Beyond classification

In supervised learning problems we are given training data

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$$

where $\mathbf{x}_i \in \mathbb{R}^d$, and so far we have only considered the case where $y_i \in \{+1, -1\}$ (or $y_i \in \{0, \ldots, K-1\}$)

What if $y_i \in \mathbb{R}$?

This problem is usually called ***regression***

You can think of regression as being an extension of classification as the number of classes grows to ∞

Regression

A ***regression model*** typically posits that our training data are realizations of a random pair (X, Y) where

$$Y = f(X) + E$$

with E representing noise and f belonging to some class of functions

Example function classes

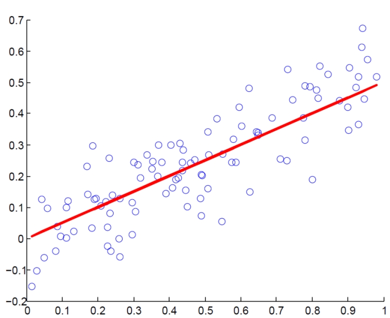
- polynomials
- sinusoids/trigonometric polynomials
- exponentials
- kernels

Linear regression

In ***linear regression***, we assume that f is an ***affine*** function, i.e.,

$$f(\mathbf{x}) = \boldsymbol{\beta}^T \mathbf{x} + \beta_0$$

where $\boldsymbol{\beta} \in \mathbb{R}^d$, $\beta_0 \in \mathbb{R}$



How can we estimate $\boldsymbol{\beta}, \beta_0$ from the training data?

Least squares

In ***least squares*** linear regression, we select $\boldsymbol{\beta}, \beta_0$ to minimize the sum of squared errors

$$\text{SSE}(\boldsymbol{\beta}, \beta_0) := \sum_{i=1}^{n} \left(y_i - \boldsymbol{\beta}^T \mathbf{x}_i - \beta_0\right)^2$$

Least squares is (arguably) the most fundamental tool in all of applied mathematics!

Legendre
(1805)

Gauss
~~(1809)~~
(1795)

Example

Suppose $d = 1$, so that x_i, β are scalars

$$\mathrm{SSE}(\beta, \beta_0) = \sum_{i=1}^{n} (y_i - \beta x_i - \beta_0)^2$$

How to minimize?

$$\frac{\partial \mathrm{SSE}}{\partial \beta_0} = -2 \sum_{i=1}^{n} (y_i - \beta x_i - \beta_0) = 0$$

$$\frac{\partial \mathrm{SSE}}{\partial \beta} = -2 \sum_{i=1}^{n} x_i (y_i - \beta x_i - \beta_0) = 0$$

Example

Rearranging these equations, we obtain

$$n\beta_0 + \sum_{i=1}^{n} \beta x_i = \sum_{i=1}^{n} y_i$$

$$\sum_{i=1}^{n} \beta_0 x_i + \sum_{i=1}^{n} \beta x_i^2 = \sum_{i=1}^{n} x_i y_i$$

or in matrix form

$$\begin{bmatrix} n & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta \end{bmatrix} = \begin{bmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{bmatrix}$$

Example

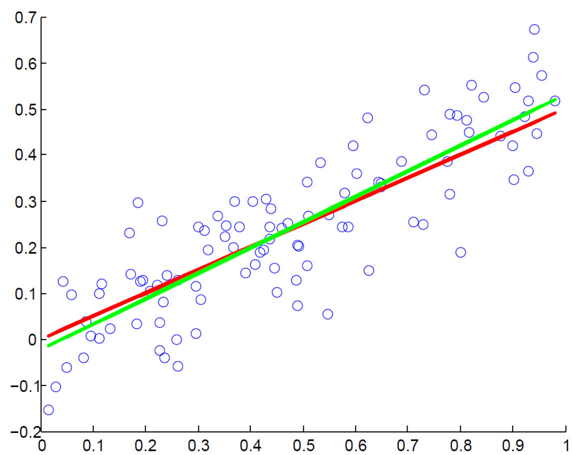
Inverting the matrix

$$\begin{bmatrix} \widehat{\beta}_0 \\ \widehat{\beta} \end{bmatrix} = \begin{bmatrix} n & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{bmatrix}$$

Setting $\bar{x} = \frac{1}{n} \sum_i x_i$ and $\bar{y} = \frac{1}{n} \sum_i y_i$, the solution to this system reduces to

$$\begin{bmatrix} \widehat{\beta}_0 \\ \widehat{\beta} \end{bmatrix} = \frac{1}{\sum_i x_i^2 - n\bar{x}^2} \begin{bmatrix} \bar{y}(\sum_i x_i^2) - \bar{x} \sum_i x_i y_i \\ \sum_i x_i y_i - n\bar{x}\bar{y} \end{bmatrix}$$

Example



General least squares

Suppose d is arbitrary. Set

$$\boldsymbol{\theta} = \begin{bmatrix} \beta_0 \\ \beta(1) \\ \vdots \\ \beta(d) \end{bmatrix}$$

Then $\mathsf{SSE}(\boldsymbol{\theta}) = \sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^T \mathbf{x}_i - \beta_0)^2 = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2$

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \boldsymbol{A} = \begin{bmatrix} 1 & x_1(1) & \cdots & x_1(d) \\ 1 & x_2(1) & \cdots & x_2(d) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n(1) & \cdots & x_n(d) \end{bmatrix}$$

General least squares

The minimizer $\widehat{\boldsymbol{\theta}}$ of this quadratic objective function is

$$\widehat{\boldsymbol{\theta}} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{y}$$

provided that $\boldsymbol{A}^T\boldsymbol{A}$ is ***nonsingular***

"Proof"

$$\begin{aligned} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 &= (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta})^T(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}) \\ &= \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{\theta}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{\theta} \end{aligned}$$

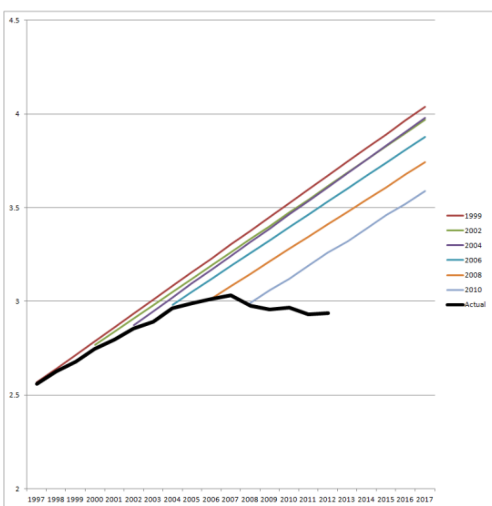
$$\nabla_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 = -2\boldsymbol{A}^T\boldsymbol{y} + 2\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{\theta} = 0$$

$$\Downarrow$$

$$\widehat{\boldsymbol{\theta}} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{y}$$

Does *linear* regression always make sense?

Official US DOT forecasts of road traffic, compared to actual



Nonlinear feature maps

Sometimes linear methods (in both regression and classification) just don't work

One way to create nonlinear estimators or classifiers is to first transform the data via a nonlinear feature map

$$\Phi : \mathbb{R}^d \to \mathbb{R}^{d'}$$

After applying Φ, we can then try applying a linear method to the transformed data $\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_n)$

Regression

In the case of regression, our model becomes

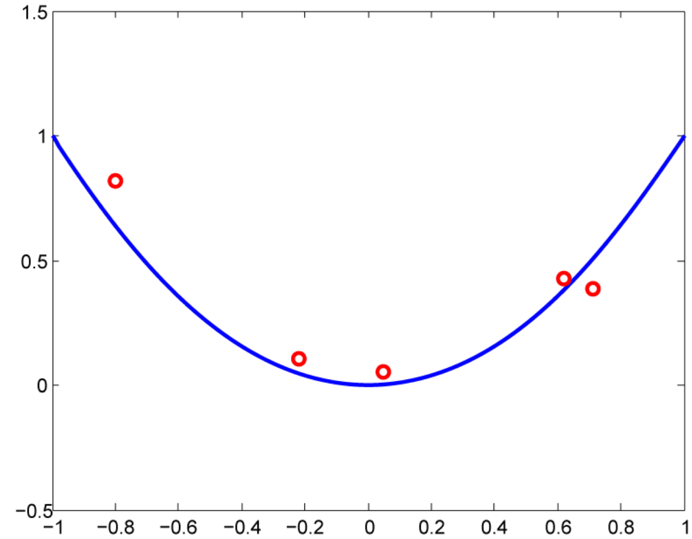
$$f(\mathbf{x}) = \boldsymbol{\beta}^T \Phi(\mathbf{x}) + \beta_0$$

where now $\boldsymbol{\beta} \in \mathbb{R}^{d'}$

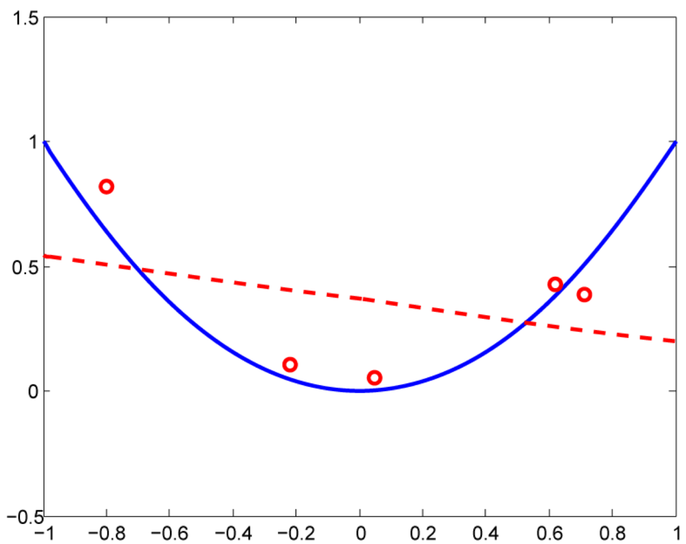
Example. Suppose $d = 1$ but $f(x)$ is a cubic polynomial. How do we find a least squares estimate of f from training data?

$$\Phi_k(x) = x^k \quad \Longrightarrow \quad \boldsymbol{A} = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 \end{bmatrix}$$

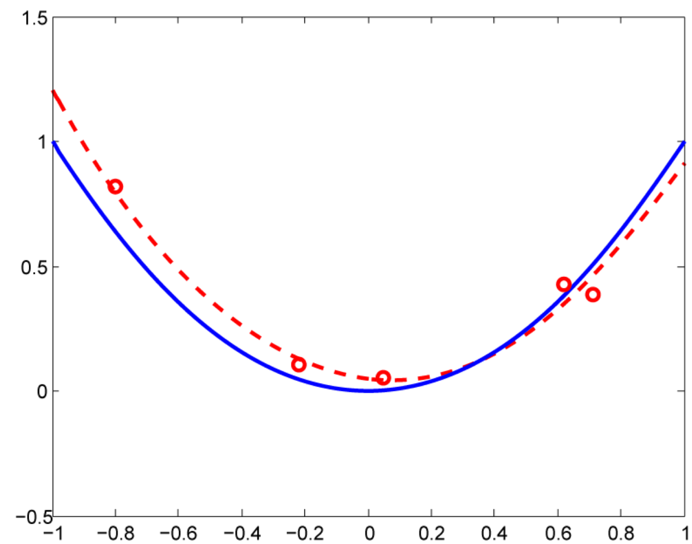
Overfitting



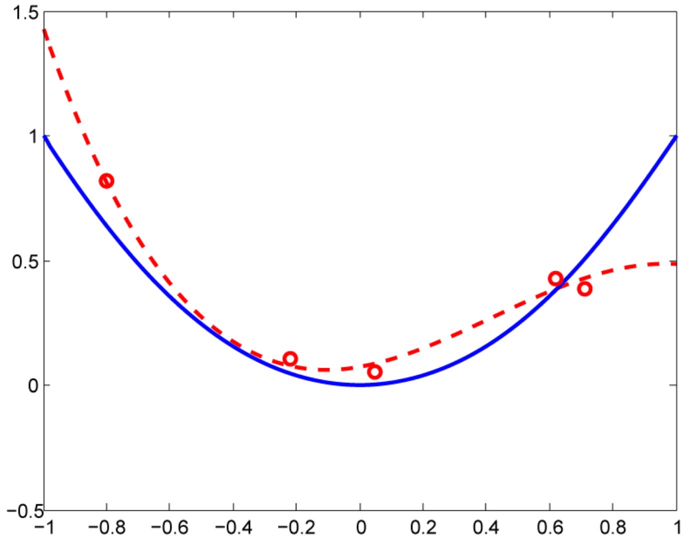
Overfitting



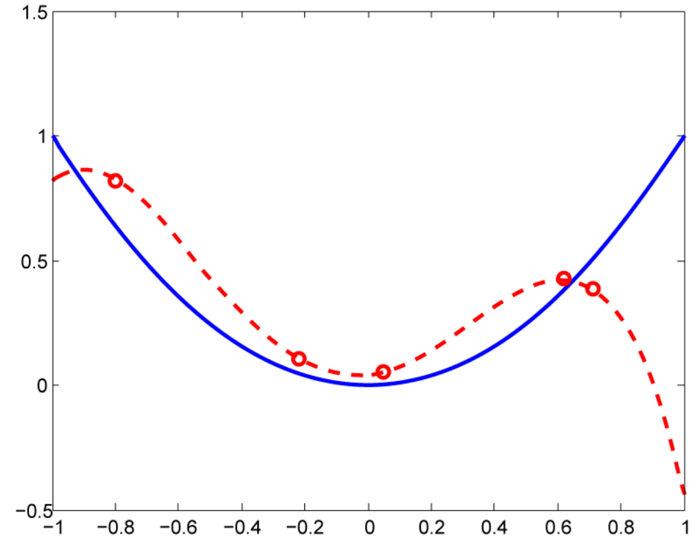
Overfitting



Overfitting



Overfitting



Is the problem just noise?

Noise in the observations can make overfitting a big problem

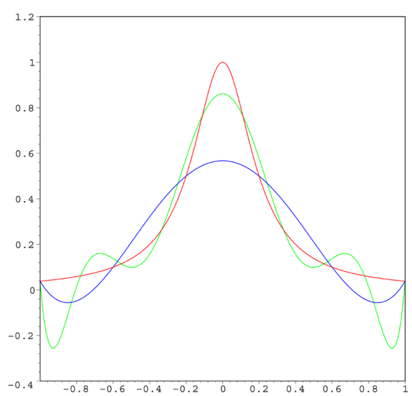
What if there is no noise?

Runge's phenomenon

Take a smooth function

- not exactly polynomial
- well approximated by a polynomial

Even in the absence of noise, fitting a higher order polynomial (interpolation) can be incredibly unstable



Regression summary

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \boldsymbol{A} = \begin{bmatrix} 1 & x_1(1) & \cdots & x_1(d) \\ 1 & x_2(1) & \cdots & x_2(d) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n(1) & \cdots & x_n(d) \end{bmatrix} \quad \boldsymbol{\theta} = \begin{bmatrix} \beta_0 \\ \beta(1) \\ \vdots \\ \beta(d) \end{bmatrix}$$

$$\text{SSE}(\boldsymbol{\theta}) = \sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^T \mathbf{x}_i - \beta_0)^2 = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2$$

Minimizer given by

$$\widehat{\boldsymbol{\theta}} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{y}$$

provided that $\boldsymbol{A}^T\boldsymbol{A}$ is ***nonsingular***

Regularization and regression

Overfitting occurs as the number of features d begins to approach the number of observations n

In this regime, we have *too many degrees of freedom*

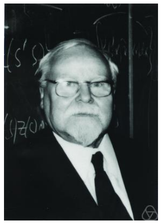
Idea: penalize candidate solutions for using too many features

One candidate regularizer: $r(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2$

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2$$

$\lambda > 0$ is a "tuning parameter" that controls the tradeoff between fit and complexity

Tikhonov regularization



This is one example of a more general technique called ***Tikhonov regularization***

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \|\boldsymbol{\Gamma}\boldsymbol{\theta}\|_2^2$$

(Note that λ has been replaced by the matrix $\boldsymbol{\Gamma}$)

Solution: Observe that

$$\begin{aligned}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \|\boldsymbol{\Gamma}\boldsymbol{\theta}\|_2^2 &= (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta})^T(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}) + \boldsymbol{\theta}^T\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\boldsymbol{\theta} \\ &= \boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{\theta}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{\theta} - 2\boldsymbol{\theta}^T\boldsymbol{A}^T\boldsymbol{y} \\ &\quad + \boldsymbol{\theta}^T\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\boldsymbol{\theta} \\ &= \boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{\theta}^T\left(\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\right)\boldsymbol{\theta} \\ &\quad - 2\boldsymbol{\theta}^T\boldsymbol{A}^T\boldsymbol{y}\end{aligned}$$

Tikhonov regularization

$$\nabla_{\boldsymbol{\theta}}\left(\boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{\theta}^T\left(\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\right)\boldsymbol{\theta} - 2\boldsymbol{\theta}^T\boldsymbol{A}^T\boldsymbol{y}\right)$$

$$= 2\left(\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\right)\boldsymbol{\theta} - 2\boldsymbol{A}^T\boldsymbol{y}$$

Setting this equal to zero and solving for $\boldsymbol{\theta}$ yields

$$\widehat{\boldsymbol{\theta}} = \left(\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{\Gamma}^T\boldsymbol{\Gamma}\right)^{-1}\boldsymbol{A}^T\boldsymbol{y}$$

Suppose $\boldsymbol{\Gamma} = \sqrt{\lambda}\boldsymbol{I}$, then

$$\widehat{\boldsymbol{\theta}} = (\underbrace{\boldsymbol{A}^T\boldsymbol{A} + \lambda\boldsymbol{I}})^{-1}\boldsymbol{A}^T\boldsymbol{y}$$

for suitable choice of λ, always well-conditioned

Ridge regression

In the context of regression, Tikhonov regularization has a special name: ***ridge regression***

Ridge regression is essentially exactly what we have been talking about, but in the special case where

$$\boldsymbol{\Gamma} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & \sqrt{\lambda} & 0 & \cdots & 0 \\ 0 & 0 & \sqrt{\lambda} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sqrt{\lambda} \end{bmatrix}$$

We are penalizing all coefficients in $\boldsymbol{\beta}$ equally, but not penalizing the offset β_0

Another take: Constrained minimization

One can use Lagrange multipliers (KKT conditions) to show that

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \|\boldsymbol{\Gamma}\boldsymbol{\theta}\|_2^2$$

is formally equivalent to

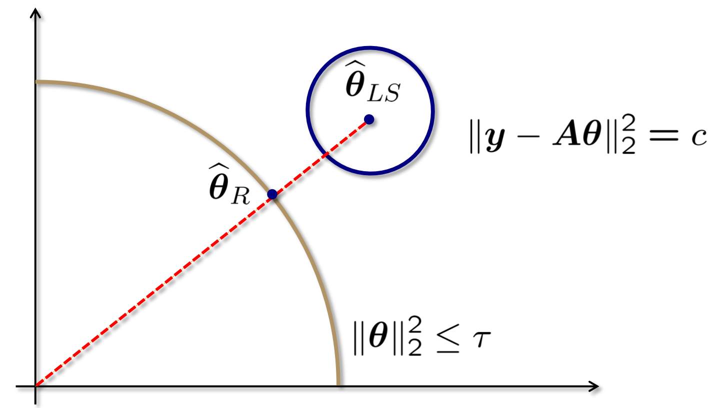
$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2$$

$$\text{subject to } \|\boldsymbol{\Gamma}\boldsymbol{\theta}\|_2^2 \le \tau$$

for a suitable choice of τ

Tikhonov versus least squares

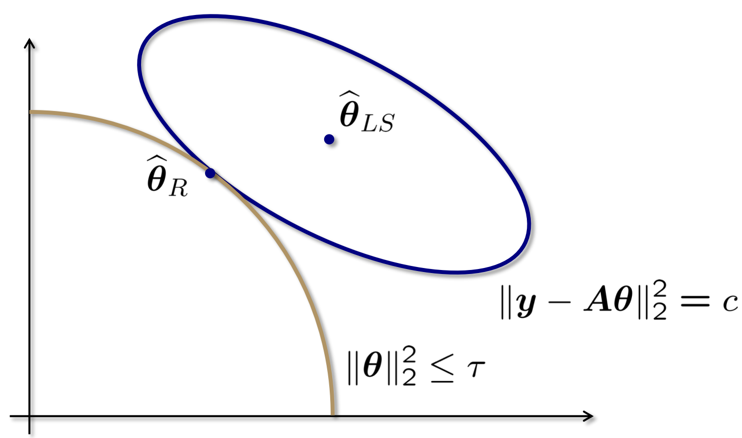
Assume $\boldsymbol{\Gamma} = \boldsymbol{I}$ and that $\boldsymbol{A}$ has orthonormal columns



Tikhonov regularization is equivalent to shrinking the least squares solution towards the origin

Tikhonov versus least squares

In general, we have this picture



Tikhonov regularization still shrinking the least squares solution towards the origin

Shrinkage estimators

Tikhonov regularization is one type of ***shrinkage estimator***

Shrinkage estimators are estimators that "shrink" the naïve estimate towards some implicit guess

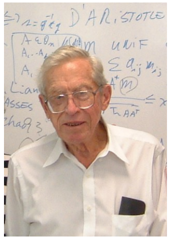
Example: How do we estimate the variance in a sample? Let $x_1, \ldots, x_n$ be n i.i.d. samples drawn according to some unknown distribution. How can we estimate the variance?

$$\widehat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \quad \Rightarrow \quad \mathbb{E}\left[\widehat{\sigma}^2\right] = \frac{n-1}{n}\sigma^2$$

This is a ***biased*** estimate (it shrinks slightly towards zero), however, it also achieves a ***lower MSE*** than the unbiased estimate

Stein's paradox

Examples where shrinkage estimators work fundamentally better than naïve estimates are much more common than you would think!



Stein's paradox (1955)

Consider the estimation problem where you observe $\boldsymbol{y} = \boldsymbol{\theta} + \boldsymbol{n}$, where $\boldsymbol{n}$ is i.i.d. Gaussian noise.

A natural estimate for $\boldsymbol{\theta}$ is $\widehat{\boldsymbol{\theta}} = \boldsymbol{y}$.

If the dimension is 3 or higher, then this is suboptimal in terms of the MSE $\mathbb{E}\left[\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2^2\right]$

One can do better by shrinking towards ***any*** guess for $\boldsymbol{\theta}$

- people usually shrink towards the origin
- a better guess leads to bigger improvements

Alternative regularizers

- Akaike information criterion (AIC)
- Bayesian information criterion (BIC)

$$r(\boldsymbol{\theta}) \approx \|\boldsymbol{\theta}\|_0 := |\mathrm{supp}(\boldsymbol{\theta})|$$

- Least absolute shrinkage and selection operator (LASSO)

$$r(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1 = \sum_j |\theta(j)|$$

  - also results in shrinkage, but where all coordinates are shrunk by the same amount
  - promotes sparsity
  - can think of $\|\boldsymbol{\theta}\|_1$ as a more computationally tractable replacement for $\|\boldsymbol{\theta}\|_0$