Ch01. Linear Regression

Hwanjo Yu

POSTECH

http://hwanjoyu.org

Outline

- Regression and linear models
- Ordinary least squares (OLS)
  - Least squared method
  - Maximum likelihood perspective
- Regularization
  - Ridge regression
  - Sparse regression (LASSO)
- Bias-Variance Dilemma

Regression?

Problem Setup

Given a set of N labeled examples, $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$ ($\mathbf{x}_n \in X \subset \mathbb{R}^D$ and $y_n \in Y \subset \mathbb{R}$), the goal is to learn a mapping

$$f(\mathbf{x}): X \to Y,$$

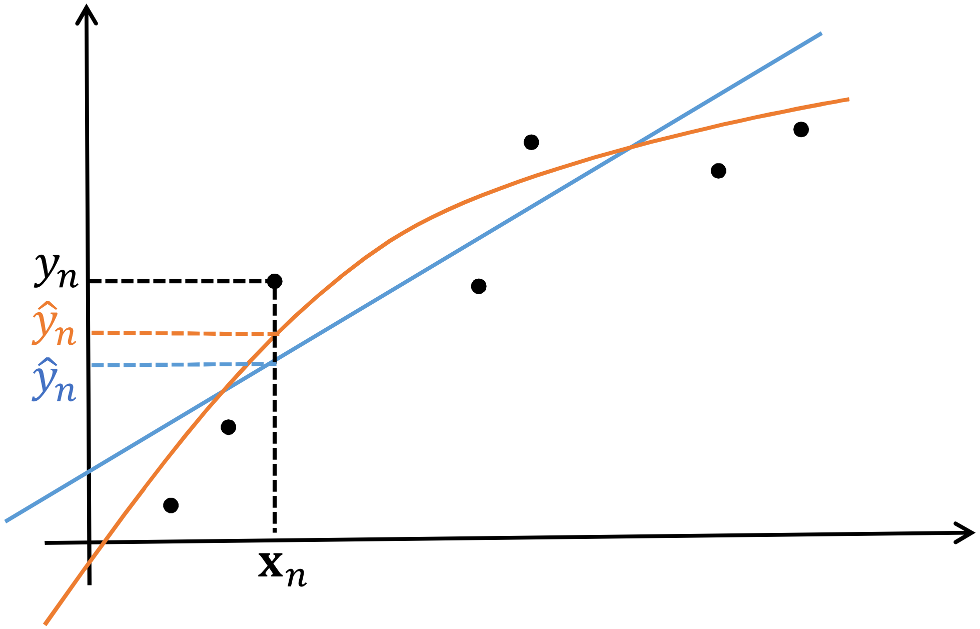
which associates $\mathbf{x}$ with y, such that we can make prediction about y^*, when a new input $\mathbf{x}^* \notin \mathcal{D}$ is provided.

- **Linear models:** $f(\mathbf{x}) = \sum_{j=1}^{M} w_j \phi_j(\mathbf{x}) + w_0$.
- **Neural networks:** $f(\mathbf{x}) = \sum_{j=1}^{M} w_j^{(2)} \phi\left(\sum_k W_{j,k}^{(1)} x_k + b_j^{(1)}\right)$.
- **Kernel regression:** $f(\mathbf{x}) = \sum_{n=1}^{N} w_n k(\mathbf{x}, \mathbf{x}_n) + w_0$.

- Regression model: $y = f(\mathbf{x}) + \epsilon$.
- $\mathbf{x}$: input, independent variable, predictor, regressor, covariate
- y: output, dependent variable, response
- $\phi_j(\mathbf{x})$: basis function, feature
- w_j: weight, coefficient, learning parameter

Why Linear Models?

$$y_n = w_1 x_{1,n} + w_2 x_{2,n} + \cdots + w_M x_{M,n} + w_0 + \epsilon_n, \qquad \forall n = 1, \ldots, N.$$



- Easy to solve (can be solved analytically)
- Interpretable (in contrast to deep learning)

Linear Regression

Linear regression refers to a model in which the conditional mean of y_n given the value of $\mathbf{x}_n$ is an affine function of $\phi(\mathbf{x}_n)$

$$f(\mathbf{x}_n) = \sum_{j=1}^{M} w_j \phi_j(\mathbf{x}_n) + w_0 \phi_0(\mathbf{x}_n) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n),$$

where $\phi_j(\mathbf{x}_n)$ are known as basis functions and

- $\mathbf{w} = [w_0, w_1, \dots, w_M]^{\mathrm{T}} \in \mathbb{R}^{M+1}$
- $\boldsymbol{\phi}(\mathbf{x}_n) = [\phi_0(\mathbf{x}_n), \phi_1(\mathbf{x}_n), \dots, \phi_M(\mathbf{x}_n)]^{\mathrm{T}} \in \mathbb{R}^{M+1}$
- $\phi_0(\mathbf{x}_n) = 1$

By using nonlinear basis functions, we allow the function $f(\mathbf{x}_n)$ to be a nonlinear function of the input vector $\mathbf{x}_n$ (but a linear function of $\boldsymbol{\phi}(\mathbf{x}_n)$).

Polynomial Regression: $f(\mathbf{x}) = \sum_{j=0}^{M} w_j \phi_j(\mathbf{x}) = \sum_{j=0}^{M} w_j \mathbf{x}^j$

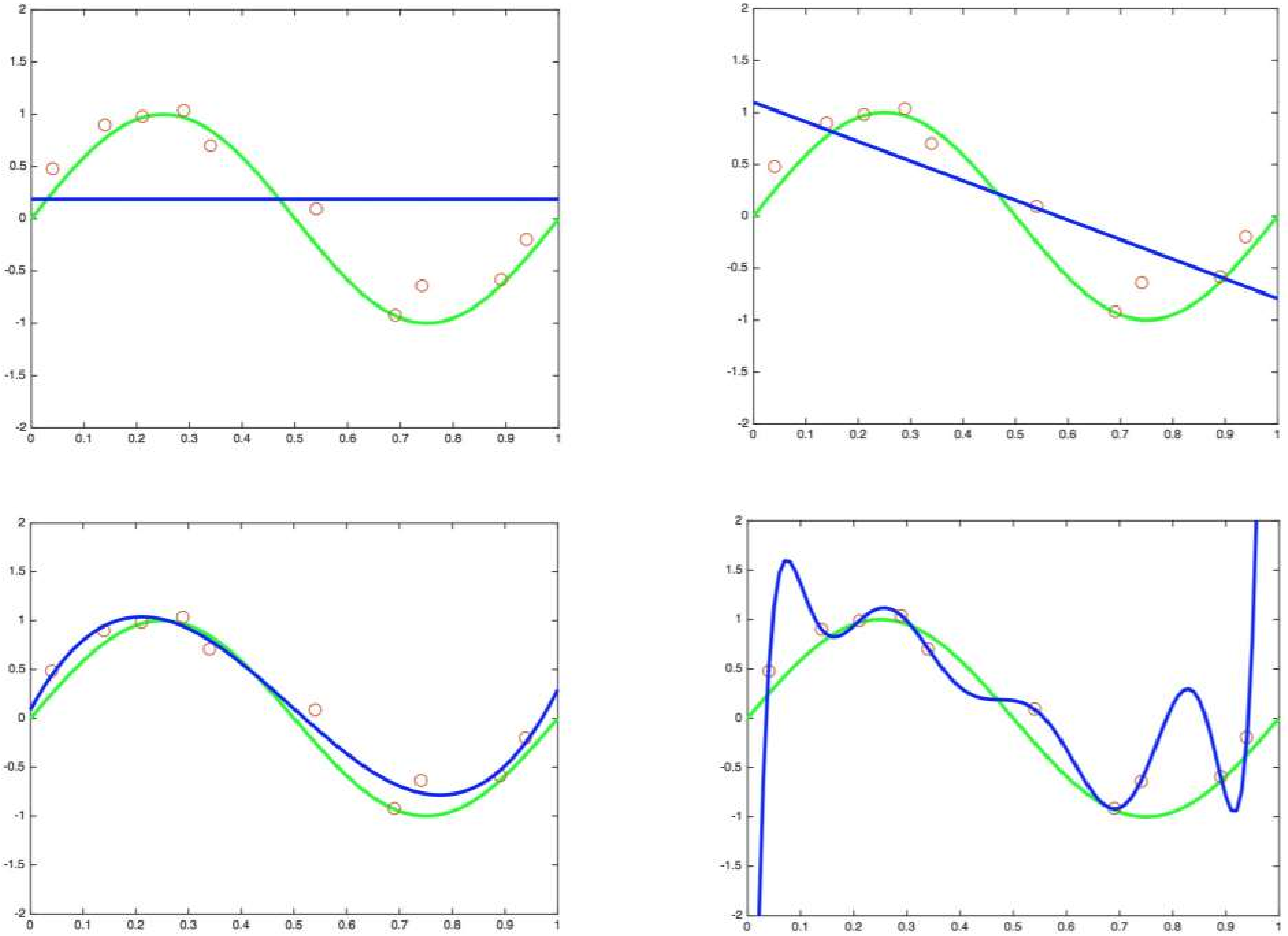


Figure: $M = 0,1,3,9$

Ordinary Least Squares (OLS)

Least Squared Method

Given a set of training data $\{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, we determine the weight vector $\mathbf{w} \in \mathbb{R}^{M+1}$ which minimizes

$$\mathcal{J}_{LS}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2 = \frac{1}{2}\left\|\mathbf{y} - \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{w}\right\|_2^2,$$

where $\mathbf{y} = [y_1, , \dots, y_N]^{\mathrm{T}} \in \mathbb{R}^N$ and $\boldsymbol{\Phi} \in \mathbb{R}^{(M+1)\times N}$ is known as the **design matrix** with $\boldsymbol{\Phi}_{j,n} = \phi_{j-1}(\mathbf{x}_n)$ for $j = 1, \dots, M+1$ and $n = 1, \dots, N$, i.e.,

$$\boldsymbol{\Phi}^{\mathrm{T}} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \dots & \phi_M(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \dots & \phi_M(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \dots & \phi_M(\mathbf{x}_N) \end{bmatrix}.$$

Note that

$$\left\|\mathbf{y} - \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{w}\right\|_2^2 = \left(\mathbf{y} - \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{w}\right)^{\mathrm{T}}\left(\mathbf{y} - \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{w}\right)$$

Least Squared Method (2)

Find the estimate $\widehat{\mathbf{w}}_{LS}$ such that

$$\mathbf{w}_{LS} = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2} \left\| \mathbf{y} - \boldsymbol{\phi}^{\mathrm{T}} \mathbf{w} \right\|_2^2$$

where both $\mathbf{y}$ and $\boldsymbol{\phi}$ are given.

How do you find the minimizer $\mathbf{w}_{LS}$?

Solve $\frac{\partial}{\partial \mathbf{w}} \left(\frac{1}{2} \left\| \mathbf{y} - \boldsymbol{\phi}^{\mathrm{T}} \mathbf{w} \right\|_2^2 \right) = 0$ for $\mathbf{w}$.

Least Squared Method (3)

Note that

$$\frac{1}{2}\left\|\mathbf{y}-\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}\right\|_2^2=\frac{1}{2}\left(\mathbf{y}-\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}\right)^{\mathrm{T}}\left(\mathbf{y}-\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}\right)$$

$$=\frac{1}{2}\left(\mathbf{y}^{\mathrm{T}}\mathbf{y}-\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}\mathbf{y}-\mathbf{y}^{\mathrm{T}}\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}+\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}\right)$$

Then, we have

$$\frac{\partial}{\partial\mathbf{w}}\left(\frac{1}{2}\left\|\mathbf{y}-\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}\right\|_2^2\right)=\boldsymbol{\phi}\,\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}-\boldsymbol{\phi}\mathbf{y}.$$

Least Squared Method (4)

Therefore, $\frac{\partial}{\partial \mathbf{w}}\left(\frac{1}{2}\left\|\mathbf{y}-\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}\right\|_2^2\right)=0$ leads to the normal equation that is of the form

$$\boldsymbol{\phi}\,\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}=\boldsymbol{\phi}\mathbf{y}.$$

Thus, LS estimate of is $\mathbf{w}$ given by

$$\mathbf{w}_{LS}=\left(\boldsymbol{\phi}\,\boldsymbol{\phi}^{\mathrm{T}}\right)^{-1}\boldsymbol{\phi}\mathbf{y}=\boldsymbol{\phi}^{\dagger}\mathbf{y}$$

where $\boldsymbol{\phi}^{\dagger}$ is known as **Moore-Penrose pseudo-inverse**

Maximum Likelihood Perspective

We consider a linear model where the target variable y_n is assumed to be generated by a deterministic function $f(\mathbf{x}_n; \mathbf{w}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n)$ with additive Gaussian noise:

$$y_n = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) + \epsilon_n,$$

for $n = 1, \dots, N$ and $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$.

In a compact form, we have

$$\mathbf{y} = \boldsymbol{\upphi}^{\mathrm{T}} \mathbf{w} + \boldsymbol{\epsilon}.$$

In other words, we model $p(\mathbf{y} | \boldsymbol{\upphi}, \mathbf{w})$ as

$$p(\mathbf{y} | \boldsymbol{\upphi}, \mathbf{w}) = \mathcal{N}\left(\boldsymbol{\upphi}^{\mathrm{T}} \mathbf{w}, \sigma^2 \mathbf{I}\right).$$

Maximum Likelihood Perspective (2)

The log-likelihood is given by

$$
\begin{aligned}
\mathcal{L} &= \log p(\mathbf{y}|\boldsymbol{\phi},\mathbf{w}) = \log \prod_{n=1}^{N} p(y_n|\phi(\mathbf{x}_n),\mathbf{w}) = \sum_{n=1}^{N} \log p(y_n|\phi(\mathbf{x}_n),\mathbf{w}) \\
&= \sum_{n=1}^{N} \log \mathcal{N}\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n),\sigma^2\right) = \sum_{n=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y_n-\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\right)^2}{2\sigma^2}} \\
&= \sum_{n=1}^{N} (\log \frac{1}{\sqrt{2\pi\sigma^2}} + \log e^{-\frac{\left(y_n-\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\right)^2}{2\sigma^2}}) = \sum_{n=1}^{N} \log \frac{1}{\sqrt{2\pi\sigma^2}} - \sum_{n=1}^{N} \frac{\left(y_n-\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\right)^2}{2\sigma^2} \\
&= \sum_{n=1}^{N} \log(2\pi\sigma^2)^{-\frac{1}{2}} - \frac{1}{\sigma^2}\frac{1}{2}\sum_{n=1}^{N}\left(y_n-\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)\right)^2 = -\frac{N}{2}\log\sigma^2 - \frac{N}{2} bg\ \ 2\pi - \sigma^{-2}\mathcal{J}_{LS}.
\end{aligned}
$$

MLE is given by

$$\mathbf{w}_{M\,L} = \underset{\mathbf{w}}{\operatorname{argmax}} \log p(\mathbf{y}|\boldsymbol{\phi},\mathbf{w})$$

Leading to

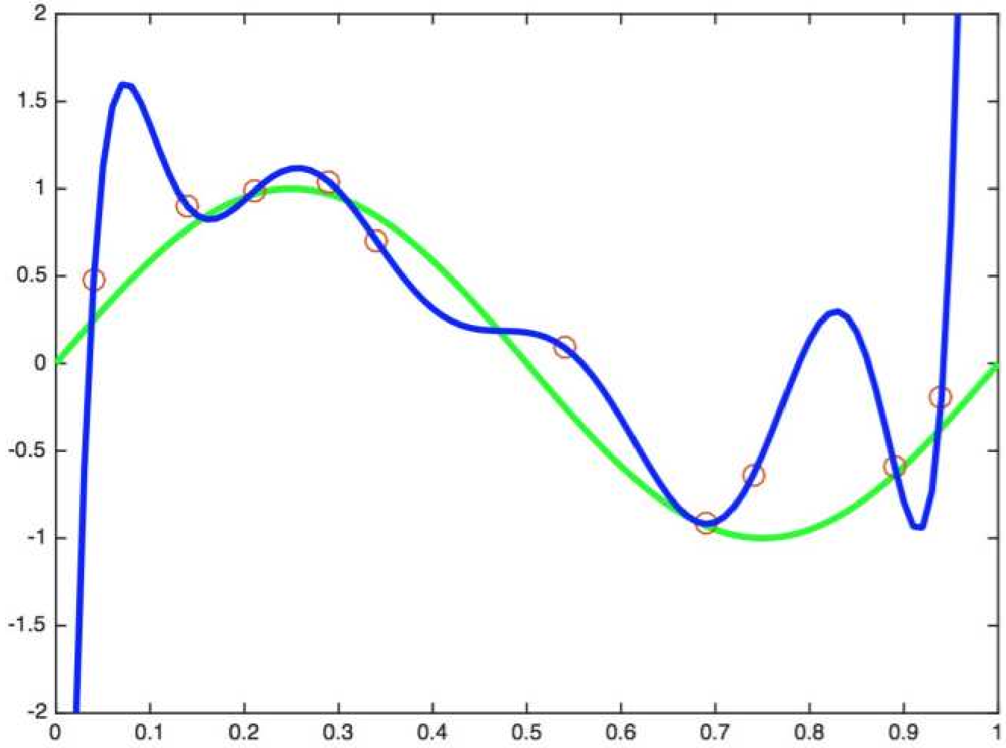
$$\mathbf{w}_{M\,L} = \mathbf{w}_{LS}$$

which we arrived at under **Gaussian noise assumption**.

Regularization

- Ridge regression: L_2 norm regularization
- LASSO: L_1 norm regularization

Why Regularization?



Improve the generalization of the learned model.

Regularization

Interested in: Inferring a function of any $\mathbf{x}$, given N examples $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$

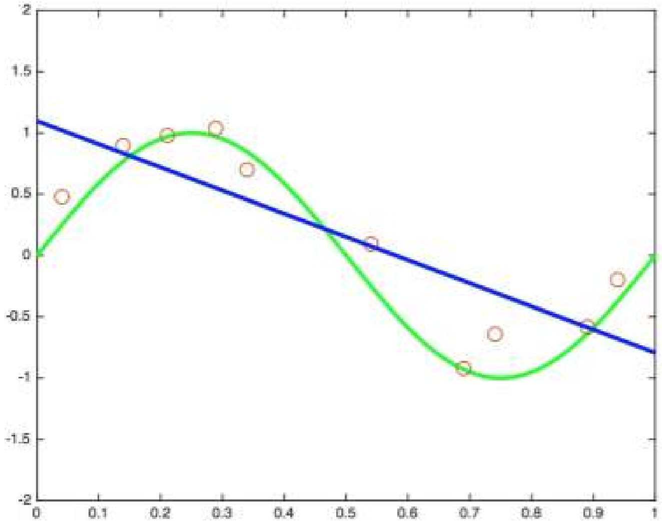
Consider a loss function $\ell(f(\mathbf{x}_n; \mathbf{w}), y_n)$. For instance, LS regression uses the square loss:

$$\sum_{n=1}^{N} \ell(f(\mathbf{x}_n; \mathbf{w}), y_n) = \frac{1}{2} \left\| \mathbf{y} - \boldsymbol{\phi}^{\mathrm{T}} \mathbf{w} \right\|_2^2$$

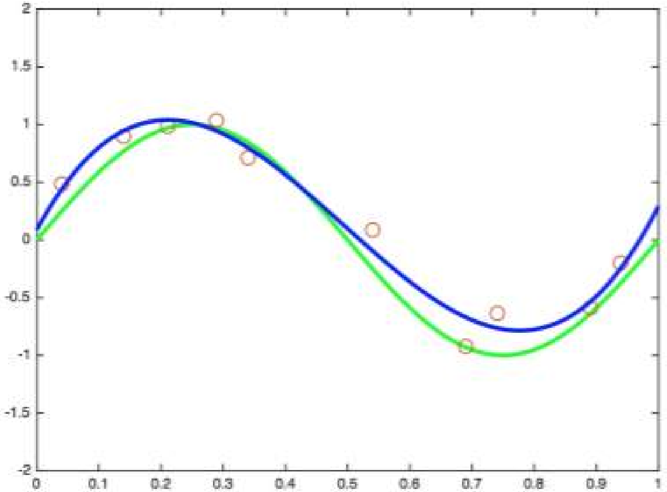
A **regularizer** (which imposes a penalty on the **complexity** of f) is added to the loss function, leading to

$$\underbrace{\sum_{n=1}^{N} \ell(f(\mathbf{x}_n; \mathbf{w}), y_n)}_{\text{loss}} + \lambda \underbrace{R(f)}_{\text{regularizer}} ,$$

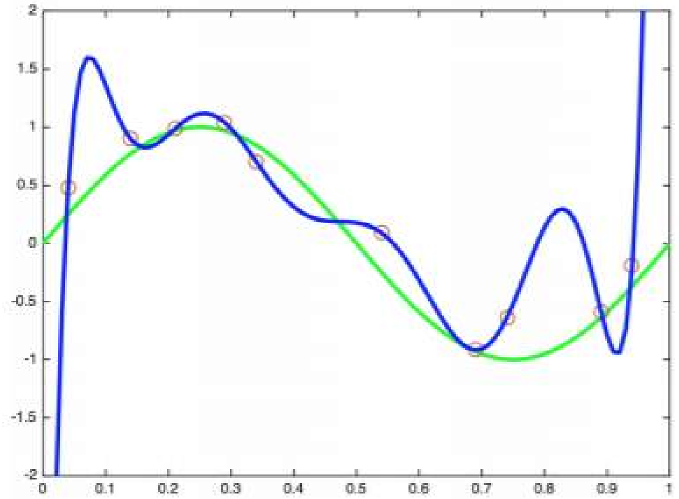
where λ controls the importance of the regularization term (**hyperparameter**)



(a) $M = 1$



(b) $M = 3$



(c) $M = 9$

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
w_0	0.1861	1.0977	0.0880	-8.1
w_1		-1.8913	9.9135	401.2
w_2			-29.8721	-6326.3
w_3			20.1642	49778.9
w_4				-222555.2
w_5				599603.0
w_6				-990507.7
w_7				980248.7
w_8				-532736.3
w_9				122122.1

(d)

Ridge Regression

Ridge Regression

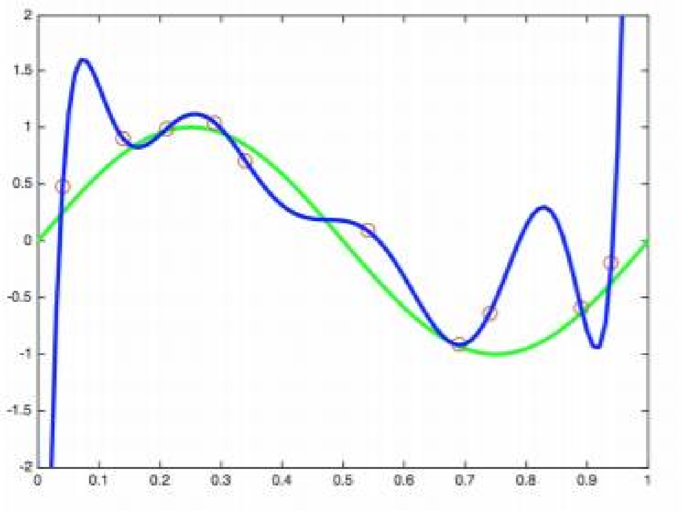
The ridge regression can be written as

$$\min_{\mathbf{w}} \frac{1}{2} \left\| \mathbf{y} - \boldsymbol{\phi}^{\mathrm{T}} \mathbf{w} \right\|_2^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2,$$

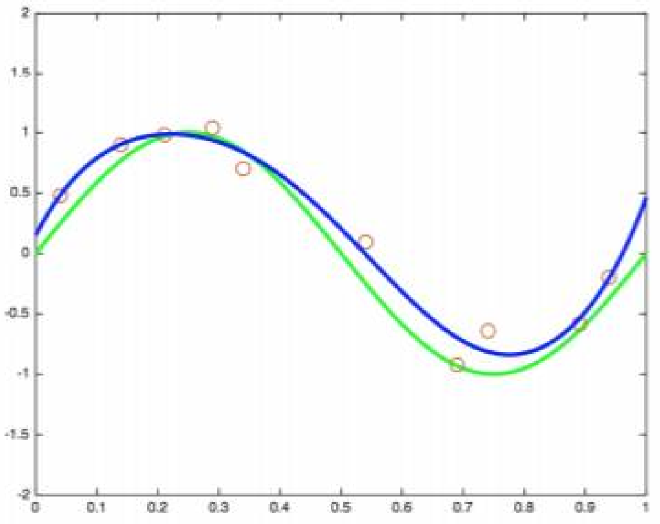
or as a bounded constrained form:

$$\min_{\mathbf{w}} \frac{1}{2} \left\| \mathbf{y} - \boldsymbol{\phi}^{\mathrm{T}} \mathbf{w} \right\|_2^2, \quad \text{s.t.} \quad \|\mathbf{w}\|_2^2 \leq B.$$

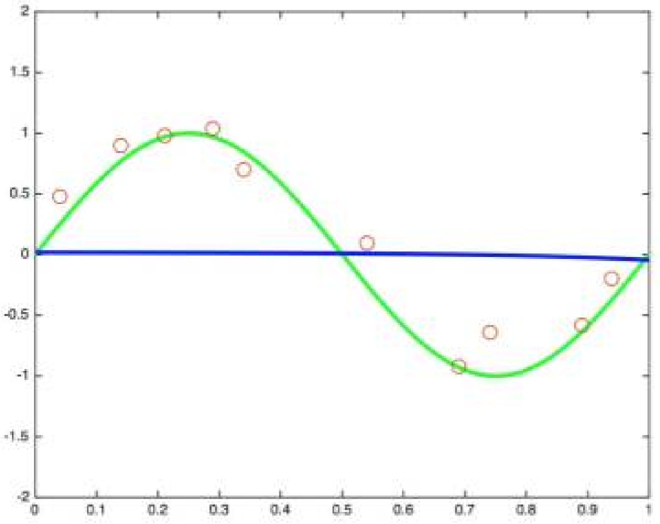
A small (tight) bound B corresponds to the penalty λ and vice versa.



(a) $\log \lambda = -\infty$ $(\lambda = 0)$



(b) $\log \lambda = -18$



(c) $\log \lambda = 0$

	$\log \lambda = -\infty$	$\log \lambda = -18$	$\log \lambda = 0$
w_0	-8.1	0.1503	0.0183
w_1	401.2	9.8564	-0.0083
w_2	-6326.3	-43.3276	-0.0112
w_3	49778.9	98.8418	-0.0101
w_4	-222555.2	-127.4478	-0.0085
w_5	599603.0	-8.6068	. -0.0071
w_6	-990507.7	139.2564	-0.0059
w_7	980248.7	19.9290	-0.0050
w_8	-532736.3	-165.8182	-0.0042
w_9	122122.1	77.6305	-0.0036

(d)

Ridge Regression

$$\min_{\mathbf{w}} \frac{1}{2}\left\|\mathbf{y}-\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}\right\|_2^2+\frac{\lambda}{2}\|\mathbf{w}\|_2^2=\frac{1}{2}\left(\mathbf{y}-\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}\right)^{\mathrm{T}}\left(\mathbf{y}-\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}\right)+\frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$=\frac{1}{2}\left(\mathbf{y}^{\mathrm{T}}\mathbf{y}-2\mathbf{y}^{\mathrm{T}}\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}+\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}\right)+\frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

Then,

$$\frac{\partial}{\partial\mathbf{w}}\left[\frac{1}{2}\left\|\mathbf{y}-\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}\right\|_2^2+\frac{\lambda}{2}\|\mathbf{w}\|_2^2\right]$$

$$=\frac{\partial}{\partial\mathbf{w}}\left[\frac{1}{2}\left(\mathbf{y}^{\mathrm{T}}\mathbf{y}-2\mathbf{y}^{\mathrm{T}}\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}+\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}\right)+\frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\right]$$

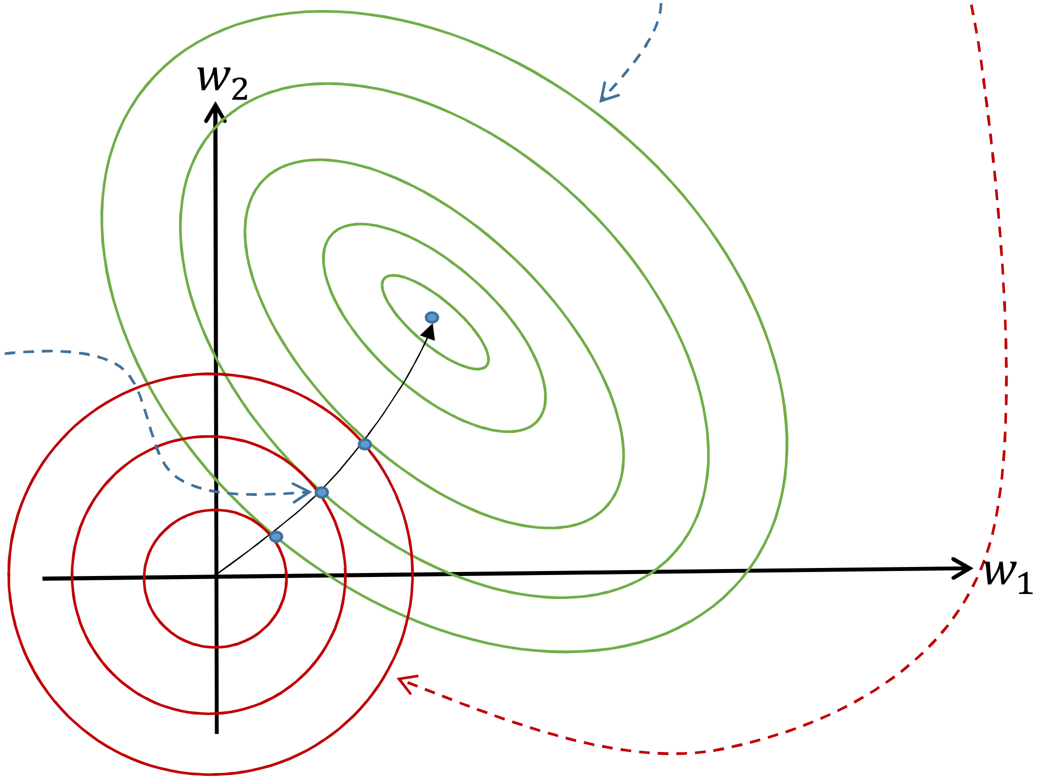
$$=-\boldsymbol{\phi}\mathbf{y}+\boldsymbol{\phi}\,\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}+\lambda\mathbf{w}=-\boldsymbol{\phi}\mathbf{y}+(\boldsymbol{\phi}\,\boldsymbol{\phi}^{\mathrm{T}}+\lambda\mathbf{I})\mathbf{w}$$

Equating to zero yields

$$\mathbf{w}_{ridge}=\left(\boldsymbol{\phi}\,\boldsymbol{\phi}^{\mathrm{T}}+\lambda\mathbf{I}\right)^{-1}\boldsymbol{\phi}\mathbf{y}$$

Ridge Regression: Illustration

Square loss + L_2 norm regularizer: $\underbrace{\frac{1}{2}\left\|\mathbf{y}-\boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}\right\|_2^2}_{\text{LS f}} + \underbrace{\frac{\lambda}{2}\|\mathbf{w}\|_2^2}_{\text{regularizer}}$



Solutions with different values of λ

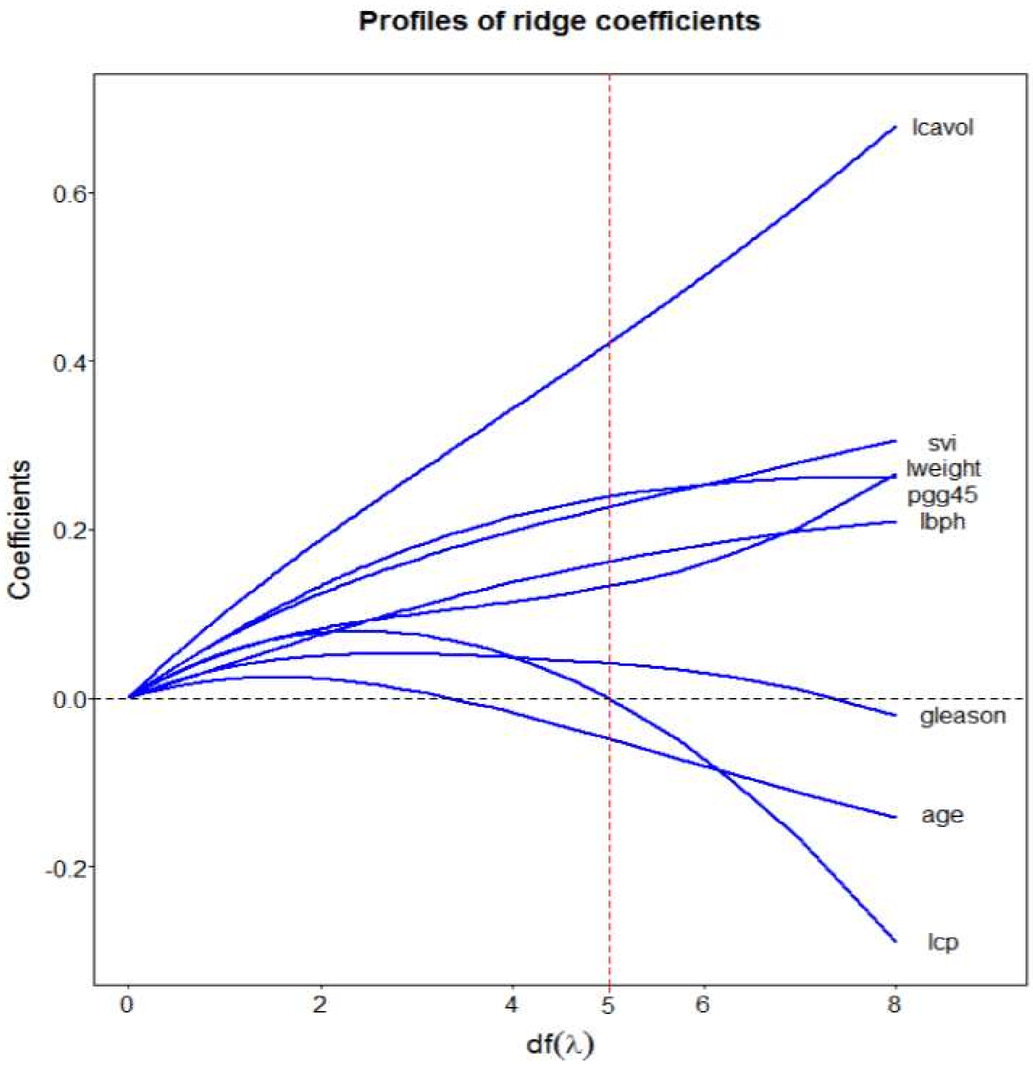
Dataset Description: Prostate Data

A study of 97 men with prostate cancer examined the correlation between (log of) PSA (prostate specific antigen) and a number of clinical measurements (lcavol, lweight, lbph, svi, lcp, gleason, pgg45) and age.

- lcavol: log-cancer volume
- lweight: log prostate weight
- age: age in years
- lbph: log benign prostatic hyperplasia
- svi: seminal vesicle invation
- lcp: log of capsular penetration
- gleason: Gleason score
- pgg45: percent of Gleason scores 4 or 5

Stamey, T.A., Kabalin, J.N., McNeal, J.E., Johnstone, I.M., Freiha, F., Redwine, E.A. and Yang, N. (1989)

Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate: II. radical prostatectomy treated patients, Journal of Urology 141(5), 1076–1083.



df(λ): effective degree of freedom ($\propto \frac{1}{\lambda}$)

LASSO

- Robert Tibshirani (1996), ”Regression shrinkage and selection via the LASSO,” Journal of the Royal Statistical Society. Series B (Methodological).

Not satisfied with OLS estimates?

- Prediction accuracy
  - Often have low bias but large variance
  - Can sometimes improve the prediction accuracy by shrinking or setting to zero some coefficients. (sacrifice a little bias to reduce the variance of the predicted values)
- Interpretability
  - Often would like to determine a smaller subset of covariates that exhibit the strongest effects.
  - The larger the number of covariates is, the less interpretable the model is.

LASSO (Least Absolute Selection and Shrinkage operator)

The LASSO regression can be written as

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \left\|\mathbf{y} - \mathbf{X}^{\mathrm{T}}\boldsymbol{\theta}\right\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1,$$

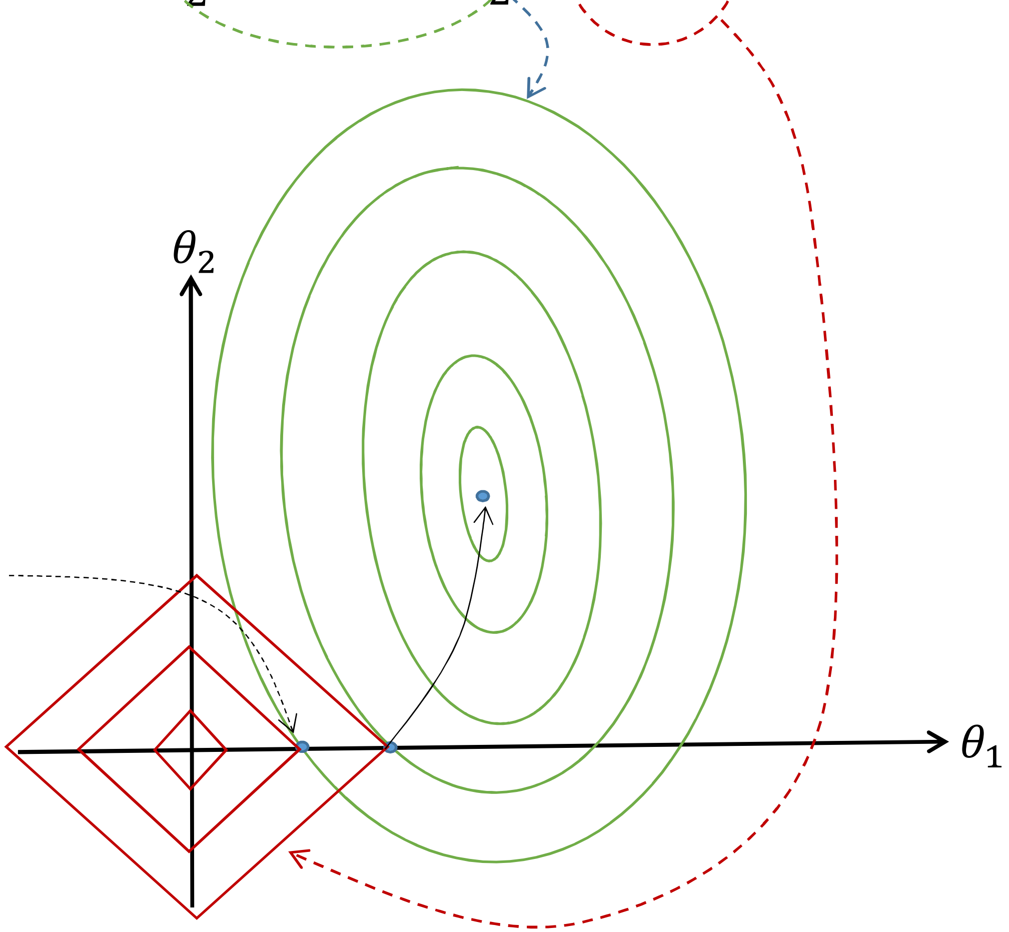
or as a bounded constrained form (a quadratic function with linear constraints):

$$\min_{\boldsymbol{\theta}} \tfrac{1}{2} \left\|\mathbf{y} - \mathbf{X}^{\mathrm{T}}\boldsymbol{\theta}\right\|_2^2, \quad \text{s.t.} \;\; \|\boldsymbol{\theta}\|_1 \leq B.$$

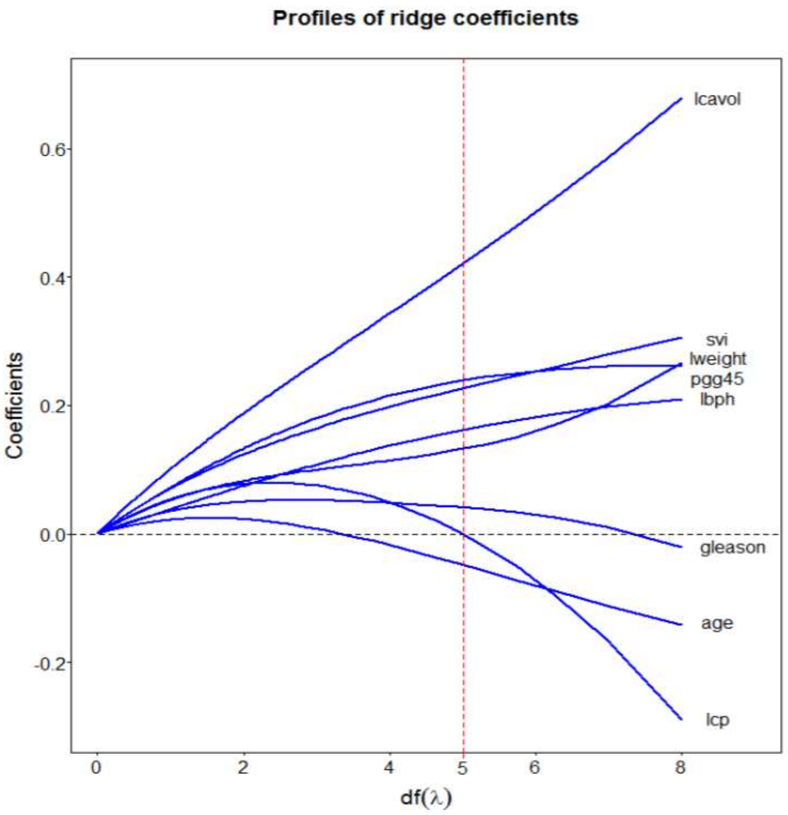
A small (tight) bound B corresponds to the penalty λ and vice versa.

LASSO: Illustration

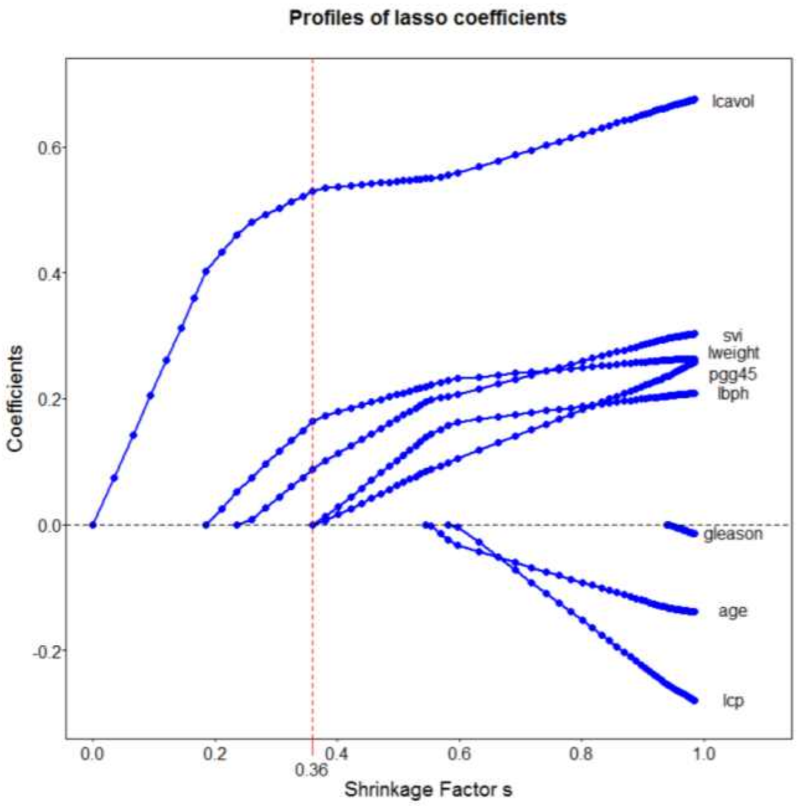
Square loss + L_1 norm regularizer: $\frac{1}{2}\left\|\mathbf{y}-\mathbf{X}^{\mathrm{T}}\boldsymbol{\theta}\right\|_2^2+\lambda\|\boldsymbol{\theta}\|_1$



Solutions with different values of λ



(a) Ridge



(a) LASSO

LASSO calculation

Now we calculate the derivative of

$$\frac{1}{2}\left\|\mathbf{y}-\mathbf{X}^{\mathrm{T}}\boldsymbol{\theta}\right\|_2^2+\lambda\|\boldsymbol{\theta}\|_1,$$

to find the LASSO solution.

$\|\boldsymbol{\theta}\|_1 = |\theta_1| + |\theta_2| + \cdots$ is not differentiable!

Coordinate descent: optimize one by one

To this end, we rewrite the objective function:

$$\mathcal{J}_{LASSO} = \frac{1}{2}\sum_{n=1}^{N}\left(y_n - \mathbf{x}_n^{\mathrm{T}}\boldsymbol{\theta}\right)^2 + \lambda\sum_{d=1}^{D}|\theta_d|$$

$$= \frac{1}{2}\sum_{n=1}^{N}\left(y_n - x_{n,d}\theta_d - \mathbf{x}_{n,-d}^{\mathrm{T}}\boldsymbol{\theta}_{-d}\right)^2 + \lambda\sum_{d'=1}^{D}|\theta_{d'}|$$

Calculate the derivative:

$$\frac{\partial}{\partial\theta_d}\mathcal{J}_{LASSO} = \sum_{n=1}^{N}\left(y_n - x_{n,d}\theta_d - \mathbf{x}_{n,-d}^{\mathrm{T}}\boldsymbol{\theta}_{-d}\right)(-x_{n,d}) + \lambda\frac{\partial|\theta_d|}{\partial\theta_d}$$

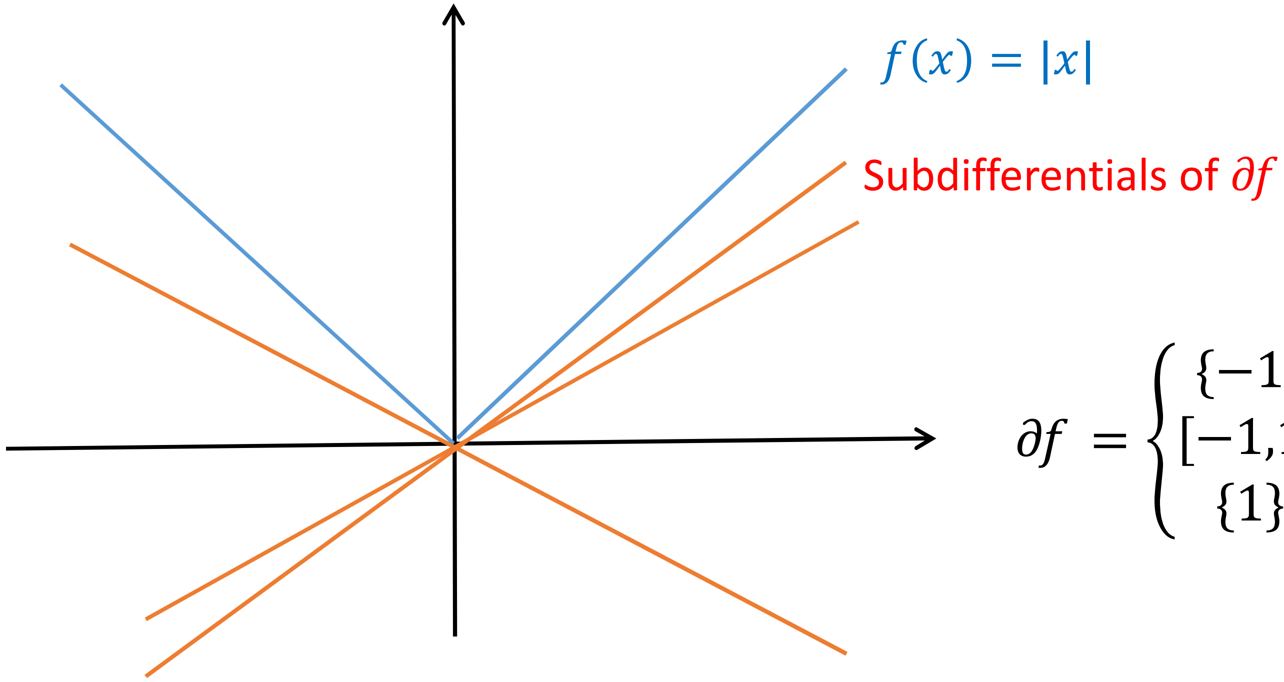
$$= \underbrace{\left(\sum_{n=1}^{N}x_{n,d}^2\right)}_{\alpha_d}\theta_d - \underbrace{\sum_{n=1}^{N}\left(y_n - \mathbf{x}_{n,-d}^{\mathrm{T}}\boldsymbol{\theta}_{-d}\right)x_{n,d}}_{\beta_d} + \lambda\frac{\partial|\theta_d|}{\partial\theta_d}$$

Subdifferentials

The subdifferential (subderivative, subgradient) $\partial f(x_0)$ of a convex function f at a point x_0 is the set defined by

$$\partial f(x_0) = \{z \in \mathbb{R} | f(x) - f(x_0) \geq z(x - x_0), \forall x \in \mathbb{R}\}.$$

As a special case, if $f(x_0)$ is differentiable, then $\partial f(x_0) = \{f'(x_0)\}$.



$$\partial f = \begin{cases} \{-1\} & \text{if } x < 0 \\ [-1,1] & \text{if } x = 0 \\ \{1\} & \text{if } x > 0 \end{cases}$$

Thus, we have

$$\partial\mathcal{J}_{LASSO} = \alpha_d\theta_d - \beta_d + \lambda\partial|\theta_d| = \begin{cases} \{\alpha_d\theta_d - \beta_d - \lambda\} & \text{if } \theta_d < 0 \\ [-\beta_d - \lambda, -\beta_d + \lambda] & \text{if } \theta_d = 0 \\ \{\alpha_d\theta_d - \beta_d + \lambda\} & \text{if } \theta_d > 0 \end{cases}$$

Thus, the estimate of θ_d given the other parameters is calculated as:

$$\hat{\theta}_d = \begin{cases} \dfrac{\beta_d + \lambda}{\alpha_d} & \text{if } \beta_d < -\lambda \\ 0 & \text{if } \beta_d \in [-\lambda, \lambda] \\ \dfrac{\beta_d - \lambda}{\alpha_d} & \text{if } \beta_d > \lambda \end{cases},$$

where

$$\alpha_d = \sum_{n=1}^{N} x_{n,d}^2 ,$$

$$\beta_d = \sum_{n=1}^{N} \left(y_n - \mathbf{x}_{n,-d}^{\mathrm{T}}\boldsymbol{\theta}_{-d}\right)x_{n,d} .$$

Shooting Algorithm

The coordinate descent algorithm for LASSO, is also knowns as shooting algorithm.

- I W. J. Fu (1998), ”Penalized regressions: The bridge versus the LASSO,” Journal of Computational and Graphical Statistics.
- I T. T. Wu and K. Lange (2008), ”Coordinate descent algorithms for LASSO penalized regression,” The Annals of Applied Statistics.

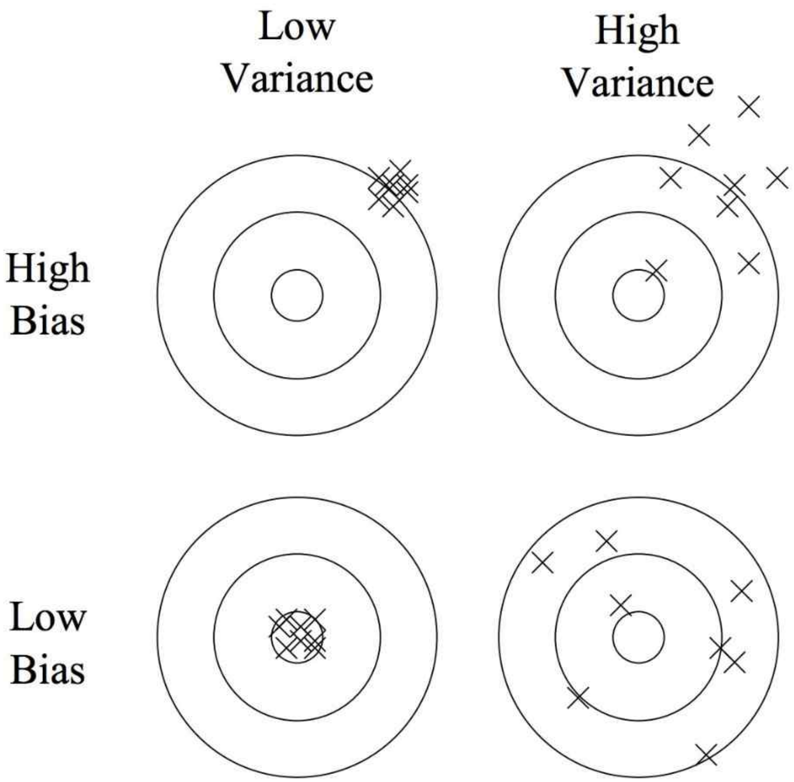
Algorithm: Coordinate Descent for Sparse Regression

```
Input: Initialize parameters $\boldsymbol{\theta}$ (e.g. use $\boldsymbol{\theta}_{LS} = \left(\mathbf{X}\mathbf{X}^{\mathrm{T}}\right)^{-1}\mathbf{X}\mathbf{y}$)
• repeat
    • for $d = 1,2,\ldots,D$ do
        • Compute $\alpha_d = \sum_{n=1}^{N} x_{n,d}^2$
        • Compute $\beta_d = \sum_{n=1}^{N} \left(y_n - \mathbf{x}_{n,-d}^{\mathrm{T}} \boldsymbol{\theta}_{-d}\right) x_{n,d}$
        • if $\beta_d < -\lambda$ then $\theta_d = \frac{\beta_d + \lambda}{\alpha_d}$
        • else if $\beta_d > \lambda$ then $\theta_d = \frac{\beta_d - \lambda}{\alpha_d}$
        • else $\theta_d = 0$
        • end if
    • end for
• until convergence
• return $\boldsymbol{\theta}_{LASSO} = [\theta_1, \ldots, \theta_D]^{\mathrm{T}}$
```

Bias-Variance Trade-off

There is a trade-off between bias and variance:

- Flexible models: low bias but high variance
- Rigid models: high bias but low variance


Low Variance
High Variance
High Bias
Low Bias