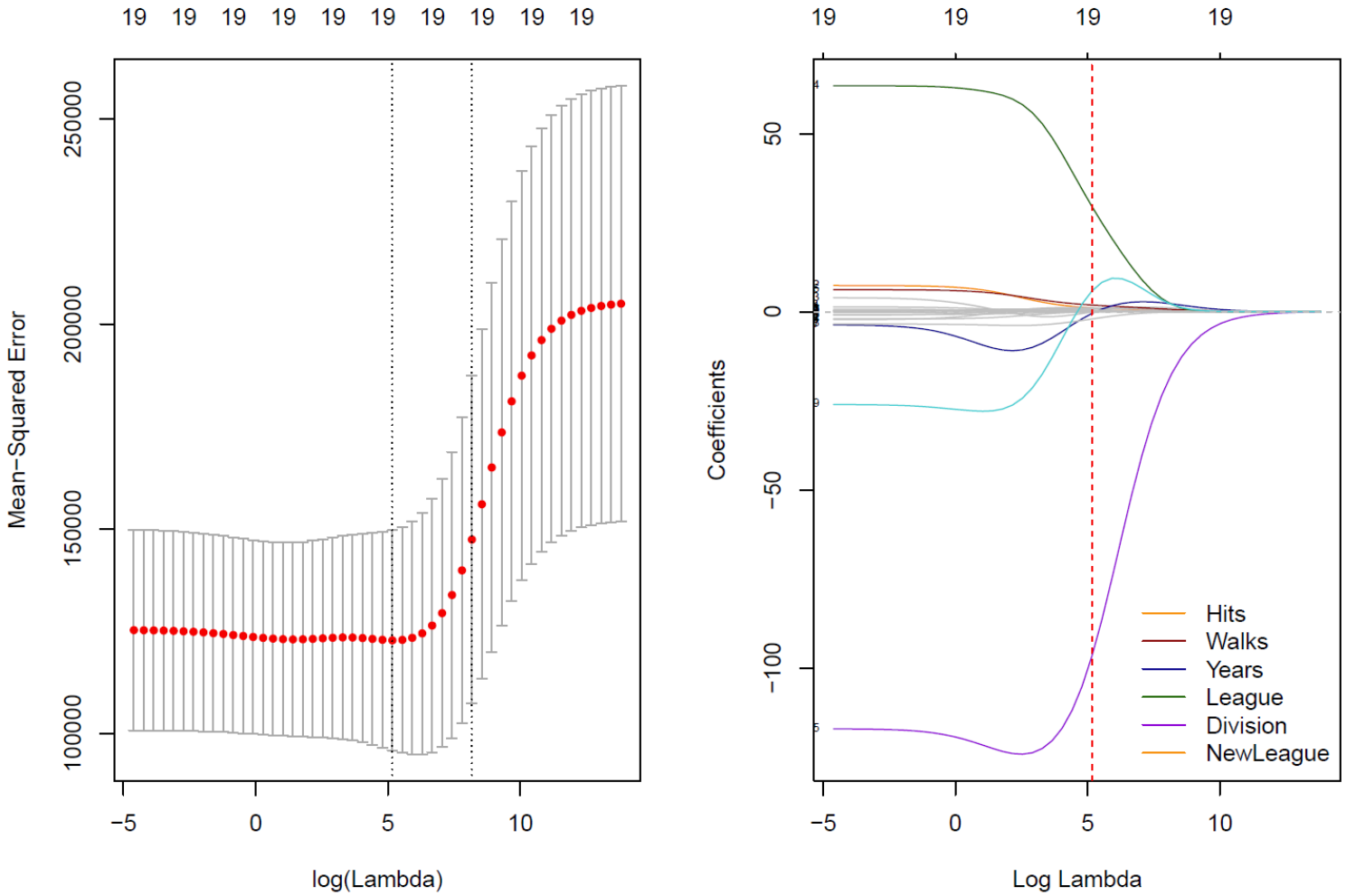
FIGURE 6.4. *The standardized ridge regression coefficients are displayed for the* `Credit` *data set, as a function of* λ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$.

Choosing λ by cross-validation

Cross-validation (CV) is a resampling approach used to estimate the test Mean Squared Error (MSE) of a model by repeatedly holding out a subset of the observations, and applying the chosen method to predict the held out outcome.

- **Leave-one-out**
- ***k*-fold**
- **Train-Test split**

Choosing λ: k-fold cross-validation



The bias-variance trade-off

The key to the improvement of Ridge regression over OLS is in the **bias-variance trade-off**: as λ increases, so does the bias, but the variance decreases by virtue of the lower flexibility

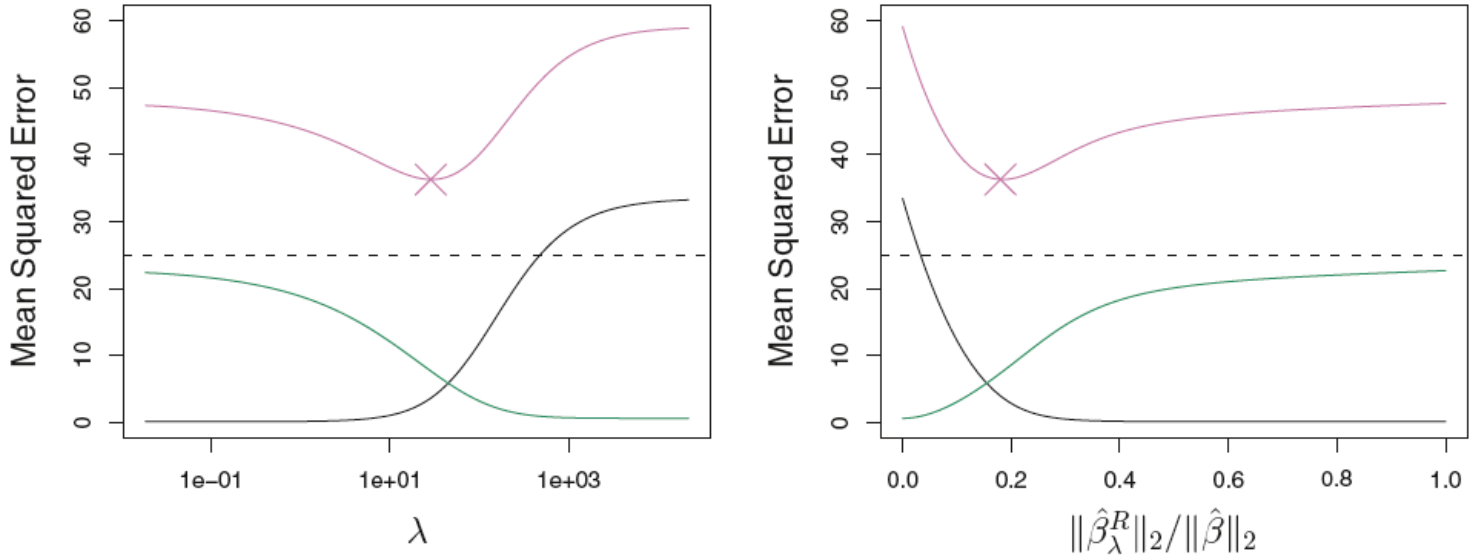
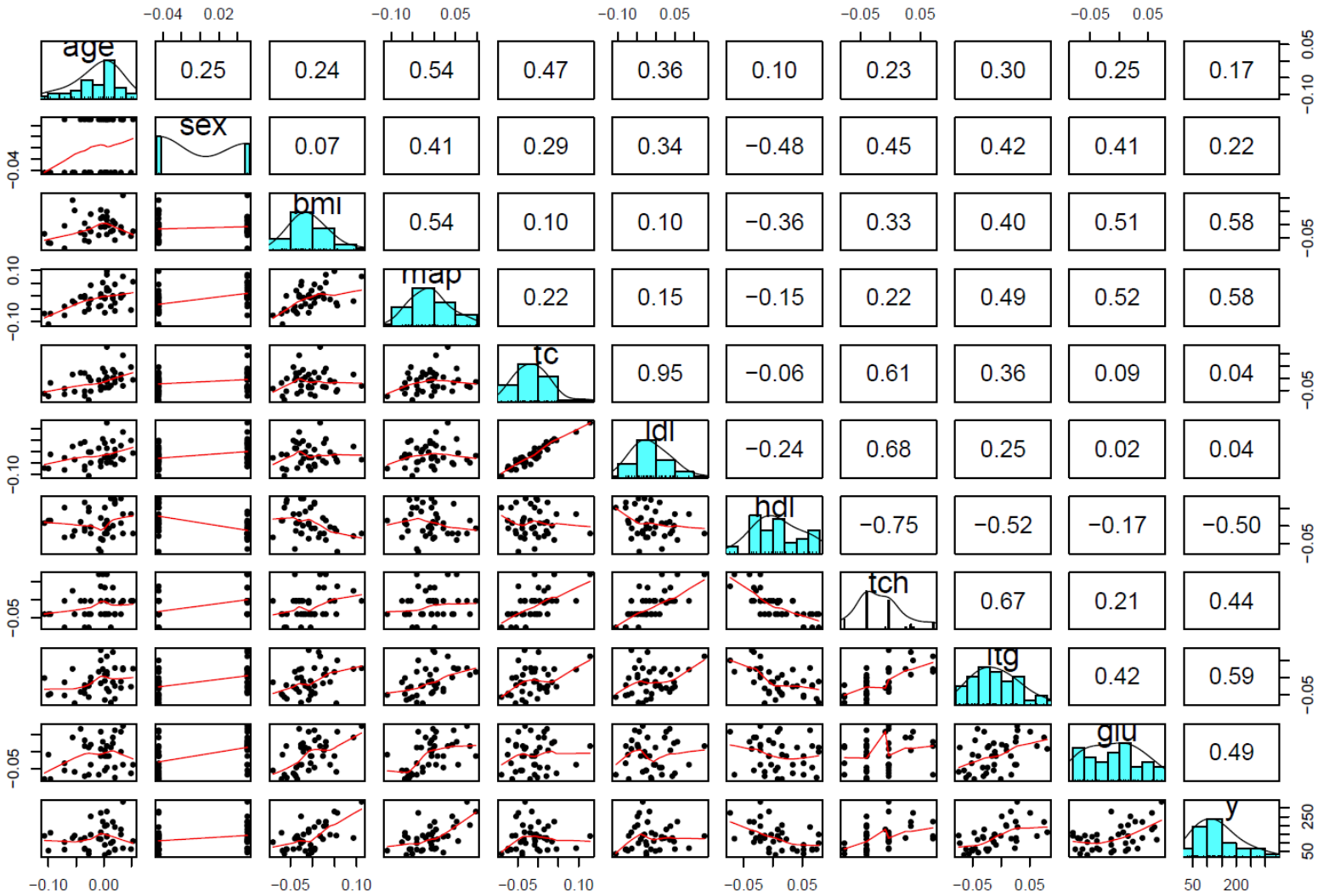
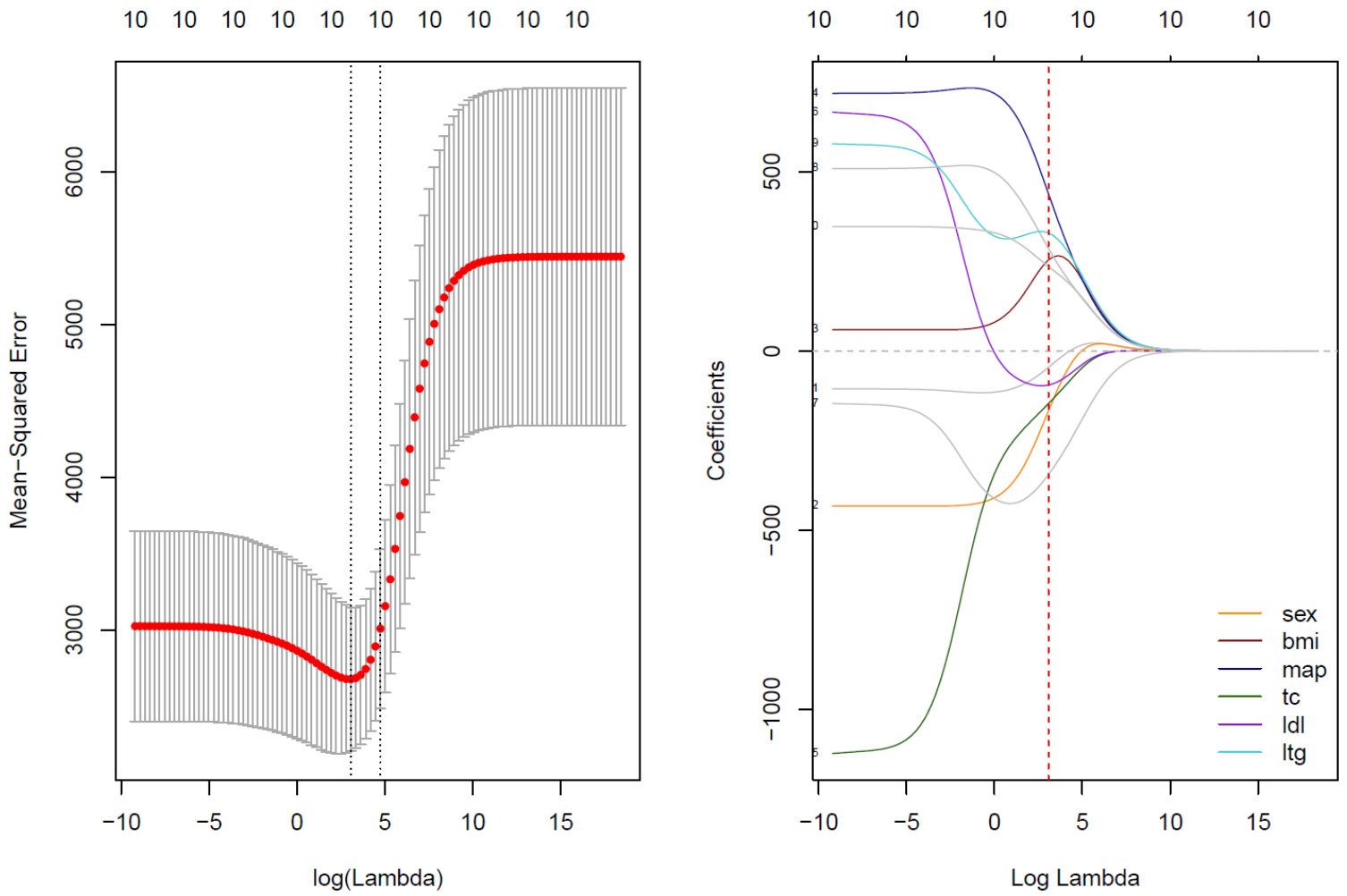


FIGURE 6.5. *Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of* λ *and* $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$. *The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.*

Example: diabetes progression $n = 40$, $p = 10$



Example: diabetes progression - Ridge



Example: diabetes progression - comparison

Table 1: Coefficient estimates

	OLS	Ridge
intercept	150.2964	147.6457
age	-103.1102	-46.0302
sex	-432.1300	-167.9041
bmi	60.8084	253.7881
map	714.0397	439.3918
tc	-1236.1539	-147.1544
ldl	764.5922	-95.2515
hdl	-106.8334	-345.4657
tch	505.2936	286.4428
ltg	617.6187	329.4664
glu	347.9019	235.6980

Ridge Summary

Ridge regression is a regularization method that can be helpful when:

- OLS coefficients may be poorly determined because of high correlation between regressors
- Extreme variability in the training data is observed, because of low sample size and/or large *p* relative to *n*.

Ridge improves over OLS by imposing a size constraint on the coefficient estimates:

- Typically has lower fit on training data than OLS, but is less prone to overfitting
- Usually generalizes better, because of higher robustness to extreme variability
- I involves hyperparameter tuning via cross-validation
- Readily extends to more general models, such as GLMs.

The LASSO Regression

"LASSO" stands for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator

The idea of constraining the size of the OLS estimates can be extended to consider different kinds of penalizations.

While the Ridge penalty encompasses the *L*2 norm of the estimates vector, the **LASSO** makes use of the *L*1 norm:

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|=RSS+\lambda\sum_{j=1}^{p}|\beta_j|$$

Variable Selection

The LASSO possesses an important property that Ridge doesn’t have: it allows for automatic **variable selection**

LASSO coefficients profiles

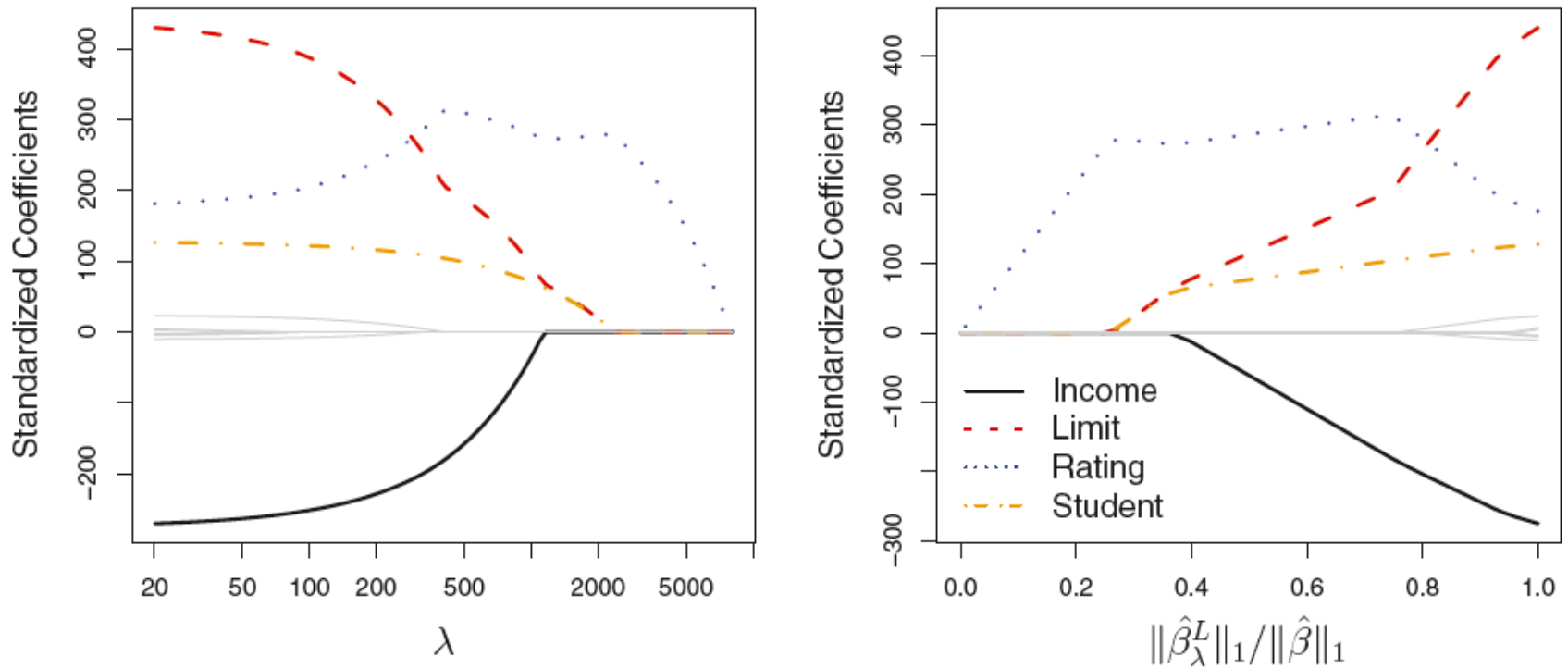
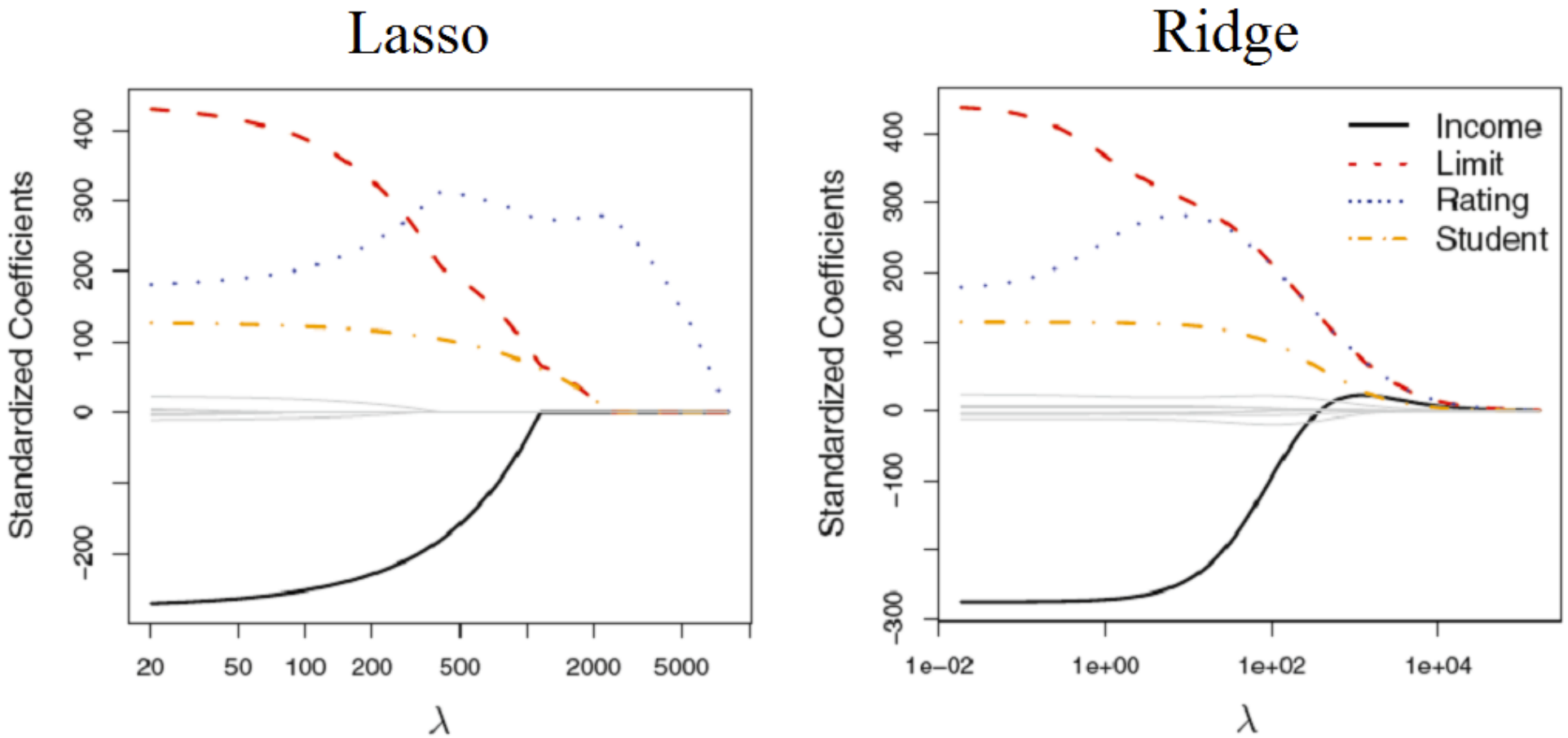


FIGURE 6.6. *The standardized lasso coefficients on the* `Credit` *data set are shown as a function of* λ *and* $\|\hat{\beta}_\lambda^L\|_1/\|\hat{\beta}\|_1$.

Comparing LASSO and Ridge



Comparing LASSO and Ridge: constraints

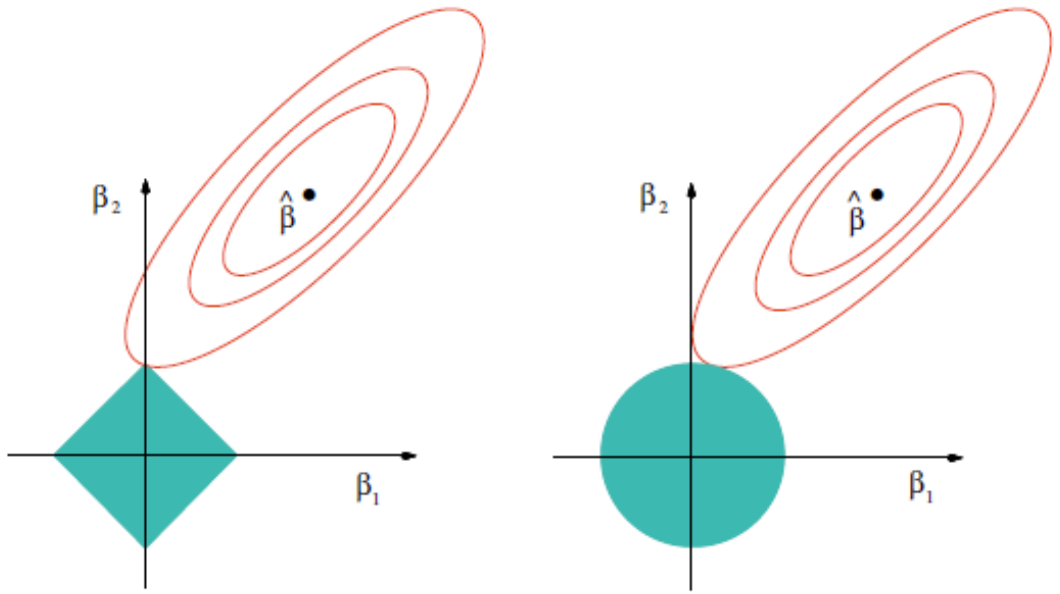
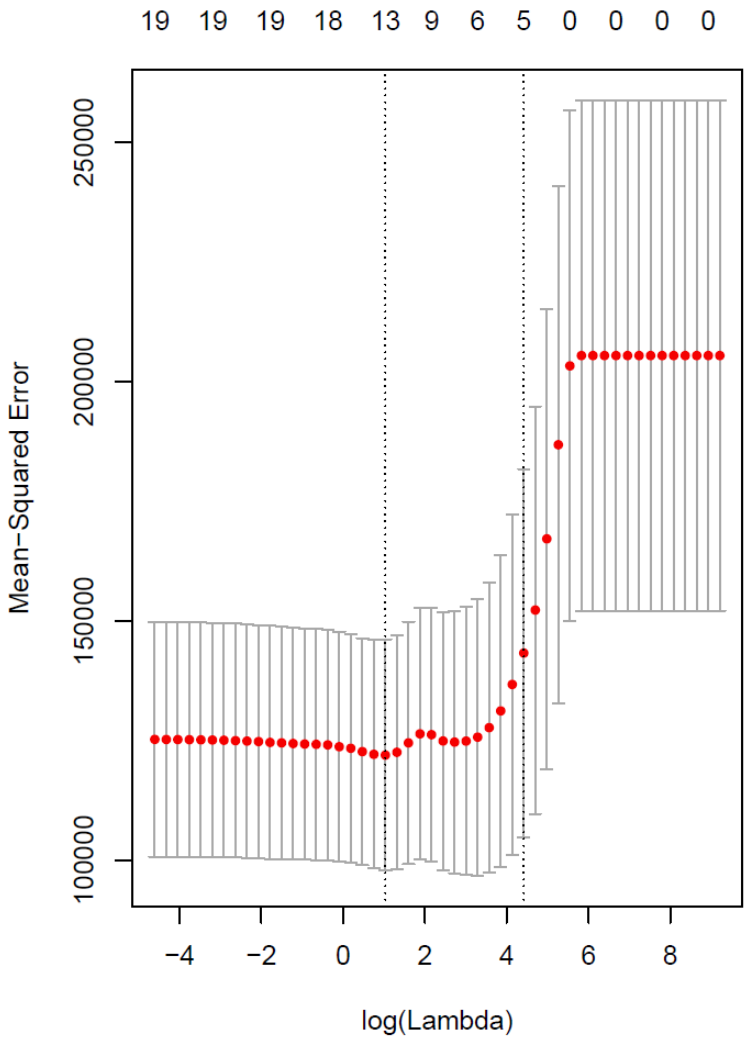
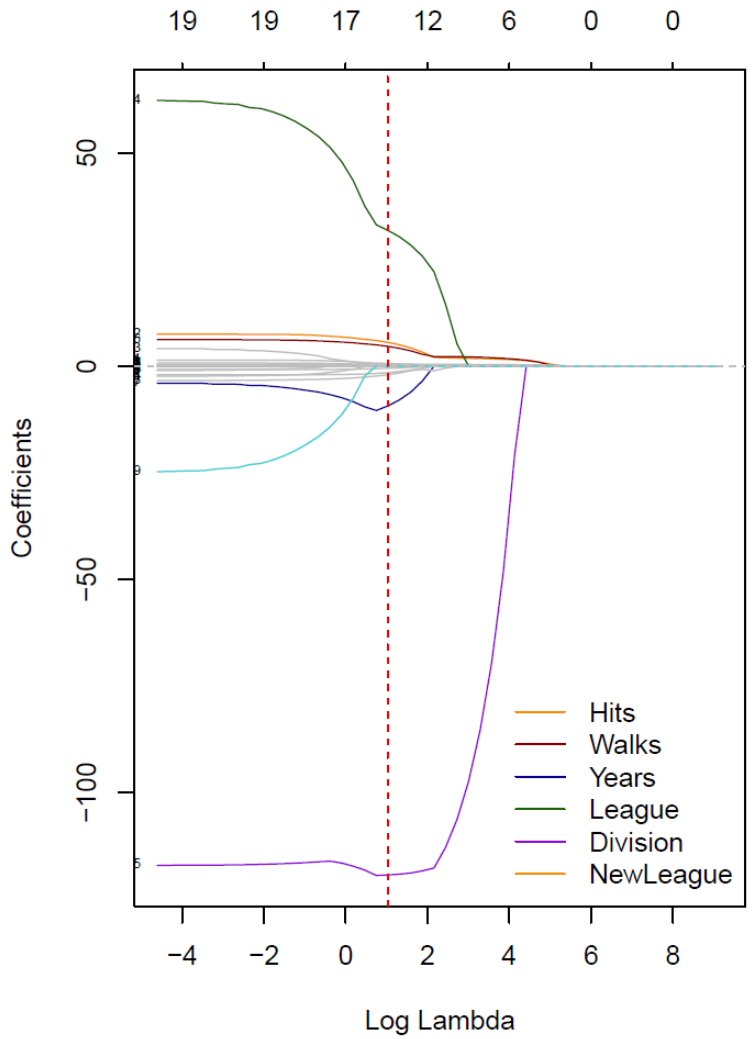


Figure 2.2 *Estimation picture for the lasso (left) and ridge regression (right). The solid blue areas are the constraint regions* $|\beta_1|+|\beta_2| \leq t$ *and* $\beta_1^2+\beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the residual-sum-of-squares function. The point* $\widehat{\beta}$ *depicts the usual (unconstrained) least-squares estimate.*

Choosing λ: cross-validation





LASSO Summary

LASSO regression is a regularization and variable selection method that can be especially helpful

- if variable selection is advisable to improve interpretability of the final model (sparsity)
- when faced with *wide* data, for which $p > n$
- for statistical and computational efficiency.

Many extensions already exist: check out the **group-LASSO** for dealing with dummy variables, and the **fused LASSO** for time series and functional data.

Difference between L1 and L2 regularization

L1 Regularization

- L1 penalizes sum of absolute value of weights.
- L1 has a sparse solution
- L1 has multiple solutions
- L1 has built in feature selection
- L1 is robust to outliers
- L1 generates model that are simple and interpretable but cannot learn complex patterns

L2 Regularization

- L2 regularization penalizes sum of square weights.
- L2 has a non sparse solution
- L2 has one solution
- L2 has no feature selection
- L2 is not robust to outliers
- L2 gives better prediction when output variable is a function of all input features
- L2 regularization is able to learn complex data patterns

More general penalties and the Elastic Nets

- The L*q* penalty

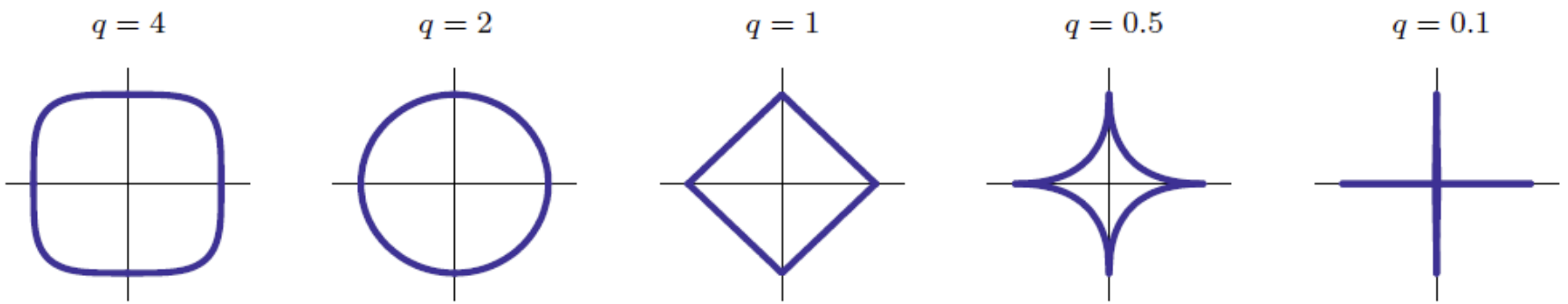


Figure 2.6 *Constraint regions* $\sum_{j=1}^{p} |\beta_j|^q \leq 1$ *for different values of* q. *For* $q < 1$, *the constraint region is nonconvex.*

q -> 0 we approach the so-called *subset selection* method

motivation for Elastic Nets

Consider the following scenarios:

- in the $p > n$ case, the LASSO can select at most n variables before it saturates
- if there is a group of variables with very high pairwise correlations, the LASSO tends to select only one variable from the group, not caring which one
- for usual $n > p$ situations, if there are high correlations between predictors, the prediction performance of LASSO is poor with respect to Ridge.
- In these situations, a more general approach is advised.

A hybrid penalty: the Elastic Nets

The LASSO sometimes does not perform well with highly correlated variables, and often performs worse than Ridge in prediction.

To overcome this limitations, a penalty that combines the $L1$ and $L2$ constraints has been developed.

An **elastic net** is a regularization and variable selection procedure that makes use of the penalty

$$\lambda \left[\frac{1}{2}(1-\alpha) \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right]$$

where $\alpha \in [0, 1]$ is called the **mixing** parameter and λ has the usual interpretation. LASSO and Ridge are special cases, respectively for $\alpha = 1$ and $\alpha = 0$.

The choice of α and λ

The mixing parameter α governs the extent to which the elastic net behaves as a Ridge or a LASSO. As $\alpha = 0$, the Ridge penalty gains more weight than the LASSO; the opposite happens when $\alpha = 1$.

In practice, one usually constructs a grid of α values. Chooses a folds configuration.

- k-fold cross-validates λ for given α
- stores the test MSE profile

The MSE profiles are then compared, and the α associated with the preferred one is chosen. The best λ within the selected profile is then used for modelling.

Joint cross-validation of α and λ is quite slow.

Elastic nets Summary

Elastic net regression is a regularization and variable selection procedure that overcomes some of the limitations of the LASSO by borrowing strength from the Ridge. Specifically, it

- allows to select more than *n* variables
- tends to jointly select or leave out groups of highly correlated variables
- improves the predictive performance w.r.t. LASSO
- is readily extendable to use with more general methods, such as GLM.

Elastic nets are especially useful when a sparse solution is either necessary or desirable (such as in $p >> n$ problems) and small groups of highly correlated predictors are present.