Classification 3: Regularization

Siva Balakrishnan
Data Mining: 36-462/36-662

January 24, 2018

Not following the book too closely today but Chapter 6 of ISL should be helpful

Recap: Logistic Regression Basics

- Logistic Regression is a *discriminative* classification method
- In the binary case, this means that we model the probability $\mathbb{P}(Y = 1|X = x)$ and use this to make classifications.
- The actual model is:

$$\mathbb{P}(Y = 1|X = x) =$$

Recap: Prediction and Decision Boundary

- We fit the model using our training data, and obtain estimates $\widehat{\beta}$.
- We predict probabilities using:
- We predict the class label using:
- The decision boundary, i.e. the boundary between points labeled 1 and 0 is:
- This decision boundary is:

Recap: Fitting the Model

- Given our training data we fit the model using:
- The likelihood function is given by:
- Unlike in linear regression we cannot simply use calculus to find the MLE. We find the maximizer using:

Recap: Linearly Separable Data

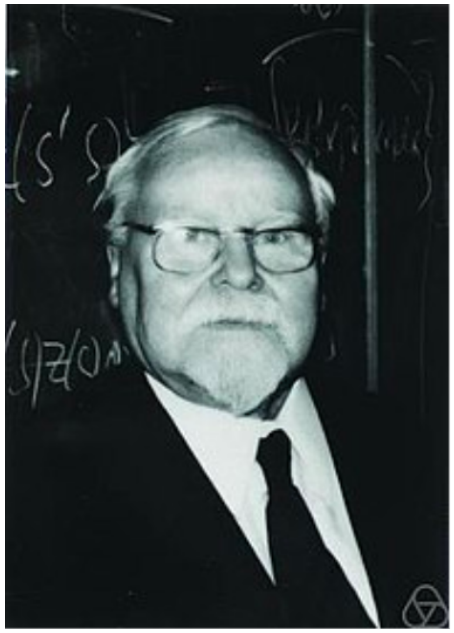
- Linearly separable data:
- When data is linearly separable, there are usually many solutions with ∞ likelihood (some better than others).
- Solving MLE will result in $\beta \to \infty$.
- Need to regularize!

Regularization Basics

- Regularization, roughly, is a set of techniques used to bias towards "lower complexity" estimates/predictions by trading-off model fit with model complexity.
- There are many different ways of trying to regularize an estimation problem and we will discuss the most important today.
- There are two primary reasons why we regularize:
  - **Improve Predictions/Estimates:**
  - **Improve Interpretability:**
  - We'll focus on the first one for a while.

Regularization History

- Regularization was introduced by a Russian geophysicist – Andrey Tikhonov.



- He was trying to solve regression problems where the solution was not unique, and found that adding regularizers increased the stability of the solution.

Regularization History

- Usual linear regression setup:

- When is the solution not unique?

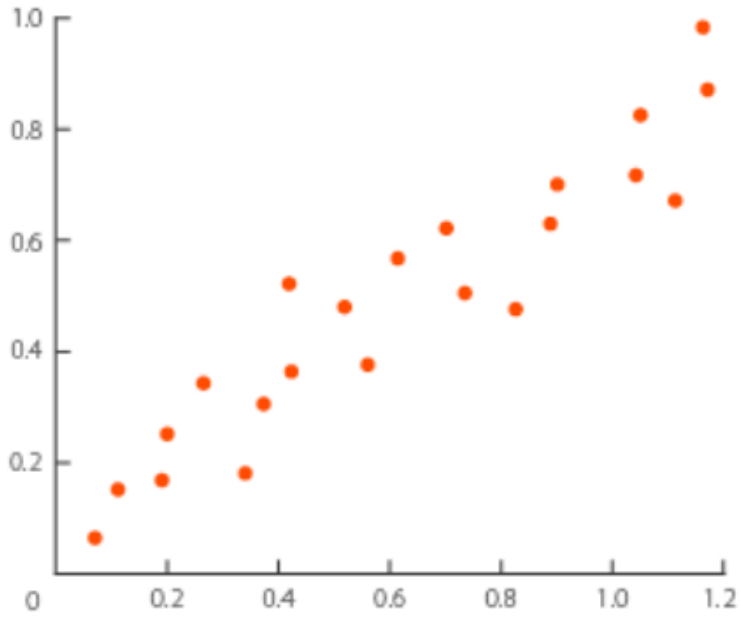
Regularization History

- Tikhonov found that solving:

  produced much more *stable* solutions (i.e. perturbing the data a little bit did not change the solution a lot).
- Solution is now always unique!
- Effectively, discovered the bias-variance tradeoff.

Non-linear Models, Over-fitting

- Suppose we just had one predictor, and observed this data:



- We could fit a line. How?
- We could fit a quadratic. How?
- ⋮

Train Error, Test Error, Model Complexity

- In typical cases, we expect to see a curve that looks like this:

Over-fitting Mathematically

- Difficult to precisely define over-fitting, but roughly, we say that we have over-fit if:
  - We choose some predictor f^*, and there is another predictor $\widetilde{f}$ such that on the training data:

and

Regularization and Over-fitting

- The basic problem is that of over-fitting. Regularization is basically a collection of different methods to try to reduce over-fitting.
- Usually models that are too complex (think high-degree polynomials in regression, or models with many features/parameters, or non-smooth estimates, or . . .) do not generalize well. Roughly, they always look good on the training data if it is not too large.
- Maybe we should prefer simpler models if they perform reasonably well since they are likely to generalize better?

Another thought experiment

- Suppose that we compared different models, as a function of the sample size. We might imagine we would see curves that looked like:

How do we regularize?

- Lots of different strategies and we'll look at the classics.
- One common idea is to penalize coefficients like Tikhonov did. Ridge regression is like least squares but shrinks the estimated coefficients towards zero. Given a response vector $y \in \mathbb{R}^n$ and a predictor matrix $X \in \mathbb{R}^{n \times p}$, the ridge regression coefficients are defined as

$$\begin{aligned}\widehat{\beta}^{\text{ridge}} &= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \\ &= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}\end{aligned}$$

- One important detail is that coefficient magnitudes only have a comparable meaning (across features) if the features are standardized (i.e. have mean 0, same length). Will return to this – but always standardize your features.

Ridge regression

$$\widehat{\beta}^{\text{ridge}} = \underset{\beta \in \mathbb{R}^p}{\text{argmin}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

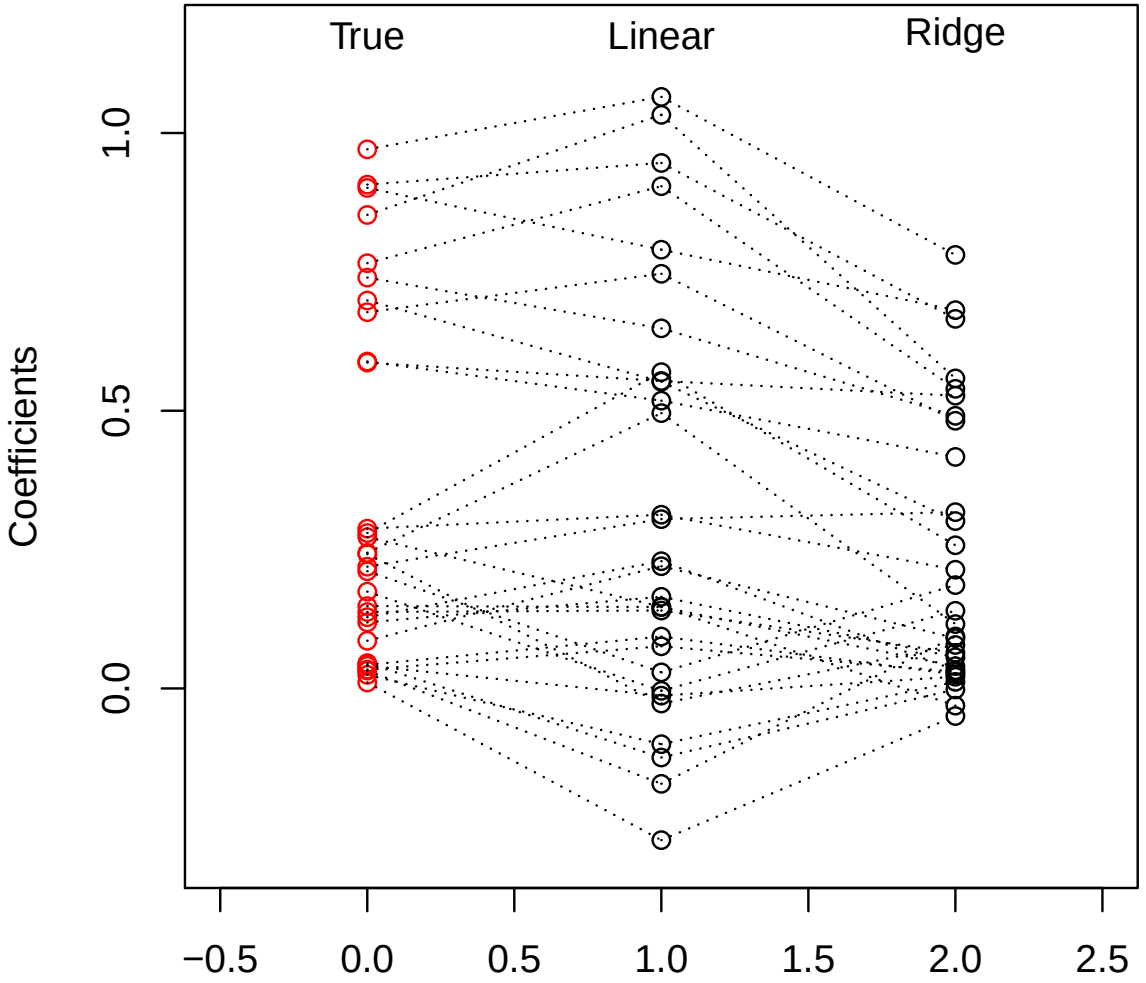
$$= \underset{\beta \in \mathbb{R}^p}{\text{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

Here $\lambda \geq 0$ is a tuning parameter, which controls the strength of the penalty term. Note that:

- When $\lambda = 0$, we get the linear regression estimate
- When $\lambda = \infty$, we get $\widehat{\beta}^{\text{ridge}} = 0$
- For λ in between, we are balancing two ideas: fitting a linear model of y on X, and shrinking the coefficients

Example: visual representation of ridge coefficients

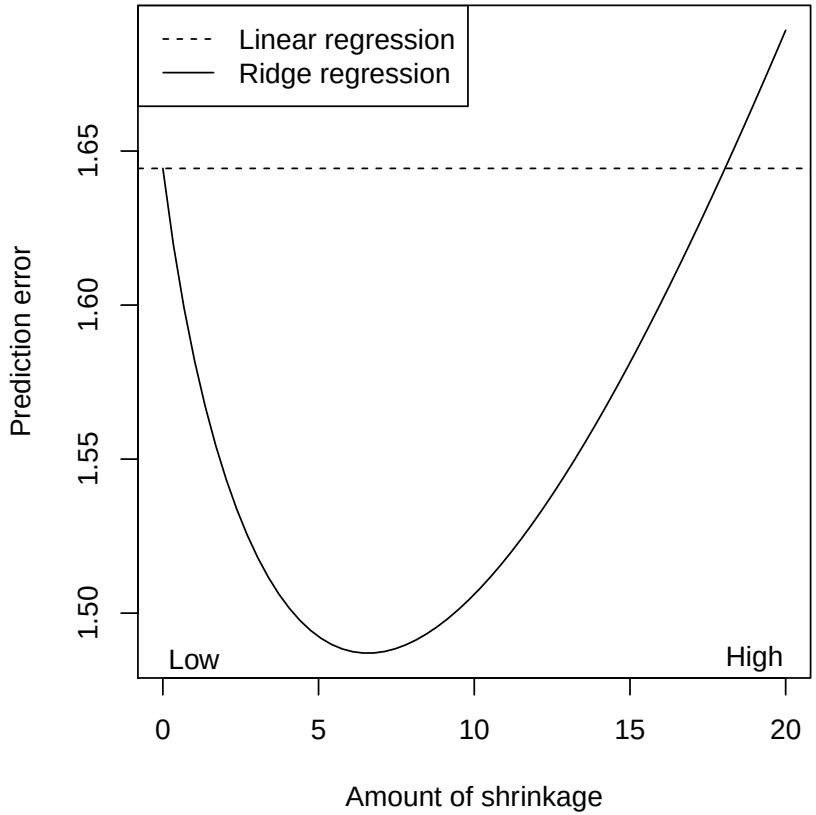
A visual representation of the ridge regression coefficients for the same example ($n = 50$, $p = 30$, and $\sigma^2 = 1$; 10 large true coefficients, 20 small) at $\lambda = 25$:



Does it work?

Recall in regression we can always write:

$$\text{prediction error} = \text{unavoidable error} + \text{bias} + \text{variance}$$



Linear regression:

Squared bias ≈ 0.006

Variance ≈ 0.627

Pred. error $\approx 1 + 0.006 + 0.627$

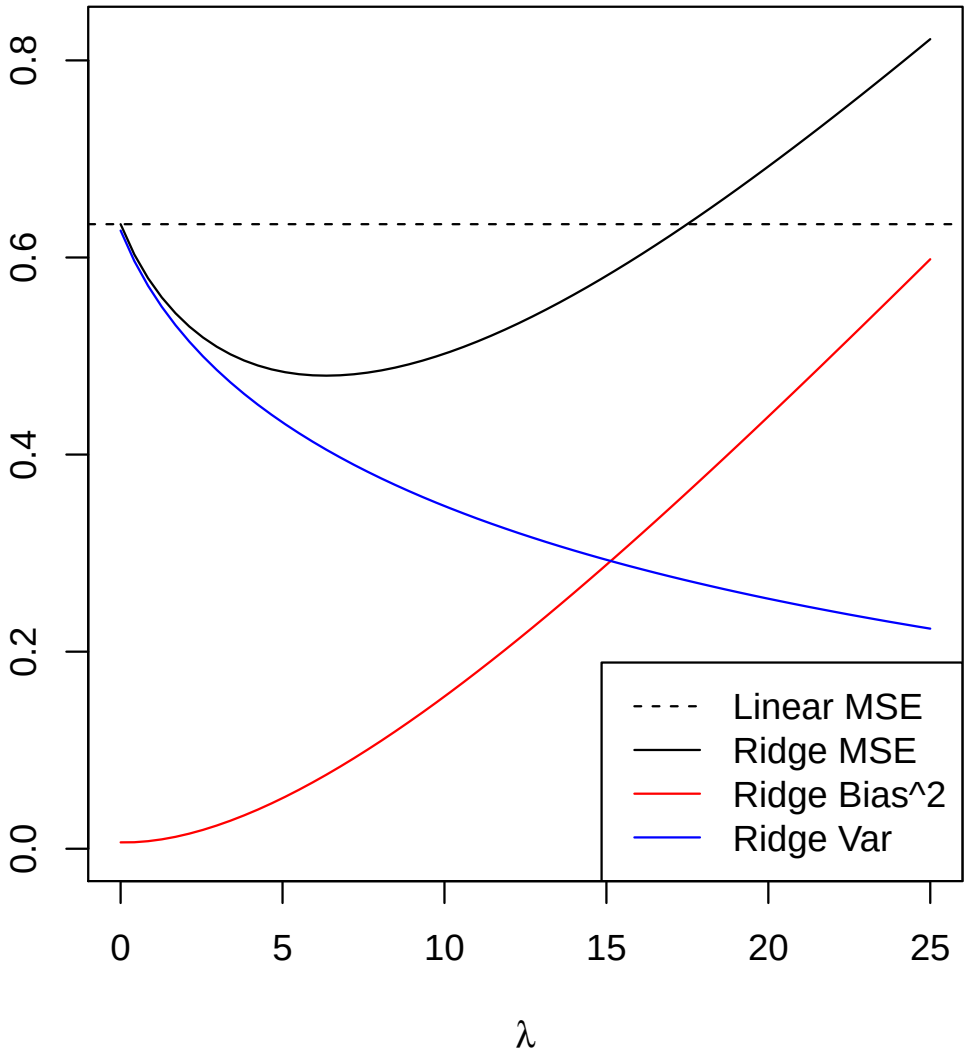
Ridge regression, at its best:

Squared bias ≈ 0.077

Variance ≈ 0.403

Pred. error $\approx 1 + 0.077 + 0.403$

Mean squared error for our last example



Notice that this looks exactly like a model complexity versus test error curve.

Other Benefits of Regularization

Some regularizers can also enhance model interpretability.

1. Suppose we are trying to predict if a patient is likely to develop prostate cancer or not. We measure 1 billion features for each patient (demographics, their DNA sequence, lifestyle factors etc.) We collect data on 10000 patients.
2. Likely to over-fit if we don't regularize properly (too many features). We try a ridge penalty on a logistic model. Might fix over-fitting but still will produce a model that is difficult to interpret – a *dense* linear combination of our 1 billion features.
3. Maybe we would like to just use predictors with 10 (or a 100) features. These models are easier to interpret.
4. How do we find the 10 (or 100) best features?

Variable selection

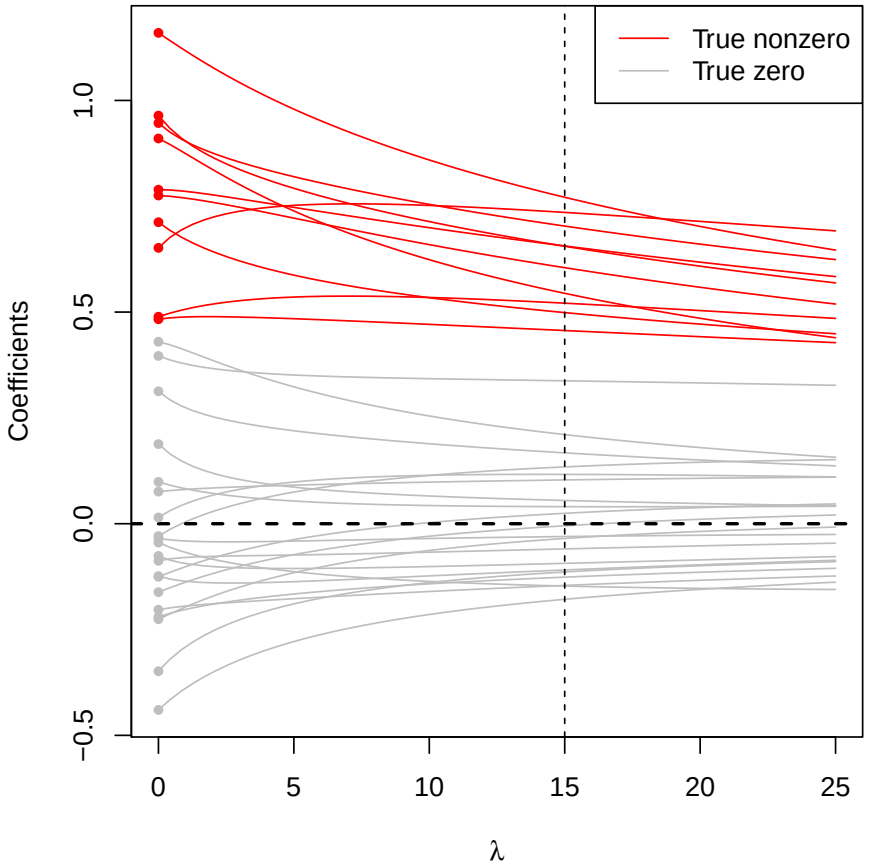
Out of many variables in our data set, only a few of them may really be useful. The rest might have zero or small coefficients.

The problem of picking out the relevant variables from a larger set is called variable selection. In the linear model setting, selecting a variable is equivalent to giving it a non-zero coefficient.

Sparse linear models (those with many zero coefficients) can be useful for model interpretability.

How does ridge regression perform if a group of the true coefficients was exactly zero?

Remember that as we vary λ we get different ridge regression coefficients, the larger the λ the more shrunken. Here we plot them again λ



The red paths correspond to the true nonzero coefficients; the gray paths correspond to true zeros. The vertical dashed line at $\lambda = 15$ marks the point above which ridge regression's MSE starts losing to that of linear regression

An important thing to notice is that the gray coefficient paths are not exactly zero; they are shrunken, but still nonzero

The Lasso

Ridge regression gave better predictions than least squares, but remained uninterpretable.

When p is large, we would like to carry out variable selection at the same time. We do this with the lasso.

The lasso will shrink the estimate, $\hat{\beta}$, while also carrying out automatic variable selection. As a result, it gives improved predictions *and* interpretable (sparse) models!

The lasso

The lasso[1] estimate is defined as

$$\begin{aligned}\widehat{\beta}^{\text{lasso}} &= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j| \\ &= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \underbrace{\|y - X\beta|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{Penalty}}\end{aligned}$$

The squared ℓ_2 penalty $\|\beta\|_2^2$ of ridge regression, has been replaced by an ℓ_1 penalty $\|\beta\|_1$. Even though these problems look similar, their solutions behave very differently

Note the name "lasso" is actually an acronym for: Least Absolute Selection and Shrinkage Operator

[1] Tibshirani (1996), "Regression Shrinkage and Selection via the Lasso"

The Lasso

$$\widehat{\beta}^{\text{lasso}} = \underset{\beta \in \mathbb{R}^p}{\text{argmin}} \ \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

The tuning parameter λ controls the strength of the penalty, and (like ridge regression):

- When $\lambda = 0$, we get:

- When $\lambda \to \infty$, we get:

For λ in between these two extremes, we are balancing two ideas: fitting a linear model of y on X, and shrinking the coefficients.

The Lasso

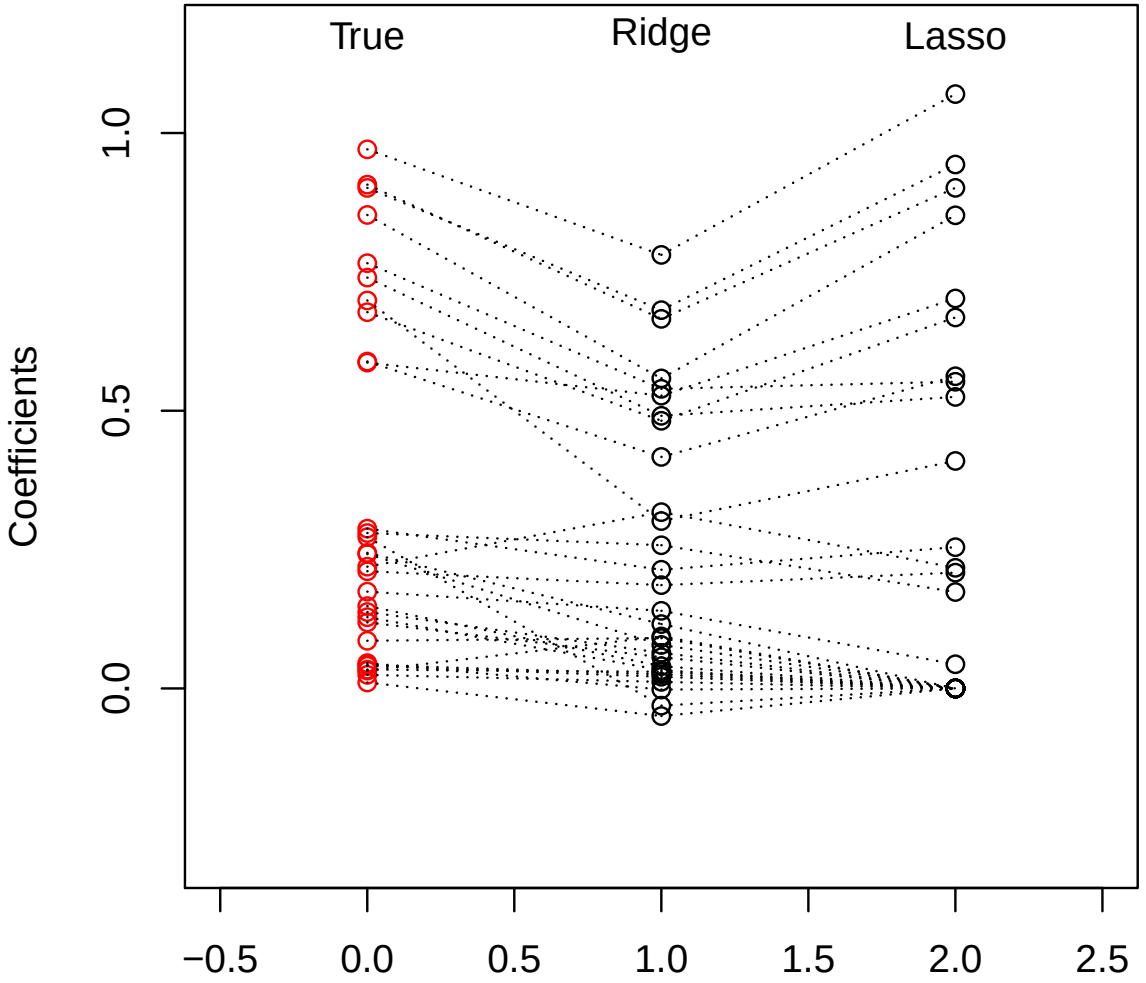
For λ in between these two extremes, we are balancing two ideas: fitting a linear model of y on X, and shrinking the coefficients.

The nature of the ℓ_1 penalty causes some coefficients to be shrunken to zero exactly at these intermediate λ values. This performs variable selection, unlike ridge regression!

As λ increases, more coefficients are set to zero, and among the nonzero coefficients, more shrinkage is employed

Example: visual representation of lasso coefficients

Our running example from last time: $n = 50$, $p = 30$, $\sigma^2 = 1$, 10 large true coefficients, 20 small. Here is a visual representation of lasso vs. ridge coefficients (with the same degrees of freedom):



Advantages of sparsity

- Interpretability: We can understand what the model relies on for prediction (understanding $\hat{f}$)
- We might gain some insight into the underlying data (though not causally) (helping to understand f)
- If we're building a predictive score, we can measure fewer things in the future (simpler $\hat{f}$ to apply later)

Important details

When including an intercept term in the model, we usually leave this coefficient unpenalized, just as we do with ridge regression. Hence the lasso problem with intercept is

$$\widehat{\beta}_0, \widehat{\beta}^{\text{lasso}} = \underset{\beta_0 \in \mathbb{R},\, \beta \in \mathbb{R}^p}{\operatorname{argmin}} \; \|y - \beta_0 \mathbf{1} - X\beta\|_2^2 + \lambda \|\beta\|_1$$

As we've seen before, if we center the columns of X, then the intercept estimate turns out to be $\widehat{\beta}_0 = \bar{y}$. Therefore we typically center y, X and don't include an intercept them

As with ridge regression, the penalty term $\|\beta\|_1 = \sum_{j=1}^p |\beta_j|$ is not fair is the predictor variables are not on the same scale. Hence, if we know that the variables are not on the same scale to begin with, we scale the columns of X (to have sample variance 1), and then we solve the lasso problem

Bias and variance of the lasso

Although we can't write down explicit formulas for the bias and variance of the lasso estimate (e.g., when the true model is linear), we know the general trend. Recall that

$$\widehat{\beta}^{\text{lasso}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

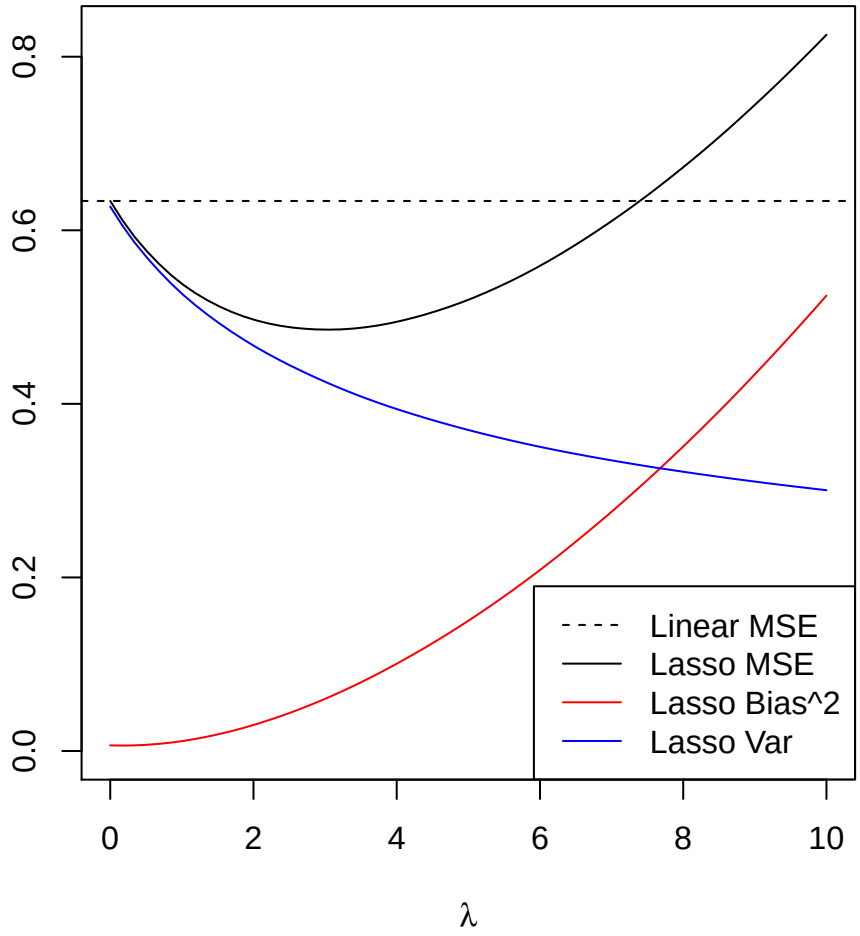
Generally speaking:

- The bias increases as λ (amount of shrinkage)
- The variance decreases as λ (amount of shrinkage)

What is the bias at $\lambda = 0$? The variance at $\lambda = \infty$?

Example: subset of small coefficients

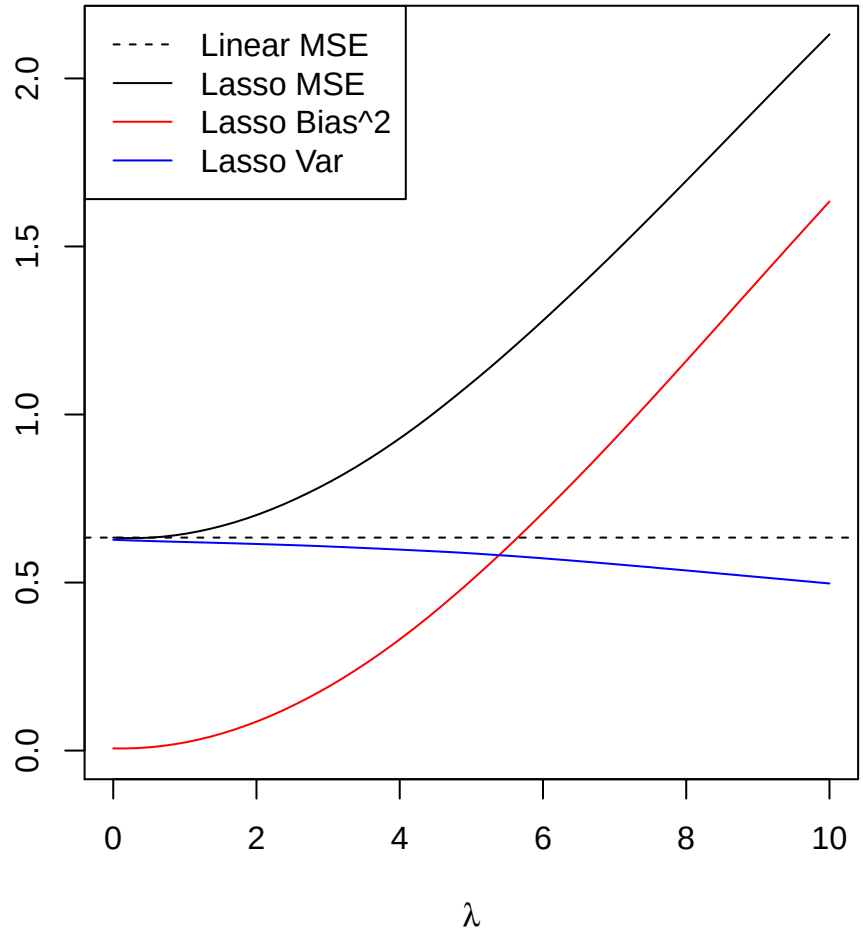
Example: $n = 50$, $p = 30$; true coefficients: 10 large, 20 small



The lasso can also be fit with `glmnet`.

Example: all moderate coefficients

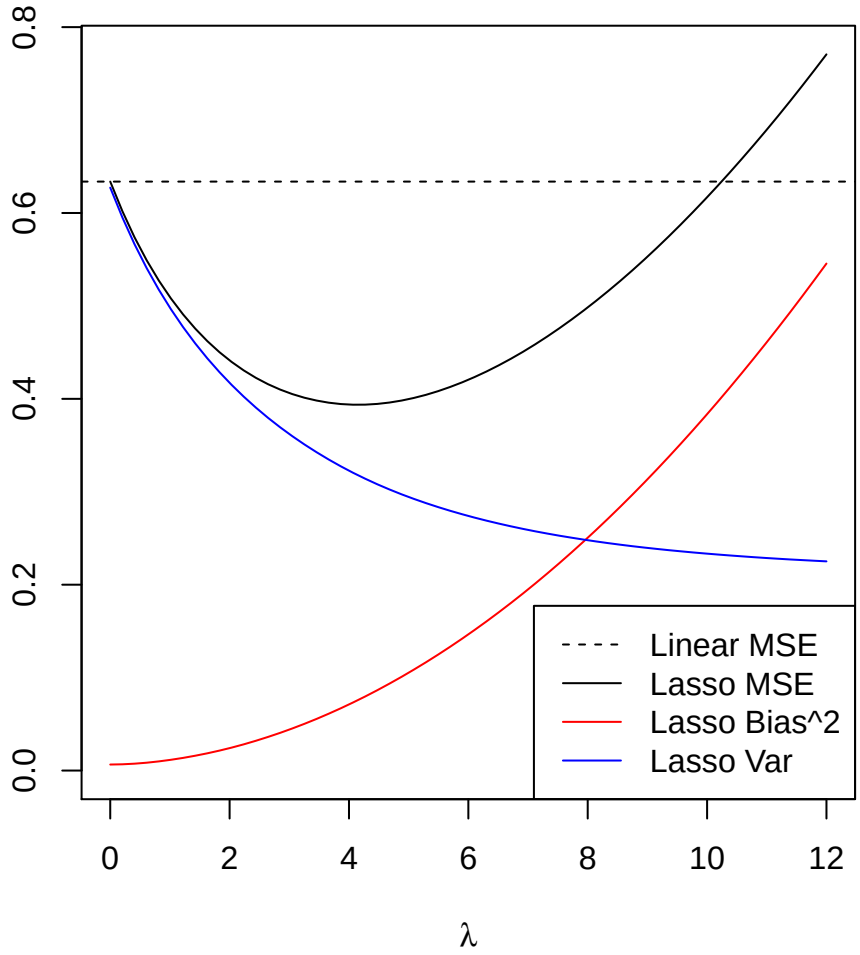
Example: $n = 50$, $p = 30$; true coefficients: 30 moderately large



Note that here, as opposed to ridge regression the variance doesn't decrease fast enough to make the lasso favorable for small λ

Example: subset of zero coefficients

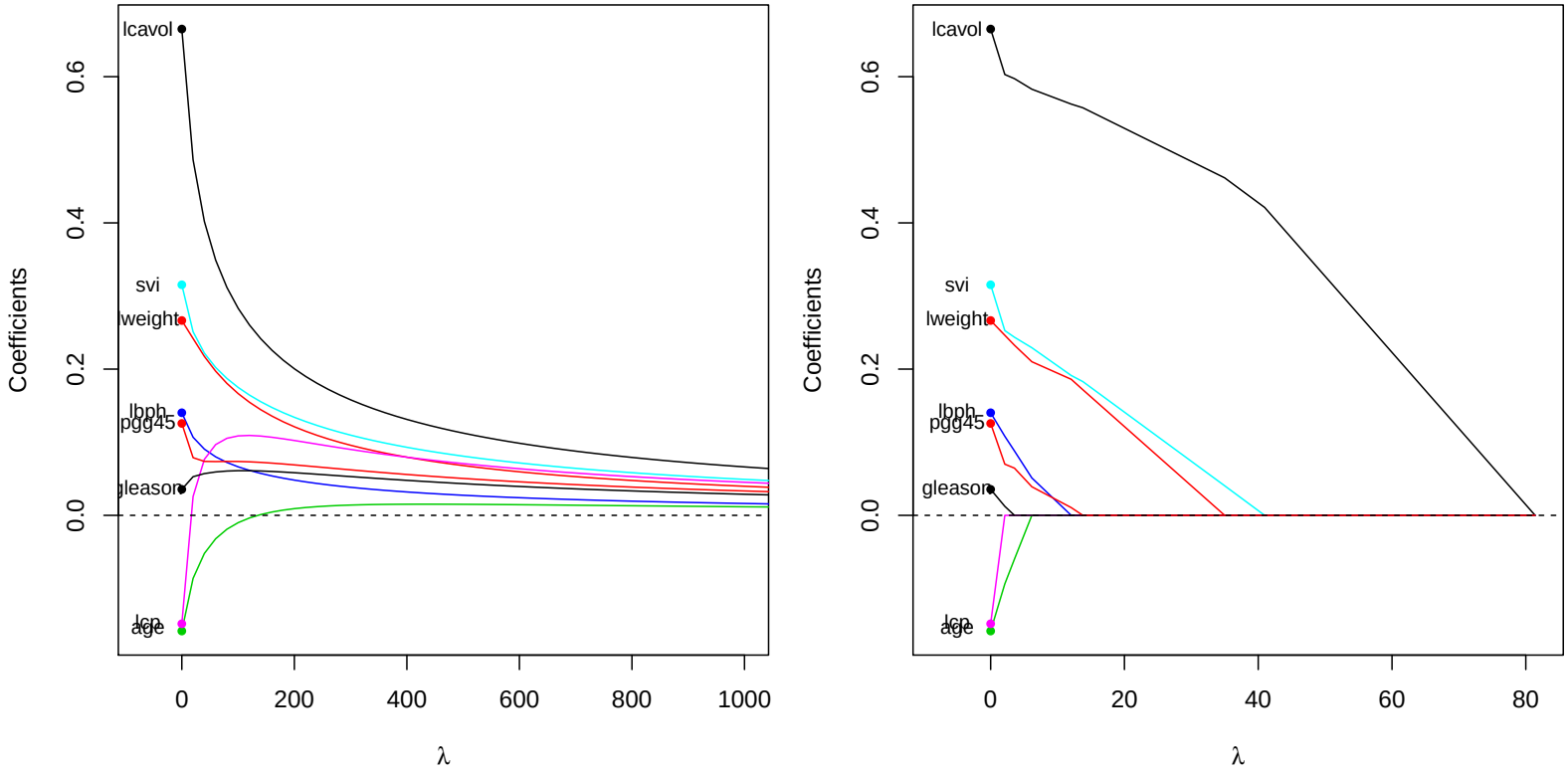
Example: $n = 50$, $p = 30$; true coefficients: 10 large, 20 zero



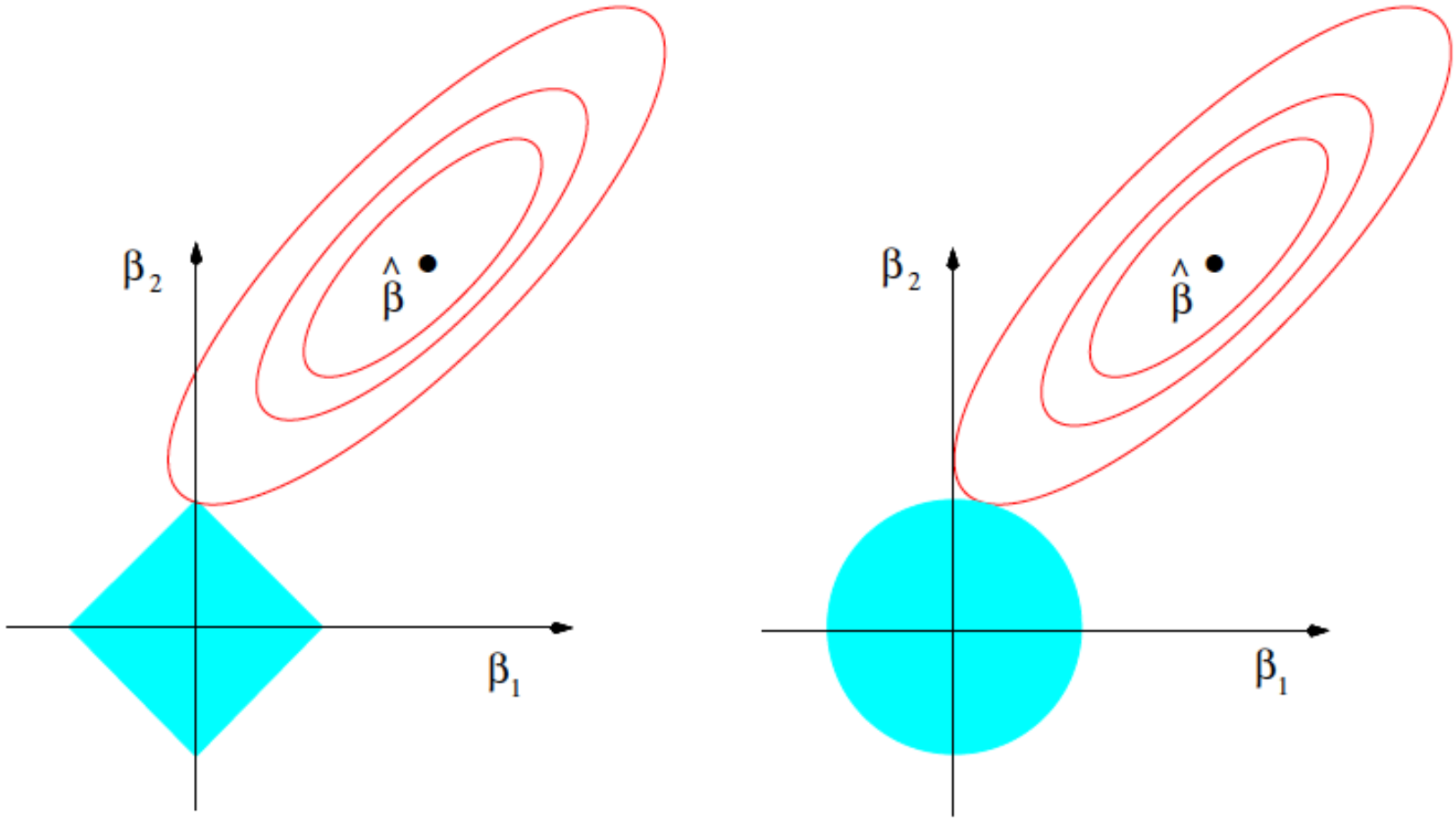
Advantage in interpretation

On top the fact that the lasso is competitive with ridge regression in terms of this prediction error, it has a big advantage with respect to interpretation. This is exactly because it sets coefficients exactly to zero, i.e., it performs variable selection in the linear model

For instance here is a picture from ESL – comparing LASSO and Ridge on a prostate cancer dataset.



Why does the lasso give zero coefficients?



(From page 71 of ESL)

Simple case: Orthogonal X

We would like to understand how the lasso and ridge regression work, as well as their differences.

Unfortunately, the lasso does not have a closed form solution. This makes it somewhat difficult to analyze.

However, in the very simple case where the columns of X are *orthogonal*, both the lasso and ridge regression have very simple forms. These forms give intuition about their behavior.

Simple case: Orthogonal X

Consider the very simple case where $X = I$. We can examine how both ridge regression and the lasso behave on this data.

This is easy, because the variables do not interact with one another, but it still gives an intuition for their more general behavior.

When $X = I$, note that the RSS becomes

$$\|y - X\beta\|_2^2 =$$

Simple case: Orthogonal X

The corresponding penalized forms then become

Ridge:

$$\hat{\beta}_{\text{ridge}} =$$

Lasso:

$$\hat{\beta}_{\text{lasso}} =$$

We can minimize separately for each i!

For ridge regression, differentiating yields

and therefore

$$\hat{\beta}_i =$$

Thus the estimates from ridge regression correspond to the least squares estimates reduced by a constant multiple.

Note that this can never give zero coefficients, and that it penalizes large coefficients quite a bit.

For the lasso, it turns out that the minimizer of

$$(y_i - \beta_i)^2 + \lambda|\beta_i|$$

is given by

$$\hat{\beta}_i =$$

Thus estimates from ridge regression correspond to shrinking the least squares estimates toward zero by an additive constant (without crossing zero).

This can give zero coefficients, and is also relatively harsher on small coefficients than larger ones. It appears to be better suited to sparse settings.

Example: visual representation of lasso coefficients

Our running example from last time: $n = 50$, $p = 30$, $\sigma^2 = 1$, 10 large true coefficients, 20 small. Here is a visual representation of lasso vs. ridge coefficients (with the same degrees of freedom):