Linear Models

DS-GA 1013 / MATH-GA 2824 Optimization-based Data Analysis

http://www.cims.nyu.edu/~cfgranda/pages/OBDA_fall17/index.html

Carlos Fernandez-Granda

Regression

The aim is to learn a function h that relates

- a response or dependent variable y
- to several observed variables $x_1, x_2, \ldots, x_p$, known as covariates, features or independent variables

The response is assumed to be of the form

$$y = h(\vec{x}) + z$$

where $\vec{x} \in \mathbb{R}^p$ contains the features and z is noise

Linear regression

The regression function h is assumed to be linear

$$y^{(i)} = \vec{x}^{(i)\,T} \vec{\beta}^* + z^{(i)}, \quad 1 \leq i \leq n$$

Our aim is to estimate $\vec{\beta}^* \in \mathbb{R}^p$ from the data

Linear regression

In matrix form

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \cdots \\ y^{(n)} \end{bmatrix} = \begin{bmatrix} \vec{x}_1^{(1)} & \vec{x}_2^{(1)} & \cdots & \vec{x}_p^{(1)} \\ \vec{x}_1^{(2)} & \vec{x}_2^{(2)} & \cdots & \vec{x}_p^{(2)} \\ \cdots & \cdots & \cdots & \cdots \\ \vec{x}_1^{(n)} & \vec{x}_2^{(n)} & \cdots & \vec{x}_p^{(n)} \end{bmatrix} \begin{bmatrix} \vec{\beta}_1^* \\ \vec{\beta}_2^* \\ \cdots \\ \vec{\beta}_p^* \end{bmatrix} + \begin{bmatrix} z^{(1)} \\ z^{(2)} \\ \cdots \\ z^{(n)} \end{bmatrix}$$

Equivalently,

$$\vec{y} = X\vec{\beta}^* + \vec{z}$$

Linear model for GDP

State	GDP (millions)	Population	Unemployment Rate
North Dakota	52 089	757 952	2.4
Alabama	204 861	4 863 300	3.8
Mississippi	107 680	2 988 726	5.2
Arkansas	120 689	2 988 248	3.5
Kansas	153 258	2 907 289	3.8
Georgia	525 360	10 310 371	4.5
Iowa	178 766	3 134 693	3.2
West Virginia	73 374	1 831 102	5.1
Kentucky	197 043	4 436 974	5.2
Tennessee	???	6 651 194	3.0

Centering

$$\vec{y}_{\text{cent}} = \begin{bmatrix} -127\,147 \\ 25\,625 \\ -71\,556 \\ -58\,547 \\ -25\,978 \\ 470 \\ -105\,862 \\ 17\,807 \end{bmatrix} \qquad X_{\text{cent}} = \begin{bmatrix} 3\,044\,121 & -1.7 \\ 1\,061\,227 & -2.8 \\ -813\,346 & 1.1 \\ -813\,825 & -5.8 \\ -894\,784 & -2.8 \\ 6508\,298 & 4.2 \\ -667\,379 & -8.8 \\ -1\,970\,971 & 1.0 \\ 634\,901 & 1.1 \end{bmatrix}$$

$$\operatorname{av}(\vec{y}) = 179\,236 \qquad \operatorname{av}(X) = \begin{bmatrix} 3\,802\,073 & 4.1 \end{bmatrix}$$

Normalizing

$$\vec{y}_{\text{norm}} = \begin{bmatrix} -0.321 \\ 0.065 \\ -0.180 \\ -0.148 \\ -0.065 \\ 0.872 \\ -0.001 \\ -0.267 \\ 0.045 \end{bmatrix} \qquad X_{\text{norm}} = \begin{bmatrix} -0.394 & -0.600 \\ 0.137 & -0.099 \\ -0.105 & 0.401 \\ -0.105 & -0.207 \\ -0.116 & -0.099 \\ 0.843 & 0.151 \\ -0.086 & -0.314 \\ -0.255 & 0.366 \\ 0.082 & 0.401 \end{bmatrix}$$

$$\text{std}\left(\vec{y}\right) = 396\ 701 \qquad \text{std}\left(X\right) = \begin{bmatrix} 7\ 720\ 656 & 2.80 \end{bmatrix}$$

Linear model for GDP

Aim: find $\vec{\beta} \in \mathbb{R}^2$ such that $\vec{y}_{\text{norm}} \approx X_{\text{norm}} \vec{\beta}$

The estimate for the GDP of Tennessee will be

$$\vec{y}^{\,\text{Ten}} = \text{av}\left(\vec{y}\right) + \text{std}\left(\vec{y}\right) \left\langle \vec{x}_{\text{norm}}^{\,\text{Ten}}, \vec{\beta} \right\rangle$$

where $\vec{x}_{\text{norm}}^{\,\text{Ten}}$ is centered using $\text{av}\left(X\right)$ and normalized using $\text{std}\left(X\right)$

Temperature predictor

A friend tells you:

I found a cool way to predict the average daily temperature in New York:
It's just a linear combination of the temperature in every other state.
I fit the model on data from the last month and a half and it's perfect!

System of equations

A is $n \times p$ and full rank

$$A\vec{b} = \vec{c}$$

- If $n < p$ the system is underdetermined: infinite solutions for any $\vec{b}$! (*overfitting*)
- If $n = p$ the system is determined: unique solution for any $\vec{b}$ (*overfitting*)
- If $n > p$ the system is overdetermined: unique solution exists only if $\vec{b} \in \mathrm{col}\,(A)$ (if there is noise, no solutions)

Least squares

For fixed $\vec{\beta}$ we can evaluate the error using

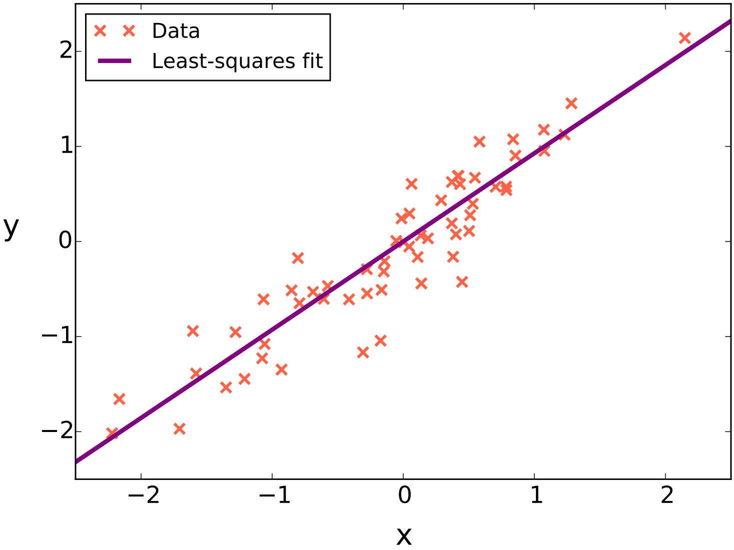
$$\sum_{i=1}^{n} \left(y^{(i)} - \vec{x}^{(i)\,T} \vec{\beta} \right)^2 = \left|\left| \vec{y} - X\vec{\beta} \right|\right|_2^2$$

The least-squares estimate $\vec{\beta}_{\text{LS}}$ minimizes this cost function

$$\begin{aligned} \vec{\beta}_{\text{LS}} &:= \arg\min_{\vec{\beta}} \quad \left|\left| \vec{y} - X\vec{\beta} \right|\right|_2 \\ &= \left(X^T X \right)^{-1} X^T \vec{y} \end{aligned}$$

if X is full rank and $n \geq p$

Least-squares fit



Least-squares solution

Let $X = USV^T$

$$\vec{y} = UU^T\vec{y} + \left(I - UU^T\right)\vec{y}$$

By the Pythagorean theorem

$$\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 =$$

Least-squares solution

Let $X = USV^T$

$$\vec{y} = UU^T\vec{y} + \left(I - UU^T\right)\vec{y}$$

By the Pythagorean theorem

$$\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \left|\left|\left(I - UU^T\right)\vec{y}\right|\right|_2^2 + \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2$$

$$\arg\min_{\vec{\beta}} \quad \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2$$

Least-squares solution

Let $X = USV^T$

$$\vec{y} = UU^T\vec{y} + \left(I - UU^T\right)\vec{y}$$

By the Pythagorean theorem

$$\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \left|\left|\left(I - UU^T\right)\vec{y}\right|\right|_2^2 + \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2$$

$$\arg\min_{\vec{\beta}} \quad \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \arg\min_{\vec{\beta}} \quad \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2$$

Least-squares solution

Let $X = USV^T$

$$\vec{y} = UU^T\vec{y} + \left(I - UU^T\right)\vec{y}$$

By the Pythagorean theorem

$$\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \left|\left|\left(I - UU^T\right)\vec{y}\right|\right|_2^2 + \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2$$

$$\begin{aligned}\arg\min_{\vec{\beta}} \quad \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 &= \arg\min_{\vec{\beta}} \quad \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2 \\ &= \arg\min_{\vec{\beta}} \quad \left|\left|UU^T\vec{y} - USV^T\vec{\beta}\right|\right|_2^2\end{aligned}$$

Least-squares solution

Let $X = USV^T$

$$\vec{y} = UU^T\vec{y} + \left(I - UU^T\right)\vec{y}$$

By the Pythagorean theorem

$$\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \left|\left|\left(I - UU^T\right)\vec{y}\right|\right|_2^2 + \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2$$

$$\begin{aligned}
\arg\min_{\vec{\beta}} \quad \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 &= \arg\min_{\vec{\beta}} \quad \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2 \\
&= \arg\min_{\vec{\beta}} \quad \left|\left|UU^T\vec{y} - USV^T\vec{\beta}\right|\right|_2^2 \\
&= \arg\min_{\vec{\beta}} \quad \left|\left|U^T\vec{y} - SV^T\vec{\beta}\right|\right|_2^2
\end{aligned}$$

Least-squares solution

Let $X = USV^T$

$$\vec{y} = UU^T\vec{y} + \left(I - UU^T\right)\vec{y}$$

By the Pythagorean theorem

$$\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 = \left|\left|\left(I - UU^T\right)\vec{y}\right|\right|_2^2 + \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2$$

$$\begin{aligned}
\arg\min_{\vec{\beta}} \quad \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 &= \arg\min_{\vec{\beta}} \quad \left|\left|UU^T\vec{y} - X\vec{\beta}\right|\right|_2^2 \\
&= \arg\min_{\vec{\beta}} \quad \left|\left|UU^T\vec{y} - USV^T\vec{\beta}\right|\right|_2^2 \\
&= \arg\min_{\vec{\beta}} \quad \left|\left|U^T\vec{y} - SV^T\vec{\beta}\right|\right|_2^2 \\
&= VS^{-1}U^T\vec{y} = \left(X^TX\right)^{-1}X^T\vec{y}
\end{aligned}$$

Linear model for GDP

The least-squares estimate is

$$\vec{\beta}_{\mathsf{LS}} = \begin{bmatrix} 1.019 \\ -0.111 \end{bmatrix}$$

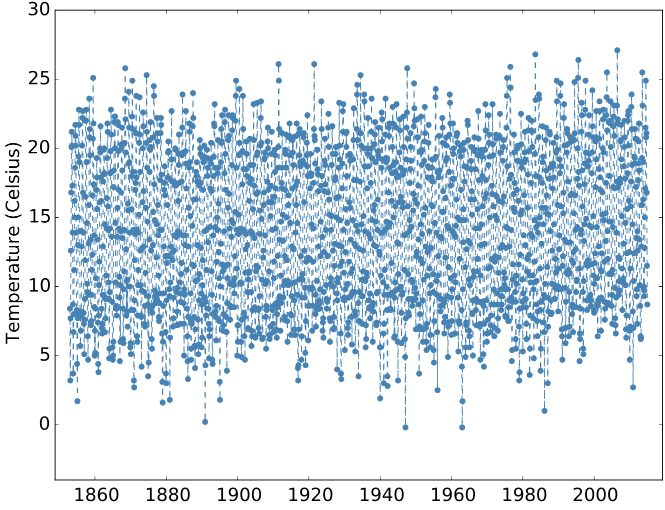
GDP roughly proportional to the population

Unemployment has a negative (linear) effect

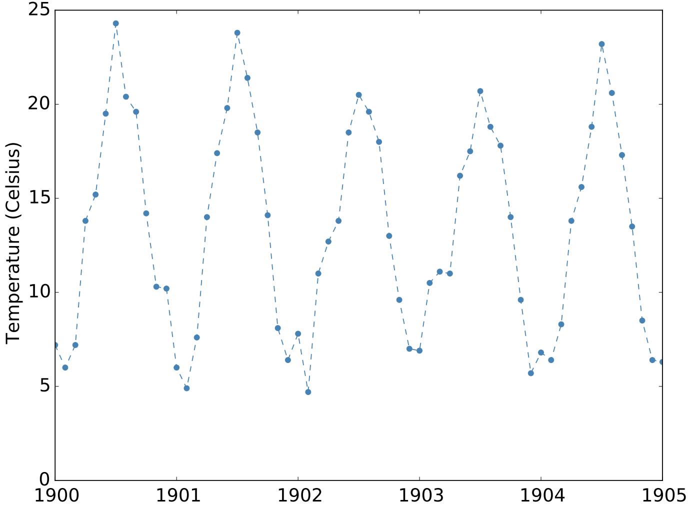
Linear model for GDP

State	GDP	Estimate
North Dakota	52 089	46 241
Alabama	204 861	239 165
Mississippi	107 680	119 005
Arkansas	120 689	145 712
Kansas	153 258	136 756
Georgia	525 360	513 343
Iowa	178 766	158 097
West Virginia	73 374	59 969
Kentucky	197 043	194 829
Tennessee	328 770	345 352

Maximum temperatures in Oxford, UK



Maximum temperatures in Oxford, UK

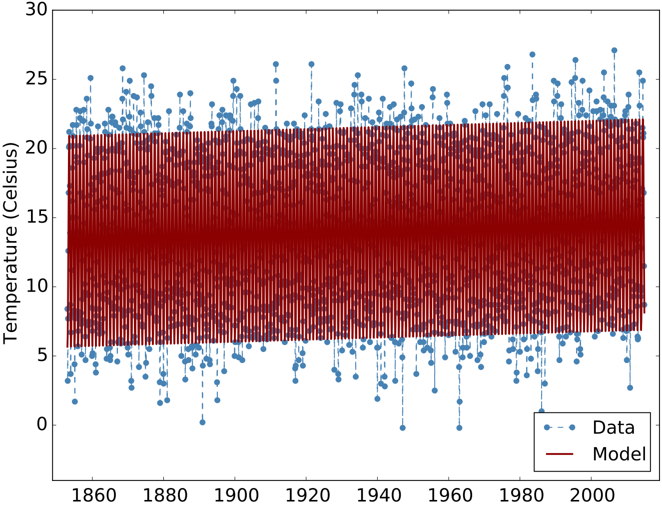


Linear model

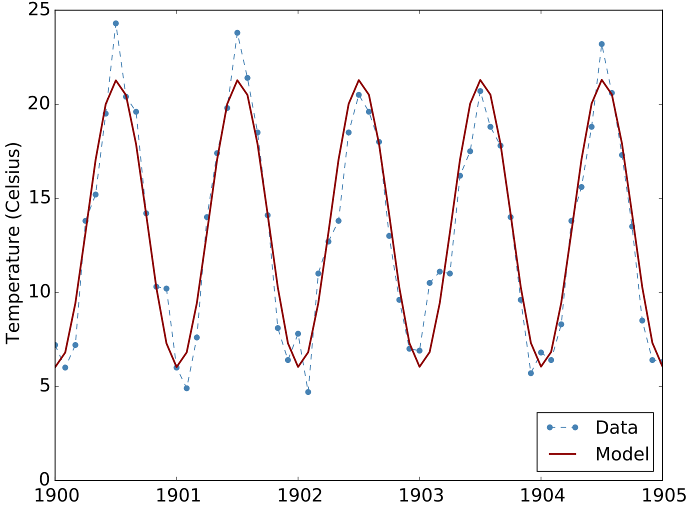
$$\vec{y}_t \approx \vec{\beta}_0 + \vec{\beta}_1 \cos\left(\frac{2\pi t}{12}\right) + \vec{\beta}_2 \sin\left(\frac{2\pi t}{12}\right) + \vec{\beta}_3 \, t$$

$1 \leq t \leq n$ is the time in months ($n = 12 \cdot 150$)

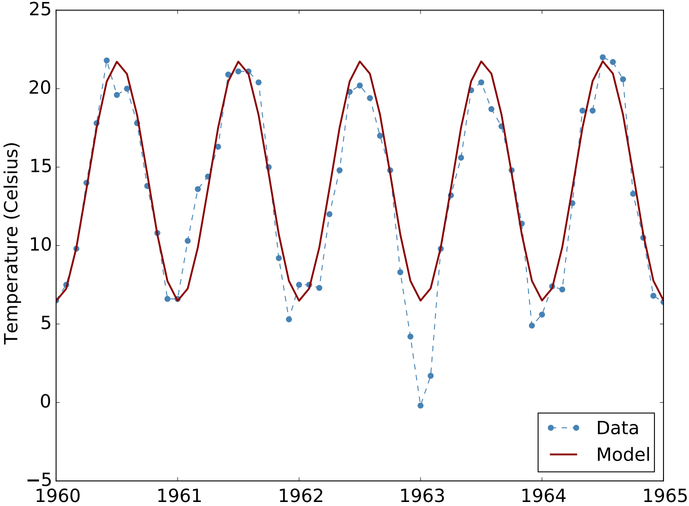
Model fitted by least squares



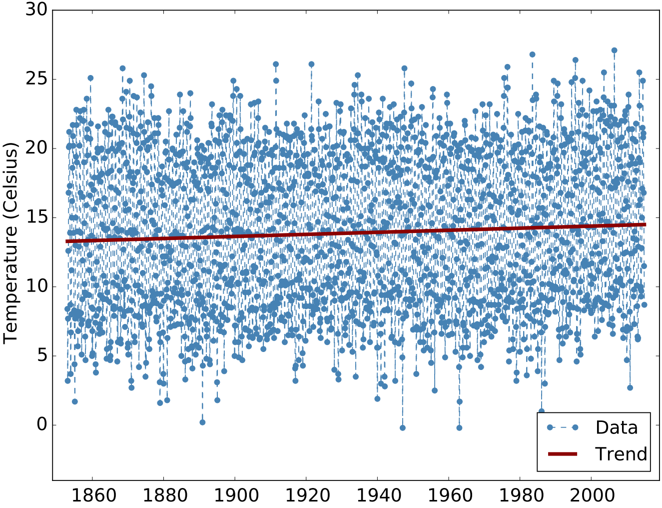
Model fitted by least squares



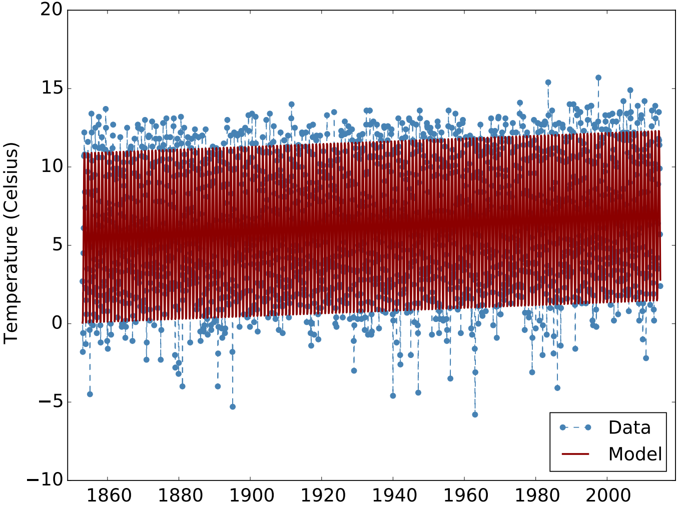
Model fitted by least squares



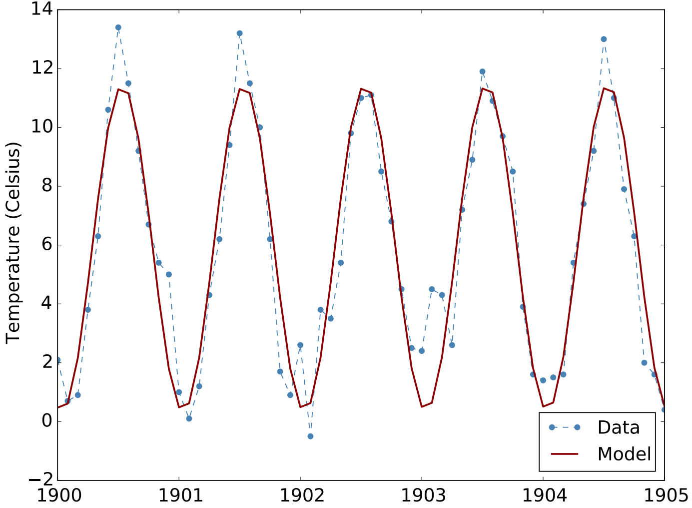
Trend: Increase of 0.75 °C / 100 years (1.35 °F)



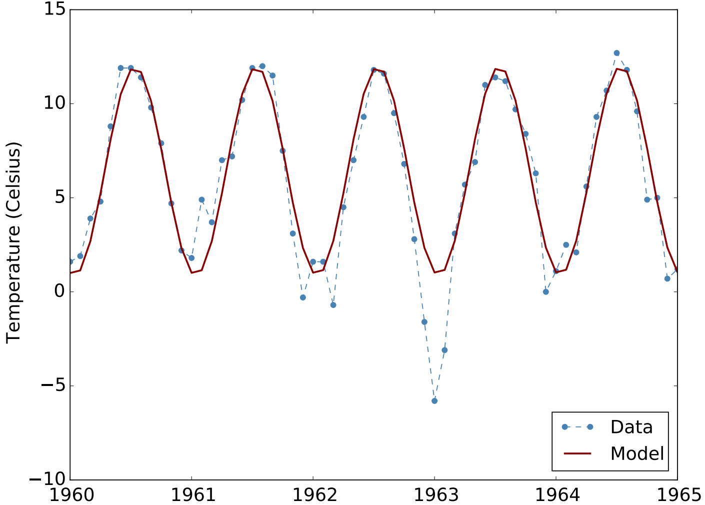
Model for minimum temperatures



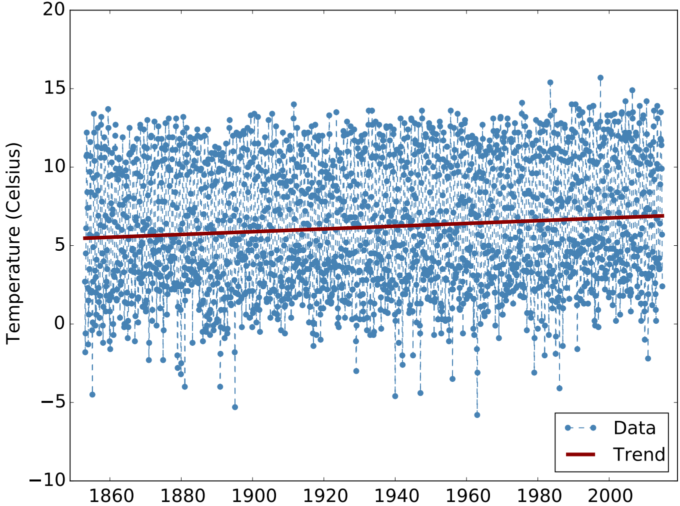
Model for minimum temperatures



Model for minimum temperatures



Trend: Increase of 0.88 °C / 100 years (1.58 °F)

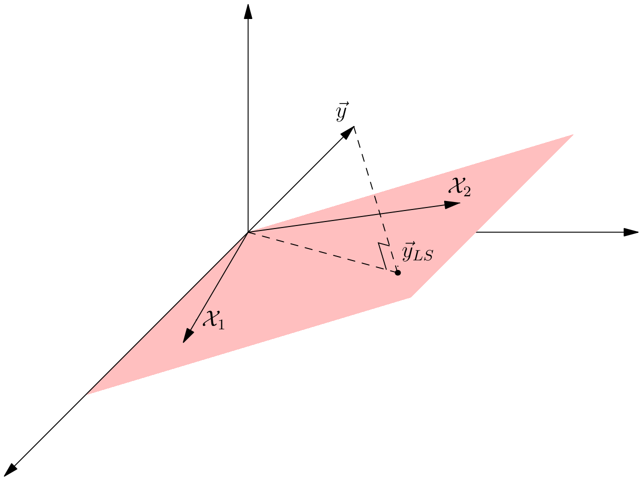


Geometric interpretation

- Any vector $X\vec{\beta}$ is in the span of the columns of X
- The least-squares estimate is the closest vector to $\vec{y}$ that can be represented in this way
- This is the projection of $\vec{y}$ onto the column space of X

$$\begin{aligned} X\vec{\beta}_{\text{LS}} &= USV^T VS^{-1} U^T \vec{y} \\ &= UU^T \vec{y} \end{aligned}$$

Geometric interpretation



Face denoising

We denoise by projecting onto:

- $\mathcal{S}_1$: the span of the 9 images from the same subject
- $\mathcal{S}_2$: the span of the 360 images in the training set

Test error:

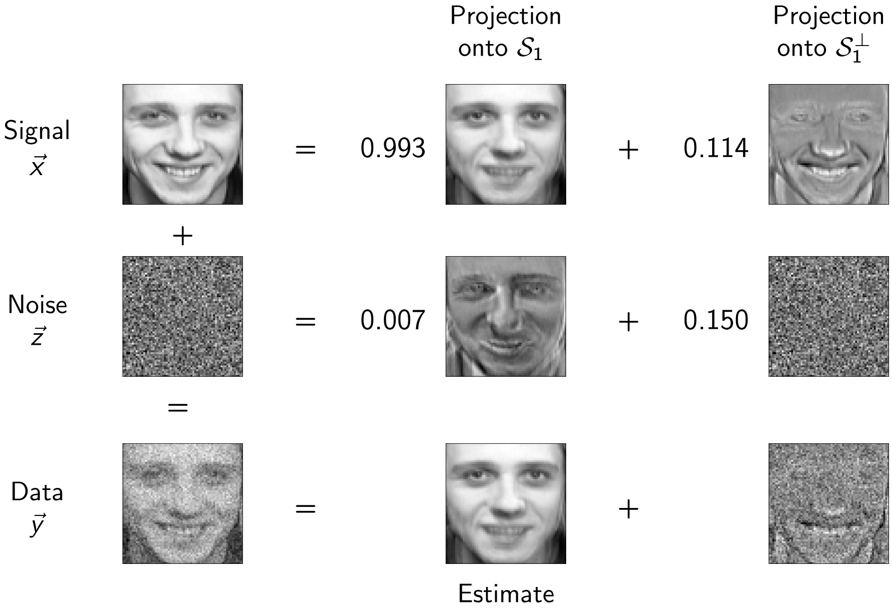
$$\frac{||\vec{x} - \mathcal{P}_{\mathcal{S}_1} \vec{y}||_2}{||\vec{x}||_2} = 0.114$$

$$\frac{||\vec{x} - \mathcal{P}_{\mathcal{S}_2} \vec{y}||_2}{||\vec{x}||_2} = 0.078$$

$\mathcal{S}_1$

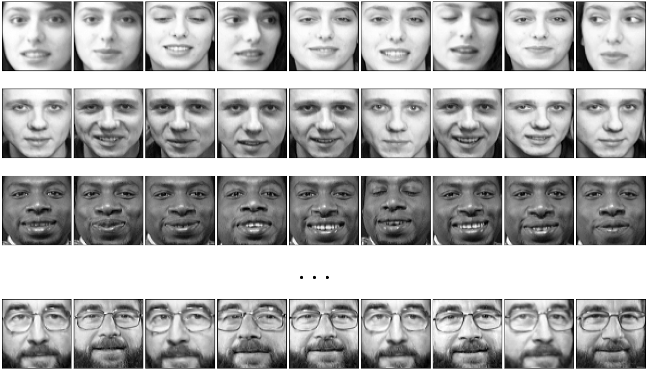
$\mathcal{S}_1 := \mathrm{span}\Bigg($  $\Bigg)$

Denoising via projection onto $\mathcal{S}_1$



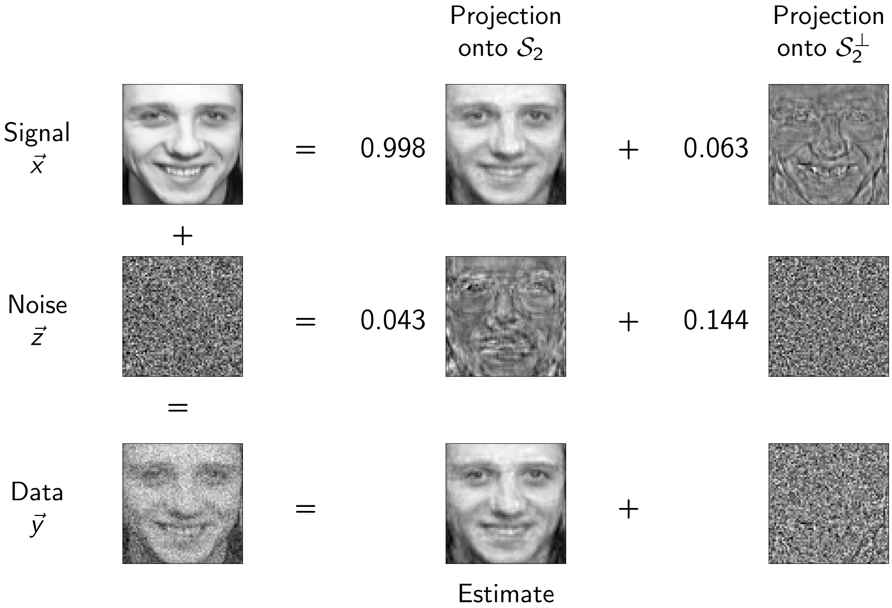
$\mathcal{S}_2$

$\mathcal{S}_2 := \mathrm{span}\Big($



$\Big)$

Denoising via projection onto $\mathcal{S}_2$



$\mathcal{P}_{\mathcal{S}_1}\,\vec{y}$ and $\mathcal{P}_{\mathcal{S}_2}\,\vec{y}$

$\vec{x}$

$\mathcal{P}_{\mathcal{S}_1}\,\vec{y}$

$\mathcal{P}_{\mathcal{S}_2}\,\vec{y}$

Lessons of Face Denoising

What does our intuition learned from Face Denoising tell us about linear regression?

Lessons of Face Denoising

What does our intuition learned from Face Denoising tell us about linear regression?

- More features = larger column space
- Larger column space = captures more of the true image
- Larger column space = captures more of the noise
- Balance between underfitting and overfitting

Linear regression

Least-squares estimation

Geometric interpretation

Probabilistic interpretation

Analysis of least-squares estimate

Noise amplification

Ridge regression

Classification

Motivation

Model data $y_1, \ldots, y_n$ as realizations of a set of random variables $\mathbf{y}_1, \ldots, \mathbf{y}_n$

The joint pdf depends on a vector of parameters $\vec{\beta}$

$$f_{\vec{\beta}}(y_1, \ldots, y_n) := f_{\mathbf{y}_1, \ldots, \mathbf{y}_n}(y_1, \ldots, y_n)$$

is the probability density of $\mathbf{y}_1, \ldots, \mathbf{y}_n$ at the observed data

Idea: Choose $\vec{\beta}$ such that the density is as high as possible

Likelihood

The likelihood is equal to the joint pdf

$$\mathcal{L}_{y_1,\ldots,y_n}\left(\vec{\beta}\right) := f_{\vec{\beta}}\left(y_1, \ldots, y_n\right)$$

interpreted as a **function of the parameters**

The log-likelihood function is the log of the likelihood $\log \mathcal{L}_{y_1,\ldots,y_n}\left(\vec{\beta}\right)$

Maximum-likelihood estimator

The likelihood quantifies how likely the data are according to the model

Maximum-likelihood (ML) estimator :

$$\begin{aligned}
\vec{\beta}_{ML}(y_1, \ldots, y_n) &:= \arg\max_{\vec{\beta}} \mathcal{L}_{y_1,\ldots,y_n}\left(\vec{\beta}\right) \\
&= \arg\max_{\vec{\beta}} \log \mathcal{L}_{y_1,\ldots,y_n}\left(\vec{\beta}\right)
\end{aligned}$$

Maximizing the log-likelihood is equivalent, and often more convenient

Probabilistic interpretation

We model the noise as an iid Gaussian random vector $\vec{\mathbf{z}}$

Entries have zero mean and variance σ^2

The data are a realization of the random vector

$$\vec{\mathbf{y}} := X\vec{\beta} + \vec{\mathbf{z}}$$

$\vec{\mathbf{y}}$ is Gaussian with mean $X\vec{\beta}$ and covariance matrix $\sigma^2 I$

Likelihood

The joint pdf of $\vec{\mathbf{y}}$ is

$$
\begin{aligned}
f_{\vec{\mathbf{y}}}\left(\vec{a}\right) &:= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}\left(\vec{a}[i] - \left(X\vec{\beta}\right)[i]\right)^2\right) \\
&= \frac{1}{\sqrt{(2\pi)^n}\sigma^n} \exp\left(-\frac{1}{2\sigma^2}\left|\left|\vec{a} - X\vec{\beta}\right|\right|_2^2\right)
\end{aligned}
$$

The likelihood is

$$
\mathcal{L}_{\vec{y}}\left(\vec{\beta}\right) = \frac{1}{\sqrt{(2\pi)^n}} \exp\left(-\frac{1}{2}\left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2\right)
$$

Maximum-likelihood estimate

The maximum-likelihood estimate is

$$\begin{aligned}
\vec{\beta}_{\mathrm{ML}} &= \arg\max_{\vec{\beta}} \mathcal{L}_{\vec{y}}\left(\vec{\beta}\right) \\
&= \arg\max_{\vec{\beta}} \log \mathcal{L}_{\vec{y}}\left(\vec{\beta}\right) \\
&= \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 \\
&= \vec{\beta}_{\mathrm{LS}}
\end{aligned}$$

Estimation error

If the data are generated according to the linear model

$$\vec{y} := X\vec{\beta}^* + \vec{z}$$

then

$$\vec{\beta}_{\text{LS}} - \vec{\beta}^*$$

Estimation error

If the data are generated according to the linear model

$$\vec{y} := X\vec{\beta}^* + \vec{z}$$

then

$$\vec{\beta}_{\text{LS}} - \vec{\beta}^* = \left(X^T X\right)^{-1} X^T \left(X\vec{\beta}^* + \vec{z}\right) - \vec{\beta}^*$$

Estimation error

If the data are generated according to the linear model

$$\vec{y} := X\vec{\beta}^* + \vec{z}$$

then

$$\begin{aligned}
\vec{\beta}_{\mathsf{LS}} - \vec{\beta}^* &= \left(X^T X\right)^{-1} X^T \left(X\vec{\beta}^* + \vec{z}\right) - \vec{\beta}^* \\
&= \left(X^T X\right)^{-1} X^T \vec{z}
\end{aligned}$$

as long as X is full rank

LS estimator is unbiased

Assume noise $\mathbf{z}$ is random and has zero mean, then

$$\mathrm{E}\left(\vec{\boldsymbol{\beta}}_{\mathsf{LS}} - \vec{\beta}^*\right)$$

LS estimator is unbiased

Assume noise **z** is random and has zero mean, then

$$\mathrm{E}\left(\vec{\boldsymbol{\beta}}_{\mathsf{LS}} - \vec{\beta}^{*}\right) = \left(X^T X\right)^{-1} X^T \mathrm{E}\left(\vec{\mathbf{z}}\right)$$

LS estimator is unbiased

Assume noise **z** is random and has zero mean, then

$$\begin{aligned} \mathrm{E}\left(\vec{\boldsymbol{\beta}}_{\mathsf{LS}} - \vec{\beta}^*\right) &= \left(X^T X\right)^{-1} X^T \mathrm{E}\left(\vec{\mathbf{z}}\right) \\ &= 0 \end{aligned}$$

The estimate is unbiased: its mean equals $\vec{\beta}^*$

Least-squares error

If the data are generated according to the linear model

$$\vec{y} := X\vec{\beta}^{*} + \vec{z}$$

then

$$\frac{\left|\left|\vec{z}\right|\right|_2}{\sigma_1} \leq \left|\left|\vec{\beta}_{\text{LS}} - \vec{\beta}^{*}\right|\right|_2 \leq \frac{\left|\left|\vec{z}\right|\right|_2}{\sigma_p}$$

σ_1 and σ_p are the largest and smallest singular values of X

Least-squares error: Proof

The error is given by

$$\vec{\beta}_{\text{LS}} - \vec{\beta}^* = (X^T X)^{-1} X^T \vec{z}.$$

How can we bound $\|(X^T X)^{-1} X^T \vec{z}\|_2$?

Singular values

The singular values of a matrix $A \in \mathbb{R}^{n \times p}$ of rank p satisfy

$$\sigma_1 = \max_{\left\{||\vec{x}||_2=1 \mid \vec{x}\in\mathbb{R}^n\right\}} ||A\vec{x}||_2$$

$$\sigma_p = \min_{\left\{||\vec{x}||_2=1 \mid \vec{x}\in\mathbb{R}^n\right\}} ||A\vec{x}||_2$$

Least-squares error

$$\vec{\beta}_{\text{LS}} - \vec{\beta}^{*} = VS^{-1}U^{T}\vec{z}$$

The smallest and largest singular values of $VS^{-1}U$ are $1/\sigma_1$ and $1/\sigma_p$, so

$$\frac{||\vec{z}||_2}{\sigma_1} \leq \left|\left|VS^{-1}U^{T}\vec{z}\right|\right|_2 \leq \frac{||\vec{z}||_2}{\sigma_p}$$

Experiment

X_{train}, X_{test}, $\vec{z}_{\text{train}}$ and β^* are sampled iid from a standard Gaussian
Data has 50 features

$$\vec{y}_{\text{train}} = X_{\text{train}}\,\vec{\beta}^* + \vec{z}_{\text{train}}$$

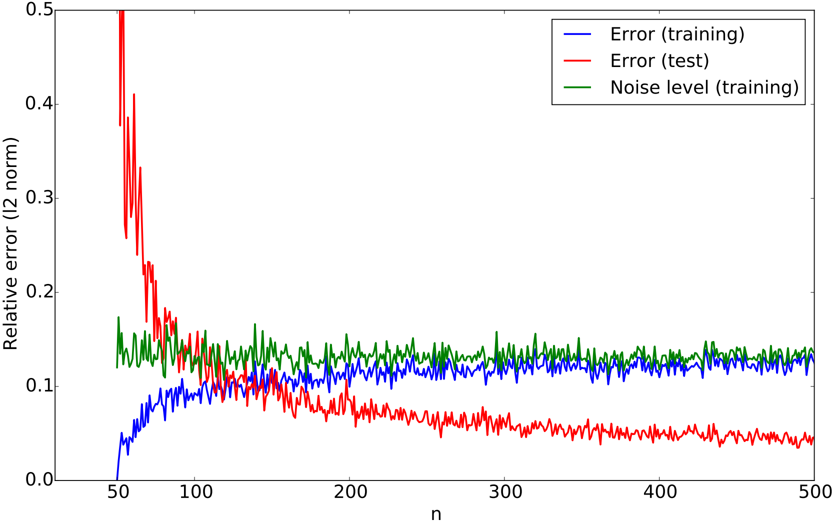
$$\vec{y}_{\text{test}} = X_{\text{test}}\,\vec{\beta}^* \qquad \text{(No Test Noise)}$$

We use $\vec{y}_{\text{train}}$ and X_{train} to compute $\vec{\beta}_{\text{LS}}$

$$\text{error}_{\text{train}} = \frac{\left|\left|X_{\text{train}}\,\vec{\beta}_{\text{LS}} - \vec{y}_{\text{train}}\right|\right|_2}{\left|\left|\vec{y}_{\text{train}}\right|\right|_2}$$

$$\text{error}_{\text{test}} = \frac{\left|\left|X_{\text{test}}\,\vec{\beta}_{\text{LS}} - \vec{y}_{\text{test}}\right|\right|_2}{\left|\left|\vec{y}_{\text{test}}\right|\right|_2}$$

Experiment



Error (training)
Error (test)
Noise level (training)
Relative error (l2 norm)
n
0.0
0.1
0.2
0.3
0.4
0.5
50
100
200
300
400
500

Experiment Questions

1. Can we approximate the relative noise level $\|\vec{z}\|_2/\|\vec{y}\|_2$?
2. Why does the training error start at 0?
3. Why does the relative training error converge to the noise level?
4. Why does the relative test error converge to zero?

Experiment Questions

1. Can we approximate the relative noise level $\|\vec{z}\|_2/\|\vec{y}\|_2$?
   $\|\vec{\beta}^*\|_2 \approx \sqrt{50}$, $\|X_{\text{train}}\vec{\beta}^*\|_2 \approx \sqrt{50n}$, $\|\vec{z}_{\text{train}}\|_2 \approx \sqrt{n}$, $\frac{1}{\sqrt{51}} \approx 0.140$
2. Why does the training error start at 0?
3. Why does the relative training error converge to the noise level?
4. Why does the relative test error converge to zero?

Experiment Questions

1. Can we approximate the relative noise level $\|\vec{z}\|_2/\|\vec{y}\|_2$?
   $\|\vec{\beta}^*\|_2 \approx \sqrt{50}$, $\|X_{\text{train}}\vec{\beta}^*\|_2 \approx \sqrt{50n}$, $\|\vec{z}_{\text{train}}\|_2 \approx \sqrt{n}$, $\frac{1}{\sqrt{51}} \approx 0.140$
2. Why does the training error start at 0?
   X is square and invertible
3. Why does the relative training error converge to the noise level?
4. Why does the relative test error converge to zero?

Experiment Questions

1. Can we approximate the relative noise level $\|\vec{z}\|_2/\|\vec{y}\|_2$?

   $\|\vec{\beta}^*\|_2 \approx \sqrt{50}$, $\|X_{\text{train}}\vec{\beta}^*\|_2 \approx \sqrt{50n}$, $\|\vec{z}_{\text{train}}\|_2 \approx \sqrt{n}$, $\frac{1}{\sqrt{51}} \approx 0.140$
2. Why does the training error start at 0?

   X is square and invertible
3. Why does the relative training error converge to the noise level?

   $\|X_{\text{train}}\vec{\beta}_{\text{LS}} - \vec{y}_{\text{train}}\|_2 = \|X_{\text{train}}(\vec{\beta}_{\text{LS}} - \vec{\beta}^*) - \vec{z}_{\text{train}}\|_2$ and $\vec{\beta}_{\text{LS}} \to \vec{\beta}$
4. Why does the relative test error converge to zero?

Experiment Questions

1. Can we approximate the relative noise level $\|\vec{z}\|_2/\|\vec{y}\|_2$?
   $\|\vec{\beta}^*\|_2 \approx \sqrt{50}$, $\|X_{\text{train}}\vec{\beta}^*\|_2 \approx \sqrt{50n}$, $\|\vec{z}_{\text{train}}\|_2 \approx \sqrt{n}$, $\frac{1}{\sqrt{51}} \approx 0.140$
2. Why does the training error start at 0?
   X is square and invertible
3. Why does the relative training error converge to the noise level?
   $\|X_{\text{train}}\vec{\beta}_{\text{LS}} - \vec{y}_{\text{train}}\|_2 = \|X_{\text{train}}(\vec{\beta}_{\text{LS}} - \vec{\beta}^*) - \vec{z}_{\text{train}}\|_2$ and $\vec{\beta}_{\text{LS}} \to \vec{\beta}$
4. Why does the relative test error converge to zero?
   We assumed no test noise, and $\vec{\beta}_{\text{LS}} \to \vec{\beta}^*$

Non-asymptotic bound

Let

$$\vec{y} := X\vec{\beta}^* + \vec{z},$$

where the entries of X and $\vec{z}$ are iid standard Gaussians

The least-squares estimate satisfies

$$\sqrt{\frac{(1-\epsilon)}{(1+\epsilon)}}\sqrt{\frac{p}{n}} \leq \left|\left|\vec{\beta}_{\text{LS}} - \vec{\beta}^*\right|\right|_2 \leq \sqrt{\frac{(1+\epsilon)}{(1-\epsilon)}}\sqrt{\frac{p}{n}}$$

with probability at least $1 - 1/p - 2\exp\left(-p\epsilon^2/8\right)$ as long as $n \geq 64p\log(12/\epsilon)/\epsilon^2$

Proof

$$\frac{\left|\left|\mathbf{U}^T\vec{z}\right|\right|_2}{\boldsymbol{\sigma_1}} \leq \left|\left|\mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T\vec{z}\right|\right|_2 \leq \frac{\left|\left|\mathbf{U}^T\vec{z}\right|\right|_2}{\boldsymbol{\sigma_p}}$$

Projection onto a fixed subspace

Let $\mathcal{S}$ be a k-dimensional subspace of $\mathbb{R}^n$ and $\vec{z} \in \mathbb{R}^n$ a vector of iid standard Gaussian noise

For any $\epsilon > 0$

$$P\left(k\left(1-\epsilon\right) < ||\mathcal{P}_{\mathcal{S}}\,\vec{z}||_2^2 < k\left(1+\epsilon\right)\right) \geq 1 - 2\exp\left(-\frac{k\epsilon^2}{8}\right)$$

Projection onto a fixed subspace

Let $\mathcal{S}$ be a k-dimensional subspace of $\mathbb{R}^n$ and $\vec{\mathbf{z}} \in \mathbb{R}^n$ a vector of iid standard Gaussian noise

For any $\epsilon > 0$

$$P\left(k\left(1-\epsilon\right) < \left|\left|\mathcal{P}_{\mathcal{S}}\,\vec{z}\right|\right|_2^2 < k\left(1+\epsilon\right)\right) \geq 1 - 2\exp\left(-\frac{k\epsilon^2}{8}\right)$$

Consequence: With probability $1 - 2\exp\left(-p\epsilon^2/8\right)$

$$\left(1-\epsilon\right)p \leq \left|\left|\mathbf{U}^T\vec{\mathbf{z}}\right|\right|_2^2 \leq \left(1+\epsilon\right)p$$

Singular values of a Gaussian matrix

Let $\mathbf{A}$ be a $n \times k$ matrix with iid standard Gaussian entries such that $n > k$

For any fixed $\epsilon > 0$, the singular values of $\mathbf{A}$ satisfy

$$\sqrt{n(1-\epsilon)} \leq \boldsymbol{\sigma}_{\mathbf{k}} \leq \boldsymbol{\sigma}_{\mathbf{1}} \leq \sqrt{n(1+\epsilon)}$$

with probability at least $1 - 1/k$ as long as

$$n > \frac{64k}{\epsilon^2} \log \frac{12}{\epsilon}$$

Proof

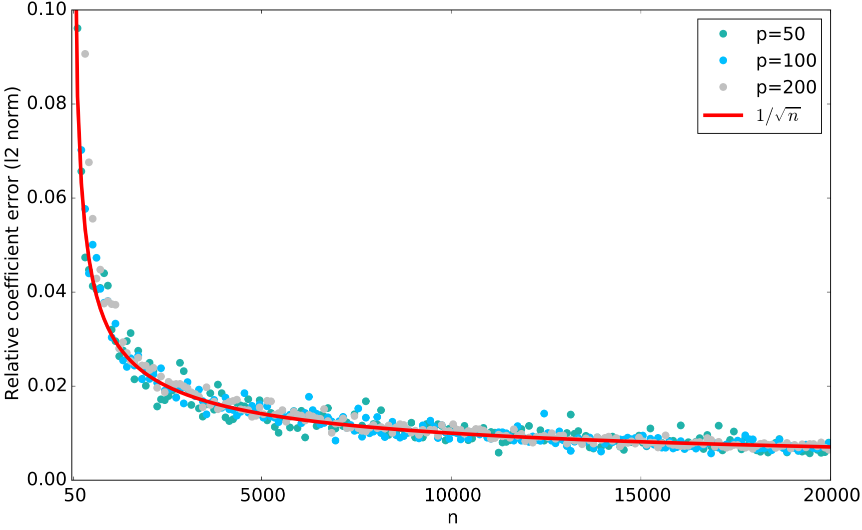
With probability $1 - 1/p$

$$\sqrt{n(1-\epsilon)} \leq \boldsymbol{\sigma_p} \leq \boldsymbol{\sigma_1} \leq \sqrt{n(1+\epsilon)}$$

as long as $n \geq 64p \log(12/\epsilon)/\epsilon^2$

Experiment: $\left\|\vec{\beta}\right\|_2 \approx p$

Plot of $\frac{\|\vec{\beta}^{*} - \vec{\beta}_{\text{LS}}\|_2}{\|\vec{\beta}^{*}\|_2}$



Linear regression

Least-squares estimation

Geometric interpretation

Probabilistic interpretation

Analysis of least-squares estimate

Noise amplification

Ridge regression

Classification

Condition number

The condition number of $A \in \mathbb{R}^{n \times p}$, $n \geq p$, is the ratio σ_1/σ_p of its largest and smallest singular values

A matrix is ill conditioned if its condition is large (almost rank defficient)

Noise amplification

Let

$$\vec{\mathbf{y}} := X\vec{\beta}^* + \vec{\mathbf{z}},$$

where $\vec{\mathbf{z}}$ is iid standard Gaussian

With probability at least $1 - 2\exp\left(-\epsilon^2/8\right)$

$$\left\|\vec{\boldsymbol{\beta}}_{\text{LS}} - \vec{\beta}^*\right\|_2 \geq \frac{\sqrt{1-\epsilon}}{\sigma_p}$$

where σ_p is the smallest singular value of X

Proof

$$\left|\left|\vec{\boldsymbol{\beta}}_{\mathsf{LS}} - \vec{\beta}^*\right|\right|_2^2$$

Proof

$$\left|\left|\vec{\boldsymbol{\beta}}_{\text{LS}} - \vec{\beta}^{*}\right|\right|_2^2 = \left|\left|VS^{-1}U^T\vec{\mathbf{z}}\right|\right|_2^2$$

Proof

$$\left|\left|\vec{\boldsymbol{\beta}}_{\text{LS}} - \vec{\beta}^*\right|\right|_2^2 = \left|\left|VS^{-1}U^T\vec{\mathbf{z}}\right|\right|_2^2$$

$$= \left|\left|S^{-1}U^T\vec{\mathbf{z}}\right|\right|_2^2 \qquad V \text{ is orthogonal}$$

Proof

$$
\begin{aligned}
\left|\left|\vec{\boldsymbol{\beta}}_{\text{LS}} - \vec{\beta}^*\right|\right|_2^2 &= \left|\left|VS^{-1}U^T\vec{\mathbf{z}}\right|\right|_2^2 \\
&= \left|\left|S^{-1}U^T\vec{\mathbf{z}}\right|\right|_2^2 \qquad V \text{ is orthogonal} \\
&= \sum_i^p \frac{\left(\vec{u}_i^T\vec{\mathbf{z}}\right)^2}{\sigma_i^2}
\end{aligned}
$$

Proof

$$
\begin{aligned}
\left|\left|\vec{\boldsymbol{\beta}}_{\text{LS}} - \vec{\beta}^*\right|\right|_2^2 &= \left|\left|VS^{-1}U^T\vec{\mathbf{z}}\right|\right|_2^2 \\
&= \left|\left|S^{-1}U^T\vec{\mathbf{z}}\right|\right|_2^2 \qquad V \text{ is orthogonal} \\
&= \sum_i^p \frac{\left(\vec{u}_i^T\vec{\mathbf{z}}\right)^2}{\sigma_i^2} \\
&\geq \frac{\left(\vec{u}_p^T\vec{\mathbf{z}}\right)^2}{\sigma_p^2}
\end{aligned}
$$

Projection onto a fixed subspace

Let $\mathcal{S}$ be a k-dimensional subspace of $\mathbb{R}^n$ and $\vec{z} \in \mathbb{R}^n$ a vector of iid standard Gaussian noise

For any $\epsilon > 0$

$$P\left(k\left(1-\epsilon\right) < ||\mathcal{P}_{\mathcal{S}}\,\vec{z}||_2^2 < k\left(1+\epsilon\right)\right) \geq 1 - 2\exp\left(-\frac{k\epsilon^2}{8}\right)$$

Projection onto a fixed subspace

Let $\mathcal{S}$ be a k-dimensional subspace of $\mathbb{R}^n$ and $\vec{\mathbf{z}} \in \mathbb{R}^n$ a vector of iid standard Gaussian noise

For any $\epsilon > 0$

$$P\left(k\left(1-\epsilon\right) < \left|\left|\mathcal{P}_{\mathcal{S}}\,\vec{z}\right|\right|_2^2 < k\left(1+\epsilon\right)\right) \geq 1 - 2\exp\left(-\frac{k\epsilon^2}{8}\right)$$

Consequence: With probability $1 - 2\exp\left(-\epsilon^2/8\right)$

$$\left(\vec{u}_p^{\,T}\vec{\mathbf{z}}\right)^2 \geq \left(1-\epsilon\right)$$

Example

Let

$$\vec{y} := X\vec{\beta}^* + \vec{z}$$

where

$$X := \begin{bmatrix} 0.212 & -0.099 \\ 0.605 & -0.298 \\ -0.213 & 0.113 \\ 0.589 & -0.285 \\ 0.016 & 0.006 \\ 0.059 & 0.032 \end{bmatrix}, \qquad \vec{\beta}^* := \begin{bmatrix} 0.471 \\ -1.191 \end{bmatrix}, \qquad \vec{z} := \begin{bmatrix} 0.066 \\ -0.077 \\ -0.010 \\ -0.033 \\ 0.010 \\ 0.028 \end{bmatrix}$$

$||z||_2 = 0.11$

Example

Condition number = 100

$$X = USV^T = \begin{bmatrix} -0.234 & 0.427 \\ -0.674 & -0.202 \\ 0.241 & 0.744 \\ -0.654 & 0.350 \\ 0.017 & -0.189 \\ 0.067 & 0.257 \end{bmatrix} \begin{bmatrix} 1.00 & 0 \\ 0 & 0.01 \end{bmatrix} \begin{bmatrix} -0.898 & 0.440 \\ 0.440 & 0.898 \end{bmatrix}$$

Example

$$\vec{\beta}_{\mathsf{LS}} - \vec{\beta}^{*}$$

Example

$$\vec{\beta}_{\text{LS}} - \vec{\beta}^{*} = VS^{-1}U^{T}\vec{z}$$

Example

$$\begin{aligned}\vec{\beta}_{\text{LS}} - \vec{\beta}^{*} &= VS^{-1}U^{T}\vec{z} \\ &= V\begin{bmatrix}1.00 & 0 \\ 0 & 100.00\end{bmatrix}U^{T}\vec{z}\end{aligned}$$

Example

$$\begin{aligned}
\vec{\beta}_{\text{LS}} - \vec{\beta}^* &= VS^{-1}U^T\vec{z} \\
&= V\begin{bmatrix} 1.00 & 0 \\ 0 & 100.00 \end{bmatrix} U^T\vec{z} \\
&= V\begin{bmatrix} 0.058 \\ 3.004 \end{bmatrix}
\end{aligned}$$

Example

$$
\begin{aligned}
\vec{\beta}_{\text{LS}} - \vec{\beta}^* &= VS^{-1}U^T\vec{z} \\
&= V \begin{bmatrix} 1.00 & 0 \\ 0 & 100.00 \end{bmatrix} U^T\vec{z} \\
&= V \begin{bmatrix} 0.058 \\ 3.004 \end{bmatrix} \\
&= \begin{bmatrix} 1.270 \\ 2.723 \end{bmatrix}
\end{aligned}
$$

Example

$$\vec{\beta}_{\mathrm{LS}} - \vec{\beta}^* = VS^{-1}U^T\vec{z}$$
$$= V \begin{bmatrix} 1.00 & 0 \\ 0 & 100.00 \end{bmatrix} U^T\vec{z}$$
$$= V \begin{bmatrix} 0.058 \\ 3.004 \end{bmatrix}$$
$$= \begin{bmatrix} 1.270 \\ 2.723 \end{bmatrix}$$

so that

$$\frac{\left|\left|\vec{\beta}_{\mathrm{LS}} - \vec{\beta}^*\right|\right|_2}{||\vec{z}||_2} = 27.00$$

Multicollinearity

Feature matrix is ill conditioned if any subset of columns is close to being linearly dependent (there is a vector *almost* in the null space)

This occurs if features are highly correlated

For any $X \in \mathbb{R}^{n \times p}$, with normalized columns, if X_i and X_j, $i \neq j$, satisfy

$$\langle X_i, X_j \rangle^2 \geq 1 - \epsilon^2$$

then the smallest singular value $\sigma_p \leq \epsilon$

Multicollinearity

Feature matrix is ill conditioned if any subset of columns is close to being linearly dependent (there is a vector *almost* in the null space)

This occurs if features are highly correlated

For any $X \in \mathbb{R}^{n \times p}$, with normalized columns, if X_i and X_j, $i \neq j$, satisfy

$$\langle X_i, X_j \rangle^2 \geq 1 - \epsilon^2$$

then the smallest singular value $\sigma_p \leq \epsilon$

Proof Idea: Consider $\|X(\vec{e}_i - \vec{e}_j)\|_2$.

Motivation

Avoid noise amplification due to multicollinearity

Problem: Noise amplification blows up coefficients

Solution: Penalize large-norm solutions when fitting the model

Adding a penalty term promoting a particular structure is called regularization

Ridge regression

For a fixed regularization parameter $\lambda > 0$

$$\vec{\beta}_{\text{ridge}} := \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2$$

Ridge regression

For a fixed regularization parameter $\lambda > 0$

$$\vec{\beta}_{\text{ridge}} := \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2$$

$$= \left(X^T X + \lambda I\right)^{-1} X^T \vec{y}$$

Ridge regression

For a fixed regularization parameter $\lambda > 0$

$$\vec{\beta}_{\text{ridge}} := \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2$$

$$= \left(X^T X + \lambda I\right)^{-1} X^T \vec{y}$$

λI increases the singular values of $X^T X$

Ridge regression

For a fixed regularization parameter $\lambda > 0$

$$\begin{aligned}\vec{\beta}_{\text{ridge}} &:= \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2 \\ &= \left(X^T X + \lambda I\right)^{-1} X^T \vec{y}\end{aligned}$$

λI increases the singular values of $X^T X$

When $\lambda \to 0$ then $\vec{\beta}_{\text{ridge}} \to \vec{\beta}_{\text{LS}}$

Ridge regression

For a fixed regularization parameter $\lambda > 0$

$$\vec{\beta}_{\text{ridge}} := \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2$$

$$= \left(X^T X + \lambda I\right)^{-1} X^T \vec{y}$$

λI increases the singular values of $X^T X$

When $\lambda \to 0$ then $\vec{\beta}_{\text{ridge}} \to \vec{\beta}_{\text{LS}}$

When $\lambda \to \infty$ then $\vec{\beta}_{\text{ridge}} \to 0$

Proof

$\vec{\beta}_{\text{ridge}}$ is the solution to a modified least-squares problem

$$\vec{\beta}_{\text{ridge}} = \arg\min_{\vec{\beta}} \left|\left| \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix} - \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \vec{\beta} \right|\right|_2^2$$

Proof

$\vec{\beta}_{\text{ridge}}$ is the solution to a modified least-squares problem

$$\begin{aligned}\vec{\beta}_{\text{ridge}} &= \arg\min_{\vec{\beta}} \left|\left| \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix} - \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \vec{\beta} \right|\right|_2^2 \\ &= \left(\begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix}^T \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \right)^{-1} \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix}^T \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix}\end{aligned}$$

Proof

$\vec{\beta}_{\text{ridge}}$ is the solution to a modified least-squares problem

$$\begin{aligned}
\vec{\beta}_{\text{ridge}} &= \arg\min_{\vec{\beta}} \left\| \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix} - \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \vec{\beta} \right\|_2^2 \\
&= \left(\begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix}^T \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \right)^{-1} \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix}^T \begin{bmatrix} \vec{y} \\ 0 \end{bmatrix} \\
&= \left(X^T X + \lambda I \right)^{-1} X^T \vec{y}
\end{aligned}$$

Modified projection

$$\vec{y}_{\text{ridge}} := X\vec{\beta}_{\text{ridge}}$$

Modified projection

$$\begin{aligned}\vec{y}_{\text{ridge}} &:= X\vec{\beta}_{\text{ridge}} \\ &= X\left(X^T X + \lambda I\right)^{-1} X^T \vec{y}\end{aligned}$$

Modified projection

$$
\begin{aligned}
\vec{y}_{\text{ridge}} &:= X\vec{\beta}_{\text{ridge}} \\
&= X\left(X^T X + \lambda I\right)^{-1} X^T \vec{y} \\
&= USV^T\left(VS^2V^T + \lambda VV^T\right)^{-1} VSU^T \vec{y}
\end{aligned}
$$

Modified projection

$$
\begin{aligned}
\vec{y}_{\text{ridge}} &:= X\vec{\beta}_{\text{ridge}} \\
&= X\left(X^T X + \lambda I\right)^{-1} X^T \vec{y} \\
&= USV^T\left(VS^2V^T + \lambda VV^T\right)^{-1} VSU^T \vec{y} \\
&= USV^T V\left(S^2 + \lambda I\right)^{-1} V^T VSU^T \vec{y}
\end{aligned}
$$

Modified projection

$$
\begin{aligned}
\vec{y}_{\text{ridge}} &:= X\vec{\beta}_{\text{ridge}} \\
&= X\left(X^T X + \lambda I\right)^{-1} X^T \vec{y} \\
&= USV^T \left(VS^2V^T + \lambda VV^T\right)^{-1} VSU^T \vec{y} \\
&= USV^T V \left(S^2 + \lambda I\right)^{-1} V^T VSU^T \vec{y} \\
&= US \left(S^2 + \lambda I\right)^{-1} SU^T \vec{y}
\end{aligned}
$$

Modified projection

$$
\begin{aligned}
\vec{y}_{\text{ridge}} &:= X\vec{\beta}_{\text{ridge}} \\
&= X\left(X^TX+\lambda I\right)^{-1}X^T\vec{y} \\
&= USV^T\left(VS^2V^T+\lambda VV^T\right)^{-1}VSU^T\vec{y} \\
&= USV^TV\left(S^2+\lambda I\right)^{-1}V^TVSU^T\vec{y} \\
&= US\left(S^2+\lambda I\right)^{-1}SU^T\vec{y} \\
&= \sum_{i=1}^{p}\frac{\sigma_i^2}{\sigma_i^2+\lambda}\langle\vec{y},\vec{u}_i\rangle\vec{u}_i
\end{aligned}
$$

Component of data in direction of $\vec{u}_i$ is shrunk by $\frac{\sigma_i^2}{\sigma_i^2+\lambda}$

Modified projection: Relation to PCA

Component of data in direction of $\vec{u}_i$ is shrunk by $\frac{\sigma_i^2}{\sigma_i^2+\lambda}$

Instead of orthogonally projecting on to the column space of X as in standard regression, we shrink and project

Which directions are shrunk the most?

Modified projection: Relation to PCA

Component of data in direction of $\vec{u}_i$ is shrunk by $\frac{\sigma_i^2}{\sigma_i^2+\lambda}$

Instead of orthogonally projecting on to the column space of X as in standard regression, we shrink and project

Which directions are shrunk the most?

The directions in the data with smallest variance

Modified projection: Relation to PCA

Component of data in direction of $\vec{u}_i$ is shrunk by $\frac{\sigma_i^2}{\sigma_i^2+\lambda}$

Instead of orthogonally projecting on to the column space of X as in standard regression, we shrink and project

Which directions are shrunk the most?

The directions in the data with smallest variance

In PCA, we delete the directions with smallest variance (i.e., shrink them to zero)

Can think of Ridge Regression as a continuous variant of performing regression on principal components

Ridge-regression estimate

If $\vec{y} := X\vec{\beta}^* + \vec{z}$

$$\vec{\beta}_{\text{ridge}} = V \begin{bmatrix} \frac{\sigma_1^2}{\sigma_1^2+\lambda} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2^2}{\sigma_2^2+\lambda} & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \frac{\sigma_p^2}{\sigma_p^2+\lambda} \end{bmatrix} V^T \vec{\beta}^* + V \begin{bmatrix} \frac{\sigma_1}{\sigma_1^2+\lambda} & 0 & \cdots & 0 \\ 0 & \frac{\sigma_2}{\sigma_2^2+\lambda} & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \frac{\sigma_p}{\sigma_p^2+\lambda} \end{bmatrix} U^T \vec{z}$$

where $X = USV^T$ and $\sigma_1, \ldots, \sigma_p$ are the singular values

For comparison,

$$\vec{\beta}_{\text{LS}} = \vec{\beta}^* + VS^{-1}U^T\vec{z}$$

Bias-variance tradeoff

Error $\vec{\beta}_{\text{ridge}} - \vec{\beta}^*$ can be divided into two terms:

Bias (depends on $\vec{\beta}^*$) and variance (depends on $\vec{z}$)

The bias equals

$$\mathrm{E}\left(\vec{\boldsymbol{\beta}}_{\text{ridge}} - \vec{\beta}^*\right) = -V \begin{bmatrix} \frac{\lambda}{\sigma_1^2+\lambda} & 0 & \cdots & 0 \\ 0 & \frac{\lambda}{\sigma_2^2+\lambda} & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & \frac{\lambda}{\sigma_p^2+\lambda} \end{bmatrix} V^T \vec{\beta}^*$$

Larger λ increases bias, but dampens noise (decreases variance)

Example

Let

$$\vec{y} := X\vec{\beta}^* + \vec{z}$$

where

$$X := \begin{bmatrix} 0.212 & -0.099 \\ 0.605 & -0.298 \\ -0.213 & 0.113 \\ 0.589 & -0.285 \\ 0.016 & 0.006 \\ 0.059 & 0.032 \end{bmatrix}, \qquad \vec{\beta}^* := \begin{bmatrix} 0.471 \\ -1.191 \end{bmatrix}, \qquad \vec{z} := \begin{bmatrix} 0.066 \\ -0.077 \\ -0.010 \\ -0.033 \\ 0.010 \\ 0.028 \end{bmatrix}$$

$||z||_2 = 0.11$

Example

$$\vec{\beta}_{\text{ridge}} - \vec{\beta}^* = V \begin{bmatrix} \frac{\lambda}{1+\lambda} & 0 \\ 0 & \frac{\lambda}{0.01^2+\lambda} \end{bmatrix} V^T \vec{\beta}^* - V \begin{bmatrix} \frac{1}{1+\lambda} & 0 \\ 0 & \frac{0.01}{0.01^2+\lambda} \end{bmatrix} U^T \vec{z}$$

Example

Setting $\lambda = 0.01$

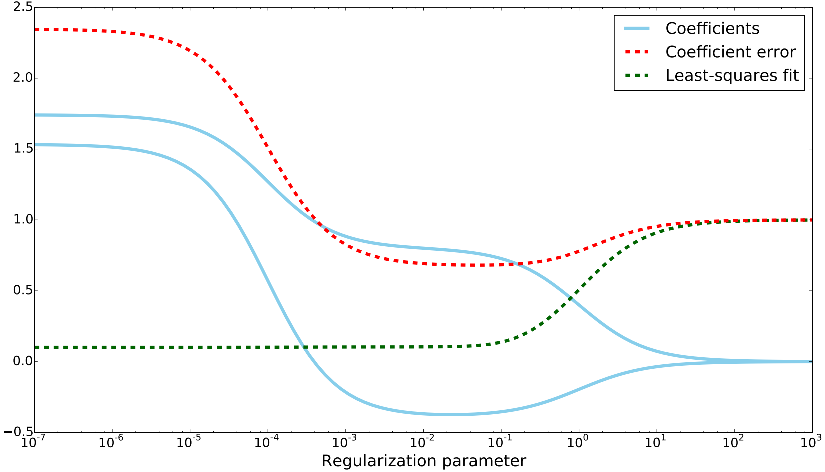
$$\begin{aligned}
\vec{\beta}_{\text{ridge}} - \vec{\beta}^* &= V \begin{bmatrix} \frac{\lambda}{1+\lambda} & 0 \\ 0 & \frac{\lambda}{0.01^2+\lambda} \end{bmatrix} V^T \vec{\beta}^* - V \begin{bmatrix} \frac{1}{1+\lambda} & 0 \\ 0 & \frac{0.01}{0.01^2+\lambda} \end{bmatrix} U^T \vec{z} \\
&= -V \begin{bmatrix} 0.001 & 0 \\ 0 & 0.99 \end{bmatrix} V^T \vec{\beta}^* + V \begin{bmatrix} 0.99 & 0 \\ 0 & 0.99 \end{bmatrix} U^T \vec{z} \\
&= \begin{bmatrix} 0.329 \\ 0.823 \end{bmatrix}
\end{aligned}$$

Example

Least-squares relative error $= 27.00$

$$\frac{\left|\left|\vec{\beta}_{\text{ridge}} - \vec{\beta}^*\right|\right|_2}{\left|\left|\vec{z}\right|\right|_2} = 7.96$$

Example



Coefficients
Coefficient error
Least-squares fit
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
10^{-7}
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Regularization parameter

Maximum-a-posteriori estimator

Is there a probabilistic interpretation of ridge regression?

Bayesian viewpoint: $\vec{\boldsymbol{\beta}}$ is modeled as random, not deterministic

The maximum-a-posteriori (MAP) estimator of $\vec{\boldsymbol{\beta}}$ given $\vec{y}$ is

$$\vec{\beta}_{MAP}(\vec{y}) := \arg\max_{\vec{\beta}} f_{\vec{\boldsymbol{\beta}} \mid \vec{\mathbf{y}}}\left(\vec{\beta} \mid \vec{y}\right),$$

$f_{\vec{\boldsymbol{\beta}} \mid \vec{\mathbf{y}}}$ is the conditional pdf of $\vec{\boldsymbol{\beta}}$ given $\vec{\mathbf{y}}$

Maximum-a-posteriori estimator

Let $\vec{y} \in \mathbb{R}^n$ be a realization of

$$\vec{\mathbf{y}} := X\vec{\boldsymbol{\beta}} + \vec{\mathbf{z}}$$

where $\vec{\boldsymbol{\beta}}$ and $\vec{\mathbf{z}}$ are iid Gaussian with mean zero and variance σ_1^2 and σ_2^2

If $X \in \mathbb{R}^{n\times m}$ is known, then

$$\vec{\beta}_{\text{MAP}} = \arg\min_{\vec{\beta}} \left|\left|\vec{y} - X\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2$$

where $\lambda := \sigma_2^2/\sigma_1^2$

What does it mean if σ_1^2 is tiny or large? How about σ_2^2?

Problem

How to calibrate regularization parameter

Cannot use coefficient error (we don't know the true value!)

Cannot minimize over training data (why?)

Solution: Check fit on new data

Cross validation

Given a set of examples

$$\left(y^{(1)}, \vec{x}^{(1)}\right), \left(y^{(2)}, \vec{x}^{(2)}\right), \ldots, \left(y^{(n)}, \vec{x}^{(n)}\right),$$

1. Partition data into a training set $X_{\text{train}} \in \mathbb{R}^{n_{\text{train}} \times p}$, $\vec{y}_{\text{train}} \in \mathbb{R}^{n_{\text{train}}}$ and a validation set $X_{\text{val}} \in \mathbb{R}^{n_{\text{val}} \times p}$, $\vec{y}_{\text{val}} \in \mathbb{R}^{n_{\text{val}}}$

2. Fit model using the training set for every λ in a set Λ

   $$\vec{\beta}_{\text{ridge}}(\lambda) := \arg\min_{\vec{\beta}} \left|\left|\vec{y}_{\text{train}} - X_{\text{train}}\vec{\beta}\right|\right|_2^2 + \lambda \left|\left|\vec{\beta}\right|\right|_2^2$$

   and evaluate the fitting error on the validation set

   $$\text{err}(\lambda) := \left|\left|\vec{y}_{\text{train}} - X_{\text{train}}\vec{\beta}_{\text{ridge}}(\lambda)\right|\right|_2^2$$

3. Choose the value of λ that minimizes the validation-set error

   $$\lambda_{\text{cv}} := \arg\min_{\lambda \in \Lambda} \text{err}(\lambda)$$

Prediction of house prices

Aim: Predicting the price of a house from

1. Area of the living room
2. Condition (integer between 1 and 5)
3. Grade (integer between 7 and 12)
4. Area of the house without the basement
5. Area of the basement
6. The year it was built
7. Latitude
8. Longitude
9. Average area of the living room of houses within 15 blocks

Prediction of house prices

Training data: 15 houses

Validation data: 15 houses

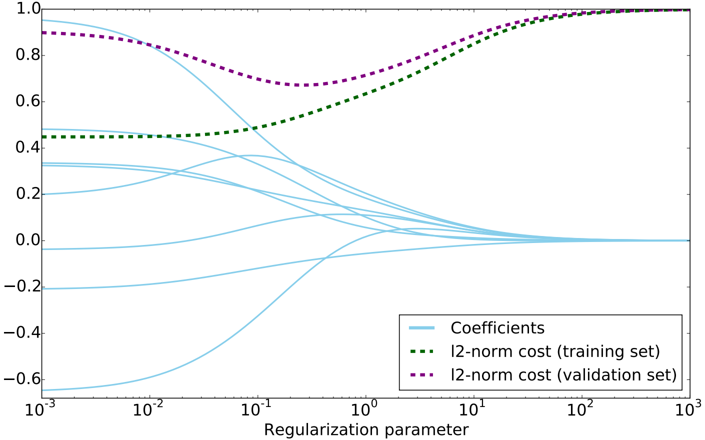
Test data: 15 houses

Condition number of training-data feature matrix: 9.94

We evaluate the relative fit

$$\frac{\left|\left|\vec{y} - X\vec{\beta}_{\text{ridge}}\right|\right|_2}{\left|\left|\vec{y}\right|\right|_2}$$

Prediction of house prices



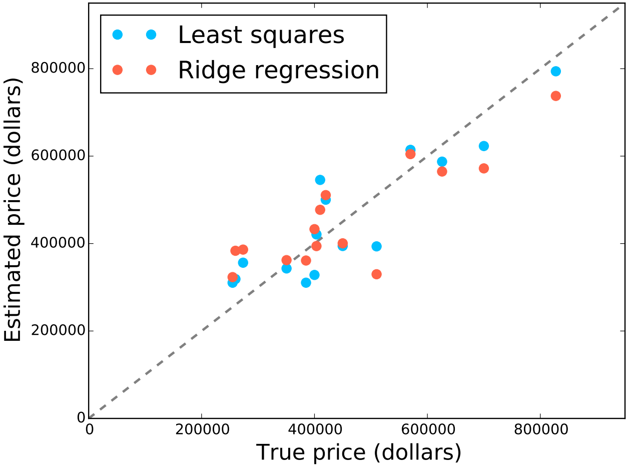
Prediction of house prices

Best λ: 0.27

Validation set error: 0.672 (least-squares: 0.906)

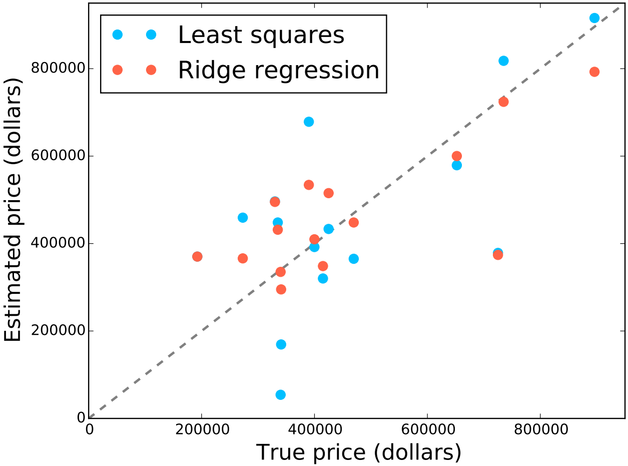
Test set error: 0.799 (least-squares: 1.186)

Training



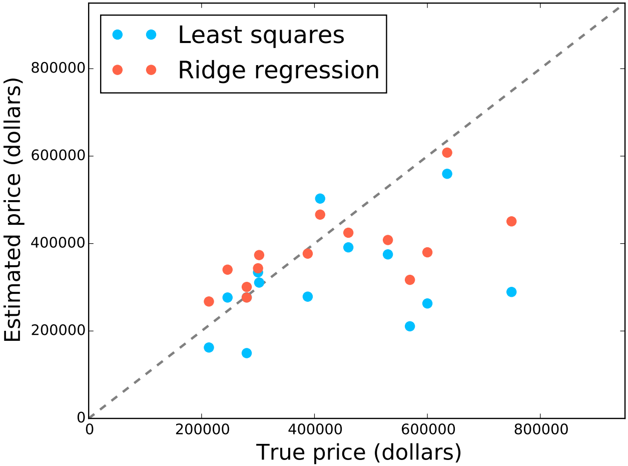
Least squares
Ridge regression
Estimated price (dollars)
True price (dollars)
0
200000
400000
600000
800000

Validation



Least squares
Ridge regression
Estimated price (dollars)
True price (dollars)
0
200000
400000
600000
800000

Test



Least squares
Ridge regression
Estimated price (dollars)
True price (dollars)
0
200000
400000
600000
800000

The classification problem

Goal: Assign examples to one of several predefined categories

We have n examples of labels and corresponding features

$$\left(y^{(1)}, \vec{x}^{(1)}\right), \left(y^{(2)}, \vec{x}^{(2)}\right), \ldots, \left(y^{(n)}, \vec{x}^{(n)}\right).$$

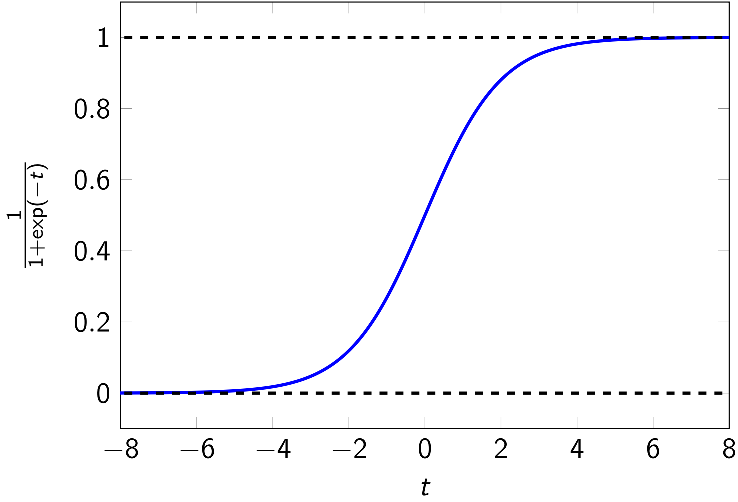
Here, we consider only two categories: labels are 0 or 1

Logistic function

Smoothed version of step function

$$g(t) := \frac{1}{1 + \exp(-t)}$$

Logistic function



Logistic regression

Generalized linear model: linear model + entrywise link function

$$y^{(i)} \approx g\left(\beta_0 + \langle \vec{x}^{(i)}, \vec{\beta} \rangle\right).$$

Maximum likelihood

If $y^{(1)}, \ldots, y^{(n)}$ are independent samples from Bernoulli random variables with parameter

$$p_{\mathbf{y}^{(\mathbf{i})}}(1) := g\left(\langle \vec{x}^{(i)}, \vec{\beta} \rangle\right)$$

where $\vec{x}^{(1)}, \ldots, \vec{x}^{(n)} \in \mathbb{R}^p$ are known, the ML estimate of $\vec{\beta}$ given $y^{(1)}, \ldots, y^{(n)}$ is

$$\vec{\beta}_{\text{ML}} := \sum_{i=1}^{n} y^{(i)} \log g\left(\langle \vec{x}^{(i)}, \vec{\beta} \rangle\right) + \left(1 - y^{(i)}\right) \log\left(1 - g\left(\langle \vec{x}^{(i)}, \vec{\beta} \rangle\right)\right)$$

Maximum likelihood

$$\mathcal{L}\left(\vec{\beta}\right) := p_{\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(n)}}\left(y^{(1)},\ldots,y^{(n)}\right)$$

Maximum likelihood

$$\begin{aligned}\mathcal{L}\left(\vec{\beta}\right) &:= p_{\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(n)}}\left(y^{(1)},\ldots,y^{(n)}\right)\\ &= \prod_{i=1}^{n} g\left(\langle\vec{x}^{(i)},\vec{\beta}\rangle\right)^{y^{(i)}}\left(1-g\left(\langle\vec{x}^{(i)},\vec{\beta}\rangle\right)\right)^{1-y^{(i)}}\end{aligned}$$

Logistic-regression estimator

$$\vec{\beta}_{\text{LR}} := \sum_{i=1}^{n} y^{(i)} \log g\left(\langle \vec{x}^{(i)}, \vec{\beta} \rangle\right) + \left(1 - y^{(i)}\right) \log\left(1 - g\left(\langle \vec{x}^{(i)}, \vec{\beta} \rangle\right)\right)$$

For a new $\vec{x}$ the logistic-regression prediction is

$$y_{\text{LR}} := \begin{cases} 1 & \text{if } g\left(\langle \vec{x}, \vec{\beta}_{LR} \rangle\right) \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

$g\left(\langle \vec{x}, \vec{\beta}_{LR} \rangle\right)$ can be interpreted as the probability that the label is 1

Iris data set

Aim: Classify flowers using sepal width and length

Two species, 5 examples each:

- *Iris setosa* (label 0): sepal lengths 5.4, 4.3, 4.8, 5.1 and 5.7, and sepal widths 3.7, 3, 3.1, 3.8 and 3.8
- *Iris versicolor* (label 1): sepal lengths 6.5, 5.7, 7, 6.3 and 6.1, and sepal widths 2.8, 2.8, 3.2, 2.3 and 2.8

Two new examples: (5.1, 3.5), (5,2)

Iris data set

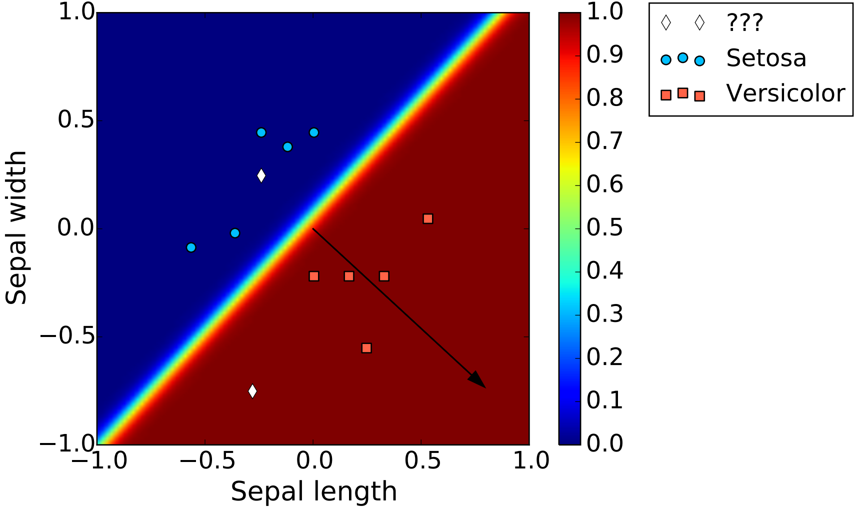
After centering and normalizing

$$\vec{\beta}_{\text{LR}} = \begin{bmatrix} 32.1 \\ -29.6 \end{bmatrix} \quad \text{and} \quad \beta_0 = 2.06$$

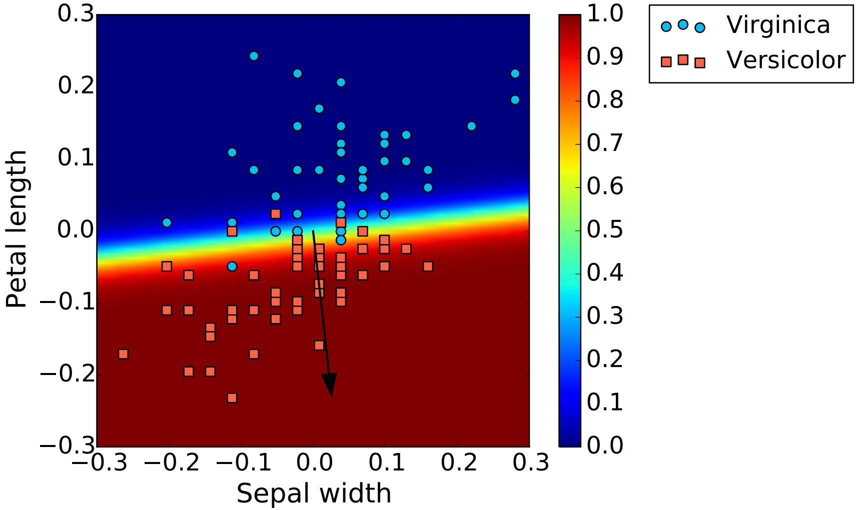
i	1	2	3	4	5
$\vec{x}^{(i)}[1]$	-0.12	-0.56	-0.36	-0.24	0.00
$\vec{x}^{(i)}[2]$	0.38	-0.09	-0.02	0.45	0.45
$\langle \vec{x}^{(i)}, \vec{\beta}_{\text{LR}} \rangle + \beta_0$	-12.9	-13.5	-8.9	-18.8	-11.0
$g\left(\langle \vec{x}^{(i)}, \vec{\beta}_{\text{LR}} \rangle + \beta_0\right)$	0.00	0.00	0.00	0.00	0.00

i	6	7	8	9	10
$\vec{x}^{(i)}[1]$	0.33	0.00	0.53	0.25	0.17
$\vec{x}^{(i)}[2]$	-0.22	-0.22	0.05	-0.05	-0.22
$\langle \vec{x}^{(i)}, \vec{\beta}_{\text{LR}} \rangle + \beta_0$	19.1	8.7	17.7	26.3	13.9
$g\left(\langle \vec{x}^{(i)}, \vec{\beta}_{\text{LR}} \rangle + \beta_0\right)$	1.00	1.00	1.00	1.00	1.00

Iris data set



Iris data set



Digit classification

MNIST data

Aim: Distinguish one digit from another

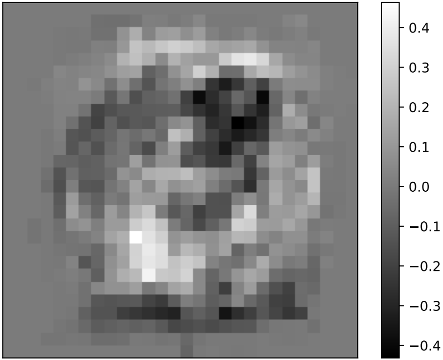
$\vec{x}_i$ is an image of a 6 or a 9

$\vec{y}_i = 1$ or $\vec{y}_i = 0$ if image i is a 6 or 9, respectively

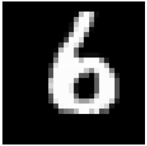
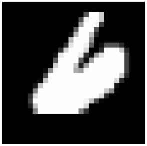
2000 training examples and 2000 test examples, each half 6 half 9

Training error rate: 0.0, Test error rate $= 0.006$

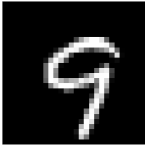
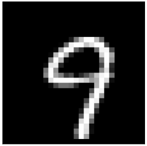
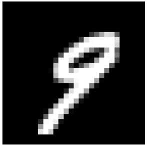
Digit classification: $\vec{\beta}$



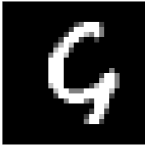
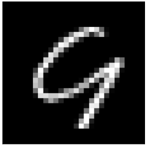
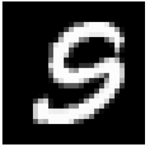
Digit classification: True Positives

$\vec{\beta}^T x$	Probability of 6	Image
20.878	1.00	
18.217	1.00	
16.408	1.00	

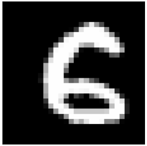
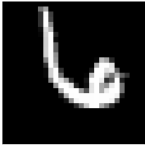
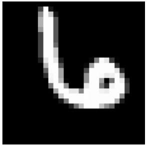
Digit classification: True Negatives

$\vec{\beta}^T x$	Probability of 6	Image
-14.71	0.00	
-15.829	0.00	
-17.02	0.00	

Digit classification: False Positives

$\vec{\beta}^T x$	Probability of 6	Image
7.612	0.9995	
0.4341	0.606	
7.822484	0.9996	

Digit classification: False Negatives

$\vec{\beta}^T x$	Probability of 6	Image
-5.984	0.0025	
-2.384	.084	
-1.164	0.238	

Digit Classification

This is a toy problem: distinguishing one digit from another is very easy

Harder is to classify any given digit

We used it to give insight into how logistic regression works

It turns out, on this simplified problem, a very easy solution for $\vec{\beta}$ gives good results. Can you guess it?

Digit Classification

Average of 6's minus average of 9's

Training error: 0.005, Test error: 0.0035