COMPSCI 589
Lecture 8: Linear Regression, Ridge, and Lasso

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).
Created with support from National Science Foundation Award# IIS-1350522.

Outline

Views on Machine Learning

Mitchell (1997): "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Substitute "training data D" for "experience E."

The Regression Task

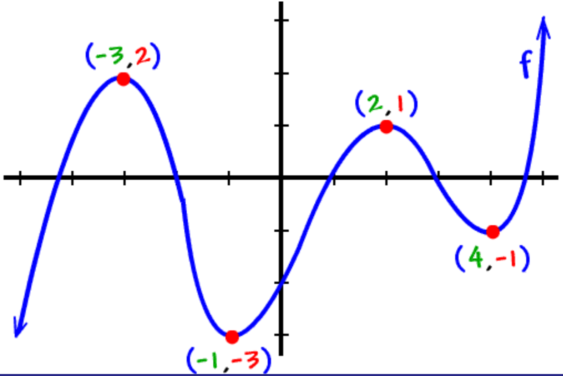
Definition: The Regression Task

Given a feature vector $\mathbf{x} \in \mathbb{R}^D$, predict it's corresponding output value $y \in \mathbb{R}$.

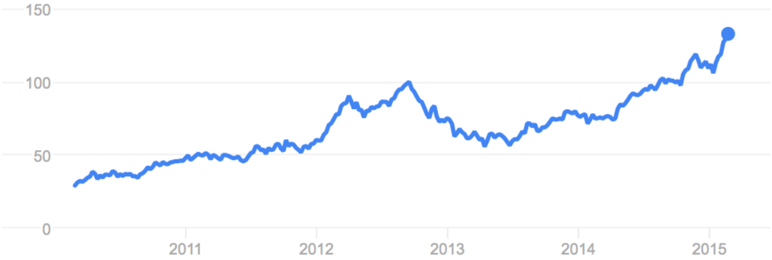
The Regression Task

Definition: The Regression Task

Given a feature vector $\mathbf{x} \in \mathbb{R}^D$, predict it's corresponding output value $y \in \mathbb{R}$.

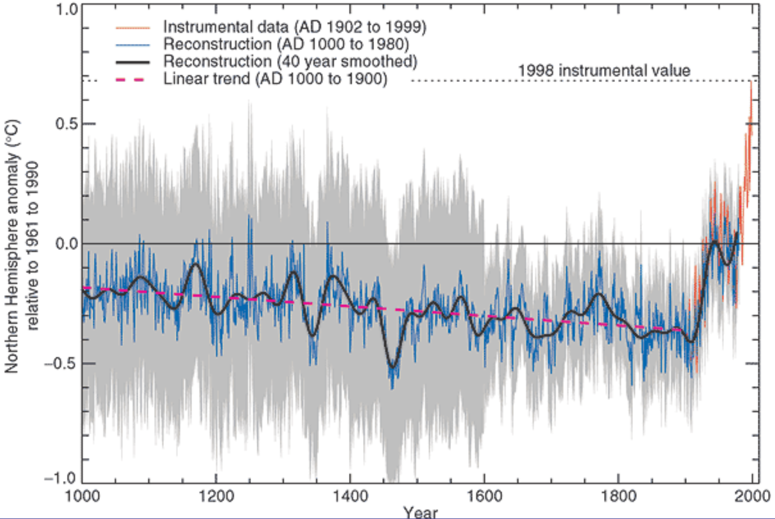


Example: Stock Prices

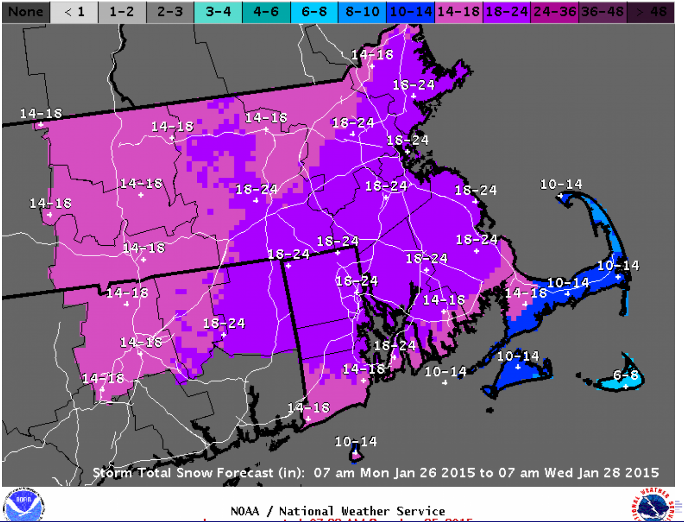


Open	130.02	Market cap	755.53B
High	133.00	P/E ratio (ttm)	17.92
Low	129.66	Dividend yield	1.41%

Example: Climate Change



Example: Weather Forecasting

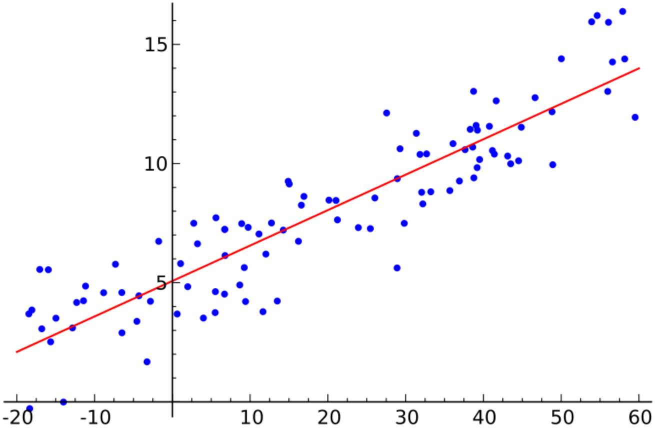


The Regression Learning Problem

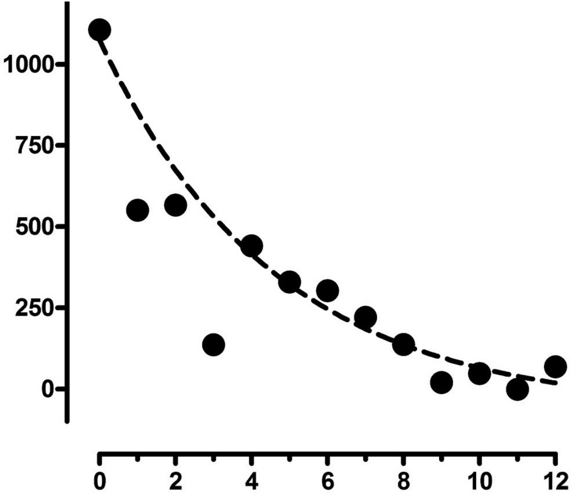
Definition: Regression Learning Problem

Given a data set of example pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1 : N\}$ where $\mathbf{x}_i \in \mathbb{R}^D$ is a feature vector and $y_i \in \mathbb{R}$ is the output, learn a function $f : \mathbb{R}^D \to \mathbb{R}$ that accurately predicts y for any feature vector $\mathbf{x}$.

Example: Linear Regression Learning



Example: Non-Linear Regression Learning



Error Measures: MSE

Definition: Mean Squared Error

Given a data set of example pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1 : N\}$ and a function $f : \mathbb{R}^D \to \mathcal{Y}$, the mean squared error of f on $\mathcal{D}$ is:

$$MSE(f, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2$$

Error Measures: MSE

Definition: Mean Squared Error

Given a data set of example pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1 : N\}$ and a function $f : \mathbb{R}^D \rightarrow \mathcal{Y}$, the mean squared error of f on $\mathcal{D}$ is:

$$MSE(f, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2$$

Related measures include:

Sum of Squared Errors: $SSE(f, \mathcal{D}) = N \cdot MSE(f, \mathcal{D})$

Risidual Sum of Squares: $RSS(f, \mathcal{D}) = N \cdot MSE(f, \mathcal{D})$

Root Mean Squared Error: $RMSE(f, \mathcal{D}) = \sqrt{MSE(f, \mathcal{D})}$

Error Measures: MAE

Definition: Mean Absolute Error

Given a data set of example pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1 : N\}$ and a function $f : \mathbb{R}^D \to \mathcal{Y}$, the mean absolute error of f on $\mathcal{D}$ is:

$$MAE(f, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} |y_i - f(\mathbf{x}_i)|$$

Outline

Linear Regression

Linear regression is a parametric regression method that assumes the relationship between y and $\mathbf{x}$ is a linear function with parameters $\mathbf{w} = [w_1, ..., w_D]^T$ and b.

Linear Regression

Linear regression is a parametric regression method that assumes the relationship between y and $\mathbf{x}$ is a linear function with parameters $\mathbf{w} = [w_1, ..., w_D]^T$ and b.

Linear Regression Function

$$f_{Lin}(\mathbf{x}) = \left(\sum_{d=1}^{D} w_d x_d\right) + b = \mathbf{x}\mathbf{w} + b$$

Linear Regression

Linear regression is a parametric regression method that assumes the relationship between y and $\mathbf{x}$ is a linear function with parameters $\mathbf{w} = [w_1, ..., w_D]^T$ and b.

Linear Regression Function

$$f_{Lin}(\mathbf{x}) = \left(\sum_{d=1}^{D} w_d x_d\right) + b = \mathbf{x}\mathbf{w} + b$$

Question: How can we learn the parameter values $\mathbf{w}$ and b?

Ordinary Least Squares Linear Regression

Ordinary least squares selects the linear regression parameters to minimize the mean squared error (MSE) on the training data set:

Ordinary Least Squares Linear Regression

Ordinary least squares selects the linear regression parameters to minimize the mean squared error (MSE) on the training data set:

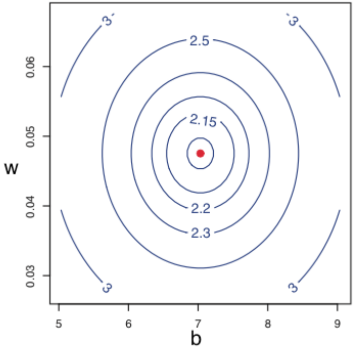
$$\mathbf{w}^*, b^* = \arg\min_{\mathbf{w},b} \frac{1}{N} \sum_{i=1}^{N} (y_i - \mathbf{x}_i \mathbf{w} + b)^2$$

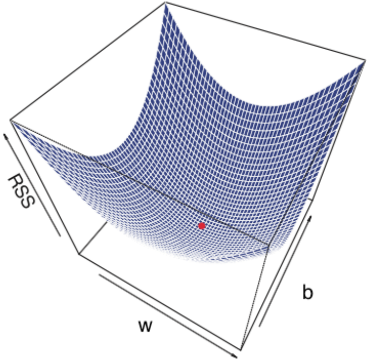
Solving OLS For One Feature

$$\arg\min_{w,b} \frac{1}{N}\sum_{i=1}^{N}(y_i - wx_i - b)^2$$

Solving OLS For One Feature

$$\arg\min_{w,b} \frac{1}{N}\sum_{i=1}^{N}(y_i - wx_i - b)^2$$





Solving OLS For One Feature

$$\arg\min_{w,b} \frac{1}{N}\sum_{i=1}^{N}(y_i - wx_i - b)^2$$

Solving OLS For One Feature

$$\arg\min_{w,b} \frac{1}{N}\sum_{i=1}^{N}(y_i - wx_i - b)^2$$

$$\frac{\partial}{\partial w}\frac{1}{N}\sum_{i=1}^{N}(y_i - wx_i - b)^2 = 0$$

$$\frac{\partial}{\partial b}\frac{1}{N}\sum_{i=1}^{N}(y_i - wx_i - b)^2 = 0$$

Solving OLS For One Feature

$$2\frac{1}{N}\sum_{i=1}^{N}(y_i - wx_i - b)x_i = 0$$

$$2\frac{1}{N}\sum_{i=1}^{N}(y_i - wx_i - b) = 0$$

Solving OLS For One Feature

$$2\frac{1}{N}\sum_{i=1}^{N}(y_i - wx_i - b)x_i = 0$$

$$2\frac{1}{N}\sum_{i=1}^{N}(y_i - wx_i - b) = 0$$

$$w\left(\sum_{i=1}^{N}x_i^2\right) + b\left(\sum_{i=1}^{N}x_i\right) = \sum_{i=1}^{N}(y_i x_i)$$

$$w\left(\sum_{i=1}^{N}x_i\right) + b(N) = \sum_{i=1}^{N}(y_i)$$

Solving OLS For One Feature

$$\begin{bmatrix} \sum_{i=1}^N x_i^2 & \sum_{i=1}^N x_i \\ \sum_{i=1}^N x_i & N \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N y_i x_i \\ \sum_{i=1}^N y_i \end{bmatrix}$$

Solving OLS For One Feature

$$\begin{bmatrix} \sum_{i=1}^N x_i^2 & \sum_{i=1}^N x_i \\ \sum_{i=1}^N x_i & N \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N y_i x_i \\ \sum_{i=1}^N y_i \end{bmatrix}$$

$$\begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N x_i^2 & \sum_{i=1}^N x_i \\ \sum_{i=1}^N x_i & N \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^N y_i x_i \\ \sum_{i=1}^N y_i \end{bmatrix}$$

General OLS Solution

Assume that $\mathbf{X}$ is a data matrix with one data case $\mathbf{x}_i \in \mathbb{R}^D$ per row, and $\mathbf{Y}$ is a column vector containing the corresponding outputs. The general OLS solution is:

General OLS Solution

Assume that $\mathbf{X}$ is a data matrix with one data case $\mathbf{x}_i \in \mathbb{R}^D$ per row, and $\mathbf{Y}$ is a column vector containing the corresponding outputs. The general OLS solution is:

$$
\begin{aligned}
\mathbf{w}^* &= \arg\min_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{x}_i\mathbf{w})^2 \\
&= \arg\min_{\mathbf{w}} \frac{1}{N}(\mathbf{Y} - \mathbf{X}\mathbf{w})^T(\mathbf{Y} - \mathbf{X}\mathbf{w}) \\
0 &= \frac{\partial}{\partial \mathbf{w}}\frac{1}{N}(\mathbf{Y} - \mathbf{X}\mathbf{w})^T(\mathbf{Y} - \mathbf{X}\mathbf{w}) \\
0 &= \mathbf{X}^T(\mathbf{Y} - \mathbf{X}\mathbf{w}) \\
\mathbf{X}^T\mathbf{X}\mathbf{w} &= \mathbf{X}^T\mathbf{Y} \\
\mathbf{w}^* &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}
\end{aligned}
$$

Connection to Probabilistic Models

This same solution can be derived as the maximum conditional likelihood estimate for the parameters of a conditional Normal model. σ^2 is the noise variance.

Connection to Probabilistic Models

This same solution can be derived as the maximum conditional likelihood estimate for the parameters of a conditional Normal model. σ^2 is the noise variance.

$$P(y|\mathbf{x}) = \mathcal{N}(y; \mathbf{xw}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y - \mathbf{xw})^2\right)$$

Connection to Probabilistic Models

This same solution can be derived as the maximum conditional likelihood estimate for the parameters of a conditional Normal model. σ^2 is the noise variance.

$$P(y|\mathbf{x}) = \mathcal{N}(y; \mathbf{xw}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y - \mathbf{xw})^2\right)$$

This view shows that OLS assumes the residuals are Normally distributed. This assumption is violated in many real world processes that have significant outliers or heavy-tailed noise.

Strengths and Limitations of OLS

- Need at least D data cases to learn a model with a D dimensional feature vector.

Strengths and Limitations of OLS

- Need at least D data cases to learn a model with a D dimensional feature vector. Otherwise inverse of $\mathbf{X}^T\mathbf{X}$ is not defined.

Strengths and Limitations of OLS

- Need at least D data cases to learn a model with a D dimensional feature vector. Otherwise inverse of $\mathbf{X}^T\mathbf{X}$ is not defined.
- Very sensitive to noise and outliers due to MSE objective function/Normally distributed residuals assumption.

Strengths and Limitations of OLS

- Need at least D data cases to learn a model with a D dimensional feature vector. Otherwise inverse of $\mathbf{X}^T\mathbf{X}$ is not defined.
- Very sensitive to noise and outliers due to MSE objective function/Normally distributed residuals assumption.
- Sensitive to co-linear features ($x_i \approx ax_j + b$).

Strengths and Limitations of OLS

- Need at least D data cases to learn a model with a D dimensional feature vector. Otherwise inverse of $\mathbf{X}^T\mathbf{X}$ is not defined.
- Very sensitive to noise and outliers due to MSE objective function/Normally distributed residuals assumption.
- Sensitive to co-linear features ($x_i \approx ax_j + b$). Otherwise inverse of $\mathbf{X}^T\mathbf{X}$ is not numerically stable.

Strengths and Limitations of OLS

- Need at least D data cases to learn a model with a D dimensional feature vector. Otherwise inverse of $\mathbf{X}^T\mathbf{X}$ is not defined.
- Very sensitive to noise and outliers due to MSE objective function/Normally distributed residuals assumption.
- Sensitive to co-linear features ($x_i \approx ax_j + b$). Otherwise inverse of $\mathbf{X}^T\mathbf{X}$ is not numerically stable.
- High bias (assumes linear relationships between the features and target).

Strengths and Limitations of OLS

- Need at least D data cases to learn a model with a D dimensional feature vector. Otherwise inverse of $\mathbf{X}^T\mathbf{X}$ is not defined.
- Very sensitive to noise and outliers due to MSE objective function/Normally distributed residuals assumption.
- Sensitive to co-linear features ($x_i \approx ax_j + b$). Otherwise inverse of $\mathbf{X}^T\mathbf{X}$ is not numerically stable.
- High bias (assumes linear relationships between the features and target).
- Computation is cubic in data dimension D.

Strengths and Limitations of OLS

- Need at least D data cases to learn a model with a D dimensional feature vector. Otherwise inverse of $\mathbf{X}^T\mathbf{X}$ is not defined.
- Very sensitive to noise and outliers due to MSE objective function/Normally distributed residuals assumption.
- Sensitive to co-linear features ($x_i \approx ax_j + b$). Otherwise inverse of $\mathbf{X}^T\mathbf{X}$ is not numerically stable.
- High bias (assumes linear relationships between the features and target).
- Computation is cubic in data dimension D.
- Variance is generally low unless there are outliers.

Outline

Regularized Linear Regression

Just as in classification, regression models require capacity control to avoid overfitting and numerical stability problems in high dimensions. This is accomplished by regularizing the weight parameters during learning.

Regularized Linear Regression

Just as in classification, regression models require capacity control to avoid overfitting and numerical stability problems in high dimensions. This is accomplished by regularizing the weight parameters during learning.

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{x}_i\mathbf{w})^2 + \lambda||\mathbf{w}||$$

$$= \arg\min_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{x}_i\mathbf{w})^2 \text{ ... st } ||\mathbf{w}|| \leq c$$

Ridge Regression

Ridge regression is the name given to regularized least squares when the weights are penalized using the square of the ℓ_2 norm $||\mathbf{w}||_2^2 = \mathbf{w}^T\mathbf{w} = \sum_{d=1}^{D} w_d^2$:

Ridge Regression

Ridge regression is the name given to regularized least squares when the weights are penalized using the square of the ℓ_2 norm $||\mathbf{w}||_2^2 = \mathbf{w}^T\mathbf{w} = \sum_{d=1}^{D} w_d^2$:

$$\begin{aligned}\mathbf{w}^* &= \arg\min_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{x}_i\mathbf{w})^2 + \lambda||\mathbf{w}||_2^2 \\ &= \arg\min_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{x}_i\mathbf{w})^2 \text{ ... st } ||\mathbf{w}||_2^2 \leq c\end{aligned}$$

Ridge Regression

Ridge regression is the name given to regularized least squares when the weights are penalized using the square of the ℓ_2 norm $||\mathbf{w}||_2^2 = \mathbf{w}^T\mathbf{w} = \sum_{d=1}^{D} w_d^2$:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{x}_i\mathbf{w})^2 + \lambda||\mathbf{w}||_2^2$$

$$= \arg\min_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{x}_i\mathbf{w})^2 \text{ ... st } ||\mathbf{w}||_2^2 \leq c$$

In this case, it is easy to show that the optimal regularized weights are:

$$\mathbf{w}^* = (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\mathbf{Y}$$

The Lasso

The Lasso is the name given to regularized least squares when the weights are penalized using the the ℓ_1 norm $||\mathbf{w}||_1 = \sum_{d=1}^{D} |w_d|$:

The Lasso

The Lasso is the name given to regularized least squares when the weights are penalized using the the ℓ_1 norm $||\mathbf{w}||_1 = \sum_{d=1}^{D} |w_d|$:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{x}_i\mathbf{w})^2 + \lambda||\mathbf{w}||_1$$

$$= \arg\min_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{x}_i\mathbf{w})^2 \text{ ... st } ||\mathbf{w}||_1 \leq c$$

The Lasso

The Lasso is the name given to regularized least squares when the weights are penalized using the the ℓ_1 norm $||\mathbf{w}||_1 = \sum_{d=1}^{D} |w_d|$:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} (y_i - \mathbf{x}_i \mathbf{w})^2 + \lambda ||\mathbf{w}||_1$$
$$= \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} (y_i - \mathbf{x}_i \mathbf{w})^2 \text{ ... st } ||\mathbf{w}||_1 \leq c$$

The Lasso problem is a quadratic programming problem. However, it can be solved efficiently for all values of λ using an algorithm called *least angle regression* (LARS). The advantage of the Lasso is that it simultaneously performs regularization and feature selection.

Lasso vs Ridge

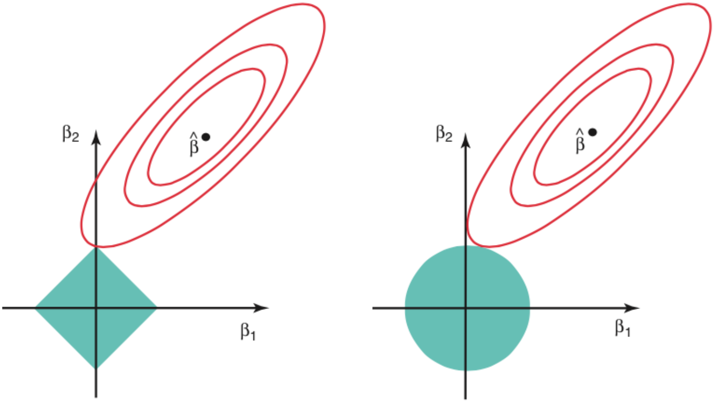


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

Strengths and Limitations of Ridge and Lasso

- Solves the problem of needing at least D data cases to learn a model with a D dimensional feature vector.

Strengths and Limitations of Ridge and Lasso

- Solves the problem of needing at least D data cases to learn a model with a D dimensional feature vector.
- Solves the problem of co-linear features ($x_i \approx ax_j + b$).

Strengths and Limitations of Ridge and Lasso

- Solves the problem of needing at least D data cases to learn a model with a D dimensional feature vector.
- Solves the problem of co-linear features ($x_i \approx ax_j + b$).
- MSE objective function still sensitive to noise and outliers, but regularization can reduce the possibility of very large weights overfitting to outliers.

Strengths and Limitations of Ridge and Lasso

- Solves the problem of needing at least D data cases to learn a model with a D dimensional feature vector.
- Solves the problem of co-linear features ($x_i \approx ax_j + b$).
- MSE objective function still sensitive to noise and outliers, but regularization can reduce the possibility of very large weights overfitting to outliers.
- Does not solve bias problem

Strengths and Limitations of Ridge and Lasso

- Solves the problem of needing at least D data cases to learn a model with a D dimensional feature vector.
- Solves the problem of co-linear features ($x_i \approx ax_j + b$).
- MSE objective function still sensitive to noise and outliers, but regularization can reduce the possibility of very large weights overfitting to outliers.
- Does not solve bias problem
- Computation for ridge is still cubic in data dimension D, but now need to determine regularization parameters. Computation for LARS is similar.

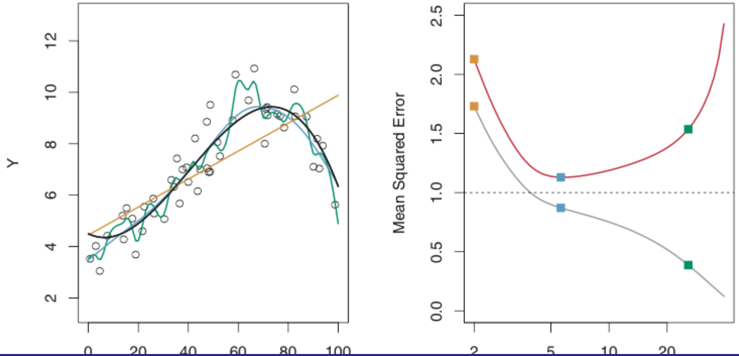
Outline

Basis Expansion

Just as with linear classification models, linear regression models can be extended to capture non-linear relationships using basis function expansions. The polynomial basis is often used for this purpose, although it is not sensible for forecasting.

Basis Expansion

Just as with linear classification models, linear regression models can be extended to capture non-linear relationships using basis function expansions. The polynomial basis is often used for this purpose, although it is not sensible for forecasting.



Strengths and Limitations of Basis Expansion

- Does solve the bias problem.

Strengths and Limitations of Basis Expansion

- Does solve the bias problem.
- MSE objective function still sensitive to noise and outliers. Basis expansions can easily overfit so need to control capacity.

Strengths and Limitations of Basis Expansion

- Does solve the bias problem.
- MSE objective function still sensitive to noise and outliers. Basis expansions can easily overfit so need to control capacity.
- Computation is cubic in the dimensionality of the basis function expansion. Can be costly.