Introduction to Machine Learning.
CSCI-UA 9473, Lecture 3.

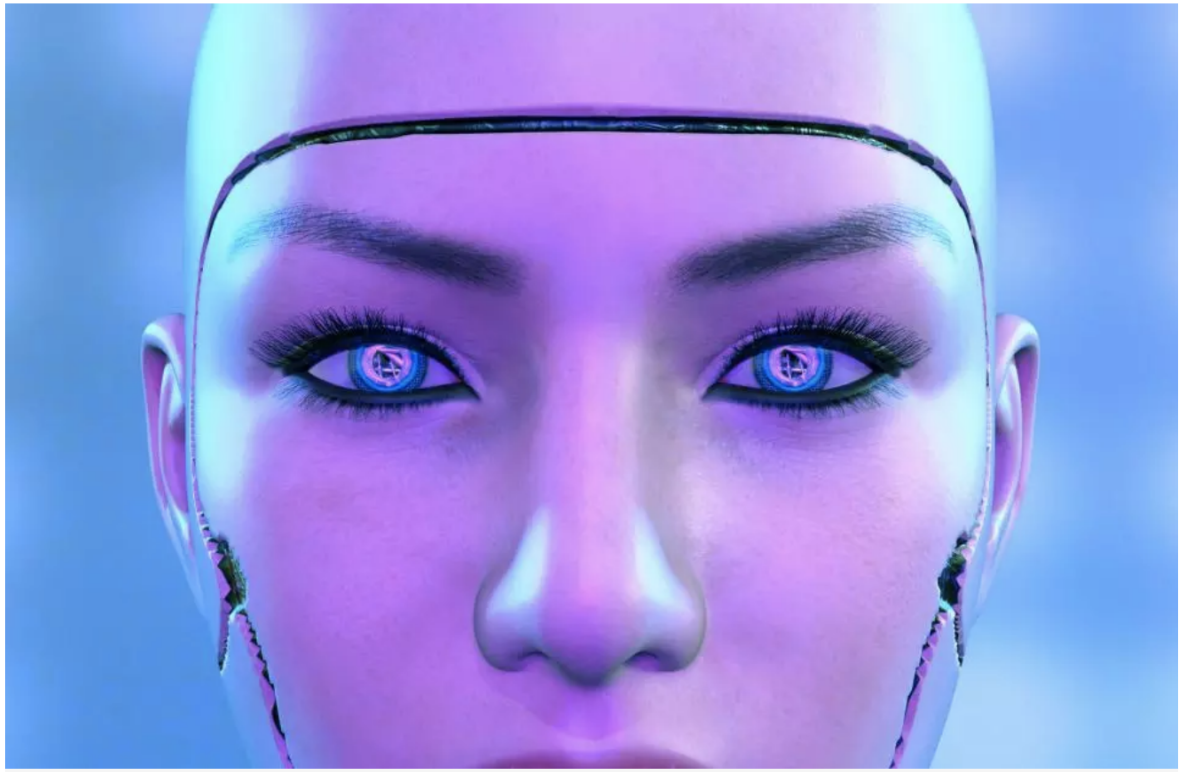
Augustin Cosse

Ecole Normale Supérieure, DMA & NYU
Fondation Sciences Mathématiques de Paris.

2018

Why study ML? One more cover from this week

Newsweek



The incoming president of the British Science Association has warned artificial intelligence is a bigger threat to national security than terrorism or climate change.

GETTY IMAGES

Fears that the rise of automation and AI, known as Fourth Industrial Revolution, will endanger jobs is also warranted, he said. His concerns are mirrored by a November 2017 report by the management consulting firm McKinsey, which estimated 50 percent of current work could be automated as soon as 2030.

A few reminders

- Supervised vs Unsupervised
  - Supervised: you are given patterns of the form $\{\boldsymbol{x}_\mu, y_\mu\}$ (both points and labels). The algorithm has to learn the model $f(\boldsymbol{x}_\mu) = y_\mu$.
  - Unsupervised: you are given data of the form $\boldsymbol{x}_\mu$, the algorithm is asked to extract some meaningful structure (such as clusters) from the data.

Previous lecture

- Data distribution in nature are often highly complex
- Learning = understand the distribution from a few samples
- Two possible statistical approaches :
  - Bayesian : maximizes the posterior and relies on the definition of a prior
  - Frequentist : no prior but estimation through repeated samples (sampling distribution)

This week

- Bayesian vs Frequentist (quick review)
- Linear Regression
- Linear classification
- Regularization
- Python !

Linear regression

- Linear models assumes that the regression function is linear in the inputs $X_1, \ldots, X_m$

$$E(Y|X) = f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$$

- Note that the model is linear in the inputs X_k but can be used for polynomial representation by introducing additional variables $X_2 = X_1^2$, $X_3 = X_1^3$, ...
- Or even other non linear transformations such as log, square root,...

Linear regression: Fitting

- Given a set of training data, $(X_1, y_1), (X_2, y_2), \ldots, (X_n, y_n)$, the most popular approach (or choice of loss) is to minimize the residual sum of squares (RSS)

$$\begin{aligned} RSS(\beta) &= \sum_{i=1}^{N} (y_i - f(X_i)) \\ &= \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} X_{i,j}\beta_j)^2 \end{aligned}$$

Linear regression: Fitting

- To fit the model β to the data (X_i, y_i), we look for the parameters that minimizes the RSS

$$RSS(\beta) = (\boldsymbol{y} - \boldsymbol{X}\beta)^T(\boldsymbol{y} - \boldsymbol{X}\beta)$$

- Here X is the N by $p-1$ matrix whose i^{th} row is defined as $[1, \boldsymbol{X}_i]$

- To find this minimum, we set the first order derivative to zero and check second order derivative to make sure we have a minimum

$$\frac{\partial \text{RSS}}{\partial \beta} = -2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\beta)$$

$$\frac{\partial^2 \text{RSS}}{\partial \beta \partial \beta^T} = 2\boldsymbol{X}^T\boldsymbol{X}$$

Linear regression: Fitting

$$\frac{\partial \text{RSS}}{\partial \beta} = -2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\beta)$$
$$\frac{\partial^2 \text{RSS}}{\partial \beta \partial \beta^T} = 2\boldsymbol{X}^T\boldsymbol{X}$$

- When all eigenvalues of $2\boldsymbol{X}^T\boldsymbol{X}$ are positive, the problem has a minimum and we can write

$$-2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\beta) = 0 \;\Rightarrow\; \beta = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Linear regression: Fitting

From Hastie, Tibshirani, Friedman, The elements of statistical learning.

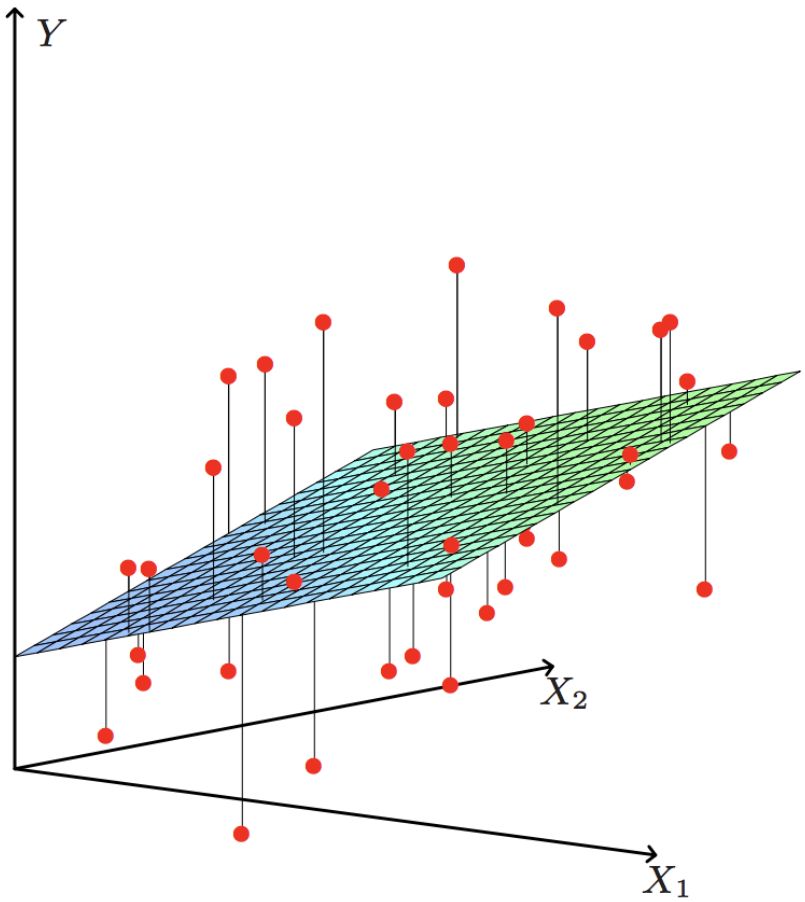


FIGURE 3.1. *Linear least squares fitting with* $X \in \mathbb{R}^2$. *We seek the linear function of* X *that minimizes the sum of squared residuals from* Y.

Intuition

- The fitted values at the training inputs are then given by

$$\hat{y} = \boldsymbol{X}\hat{\beta} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

- The idea behind the linear (RSS) regression model is that we want the residuals to be orthogonal to the subspace spanned by the X_i
- If most of the variability of the input data occurs in a given direction, we want to minimize the error along that direction
- So we ask $\boldsymbol{X}^T(y - \boldsymbol{X}\beta) = 0$

Gauss Markov and the Bias variance tradeoff (I)

- Assume we want to estimate any linear function of β (can be β itself)
- The RSS/LS estimate for $a^T\beta$ is

$$\hat{\theta} = a^T\hat{\beta} = a^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

- If the linear model is correct ($\mathbb{E}y = \boldsymbol{X}\beta$)

$$\begin{aligned}\mathbb{E}\left\{a^T\hat{\beta}\right\} &= \mathbb{E}\left\{a^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}\right\} \\ &= a^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}\beta \\ &= a^T\beta\end{aligned}$$

- We say that the estimator is unbiased

Gauss Markov and the Bias variance tradeoff (II)

- Gauss-Markov: for any other linear (in $\boldsymbol{y}$) estimator $\tilde{\theta} = \boldsymbol{c}^T \boldsymbol{y}$ that is unbiased, we necessarily have

$$\text{Var}(\hat{\theta}) = \text{Var}(\boldsymbol{a}^T \hat{\beta}) \leq \text{Var}(\tilde{\theta}) = \text{Var}(\boldsymbol{c}^T \boldsymbol{y})$$

- Is the minimal variance always a good idea ?
- For a given estimator, the mean square error (MSE) is defined as

$$\begin{aligned} \text{MSE}(\tilde{\theta}) &= \mathbb{E}\left\{(\tilde{\theta} - \theta)^2\right\} \\ &= \text{Var}(\tilde{\theta}) + \left(\mathbb{E}\left\{\tilde{\theta} - \theta\right\}\right)^2 \end{aligned}$$

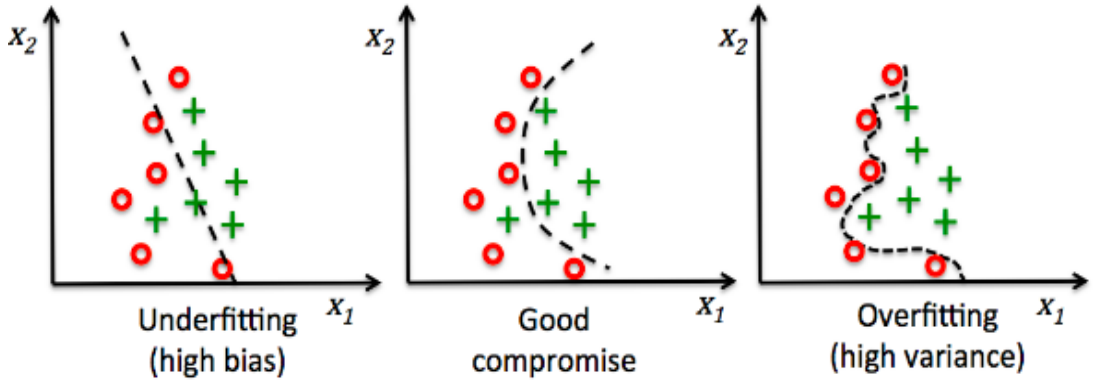
- Least squares estimator has the smallest variance among all estimators with no bias but there might exists estimators with bias that have much smaller MSE

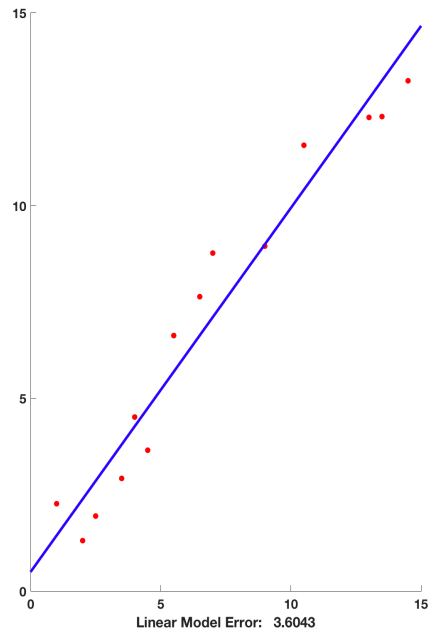
Gauss Markov and the Bias variance tradeoff (II)

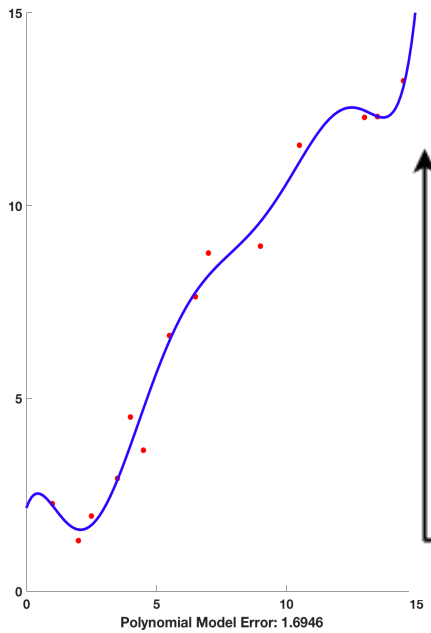
- For the particular choice $\tilde{\theta} = f(x_0) = x_0^T \beta$ (x_0 in the test set), we get a measure of the error on future predictions

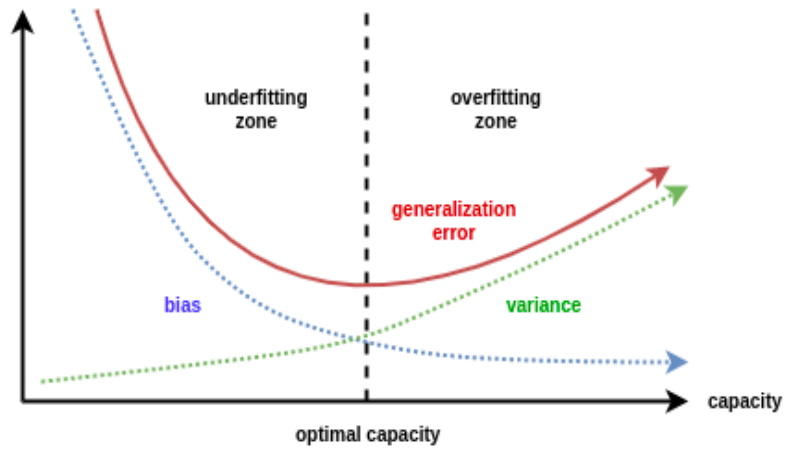
$$E(\beta) = \text{Bias}^2(\hat{f}(x_0)) + \text{Var}(\hat{f}(x_0))$$

- LS estimator has smallest variance among all estimators with no bias
- But this variance (and thus the prediction error) can be large..
- This is the case when variables are correlated. Then a large β_ℓ in one variable can be canceled by another large negative β_k multiplying a correlated variable. Exact 0 bias might introduce large variance.


X_2
X_1
Underfitting
(high bias)
X_2
X_1
Good
compromise
X_2
X_1
Overfitting
(high variance)


15
10
5
0
0
5
10
15
Linear Model Error: 3.6043


15
10
5
0
0
5
10
15
Polynomial Model Error: 1.6946


underfitting
zone
overfitting
zone
generalization
error
bias
variance
capacity
optimal capacity

Two issues

- Two reasons why linear regression is often not satisfying
  - Prediction accuracy. Linear regression models have low bias (good on average) but large variance
  - Interpretation. When doing prediction it would be good to target a smaller subset of the data which contains the meaningful information about future values
- Two possible approaches
  - Subset selection
  - Shrinkage methods

Can we do better: Subset selection

- Consider data of dimension d, $\{(X_\mu, y_\mu)\}_\mu$ with $X \in \mathbb{R}^d$
- Best subset selection: for each $k \in \{1, 2, \ldots, p\}$, select the subset S of size k which gives the smallest residual SOS for β_S

$$\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j \in S} X_{i,j}\beta_j)$$

- Main problem with subset selection : hard in large dimension (requires enumerating all subsets)

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\text{argmin}} \sum_{\beta\ i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

$$\text{subject to } \|\boldsymbol{\beta}\|_{\ell_0} \leq k$$

- One efficient algorithm for $p \leq 40$: Leaps and Bound procedure

Can we do better: Subset selection (II)

- Can we find a more tractable alternative to the best subset ?
- Start with the constant model β_0, then add to the model the predictor β_k that most improves the fit.
- This (greedy) idea is known as Forward Stepwise selection
- Usually improves over best subset selection on at least two factors
  - Computational. Clearly for large p finding the best subset is intractable. Computing the forward stepwise sequence is feasible.
  - Statistical: exactly finding the best subset will be biased by the training set and is likely to lead high variance, whether the forward stepwise approach which is more constrained will have lower variance

Can we do better: Subset selection (II)

- The method comes in two flavors
  - Forward Stepwise selection (start from the intercept and gradually add more predictors)
  - Backward Stepwise selection (start from the full set of predictors and gradually remove the coefficient which has the least impact on the fit = coefficient with the smallest Z score)

$$\text{(Z-Score)} \quad Z_j = \frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{v_j}}$$

where v_j is the j^{th} diagonal element of $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ and $\hat{\sigma}^2$ is the empirical variance of the y_i

$$\hat{\sigma}^2 = \frac{1}{N-p-1}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Here $\hat{y}_i$ are the predictions.

Can we do better: Shrinkage methods

- Introducing a small bias, might lead to a decrease in the Variance and by extension to the prediction error
- How : Penalize large values of β
- Several approaches:
  - Ridge regression
  - LASSO

Ridge regression

- Ridge regression imposes a SOS penalty on the size of the regression coefficients

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\text{argmin}}_{\beta} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

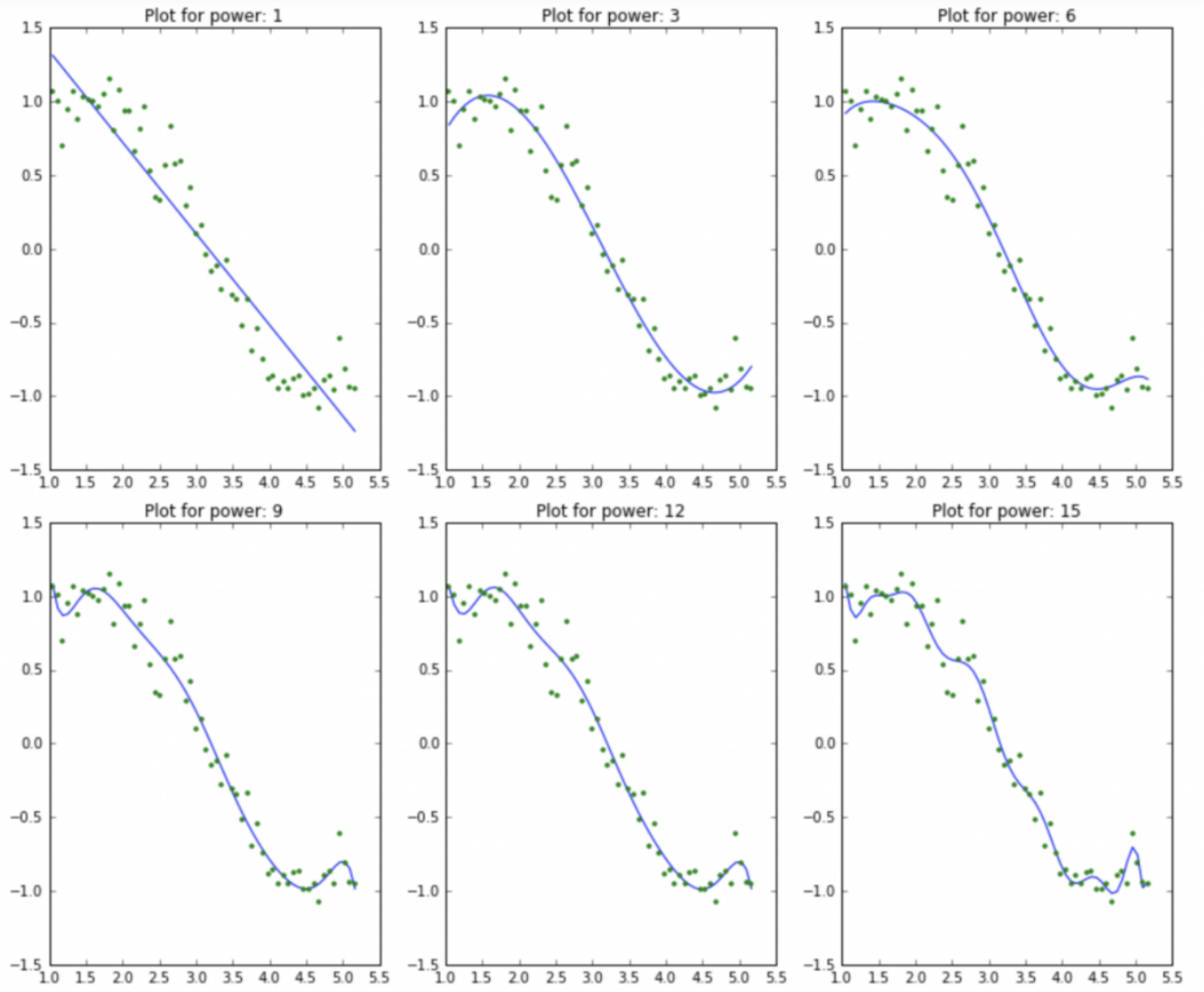
An alternative reformulation is

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\text{argmin}}_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

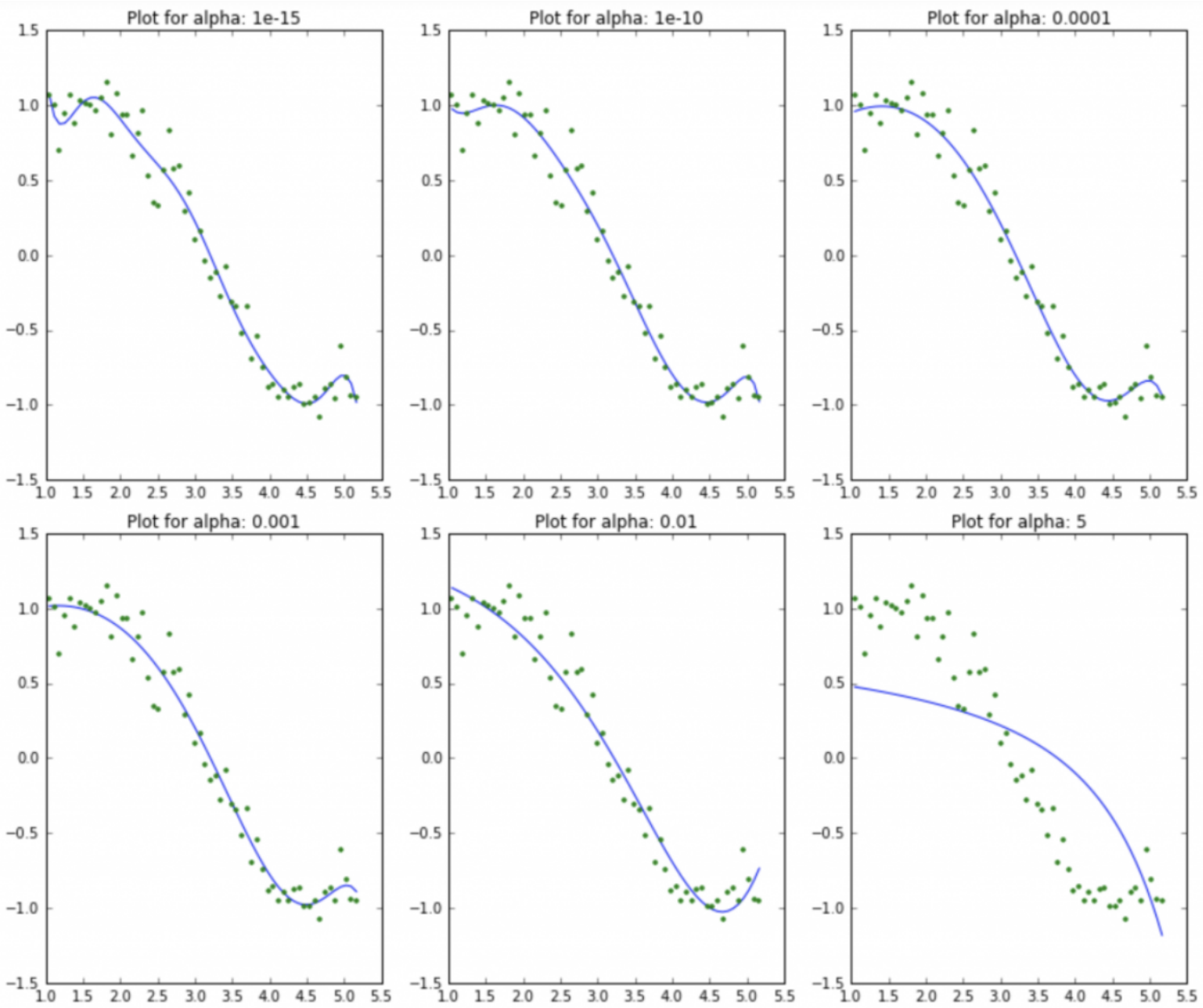
$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 \leq t$$

- This approach is known as weight decay in Neural networks

from A Jain, *A Complete Tutorial on Ridge and Lasso Regression in Python*



from A Jain, *A Complete Tutorial on Ridge and Lasso Regression in Python*



Basis Pursuit/LASSO

- The Lasso estimate is defined as

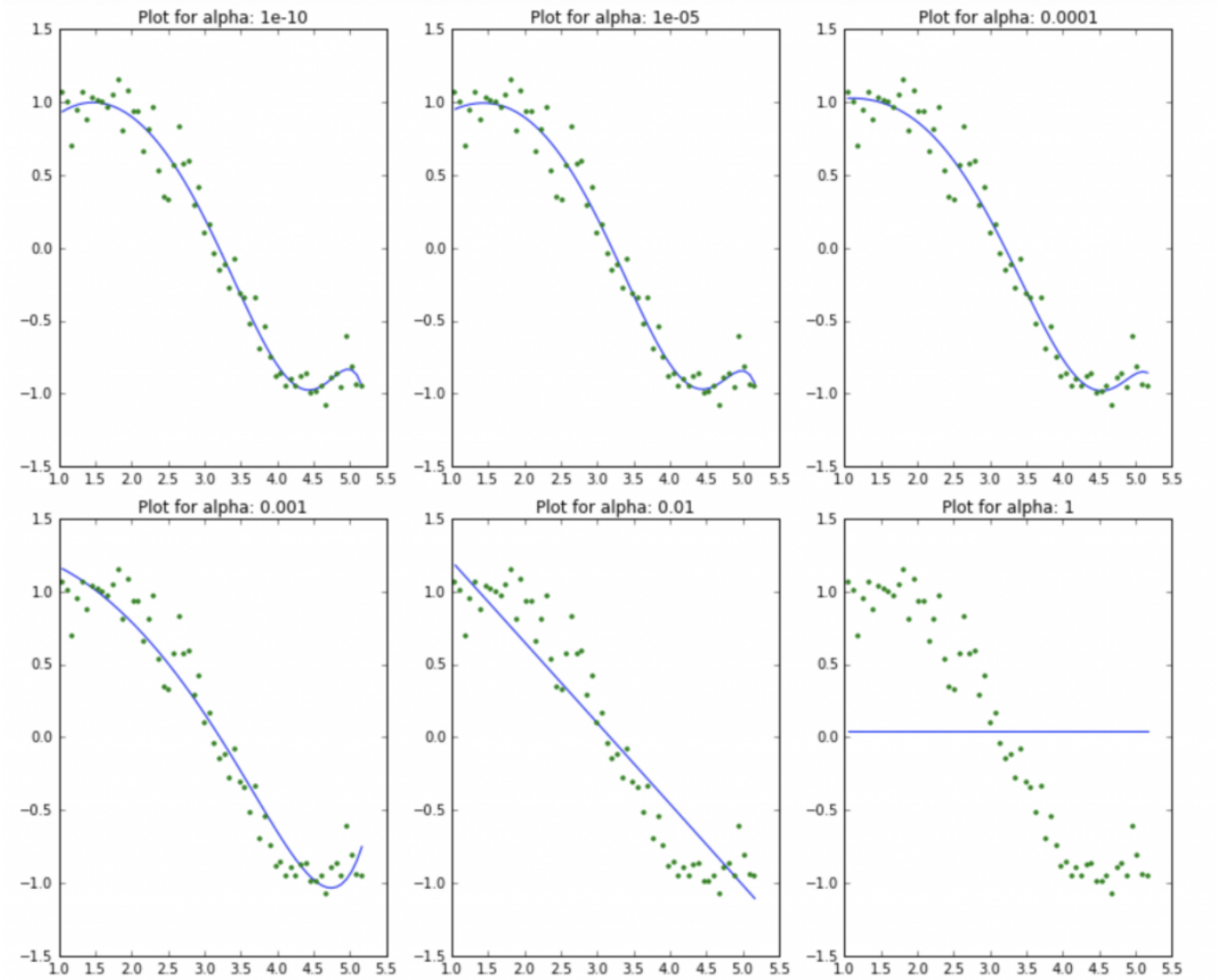
$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\text{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

An alternative reformulation is

$$\hat{\beta}^{\text{lasso}} = \underset{x}{\text{argmin}} \underset{\beta}{} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

$$\text{subject to } \sum_{j=1}^{p} |\beta_j| \leq t$$

from A Jain, *A Complete Tutorial on Ridge and Lasso Regression in Python*



LASSO vs Ridge

- Because of the nature of the constraint, the LASSO does some sort of subset selection
- When t is chosen sufficiently small, some of the coefficient that were small in Ridge regression will be set exactly to zero by the Lasso formulation.

LASSO vs Ridge (H, T, F)

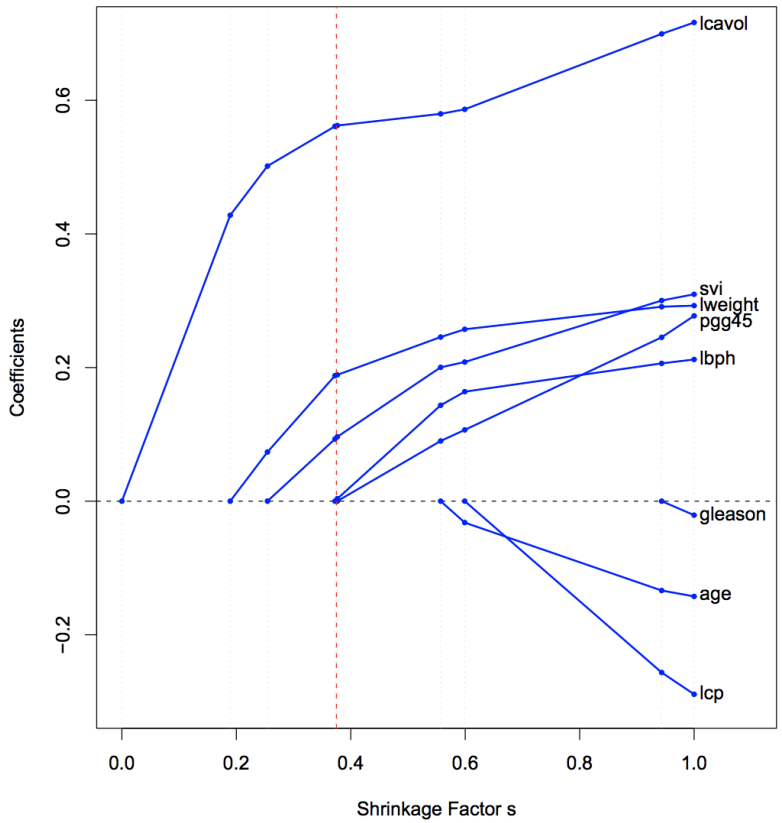


FIGURE 3.10. *Profiles of lasso coefficients, as the tuning parameter t is varied. Coefficients are plotted versus $s = t/\sum_1^p |\hat{\beta}_j|$. A vertical line is drawn at $s = 0.36$, the value chosen by cross-validation. Compare Figure 3.8 on page 65; the lasso profiles hit zero, while those for ridge do not. The profiles are piece-wise linear, and so are computed only at the points displayed; see Section 3.4.4 for details.*

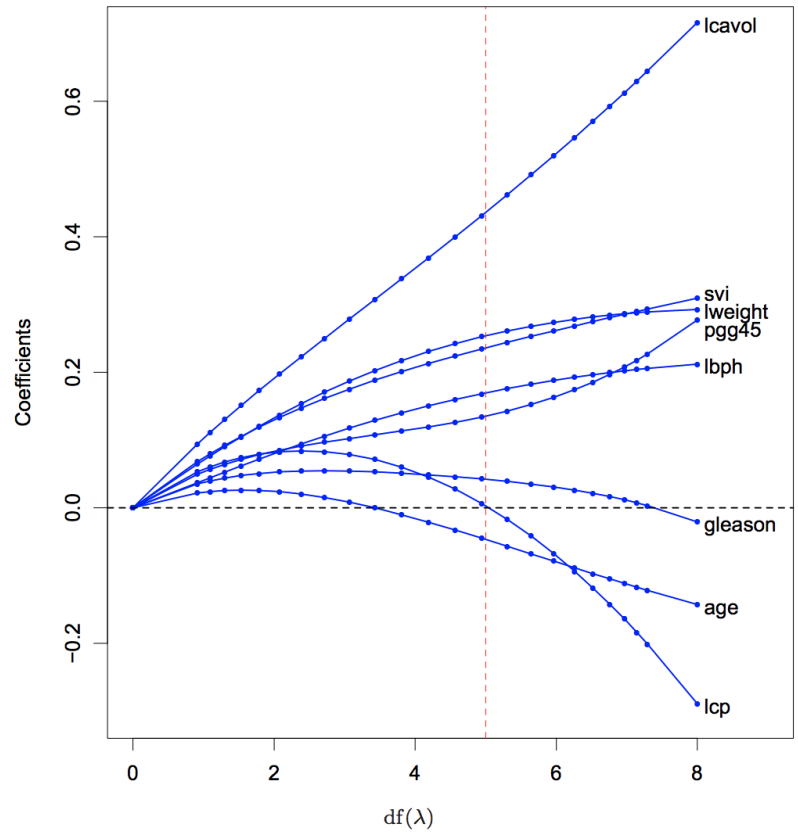


FIGURE 3.8. *Profiles of ridge coefficients for the prostate cancer example, as the tuning parameter λ is varied. Coefficients are plotted versus $\mathrm{df}(\lambda)$, the effective degrees of freedom. A vertical line is drawn at $\mathrm{df} = 5.0$, the value chosen by cross-validation.*

LASSO vs Ridge (H, T, F)

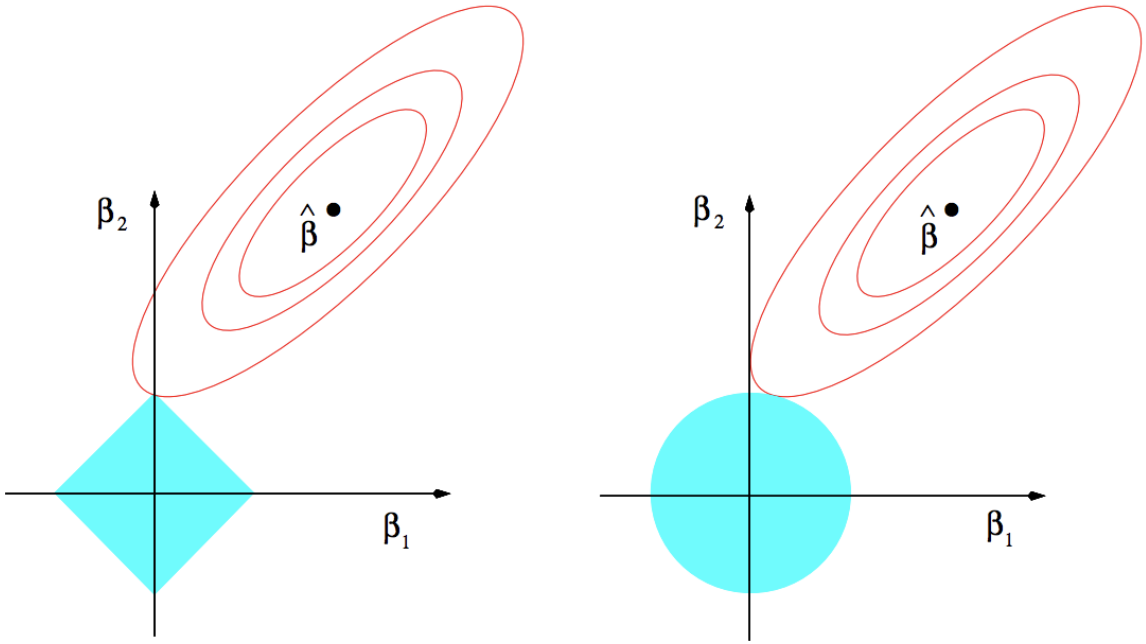


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

Going back to frequentist vs Bayesian

- Assume that we measure labels which can be expressed as a linear function of the data up to some gaussian noise

$$y_j = f(X_{ij}) + \sigma_j, \quad \sigma_i \sim \mathcal{N}(0, \sigma^2)$$

- In particular, we now know that y_j has a gaussian distribution with mean $f(X_{ij})$ and variance σ^2

$$p(Y|X, \beta) \propto \prod_j \exp\left(-\frac{|y_j - f(X_j)|^2}{2\sigma^2}\right) \tag{1}$$

- taking the log, we get the log-likelihood

$$-\log p(Y|X, \beta) = \sum_{j=1}^{N} \frac{|y_j - f(X_j)|^2}{2\sigma^2}$$

Going back to frequentist vs Bayesian

- This (i.e RSS) approach corresponds to a uniform prior on the parameters.
- Now we want to add a prior.
- Let us assume for example that the weights should follow independent Gaussian distributions

$$\beta \sim \mathcal{N}(0, \frac{1}{\lambda} I)$$

$$P(\beta) = \left(\frac{\lambda}{2\pi}\right)^{n/2} \exp\left(-\frac{\lambda}{2}\beta^T\beta\right) \tag{2}$$

Going back to frequentist vs Bayesian

- If we add this to the MLE framework, and take the log, we get the (Bayesian) MAP estimator

$$\hat{\beta}_{MAP,gaussian} = \underset{\beta}{\text{argmin}} \sum_{j=1}^{N} (y_j - f(X_j))^2 + \frac{\lambda}{2} \beta^T \beta$$

- Does that remind you something?
- The ridge regression estimate is the Bayesian estimator with Gaussian prior,

$$\hat{\beta}_{MAP,gaussian} = \hat{\beta}_{ridge}$$

Going back to frequentist vs Bayesian

- Instead of a Gaussian prior, assume that the weights follow independent Laplace distributions

$$\boldsymbol{\beta} \sim \text{Laplace}(0, \lambda \boldsymbol{I})$$

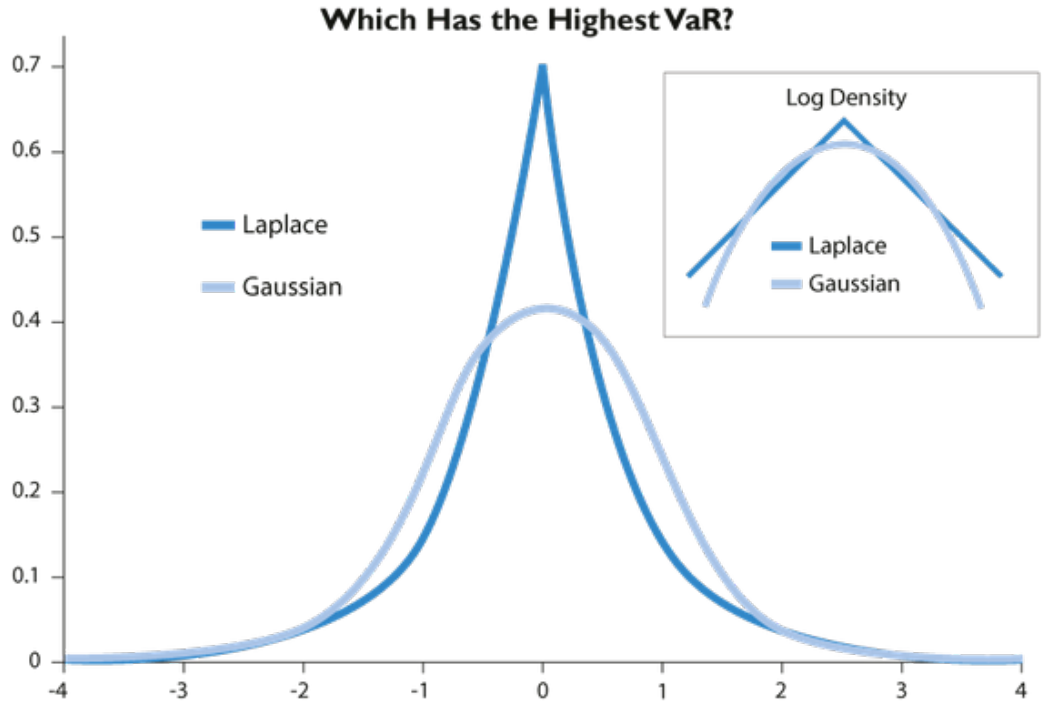
$$P(\boldsymbol{\beta}) = 2\lambda \exp(-\lambda|\beta|)$$

- let us substitute this in the expression for the posterior $p(y|X, \beta)p(\beta)$, and take the log

$$\hat{\beta}_{MAP,Laplace} = \underset{\beta}{\text{argmin}} \sum_{j=1}^{N}(y_j - f(X_j))^2 + \frac{\lambda}{2}\sum_{i=1}^{d}|\beta_i|$$

- The LASSO estimator is equivalent to do Bayesian inference with a Laplace prior, $\hat{\beta}_{MAP,Laplace} = \hat{\beta}_{LASSO}$

Going back to frequentist vs Bayesian



Many other choices of priors/regularizers are possible

- We can generalize the LASSO and Ridge regression models to other "log prior densities"

$$\hat{\beta} = \underset{\beta}{\text{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^p \right\}$$

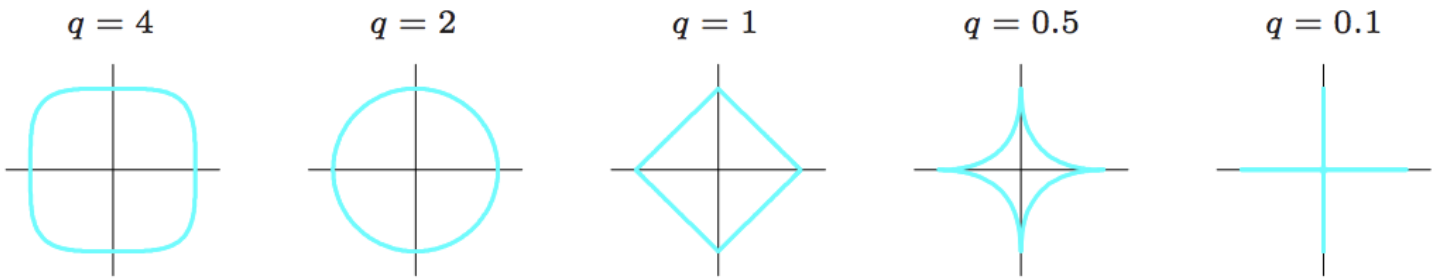


FIGURE 3.12. *Contours of constant value of* $\sum_j |\beta_j|^q$ *for given values of* q.

From Hastie, Tibshirani, Friedman