Foundations of Data Science, Fall 2020

6. Regularisation, Validation

Prof. Dan Olteanu

DaST Data • (Systems+Theory)

University of Zurich[UZH]

Oct 13, 2020

https://lms.uzh.ch/url/RepositoryEntry/16830890400

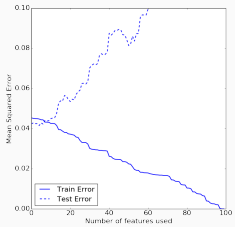
https://uzh.zoom.us/j/96690150974?pwd=cnZmMTduWUtCeWoxYW85Z3RMYnpTZz09

How Does Overfitting Occur?

Recall our previous experiment: Dataset $\mathbf{X}$ with $D = 100$ features and $N = 100$ data points

Every entry of $\mathbf{X}$ is drawn from $\mathcal{N}(0, 1)$

We let $y_i = x_{i,1} + \mathcal{N}(0, 0.2^2)$, i.e., the label y_i only depends on the first feature



Ridge Regression

Suppose we have data $\mathbf{X} \in \mathbb{R}^{N \times D}$, where $D \gg N$

One idea to avoid overfitting is to add a penalty term for weights

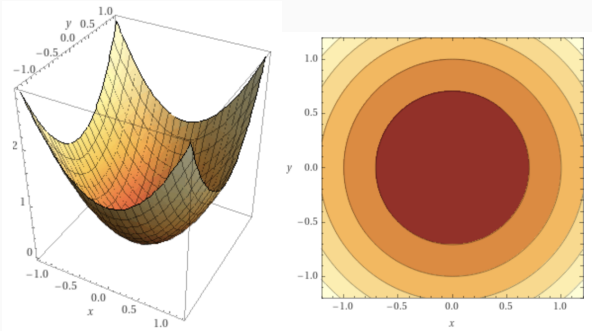
Least Squares Estimate Objective

$$\mathcal{L}(\mathbf{w}) = (\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y})$$

Ridge Regression Objective

$$\mathcal{L}_{\text{ridge}}(\mathbf{w}) = (\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda \sum_{i=1}^{D} w_i^2$$

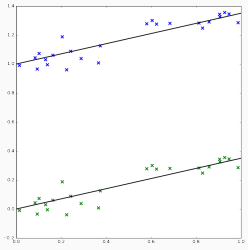
Plots of Square Function $x^2 + y^2$



No Penalty Needed for the Constant Term w_0

We add a penalty term for all weights but w_0 to control model complexity

The effect of varying w_0 is output translation and not on model complexity



Example of Scaling and Translation

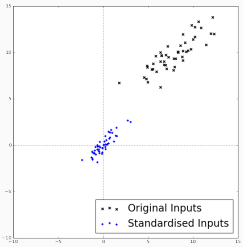
Should translating and scaling inputs contribute to model complexity?

- Suppose $\widehat{y} = w_0 + w_1 x$
- Supose x is temperature in $^\circ C$ and x' in $^\circ F$: $x = x'\frac{5}{9} - \frac{160}{9}$
- So $\widehat{y} = w_0 + w_1\left(x'\frac{5}{9} - \frac{160}{9}\right) = \left(w_0 - \frac{160}{9}w_1\right) + \frac{5}{9}w_1 x'$
- In one case "model complexity" is w_1^2, in the other it is $\frac{25}{81}w_1^2 < \frac{w_1^2}{3}$

Should try and avoid dependence on scaling and translation of variables $\Rightarrow$ Standardise the input

Standardising Input

- Feature standardisation: values drawn from normal distribution with mean 0 and variance 1
- Exercise Sheet 3: If we center the outputs, *i.e.,* their mean is 0, then w_0 becomes 0
- Now find $\mathbf{w}$ that minimises the modified objective function: $\mathcal{L}_{\text{ridge}}(\mathbf{w}) = (\mathbf{Xw} - \mathbf{y})^T(\mathbf{Xw} - \mathbf{y}) + \lambda \mathbf{w}^T\mathbf{w}$



Deriving Estimate for Ridge Regression

Suppose the data $\mathbf{X} \in \mathbb{R}^{N \times D}$ with inputs standardised and output centred

We want to derive the expression for $\mathbf{w}$ that minimises

$$\begin{aligned}\mathcal{L}_{\text{ridge}}(\mathbf{w}) &= (\mathbf{Xw} - \mathbf{y})^T(\mathbf{Xw} - \mathbf{y}) \quad + \lambda \mathbf{w}^T\mathbf{w} \\ &= \mathbf{w}^T\mathbf{X}^T\mathbf{Xw} - 2\mathbf{y}^T\mathbf{Xw} + \mathbf{y}^T\mathbf{y} + \lambda \mathbf{w}^T\mathbf{w}\end{aligned}$$

Let's take the gradient of the objective with respect to $\mathbf{w}$

$$\begin{aligned}\nabla_{\mathbf{w}}\mathcal{L}_{\text{ridge}} &= 2(\mathbf{X}^T\mathbf{X})\mathbf{w} - 2\mathbf{X}^T\mathbf{y} + 2\lambda\mathbf{w} \\ &= 2\left(\left(\mathbf{X}^T\mathbf{X} + \lambda \mathbf{I}_D\right)\mathbf{w} - \mathbf{X}^T\mathbf{y}\right)\end{aligned}$$

Set the gradient to $\mathbf{0}$ and solve for $\mathbf{w}$

$$\left(\mathbf{X}^T\mathbf{X} + \lambda \mathbf{I}_D\right)\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

$$\mathbf{w}_{\text{ridge}} = \left(\mathbf{X}^T\mathbf{X} + \lambda \mathbf{I}_D\right)^{-1}\mathbf{X}^T\mathbf{y}$$

Alternative Formulation of Ridge Regression

Ridge Regression: Least Squares Estimate + Penalty Term using ℓ_2 norm for the parameter vector $\mathbf{w}$

$$\mathcal{L}_{\text{ridge}}(\mathbf{w}) = (\mathbf{Xw} - \mathbf{y})^T(\mathbf{Xw} - \mathbf{y}) + \lambda \mathbf{w}^T\mathbf{w}$$

Alternative formulation as a constrained optimisation problem:

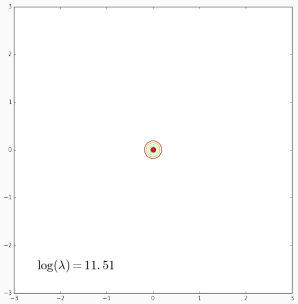
Minimise $(\mathbf{Xw} - \mathbf{y})^T(\mathbf{Xw} - \mathbf{y})$ subject to $\mathbf{w}^T\mathbf{w} \leq R$

The former is the Lagrangian formulation of the latter!

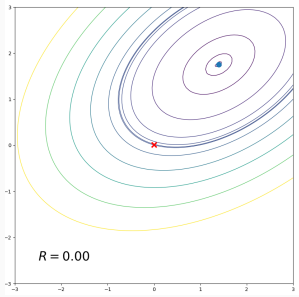
What is the relationship between λ and R?

Solution to Ridge Regression as Function of λ and R

Minimise $(\mathbf{Xw} - \mathbf{y})^T(\mathbf{Xw} - \mathbf{y}) + \lambda \mathbf{w}^T\mathbf{w}$

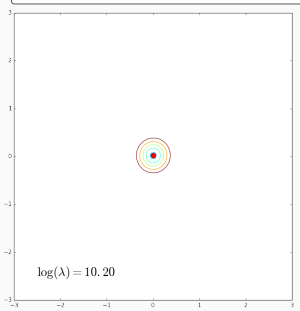


Minimise $(\mathbf{Xw} - \mathbf{y})^T(\mathbf{Xw} - \mathbf{y})$ s.t. $\mathbf{w}^T\mathbf{w} \leq R$

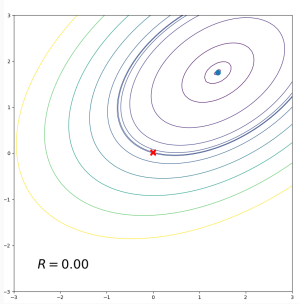


Solution to Ridge Regression as Function of λ and R

Minimise $(\mathbf{Xw} - \mathbf{y})^T(\mathbf{Xw} - \mathbf{y}) + \lambda \mathbf{w}^T\mathbf{w}$

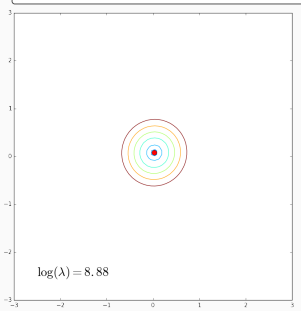


Minimise $(\mathbf{Xw} - \mathbf{y})^T(\mathbf{Xw} - \mathbf{y})$ s.t. $\mathbf{w}^T\mathbf{w} \leq R$

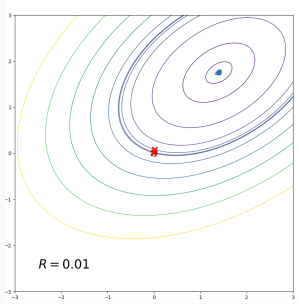


Solution to Ridge Regression as Function of λ and R

Minimise $(\mathbf{Xw} - \mathbf{y})^T(\mathbf{Xw} - \mathbf{y}) + \lambda \mathbf{w}^T\mathbf{w}$

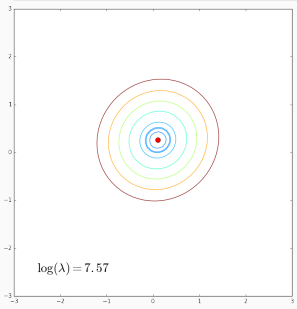


Minimise $(\mathbf{Xw} - \mathbf{y})^T(\mathbf{Xw} - \mathbf{y})$ s.t. $\mathbf{w}^T\mathbf{w} \leq R$

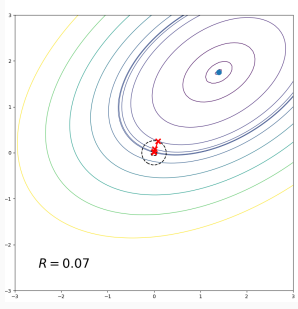


Solution to Ridge Regression as Function of λ and R

Minimise $(\mathbf{Xw}-\mathbf{y})^T(\mathbf{Xw}-\mathbf{y})+\lambda\mathbf{w}^T\mathbf{w}$

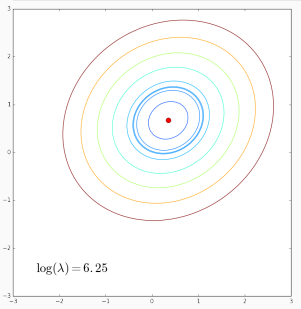


Minimise $(\mathbf{Xw}-\mathbf{y})^T(\mathbf{Xw}-\mathbf{y})$ s.t. $\mathbf{w}^T\mathbf{w} \leq R$

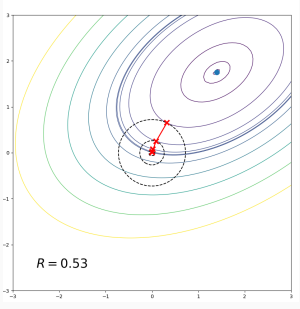


Solution to Ridge Regression as Function of λ and R

Minimise $(\mathbf{Xw}-\mathbf{y})^T(\mathbf{Xw}-\mathbf{y})+\lambda\mathbf{w}^T\mathbf{w}$

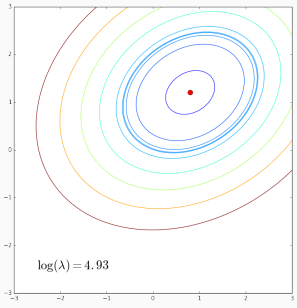


Minimise $(\mathbf{Xw}-\mathbf{y})^T(\mathbf{Xw}-\mathbf{y})$ s.t. $\mathbf{w}^T\mathbf{w} \leq R$

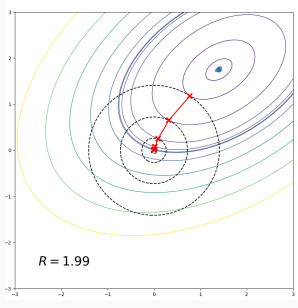


Solution to Ridge Regression as Function of λ and R

Minimise $(\mathbf{Xw}-\mathbf{y})^T(\mathbf{Xw}-\mathbf{y})+\lambda\mathbf{w}^T\mathbf{w}$

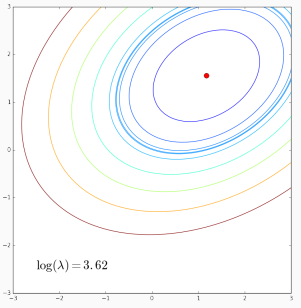


Minimise $(\mathbf{Xw}-\mathbf{y})^T(\mathbf{Xw}-\mathbf{y})$ s.t. $\mathbf{w}^T\mathbf{w} \leq R$

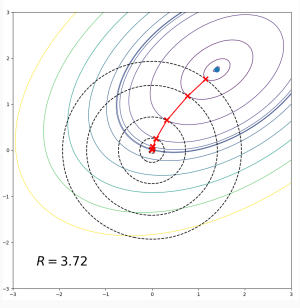


Solution to Ridge Regression as Function of λ and R

Minimise $(\mathbf{Xw}-\mathbf{y})^T(\mathbf{Xw}-\mathbf{y})+\lambda\mathbf{w}^T\mathbf{w}$

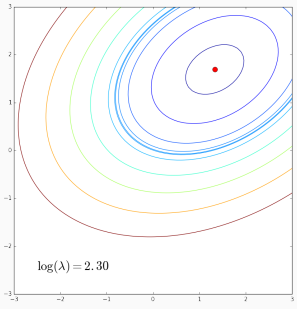


Minimise $(\mathbf{Xw}-\mathbf{y})^T(\mathbf{Xw}-\mathbf{y})$ s.t. $\mathbf{w}^T\mathbf{w} \leq R$

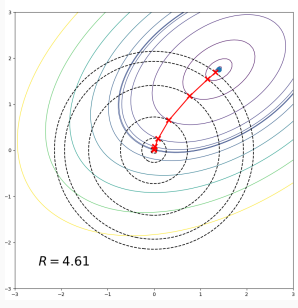


Solution to Ridge Regression as Function of λ and R

Minimise $(\mathbf{Xw}-\mathbf{y})^T(\mathbf{Xw}-\mathbf{y})+\lambda\mathbf{w}^T\mathbf{w}$



Minimise $(\mathbf{Xw}-\mathbf{y})^T(\mathbf{Xw}-\mathbf{y})$ s.t. $\mathbf{w}^T\mathbf{w} \leq R$

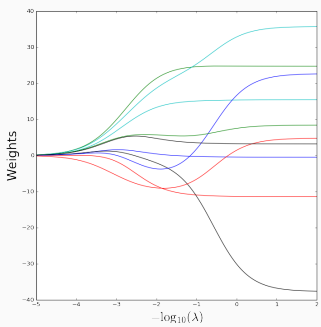


Ridge Regression: Effect of λ on Weights

λ decreasing

$\Rightarrow$

Magnitudes of weights start increasing



Summary: Ridge Regression

In ridge regression, in addition to the residual sum of squares we penalise the sum of squares of weights

Ridge Regression Objective

$$\mathcal{L}_{\text{ridge}}(\mathbf{w}) = (\mathbf{Xw} - \mathbf{y})^{\mathsf{T}} (\mathbf{Xw} - \mathbf{y}) + \lambda \mathbf{w}^{\mathsf{T}} \mathbf{w}$$

This is also called ℓ_2-regularization or weight-decay

Penalising weights "encourages fitting signal rather than just noise"

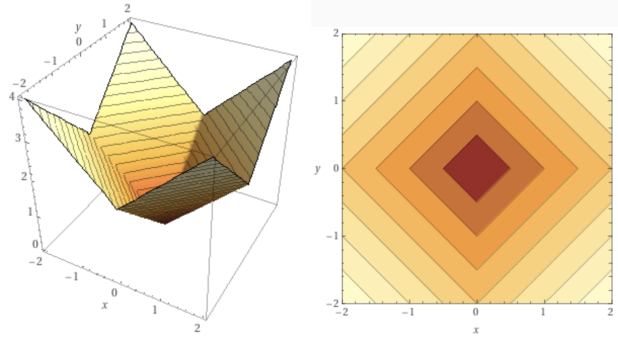
LASSO: Least Absolute Shrinkage and Selection Operator

Lasso Objective

$$\mathcal{L}_{\text{lasso}}(\mathbf{w}) = (\mathbf{Xw} - \mathbf{y})^{\mathsf{T}} (\mathbf{Xw} - \mathbf{y}) + \lambda \sum_{i=1}^{D} |w_i|$$

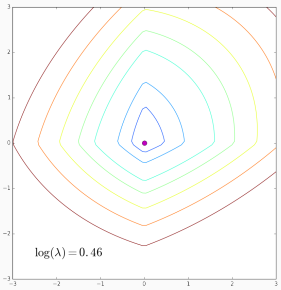
- As with ridge regression, there is a penalty on the weights
- The absolute value function does not allow for a simple closed-form expression (ℓ_1-regularization)
- However, there are advantages to using the lasso as we shall see next

Plots of Absolute Function $|x| + |y|$

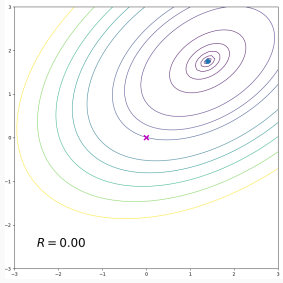


The Lasso : Optimisation

Minimise $(\mathbf{Xw} - \mathbf{y})^{\mathsf{T}} (\mathbf{Xw} - \mathbf{y}) + \lambda \sum_{i=1}^{D} |w_i|$

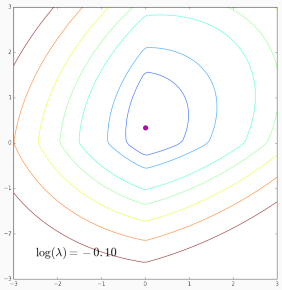


Minimise $(\mathbf{Xw} - \mathbf{y})^{\mathsf{T}} (\mathbf{Xw} - \mathbf{y})$ s.t. $\sum_{i=1}^{D} |w_i| \leq R$

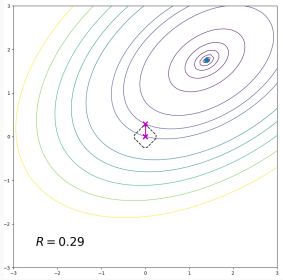


The Lasso : Optimisation

Minimise $(\mathbf{Xw} - \mathbf{y})^{\mathsf{T}} (\mathbf{Xw} - \mathbf{y}) + \lambda \sum_{i=1}^{D} |w_i|$

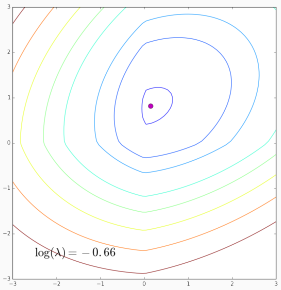


Minimise $(\mathbf{Xw} - \mathbf{y})^{\mathsf{T}} (\mathbf{Xw} - \mathbf{y})$ s.t. $\sum_{i=1}^{D} |w_i| \leq R$

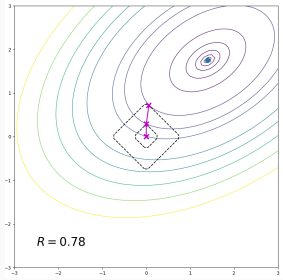


The Lasso : Optimisation

Minimise $(\mathbf{Xw} - \mathbf{y})^{\mathsf{T}} (\mathbf{Xw} - \mathbf{y}) + \lambda \sum_{i=1}^{D} |w_i|$

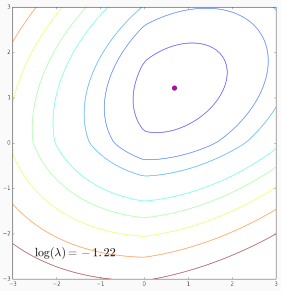


Minimise $(\mathbf{Xw} - \mathbf{y})^{\mathsf{T}} (\mathbf{Xw} - \mathbf{y})$ s.t. $\sum_{i=1}^{D} |w_i| \leq R$

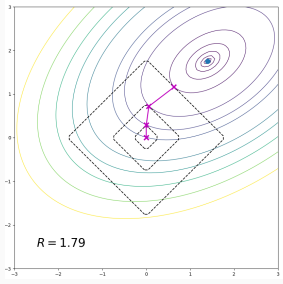


The Lasso : Optimisation

Minimise $(\mathbf{Xw}-\mathbf{y})^{\mathsf{T}}(\mathbf{Xw}-\mathbf{y})+\lambda\sum_{i=1}^{D}|w_i|$

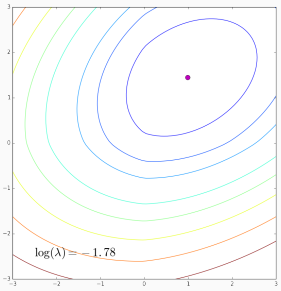


Minimise $(\mathbf{Xw}-\mathbf{y})^{\mathsf{T}}(\mathbf{Xw}-\mathbf{y})$ s.t. $\sum_{i=1}^{D}|w_i|\leq R$

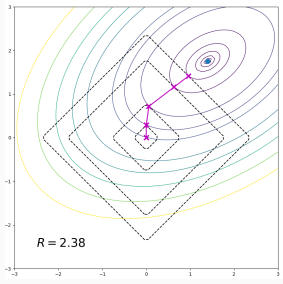


The Lasso : Optimisation

Minimise $(\mathbf{Xw}-\mathbf{y})^{\mathsf{T}}(\mathbf{Xw}-\mathbf{y})+\lambda\sum_{i=1}^{D}|w_i|$

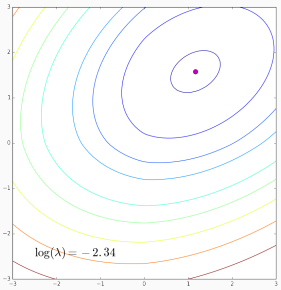


Minimise $(\mathbf{Xw}-\mathbf{y})^{\mathsf{T}}(\mathbf{Xw}-\mathbf{y})$ s.t. $\sum_{i=1}^{D}|w_i|\leq R$

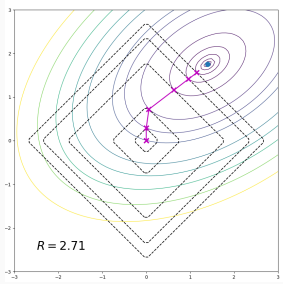


The Lasso : Optimisation

Minimise $(\mathbf{Xw}-\mathbf{y})^{\mathsf{T}}(\mathbf{Xw}-\mathbf{y})+\lambda\sum_{i=1}^{D}|w_i|$

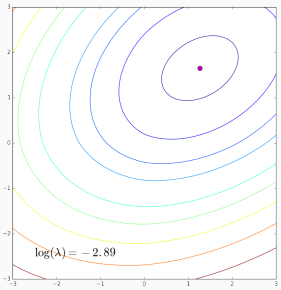


Minimise $(\mathbf{Xw}-\mathbf{y})^{\mathsf{T}}(\mathbf{Xw}-\mathbf{y})$ s.t. $\sum_{i=1}^{D}|w_i|\leq R$

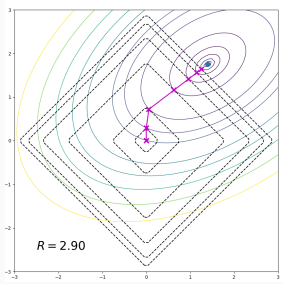


The Lasso : Optimisation

Minimise $(\mathbf{Xw}-\mathbf{y})^{\mathsf{T}}(\mathbf{Xw}-\mathbf{y})+\lambda\sum_{i=1}^{D}|w_i|$

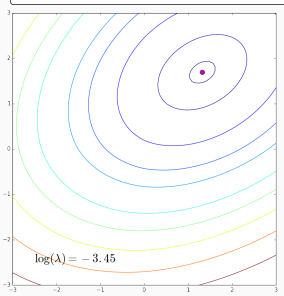


Minimise $(\mathbf{Xw}-\mathbf{y})^{\mathsf{T}}(\mathbf{Xw}-\mathbf{y})$ s.t. $\sum_{i=1}^{D}|w_i|\leq R$

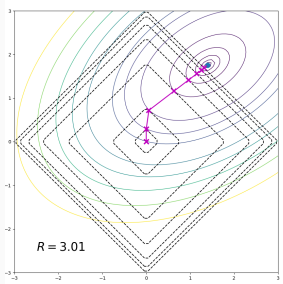


The Lasso : Optimisation

Minimise $(\mathbf{Xw}-\mathbf{y})^{\mathsf{T}}(\mathbf{Xw}-\mathbf{y})+\lambda\sum_{i=1}^{D}|w_i|$

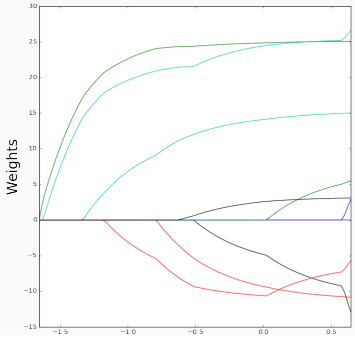


Minimise $(\mathbf{Xw}-\mathbf{y})^{\mathsf{T}}(\mathbf{Xw}-\mathbf{y})$ s.t. $\sum_{i=1}^{D}|w_i|\leq R$

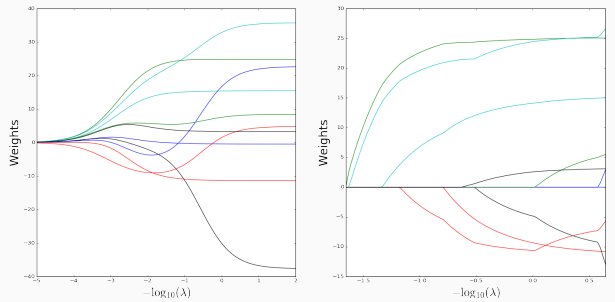


The Lasso Paths

λ decreasing
$\Rightarrow$
Magnitudes of weights start increasing



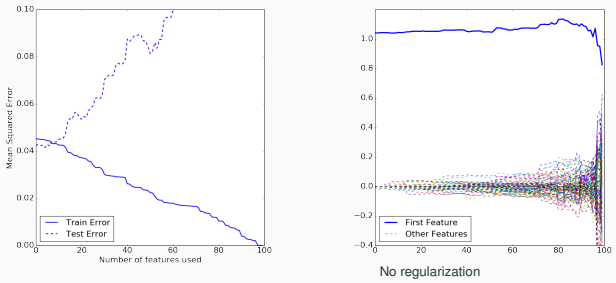
Comparing Ridge Regression and the Lasso



When using the Lasso, weights are often exactly 0. Thus, Lasso gives sparse models.

Back to the Overfitting Experiment

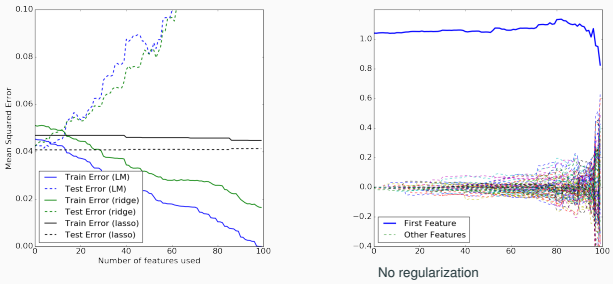
Dataset $\mathbf{X} \in \mathbb{R}^{100\times100}$, every data point drawn from $\mathcal{N}(0,1)$. We let $y_i = x_{i,1} + \mathcal{N}(0, 0.2^2)$



No regularization

Back to the Overfitting Experiment

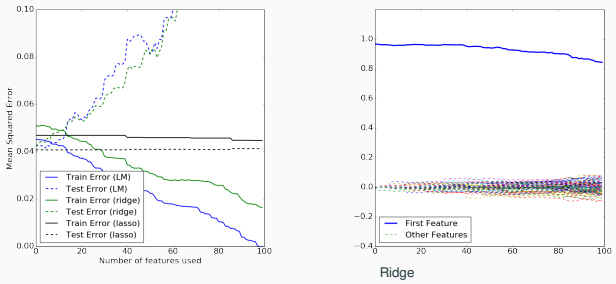
Dataset $\mathbf{X} \in \mathbb{R}^{100\times100}$, every data point drawn from $\mathcal{N}(0,1)$. We let $y_i = x_{i,1} + \mathcal{N}(0, 0.2^2)$



No regularization

Back to the Overfitting Experiment

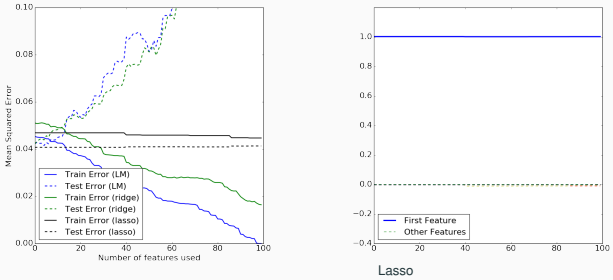
Dataset $\mathbf{X} \in \mathbb{R}^{100\times100}$, every data point drawn from $\mathcal{N}(0,1)$. We let $y_i = x_{i,1} + \mathcal{N}(0, 0.2^2)$



Ridge

Back to the Overfitting Experiment

Dataset $\mathbf{X} \in \mathbb{R}^{100\times100}$, every data point drawn from $\mathcal{N}(0,1)$. We let $y_i = x_{i,1} + \mathcal{N}(0, 0.2^2)$



Lasso

How to Choose Hyper-parameters?

- So far, we learned how to estimate the parameters $\mathbf{w}$
- For Ridge Regression or Lasso, we need to choose λ
- If we perform basis expansion
  - For kernels, we need to pick the width parameter γ
  - For polynomials, we need to pick degree d
- For more complex models there may be even more hyper-parameters

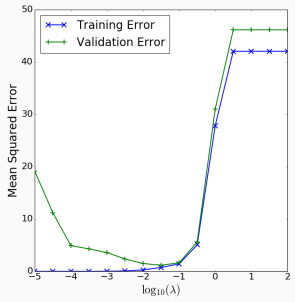
Validation Error

- Divide the data into parts: training, validation (and testing)
  - Typically, split the data as 80% for training, 20% for validation
- Train model using training set and evaluate on validation set
- Pick the value of λ that minimises the validation error

λ	training error	validation error
10^{-5}	0.000057	19.144985
10^{-4}	0.000181	4.914340
10^{-3}	0.011502	3.570725
10^{-2}	0.266629	1.483503
10^{-1}	1.432287	1.611345
10^{0}	27.823336	31.027190
10^{1}	42.058130	46.154464
10^{2}	42.058130	46.154464

Training and Validation Curves

- Plot training and validation error vs λ for Lasso
  - Dataset with 2000 points and 2000 features
- Validation error curve is U-shaped
  - Left: overfitting
    Low training error and high validation error
  - Right: underfitting
    High training error and high validation error
- Pick hyper-parameter λ at the bottom of the curve



What if We have Many Hyper-Parameters?

Main (trivial) approach: Grid search

- Assume a small domain D_i for each hyper-parameter λ_i ($1 \leq i \leq k$)
- Iterate over all possible combinations of hyper-parameter values $D_1 \times \cdots \times D_k$
- Perform cross-validation for each such combination
- Pick the combination with the lowest validation error

K-Fold Cross Validation

When data is scarce, instead of splitting as training and validation, divide data into K folds (parts):

- Use $K-1$ folds for training and 1 fold as validation
  - Commonly set $K = 5$ or $K = 10$
  - When $K =$ the number of datapoints, it is called LOOCV (Leave one out cross validation)
- Validation error for fixed hyper-parameter values: average over all runs

Overfitting on the Validation Set

Suppose we do all the right things

1. Train the model on the training set
2. Choose hyper-parameters using k-fold validation
3. Test on the test set (real world), and your error is unacceptably high!

What should we do now?

Kaggle Leaderboard Mechanism

Kaggle competitions use a leaderboard mechanism $\approx$ classic hold-out method

- Training set – publicly available
- Hold-out set – publicly available *without* the labels
  Split randomly into 30% hold-out labels and 70% test labels
- Valid submission = List of predicted labels, one for each point in the testing set
- Score function, e.g., misclassification rate, maps the submission to [0,1]
- Public ranking given by the score on the hold-out labels (without the test labels)
- Private ranking given by the score on the test labels

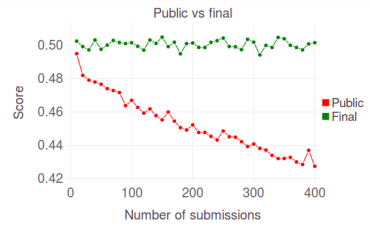
Teams submit repeatedly for public ranking for quite some time and before the final private ranking

Winning Kaggle without Reading the Data!

Suppose the task is to predict N binary labels, i.e., $\mathbf{y} \in \{0,1\}^N$

Algorithm Wacky Boosting:

1. Pick $\mathbf{y}_1, \ldots, \mathbf{y}_K \in \{0,1\}^N$ randomly
2. Select $\mathbf{Y} = \{\mathbf{y}_i \mid \text{score}(\mathbf{y}_i) < 0.5\}$
3. Output $\widehat{\mathbf{y}} = \text{majority}(\mathbf{Y})$



Source `http://blog.mrtz.org/2015/03/09/competition.html`

The Problem with Kaggle Leaderboard

There is statistical dependence between the hold-out data and the submission!

- Submissions may incorporate information about the hold-out labels released through the leaderboard
- Due to this feedback loop, the public score is no longer an unbiased estimate of the score
- Eventually, the submissions will overfit to the hold-out set

How to deal with it?

- Limiting rate of re-submissions
- Control bit precision numbers
- Winners determined on a separate test set

Example of disconnect between theory of static data analysis and practice of interactive data analysis