Machine Learning - MT 2017

6 Regularization, Validation, Model Selection

Christoph Haase

University of Oxford

October 20, 2017

Outline

Ridge Regression and Lasso

Model Selection

Ridge Regression

Suppose we have data $\langle(\mathbf{x}_i, y_i)\rangle_{i=1}^{N}$, where $\mathbf{x} \in \mathbb{R}^D$ with $D \gg N$

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

Least Squares Estimate Objective

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

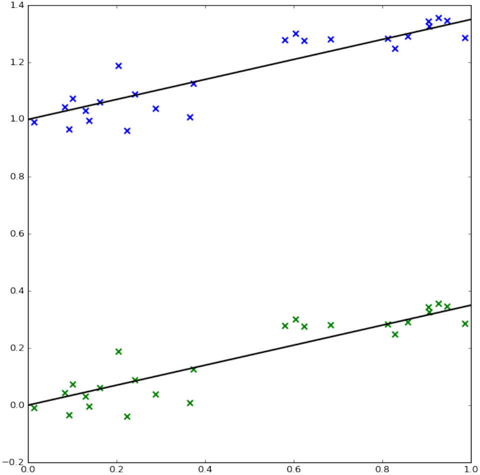
Ridge Regression Objective

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

Ridge Regression

We add a penalty term for weights to control model complexity

Should not penalise the constant term w_0 for being large



Ridge Regression

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 $^\circ F$

So $\widehat{y} = \left(w_0 - \frac{160}{9} w_1\right) + \frac{5}{9} w_1 x'$

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

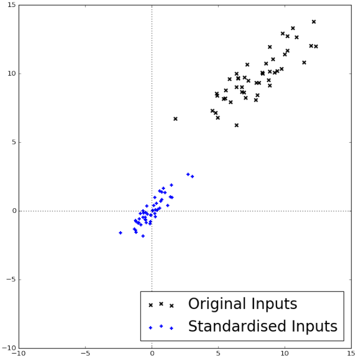
Ridge Regression

Before optimising the ridge objective, it's a good idea to standardise all inputs (mean 0 and variance 1)

If in addition, we center the outputs, *i.e.,* the outputs have mean 0, then the constant term is unnecessary (Exercise on Sheet 2)

Then find $\mathbf{w}$ that minimises the objective function

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



Deriving Estimate for Ridge Regression

Suppose the data $\langle(\mathbf{x}_i, y_i)\rangle_{i=1}^{N}$ with inputs standardised and output centered

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

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

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

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

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

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

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

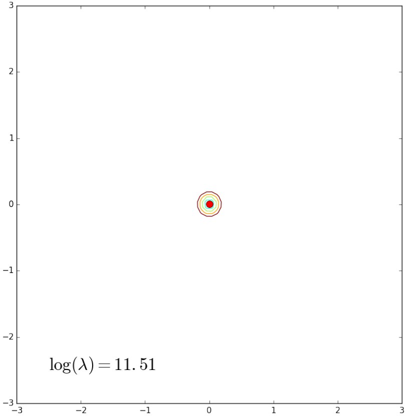
Ridge Regression

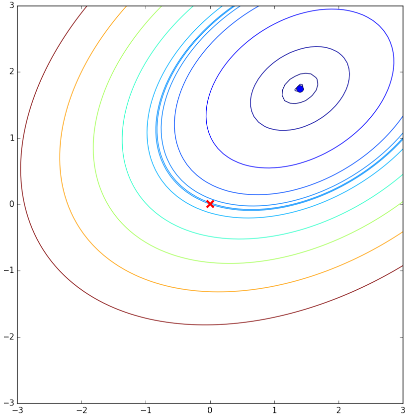
Minimise

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

Minimise $(\mathbf{Xw}-\mathbf{y})^{\mathsf{T}}(\mathbf{Xw}-\mathbf{y})$

subject to $\mathbf{w}^{\mathsf{T}}\mathbf{w} \leq R$

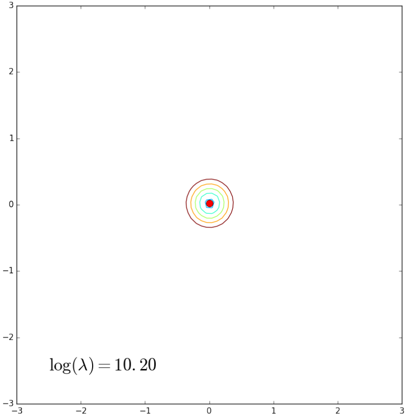




Ridge Regression

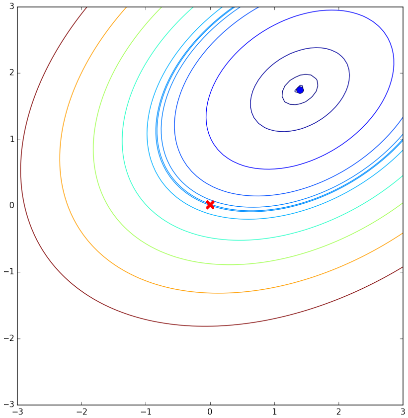
Minimise

$$(\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathsf{T}} (\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda \mathbf{w}^{\mathsf{T}} \mathbf{w}$$



Minimise $(\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathsf{T}} (\mathbf{X}\mathbf{w} - \mathbf{y})$

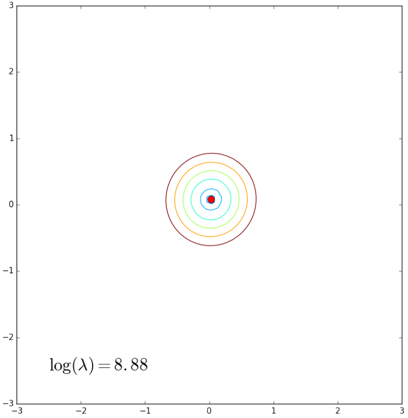
subject to $\mathbf{w}^{\mathsf{T}} \mathbf{w} \leq R$



Ridge Regression

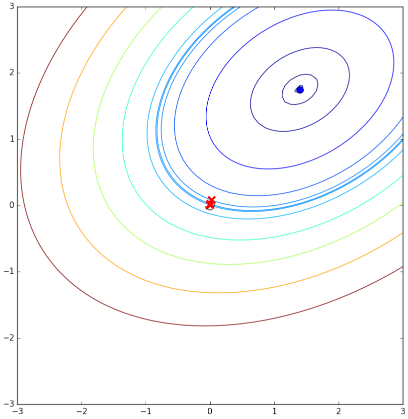
Minimise

$$(\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathsf{T}} (\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda \mathbf{w}^{\mathsf{T}} \mathbf{w}$$



Minimise $(\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathsf{T}} (\mathbf{X}\mathbf{w} - \mathbf{y})$

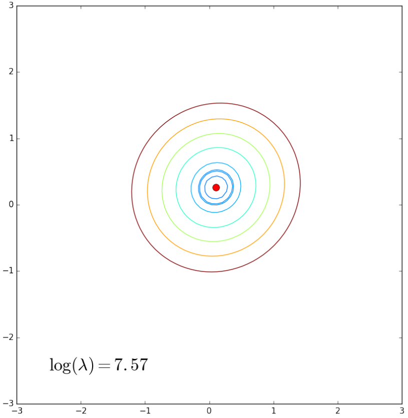
subject to $\mathbf{w}^{\mathsf{T}} \mathbf{w} \leq R$



Ridge Regression

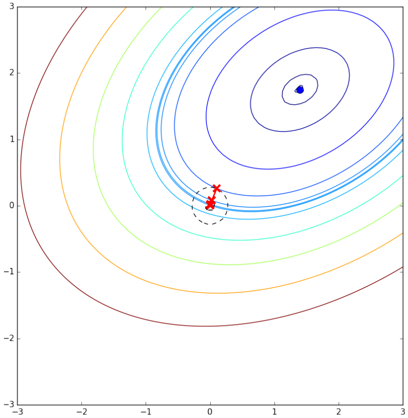
Minimise

$$(\mathbf{X}\mathbf{w}-\mathbf{y})^{\mathsf{T}}(\mathbf{X}\mathbf{w}-\mathbf{y})+\lambda\mathbf{w}^{\mathsf{T}}\mathbf{w}$$



Minimise $(\mathbf{X}\mathbf{w}-\mathbf{y})^{\mathsf{T}}(\mathbf{X}\mathbf{w}-\mathbf{y})$

subject to $\mathbf{w}^{\mathsf{T}}\mathbf{w} \leq R$



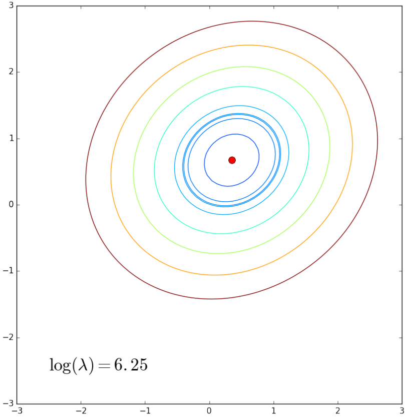
Ridge Regression

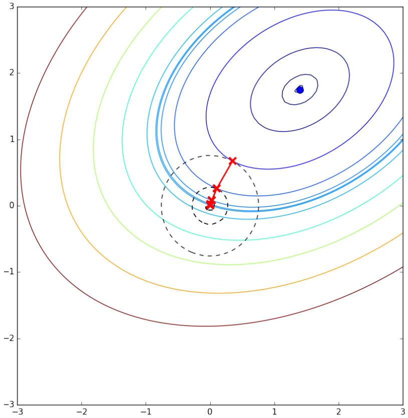
Minimise

$$(\mathbf{X}\mathbf{w}-\mathbf{y})^{\mathsf{T}}(\mathbf{X}\mathbf{w}-\mathbf{y})+\lambda\mathbf{w}^{\mathsf{T}}\mathbf{w}$$

Minimise $(\mathbf{X}\mathbf{w}-\mathbf{y})^{\mathsf{T}}(\mathbf{X}\mathbf{w}-\mathbf{y})$

subject to $\mathbf{w}^{\mathsf{T}}\mathbf{w} \le R$





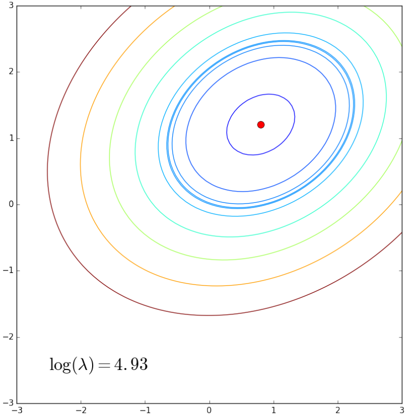
Ridge Regression

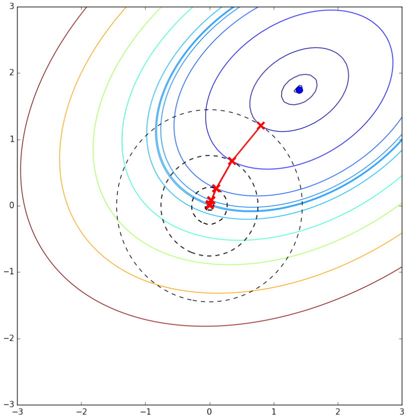
Minimise

$(\mathbf{X}\mathbf{w}-\mathbf{y})^{\mathsf{T}}(\mathbf{X}\mathbf{w}-\mathbf{y})+\lambda\mathbf{w}^{\mathsf{T}}\mathbf{w}$

Minimise $(\mathbf{X}\mathbf{w}-\mathbf{y})^{\mathsf{T}}(\mathbf{X}\mathbf{w}-\mathbf{y})$

subject to $\mathbf{w}^{\mathsf{T}}\mathbf{w} \leq R$





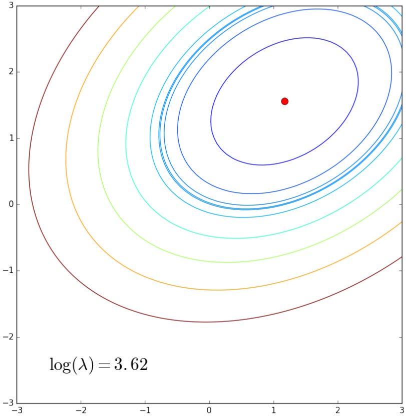
Ridge Regression

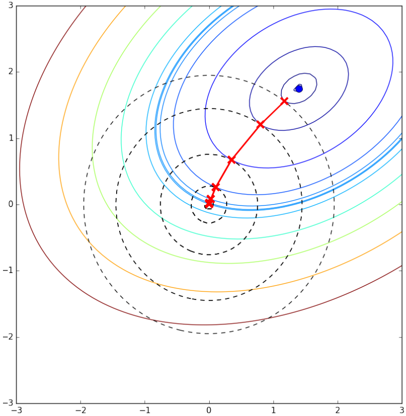
Minimise

$(\mathbf{X}\mathbf{w} - \mathbf{y})^\mathsf{T}(\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda \mathbf{w}^\mathsf{T}\mathbf{w}$

Minimise $(\mathbf{X}\mathbf{w} - \mathbf{y})^\mathsf{T}(\mathbf{X}\mathbf{w} - \mathbf{y})$

subject to $\mathbf{w}^\mathsf{T}\mathbf{w} \leq R$





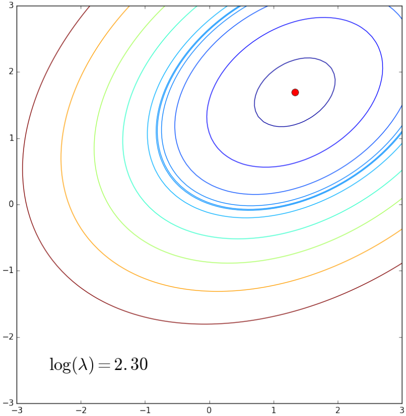
Ridge Regression

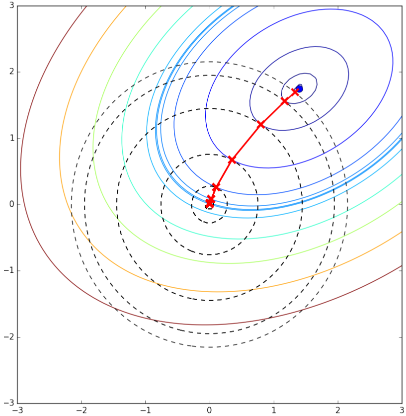
Minimise

$(\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathsf{T}} (\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda \mathbf{w}^{\mathsf{T}} \mathbf{w}$

Minimise $(\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathsf{T}} (\mathbf{X}\mathbf{w} - \mathbf{y})$

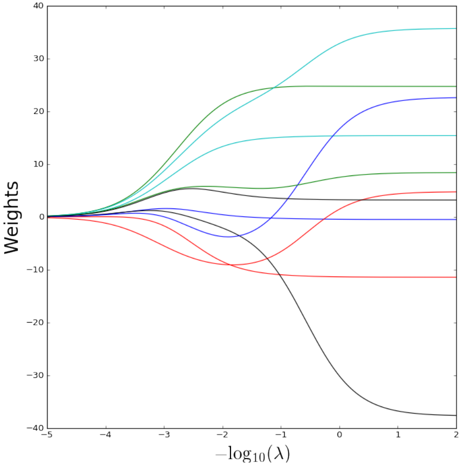
subject to $\mathbf{w}^{\mathsf{T}} \mathbf{w} \leq R$





Ridge Regression

As we decrease λ the 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

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

This is also called ℓ_2-regularization or weight-decay

Penalising weights "encourages fitting signal rather than just noise"

The Lasso

Lasso (least absolute shrinkage and selection operator) minimises the following objective function

Lasso Objective

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

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

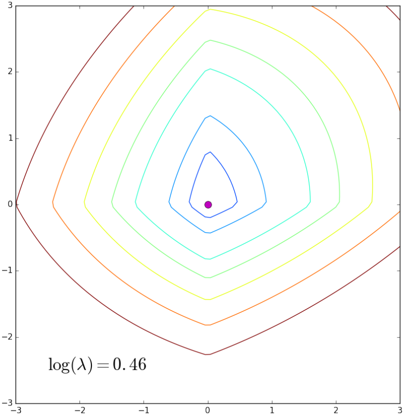
The Lasso : Optimization

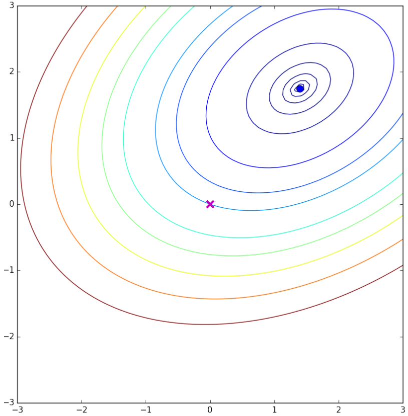
Minimise

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

Minimise $(\mathbf{X}\mathbf{w} - \mathbf{y})^\mathsf{T} (\mathbf{X}\mathbf{w} - \mathbf{y})$

subject to $\sum_{i=1}^{D} |w_i| \leq R$





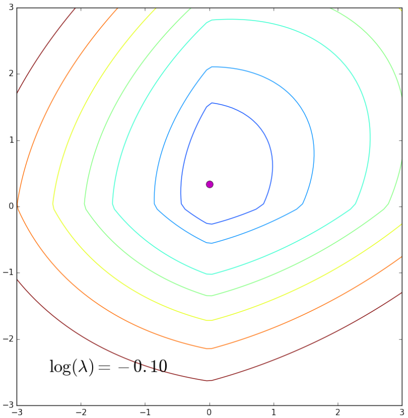
The Lasso : Optimization

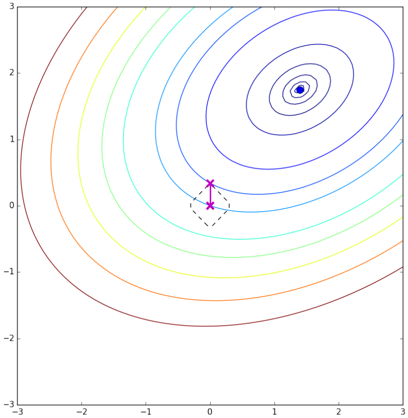
Minimise

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

Minimise $(\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathsf{T}} (\mathbf{X}\mathbf{w} - \mathbf{y})$

subject to $\sum_{i=1}^{D} |w_i| \leq R$





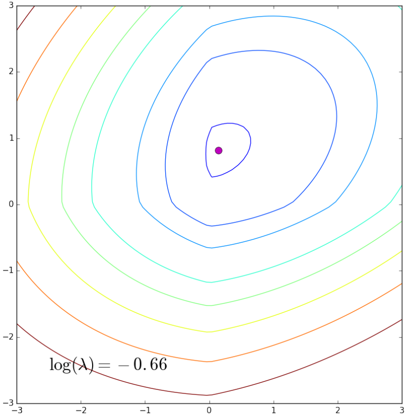
The Lasso : Optimization

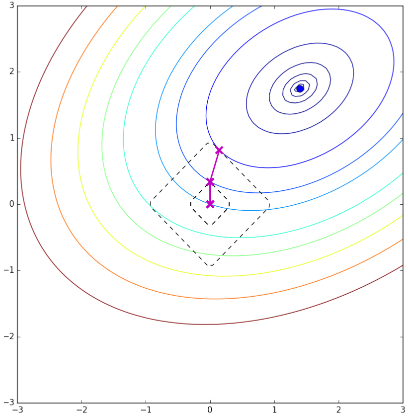
Minimise

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

Minimise $(\mathbf{X}\mathbf{w}-\mathbf{y})^{\mathsf{T}}(\mathbf{X}\mathbf{w}-\mathbf{y})$

subject to $\sum_{i=1}^{D}|w_i|\leq R$





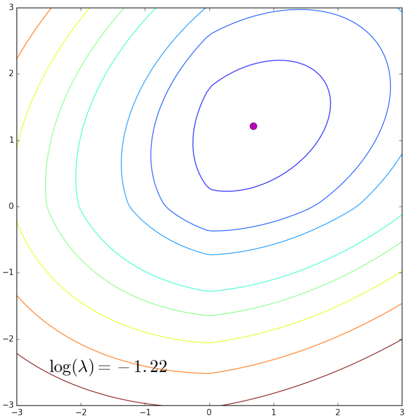
The Lasso : Optimization

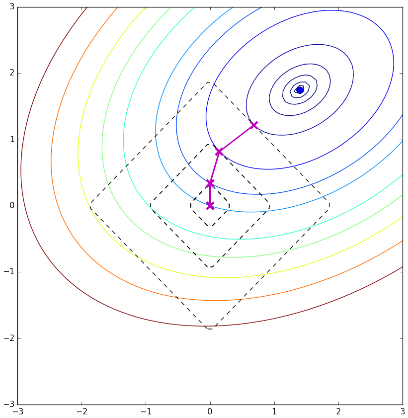
Minimise

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

Minimise $(\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathsf{T}} (\mathbf{X}\mathbf{w} - \mathbf{y})$

subject to $\sum_{i=1}^{D} |w_i| \leq R$





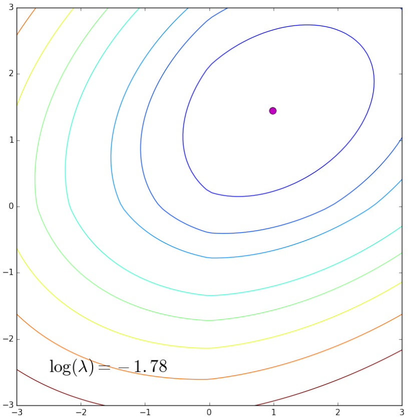
The Lasso : Optimization

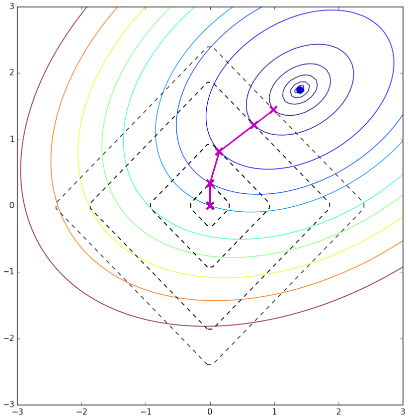
Minimise

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

Minimise $(\mathbf{Xw} - \mathbf{y})^{\mathsf{T}} (\mathbf{Xw} - \mathbf{y})$

subject to $\sum_{i=1}^{D} |w_i| \leq R$





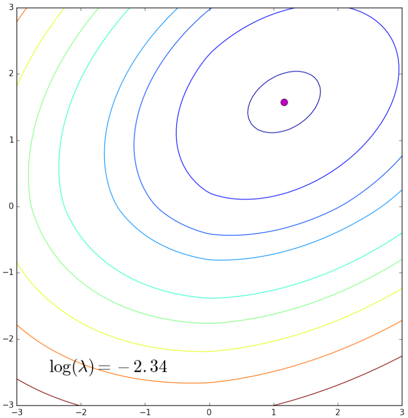
The Lasso : Optimization

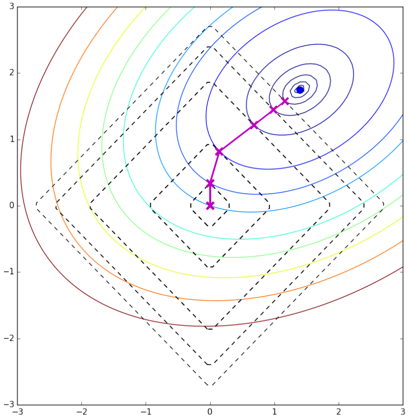
Minimise

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

Minimise $(\mathbf{Xw} - \mathbf{y})^\mathsf{T} (\mathbf{Xw} - \mathbf{y})$

subject to $\sum_{i=1}^{D} |w_i| \leq R$





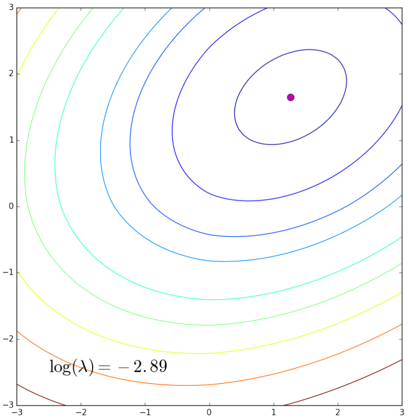
The Lasso : Optimization

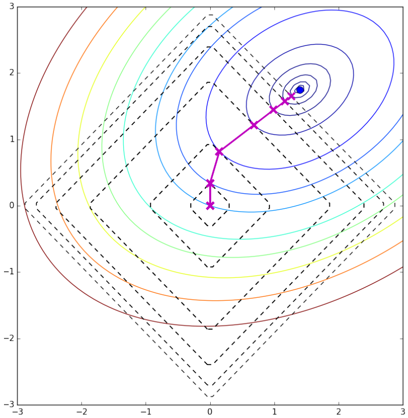
Minimise

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

Minimise $(\mathbf{Xw}-\mathbf{y})^{\mathsf{T}}(\mathbf{Xw}-\mathbf{y})$

subject to $\sum_{i=1}^{D}|w_i| \leq R$





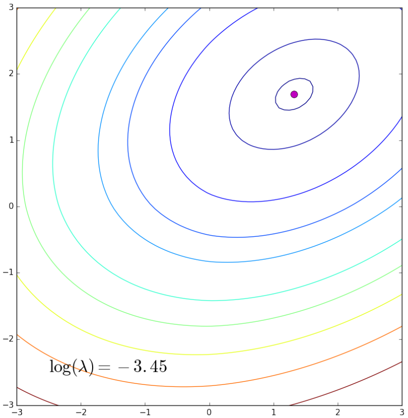
The Lasso : Optimization

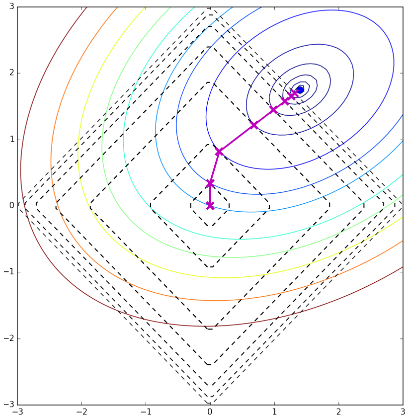
Minimise

$$\left(\mathbf{X}\mathbf{w} - \mathbf{y}\right)^{\mathsf{T}} \left(\mathbf{X}\mathbf{w} - \mathbf{y}\right) + \lambda \sum_{i=1}^{D} |w_i|$$

Minimise $\left(\mathbf{X}\mathbf{w} - \mathbf{y}\right)^{\mathsf{T}} \left(\mathbf{X}\mathbf{w} - \mathbf{y}\right)$

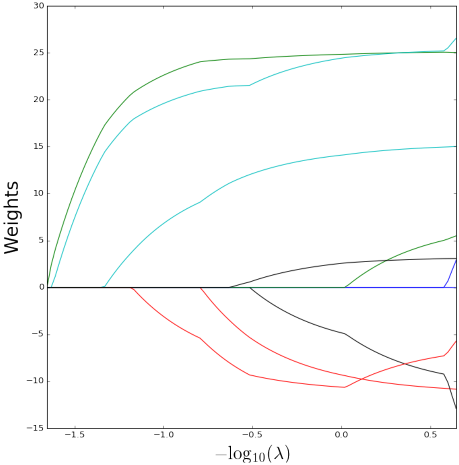
subject to $\sum_{i=1}^{D} |w_i| \leq R$



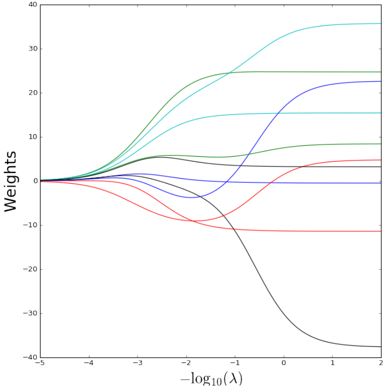


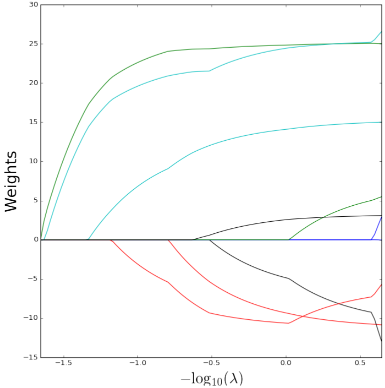
The Lasso Paths

As we decrease λ the magnitudes of weights start increasing



Comparing Ridge Regression and the Lasso





When using the Lasso, weights are often exactly 0.

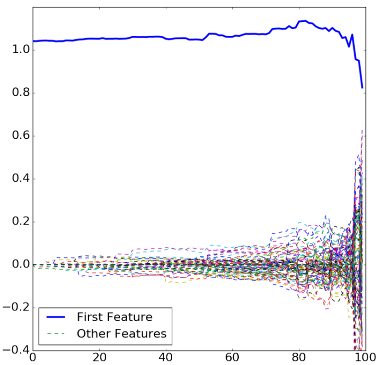
Thus, Lasso gives sparse models.

Overfitting: How does it occur?

We have $D = 100$ and $N = 100$ so that $\mathbf{X}$ is 100×100

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

$y_i = x_{i,1} + \mathcal{N}(0, \sigma^2)$, for $\sigma = 0.2$



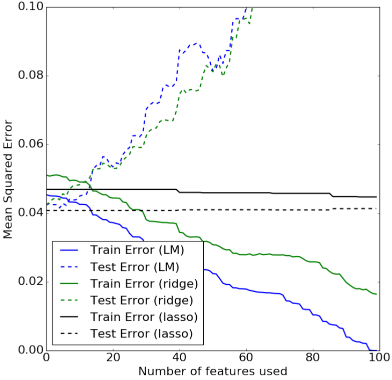
No regularization

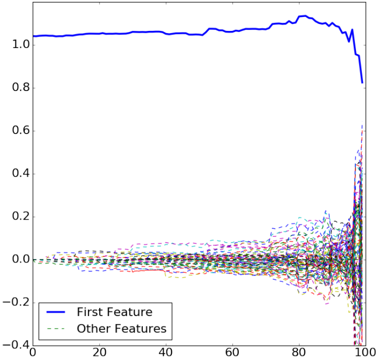
Overfitting: How does it occur?

We have $D = 100$ and $N = 100$ so that $\mathbf{X}$ is 100×100

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

$y_i = x_{i,1} + \mathcal{N}(0, \sigma^2)$, for $\sigma = 0.2$





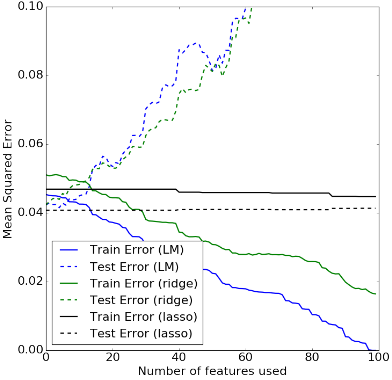
No regularization

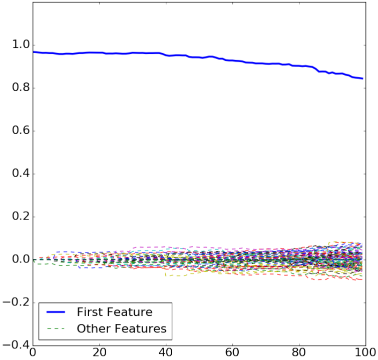
Overfitting: How does it occur?

We have $D = 100$ and $N = 100$ so that $\mathbf{X}$ is 100×100

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

$y_i = x_{i,1} + \mathcal{N}(0, \sigma^2)$, for $\sigma = 0.2$





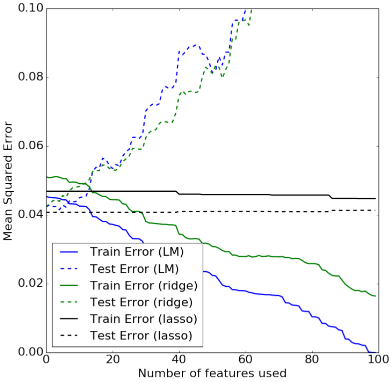
Ridge

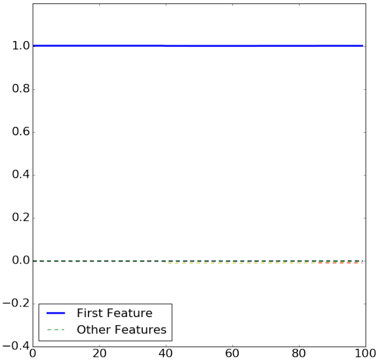
Overfitting: How does it occur?

We have $D = 100$ and $N = 100$ so that $\mathbf{X}$ is 100×100

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

$y_i = x_{i,1} + \mathcal{N}(0, \sigma^2)$, for $\sigma = 0.2$





Lasso

Outline

Ridge Regression and Lasso

Model Selection

How to Choose Hyper-parameters?

- So far, we were just trying 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 more hyperparameters

Using a Validation Set

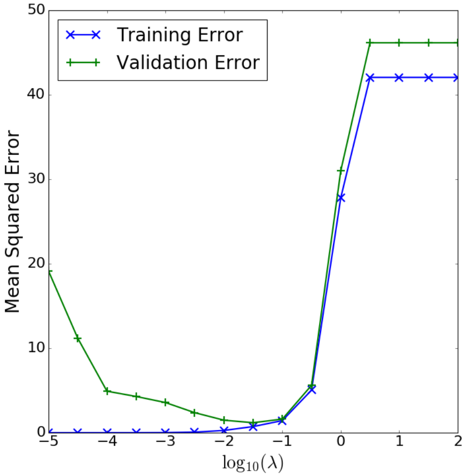
- Divide the data into parts: training, validation (and testing)
- Grid Search: Choose values for the hyperparameters from a finite set
- Train model using training set and evaluate on validation set

λ	training error(%)	validation error(%)
0.01	0	89
0.1	0	43
1	2	12
10	10	8
100	25	27

- Pick the value of λ that minimises validation error
- Typically, split the data as 80% for training, 20% for validation

Training and Validation Curves

- Plot of training and validation error vs λ for Lasso
- Validation error curve is U-shaped



K-Fold Cross Validation

When data is scarce, instead of splitting as training and validation:

- Divide data into K parts
- Use $K-1$ parts for training and 1 part as validation
- Commonly set $K=5$ or $K=10$
- When $K=N$ (the number of datapoints), it is called LOOCV (Leave one out cross validation)

Run 1	train	train	train	train	valid
Run 2	train	train	train	valid	train
Run 3	train	train	valid	train	train
Run 4	train	valid	train	train	train
Run 5	valid	train	train	train	train

Overfitting on the Validation Set

Suppose you do all the right things

- Train on the training set
- Choose hyperparameters using proper validation
- Test on the test set (real world), and your error is unacceptably high!

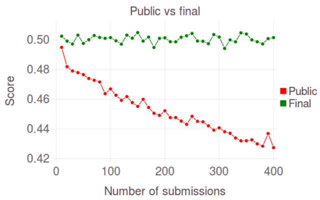
What would you do?

Winning Kaggle without reading the data!

Suppose the task is to predict N binary labels

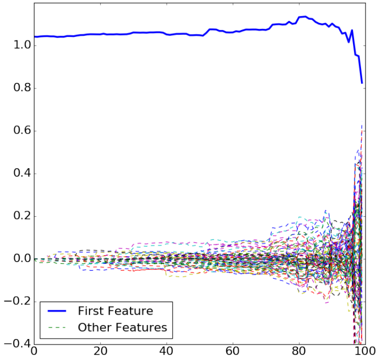
Algorithm (Wacky Boosting):

1. Choose $\mathbf{y}^1, \ldots, \mathbf{y}^k \in \{0,1\}^N$ randomly
2. Set $I = \{i \mid \mathrm{accuracy}(\mathbf{y}^i) > 51\%\}$
3. Output $\widehat{y}_j = \mathrm{majority}\{y_j^i \mid i \in I\}$



Source blog.mrtz.org

Feature Selection



- Recall that small training set with many features is prone to overfitting
- What if we discard irrelevant features and using training set with fewer features?
- Problem: there are 2^n subsets of features

Feature Selection

Forward search is a generic (i.e. learning algorithm independent) greedy approach to identify relevant features:

- 1 Set set of selected features to $F := \emptyset$
- 2 Repeat the following until $F = \{1, \ldots, n\}$:
  - Set $F_i := F \cup \{i\}$ for $i \in \{1, \ldots, n\} \setminus F$
  - Evaluate generalization error when using only features from F_i
  - Set new F to the best feature subset found
- 3 Return best overall feature subset found

Still requires $O(n^2)$ calls to underlying learning algorithm

Feature Selection

Filter feature selection is computationally more lightweight:

- Only keep feature x_i that provide information about y
- For instance, use mutual information as criterion:

$$I(x_i, y) = \sum_{x_i \in X} \sum_{y \in Y} p(x_i, y) \cdot \log \frac{p(x_i, y)}{p(x_i) \cdot p(y)}$$

- Retain top k features

Next Time

- Ridge Regression viewed through the Bayesian approach to Machine Learning
- Preparation for optimization
- Lecture takes place in the University Museum