Lecture 2 Linear Regression and Regularization

Shiwei Lan[1]

[1]School of Mathematical and Statistical Sciences
Arizona State University

STP598 Machine Learning and Deep Learning
Fall 2021

Table of Contents

- Observe a collection of i.i.d. training data

$$\mathcal{D}_n = \{x_i, y_i\}_{i=1}^n$$

where each x_i is a p dimensional vector (prediction variables, covariates, features, inputs), i.e.

$$x_i = (x_{i1}, \ldots, x_{ip})^\mathsf{T}$$

and $y_i \in \mathbb{R}$ is a continuous response (outcome, output).

- We want to estimate $f(X)$ using the training data to describe the relationship between X and Y.

- To clarify some other notations:
- $\mathbf{x}_j$ is an n dimensional vector of the jth feature, i.e.

$$\mathbf{x}_j = (x_{1j}, x_{2j}, \ldots, x_{nj})^{\mathrm{T}}$$

- The design matrix $\mathbf{X}$ is $n \times p$ dimensional,

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p)$$

- To estimate $f(X)$, we need to define a criterion for a good estimator, $\widehat{f}(\cdot)$.
- We define a loss function L that measures the discrepancies between Y and $f(X)$. For regression, a commonly used loss function is the squared error loss:

$$L\big(Y, f(X)\big) = \big(Y - f(X)\big)^2.$$

- Risk is the expected loss over the entire population

$$\mathrm{R}(f) = \mathrm{E}\left[L\big(Y, f(X)\big)\right] = \mathrm{E}\left[\big(Y - f(X)\big)^2\right].$$

Minimizing the Empirical Risk

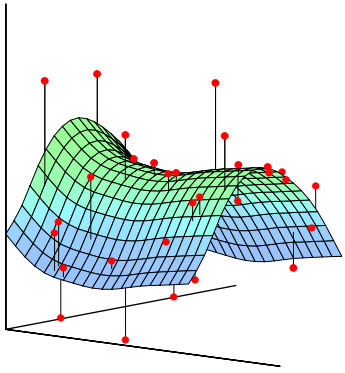
- In practice, we cannot directly calculate the risk, however, with the observed training data $\mathcal{D}_n$, we can calculate the empirical risk, which is simply replacing the expectation with the average over n training samples.

$$\mathrm{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} L\big(y_i, f(x_i)\big) = \frac{1}{n}\sum_{i=1}^{n} \big(y_i - f(x_i)\big)^2.$$

- We search for a function $\widehat{f}$ (in a certain space $\mathcal{F}$) to minimize the empirical risk on the training dataset

$$\begin{aligned}\widehat{f} &= \arg\min_{f\in\mathcal{F}} \mathrm{R}_n(f) \\ &= \arg\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \big(y_i - f(x_i)\big)^2.\end{aligned}$$

Minimizing the Empirical Risk



from ESL textbook

- A linear regression model describes the dependence between X and Y by

$$\begin{aligned} Y &= X^{\mathsf{T}}\boldsymbol{\beta} + \epsilon \\ &= \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon \end{aligned}$$

  where $\mathrm{E}(\epsilon) = 0$, $\mathrm{Var}(\epsilon) = \sigma^2$ and $\epsilon \perp X$.

- Given the training data $\mathcal{D}_n$, we express the regression model in the matrix form

$$\mathbf{y}_{n\times 1} = \mathbf{X}_{n\times p}\boldsymbol{\beta}_{p\times 1} + \mathbf{e}_{n\times 1}$$

  where $\mathbf{X}_{n\times p}$ is called the design matrix with each row representing one subject.

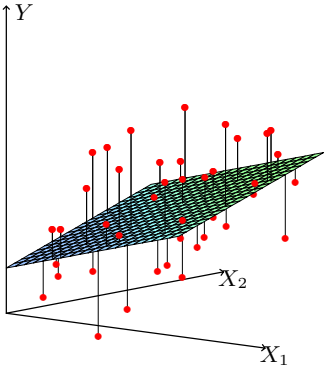
- Intercept can be included by setting the first column of $\mathbf{X}$ to be 1.

- Now, estimating f comes down to estimating $\boldsymbol{\beta}$.
- Based on our previous definition of the empirical risk, we solve for $\boldsymbol{\beta}$ that minimizes the residual sum of squares (RSS)

$$\begin{aligned}\text{RSS} &= \sum_{i=1}^{n} \left(y_i - x_{i1}\beta_1 - \cdots - x_{ip}\beta_p \right)^2 \\ &= \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \\ &= \left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)^{\mathsf{T}} \left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)\end{aligned}$$

- The ordinary least squares estimator (OLS) is

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)^{\mathsf{T}} \left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right)$$



from ESL textbook

Estimating β

- To estimate $\boldsymbol{\beta}$, we set the derivative equal to 0

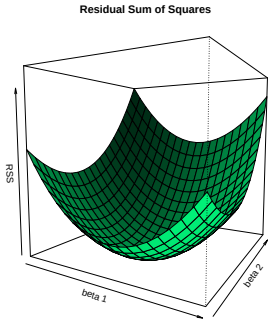
$$\frac{\partial \text{RSS}}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^{\mathsf{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0$$
$$\Longrightarrow \quad \mathbf{X}^{\mathsf{T}}\mathbf{y} = \mathbf{X}^{\mathsf{T}}\mathbf{X}\boldsymbol{\beta}$$

  which is commonly known as the normal equation.
- $\mathbf{X}$ full rank $\Longleftrightarrow$ $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ invertible
- We then have, if $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ is invertible,

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y}.$$

A Convex Problem

- There are many different ways to view a linear regression.
- One way is to view it as a convex optimization problem, which helps understand Lasso and Ridge.
- When $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ is invertible, the RSS is a strictly convex function of $\boldsymbol{\beta}$



Hat Matrix

- The fitted values (i.e., prediction at the n observed data points) are

$$\widehat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} \doteq \mathbf{H}_{n\times n}\mathbf{y}$$

- The "hat matrix"

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$$

  is a project matrix that projects onto the column space of $\mathbf{X}$.
  - symmetric: $\mathbf{H}^\mathsf{T} = \mathbf{H}$
  - idempotent: $\mathbf{HH} = \mathbf{H}$

Residuals

- The residual $\mathbf{r}$ is defined as

$$\begin{aligned}\widehat{\mathbf{e}} = \mathbf{r}_{n\times 1} &= \mathbf{y} - \widehat{\mathbf{y}} \\ &= (\mathbf{I} - \mathbf{H})\mathbf{y}\end{aligned}$$

- $\mathbf{r}$ can be used to estimate the error variance

$$\widehat{\sigma}^2 = \frac{1}{n-p}\sum_{i=1}^{n} r_i^2 = \frac{\text{RSS}}{n-p}$$

Vector Space Interpretation

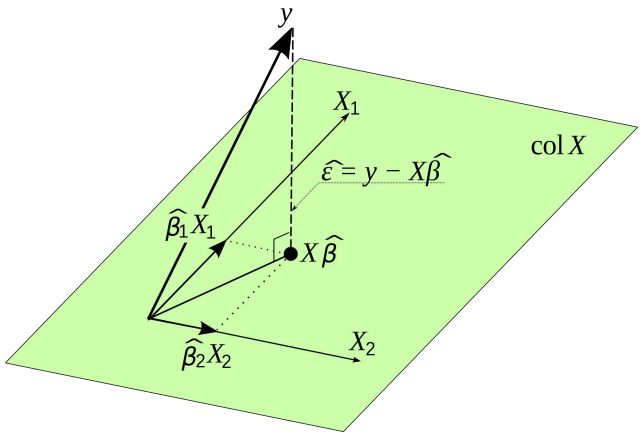


Figure from Wikipedia

- The essence of LS is to decompose the data vector $\mathbf{y}$ into two orthogonal vectors

$$\begin{aligned}\mathbf{y} &= \mathbf{H}\mathbf{y} + (\mathbf{I} - \mathbf{H})\mathbf{y} \\ &= \widehat{\mathbf{y}} + \mathbf{r}\end{aligned}$$

- Note that since $\mathbf{H}$ is a projection matrix, $\mathbf{r}$ is orthogonal to each column of $\mathbf{X}$, i.e.,

$$\mathbf{X}^{\mathsf{T}}\mathbf{r} = \mathbf{0}_{p\times 1}.$$

Properties of $\widehat{\beta}$

- If the samples are indeed generated from a linear model

$$Y = X^{\mathsf{T}}\boldsymbol{\beta} + \epsilon,$$

where the errors ϵ_i are i.i.d., independent of X, with $\mathrm{E}(\epsilon_i) = 0$ and $\mathrm{Var}(\epsilon_i) = \sigma^2$.

- Then $\widehat{\boldsymbol{\beta}}$ is unbiased: $\mathrm{E}(\widehat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$
- Variance-covariance

$$\begin{aligned}\mathrm{Var}(\widehat{\boldsymbol{\beta}}) &= \mathrm{Var}\big((\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y}\big)\\ &=?\end{aligned}$$

Properties of $\widehat{\beta}$

- By the Gauss-Markov Theorem, $\widehat{\beta}$ is the best linear unbiased estimator (BLUE)
- If the errors are generated from a Gaussian distribution, then $\widehat{\beta}$ is also the minimum variance unbiased estimator (MVUE)
- However, based on our understanding of the bias-variance trade-off, we could sacrifice the unbiasedness to trade for a large reduction in variance. Then the overall prediction error may perform better.

Dealing with large p

- In many applications nowadays, we have many explanatory variables, i.e., p is large or even $p \gg n$.
  - There are more than 20,000 human protein-coding genes
  - About 10 million single nucleotide polymorphisms (SNPs)
  - Number of subjects, n, is usually in hundreds or thousands
- In some applications, the key question is to identify a subset of X variables that are most relevant to Y
- Let's examine the training and testing errors from a linear model

Training and Testing Data

- Training data $\mathcal{D}_n = \{x_i, y_i\}_{i=1}^n$
- Suppose $\{x_i, y_i^*\}_{i=1}^n$ is an independent (imaginary) testing dataset collected at the same location x_i's (aka, in-sample prediction
- Assume that the data are generated from

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{e} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$$
$$\mathbf{y}^* = \boldsymbol{\mu} + \mathbf{e}^* = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}^*$$

  where both $\mathbf{y}$ and $\mathbf{y}^*$ are $n \times 1$ response vectors, $\mathbf{e}$ and $\mathbf{e}^*$ are i.i.d. error terms with mean 0 and variance σ^2.
- The true model is indeed linear!
- Goal: What is the best model that predicts $\mathbf{y}^*$?

Testing Error

$$
\begin{aligned}
\mathrm{E}[\text{Test Err}] &= \mathrm{E}\|\mathbf{y}^* - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2 \\
&= \mathrm{E}\|(\mathbf{y}^* - \mathbf{X}\boldsymbol{\beta}) + (\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\widehat{\boldsymbol{\beta}})\|^2 \\
&= \mathrm{E}\|\mathbf{e}^*\|^2 + \mathrm{E}\|\mathbf{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\|^2 \\
&= n\sigma^2 + \mathrm{E}\big[\mathrm{Trace}\big((\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})\big)\big] \\
&= n\sigma^2 + \mathrm{Trace}\big(\mathbf{X}^{\mathsf{T}}\mathbf{X}\mathrm{Cov}(\widehat{\boldsymbol{\beta}})\big) \\
&= n\sigma^2 + p\sigma^2
\end{aligned}
$$

- We used the properties:
  - $\mathrm{Trace}(ABC) = \mathrm{Trace}(CBA)$
  - $\mathrm{E}(\mathrm{Trace}(A)) = \mathrm{Trace}(\mathrm{E}(A))$

Training Error

$$
\begin{aligned}
\mathrm{E}[\text{Train Err}] &= \mathrm{E}\|\mathbf{y} - \widehat{\mathbf{y}}\|^2 = \mathrm{E}\|(\mathbf{I} - \mathbf{H})\mathbf{y}\|^2 \\
&= \mathrm{E}\|(\mathbf{I} - \mathbf{H})(\mathbf{X}\boldsymbol{\beta} + \mathbf{e})\|^2 \\
&= \mathrm{E}\|(\mathbf{I} - \mathbf{H})\mathbf{e}\|^2 \\
&= \mathrm{E}\big[\mathrm{Trace}\big(\mathbf{e}^{\mathsf{T}}(\mathbf{I} - \mathbf{H})^{\mathsf{T}}(\mathbf{I} - \mathbf{H})\mathbf{e}\big)\big] \\
&= \mathrm{Trace}\big((\mathbf{I} - \mathbf{H})^{\mathsf{T}}(\mathbf{I} - \mathbf{H})\mathrm{Cov}(\mathbf{e})\big) \\
&= (n - p)\sigma^2
\end{aligned}
$$

- We used the property:
  - $\mathbf{HX} = \mathbf{X}$

Training vs. Testing error

- Summary:
  - testing error: $n\sigma^2 + p\sigma^2$
  - training error: $(n-p)\sigma^2$
- The expected testing error increase with p and the expected training error decreases with p.
- When p gets large, this is a big trouble. Consider the case $p = n$, this is equivalent to 1NN.
- Can we just select a few number of variables to reduce p?
- What could be the consequences?

Variable Selection

- Variable/model selection may improve
  - Prediction accuracy
  - Interpretability
- However, this may also increase bias (we did not discuss them in the previous derivation) because we are taking the risk of removing some important variables.
- Overall, this is a difficult task.
  - No natural ordering of importance for the variables
  - The role of a variable needs be measured conditioning on others, high correlation causes trouble
  - It is essential to check all possible combinations, however, this may be computationally expansive

Motivation

- If we compare the two errors:
  - testing error: $n\sigma^2 + p\sigma^2$
  - training error: $(n-p)\sigma^2$

  we have:

$$\text{testing error} = \text{training error} + 2p\sigma^2$$

- Training error (RSS) is always computable, and we can estimate σ^2 using $\widehat{\sigma}^2$.
- Hence, how about searching for a model that minimizes

$$\text{RSS} + 2\widehat{\sigma}^2_{\text{full}} \cdot p$$

- $\widehat{\sigma}^2_{\text{full}}$ can be estimated using the full model, with all variables.
- The method is called Mallows' C_p (Mallows 1973)

Model Selection Criteria

- Model selection is usually done in the following way
  1. Give each fitted model a score (goodness-of-fit)
  2. Design an algorithm to find the model with the best score
- The score of a fitted model usually takes the the form

$$\text{goodness-of-fit} + \text{model-complexity}$$

- The first term will decrease as the model gets more complicated (recall 1NN, or linear model with $p = n$)
- The second term increases with the number of predictors used, which prefers "smaller" models

Model Selection Criteria

- Popular choices of scores:
  - Mallows' C_p (Mallows 1973): RSS $+ 2\widehat{\sigma}^2_{\text{full}} \cdot p$
  - AIC (Akaike 1970): -2 Log-likelihood $+ 2 \cdot p$
  - BIC (Schwarz, 1978): -2 Log-likelihood $+ \log n \cdot p$
- AIC is motivated from the Kullback–Leibler divergence; BIC is motivated from Bayesian posterior.
- C_p performs similarly to AIC.
- When n is large, adding one predictor costs a lot more in BIC than AIC (or C_p). So AIC tends to pick a larger model than BIC.

Bias-Variance Trade-Off

- Recall our previous analysis of the training and testing errors with $\mathbf{y}$ and $\mathbf{y}^*$, no bias term was involved.
- This is because we assume that the true model is linear, and we always include all the necessary variables.
- What will happen if linear model is wrong? or we eliminated some true variables?
- "All models are wrong, but some are useful."

George E. P. Box, (1919 - 2013)

- Now, lets assume that the model is not necessarily a linear model, i.e.,

$$\mathbf{y} = f(\mathbf{X}) + \mathbf{e} = \boldsymbol{\mu} + \mathbf{e}$$
$$\mathbf{y}^* = f(\mathbf{X}) + \mathbf{e} = \boldsymbol{\mu} + \mathbf{e}^*$$

- But we don't have $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$. However, we still perform a linear regression.
- Note that $\boldsymbol{\mu}$ is a vector of n elements, the best linear model is essentially projecting this mean vector onto the column space defined by $\mathbf{X}$. Hence, the best linear model to describe this $\mathbf{H}\boldsymbol{\mu}$ — projecting the mean vector onto the column space of $\mathbf{X}$.
- This will introduce bias as long as $\mathbf{H}\boldsymbol{\mu} \neq \boldsymbol{\mu}$.

Justification of Mallows' C_p

$$\begin{aligned}
\mathrm{E}[\text{Test Err}] &= \mathrm{E}\|\mathbf{y}^* - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2 = \|\mathbf{y}^* - \mathbf{H}\mathbf{y}\|^2 \\
&= \mathrm{E}\|(\mathbf{y}^* - \boldsymbol{\mu}) + (\boldsymbol{\mu} - \mathbf{H}\boldsymbol{\mu}) + (\mathbf{H}\boldsymbol{\mu} - \mathbf{H}\mathbf{y})\|^2 \\
&= \mathrm{E}\|\mathbf{y}^* - \boldsymbol{\mu}\|^2 + \mathrm{E}\|\boldsymbol{\mu} - \mathbf{H}\boldsymbol{\mu}\|^2 + \mathrm{E}\|\mathbf{H}\boldsymbol{\mu} - \mathbf{H}\mathbf{y}\|^2 \\
&= \mathrm{E}\|\mathbf{e}^*\|^2 + \mathrm{E}\|\boldsymbol{\mu} - \mathbf{H}\boldsymbol{\mu}\|^2 + \mathrm{E}\|\mathbf{H}\mathbf{e}\|^2 \\
&= n\sigma^2 + \mathrm{Bias}^2 + p\sigma^2
\end{aligned}$$

$$\begin{aligned}
\mathrm{E}[\text{Train Err}] &= \mathrm{E}\|\mathbf{y} - \widehat{\mathbf{y}}\|^2 = \mathrm{E}\|(\mathbf{I} - \mathbf{H})\boldsymbol{\mu} + (\mathbf{I} - \mathbf{H})\mathbf{e}\|^2 \\
&= \mathrm{E}\|(\mathbf{I} - \mathbf{H})\boldsymbol{\mu}\|^2 + \mathrm{E}\|(\mathbf{I} - \mathbf{H})\mathbf{e}\|^2 \\
&= \mathrm{Bias}^2 + (n - p)\sigma^2
\end{aligned}$$

Hence, we still have Test Err = Train Err + $2\sigma^2 p$.

Model Selection Algorithm

- Basic idea:
  - Pick a penalty for model complexity (Mallows' C_p, AIC or BIC)
  - Try models with different variables
  - For each model, calculate the sum of goodness-of-fit and the penalty for model complexity
  - Compare all candidates, and pick the best one
- Note: When comparing two models with the same number of variables, only the goodness-of-fit measure matters.
- Commonly used algorithms: best subset selection; backward/forward selection.

Best Subset Selection

- Best subset selection is a level-wise search algorithm, which returns the global optimal solution for a given model size.
- Only feasible for p not very large (< 50)
- Algorithm:
  1). For each $k = 1, \ldots, p$, check 2^k possible combinations, and find the model with smallest RSS
     - The penalty term is the same for models with the same size
  2). To choose the best k, use model selection criteria

Stepwise Regression

- Greedy algorithms: fast, but only return a local optimal solution (which might be good enough in practice).
  - Backward: start with the full model and sequentially delete predictors until the score does not improve.
  - Forward: start with the null model and sequentially add predictors until the score does not improve.
  - Stepwise: consider both deleting and adding one predictor at each stage.

Table of Contents

Motivation

- Best subset selection
  - Computationally expensive; not feasible when p is large
- Forward/backward selection
  - No guarantee to find the best global sub-model
  - The selection process is discrete ("add" or "drop"). The result highly depends on the inclusion/exclusion criterion.

Motivation

- The OLS estimator is a linear function of $\mathbf{y}$, and it is the BLUE.
- Recall that the prediction accuracy is

$$\text{Irreducible Error} + \text{Bias}^2 + \text{Variance}$$

- Generally, by regularizing (shrinking, penalizing) the estimator in some way, we can create a new estimator
  - The estimator is biased
  - The variance is reduced
  - Overall, we can have a better prediction accuracy

- ℓ_2 penalty: Ridge regression
- ℓ_1 penalty: Lasso

Overview

- Definition of the Ridge regression
- How to derive the solution through connections with PCA?
- Effect of shrinkage and the degrees of freedom
- Selecting the tuning parameter

Ridge Regression

Penalizing the square of the coefficients

$$\widehat{\boldsymbol{\beta}}^{\text{ridge}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2$$

- proposed by Hoerl and Kennard (1970); Tikhonov (1943)
- $\lambda \geq 0$ is a tuning parameter (penalty level) that controls the amount of shrinkage
- penalizing the ℓ_2 norm of $\boldsymbol{\beta}$, hence is called the ℓ_2 penalty
- the coefficients $\widehat{\boldsymbol{\beta}}^{\text{ridge}}$ are shrunken towards 0

Solution for Ridge Regression

- We can also write the Ridge regression in matrix form:

$$\text{minimize} \quad (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathsf{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{\beta}$$

- Similar to solving the linear regression, by taking the derivative of $\boldsymbol{\beta}$, we have the normal equation

$$\begin{aligned} \mathbf{0} &= -2\mathbf{X}^{\mathsf{T}}\mathbf{y} + 2\mathbf{X}^{\mathsf{T}}\mathbf{X}\boldsymbol{\beta} + 2\lambda\boldsymbol{\beta} \\ \Longrightarrow \mathbf{X}^{\mathsf{T}}\mathbf{y} &= \left(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda\mathbf{I}\right)\boldsymbol{\beta} \\ \Longrightarrow \quad \boldsymbol{\beta} &= \left(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y} \end{aligned}$$

- Why this helps fitting a linear model?

The Effect of Ridge Regression

- The Ridge regression is frequently used for addressing highly correlated variables
- When some variables are linearly correlated (e.g., $p > n$) $\mathbf{X}$ do not have full column rank
- This makes $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ singular, hence inverting this matrix becomes impossible
- However, $\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda\mathbf{I}$ is always full ranked

The Effect of Ridge Regression

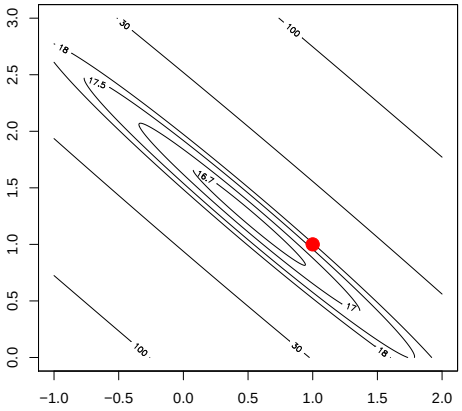
- Highly correlated variables makes the estimation unstable
- If $\mathbf{X}^T\mathbf{X}$ is close to singular,

$$\det(\mathbf{X}^T\mathbf{X}) \to 0 \quad \Rightarrow \quad \det((\mathbf{X}^T\mathbf{X})^{-1}) \to \infty$$

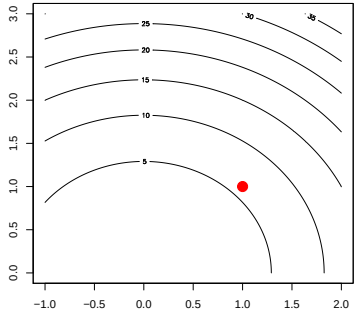
- Since $\text{Var}(\widehat{\boldsymbol{\beta}}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2$, the variance of $\widehat{\boldsymbol{\beta}}$ (or certain combinations of $\widehat{\boldsymbol{\beta}}$) is extremely large.
- Trade that variance with some bias?

Optimization Point-of-view

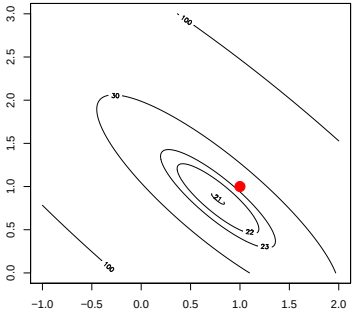
- The instability of having highly correlated variables can also be explained by the lack of convexity of the objective function
- The objective function of the OLS estimator is almost flat alone certain combinations of the $\boldsymbol{\beta}$ parameters
- The optimal solution is greatly affected by the random errors
- The Ridge penalty $\lambda \boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{\beta}$ forces some convexity



OLS loss function $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathsf{T}}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$



Ridge penalty: $\lambda\beta^{\mathsf{T}}\beta$



Ridge objective function

Understanding the Shrinkage

- Suppose we have an orthonormal design matrix ($\mathbf{X}^T\mathbf{X} = \mathbf{I}$), then $\widehat{\boldsymbol{\beta}}^{\text{ols}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{X}^T\mathbf{y}$ and

$$\begin{aligned}\widehat{\boldsymbol{\beta}}^{\text{ridge}} =& (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \\ =& (\mathbf{I} + \lambda\mathbf{I})^{-1}\widehat{\boldsymbol{\beta}}^{\text{ols}} \\ =& (1+\lambda)^{-1}\widehat{\boldsymbol{\beta}}^{\text{ols}},\end{aligned}$$

- This means that we just need to shrink each element of $\widehat{\boldsymbol{\beta}}^{\text{ols}}$ by a factor of $(1+\lambda)^{-1}$, i.e.,

$$\widehat{\boldsymbol{\beta}}_j^{\text{ridge}} = \frac{1}{1+\lambda}\widehat{\boldsymbol{\beta}}_j^{\text{ols}}, \text{ for all } j$$

Understanding the Shrinkage

- How about bias and variance under the orthonormal design
- $\text{Var}(\widehat{\boldsymbol{\beta}}_j^{\text{ridge}}) = \frac{1}{(1+\lambda)^2}\text{Var}(\widehat{\boldsymbol{\beta}}_j^{\text{ols}})$ (reduced from OLS!)
- $\text{Bias}(\widehat{\boldsymbol{\beta}}_j^{\text{ridge}}) = \frac{-\lambda}{1+\lambda}\boldsymbol{\beta}_j$ (biased!)
- There always exists a λ such that the prediction error of $\widehat{\boldsymbol{\beta}}^{\text{ridge}}$ is smaller than $\widehat{\boldsymbol{\beta}}^{\text{ols}}$

Understanding the Shrinkage

- When the columns of $\mathbf{X}$ are not orthogonal, we can utilize PCA
- The relationship between Ridge and PCA can be understood by (assuming $\mathbf{X}$ centered) decomposing the covariance matrix

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{D}^2\mathbf{V}^T$$

- This means $(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1} = \mathbf{V}(\mathbf{D}^2 + \lambda\mathbf{I})^{-1}\mathbf{V}^T$
- The Ridge fitted value $\widehat{\mathbf{y}}$ can be calculated as (since $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$)

$$\begin{aligned}\widehat{\mathbf{y}} = \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ridge}} &= \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{U}\mathbf{D}(\mathbf{D}^2 + \lambda\mathbf{I})^{-1}\mathbf{D}^T\mathbf{U}^T\mathbf{y} \\ &= \sum_{j=1}^{p}\mathbf{u}_j\left(\frac{d_j^2}{d_j^2 + \lambda}\mathbf{u}_j^T\mathbf{y}\right)\end{aligned}$$

Understanding the Shrinkage

- Hence, Ridge regression can be understood as
  (1) Perform principle component analysis of $\mathbf{X}$
  (2) Treat the principle components $\mathbf{u}_j$'s as new independent variables and project $\mathbf{y}$ onto the them: $\mathbf{u}_j^T\mathbf{y}$ for each j
  (3) Shrink the projections using the factor $d_j^2/(d_j^2+\lambda)$
- Directions with smaller eigenvalues d_j get more relative shrinkage.
- The ridge fitted value of $\widehat{\mathbf{y}}$ is the sum of p shrunk projections.

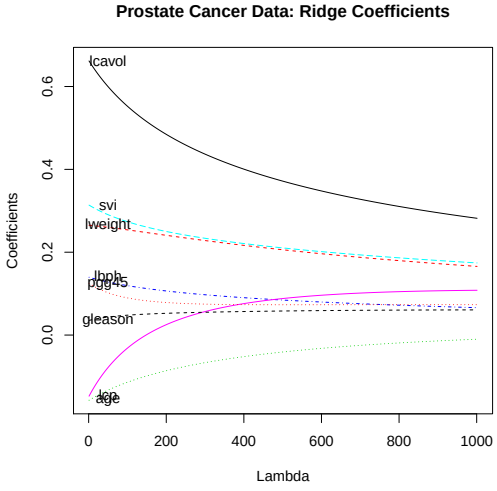
- The Ridge regression solution is not invariant with respect to the scale of the predictors!
- The scale of variables determines d_j's, hence affect the shrinkage.
- A standard procedure: perform centering and scaling on $\mathbf{X}$, perform centering on $\mathbf{y}$, and fit linear regression on the normalized data without intercept. The parameters on the original scale can be reversely solved.
- The intercept term is not penalized.
- Some packages (e.g. "glmnet" package, and lm.ridge function in MASS package) handles the centering and scaling automatically.

- We need to tune the penalty term λ in a Ridge regression
- Cross-validation is possible, however, we also have some easier approach because Ridge regression, similar to linear regression, has some nice properties.
- The procedure is called GCV (generalized cross-validation)

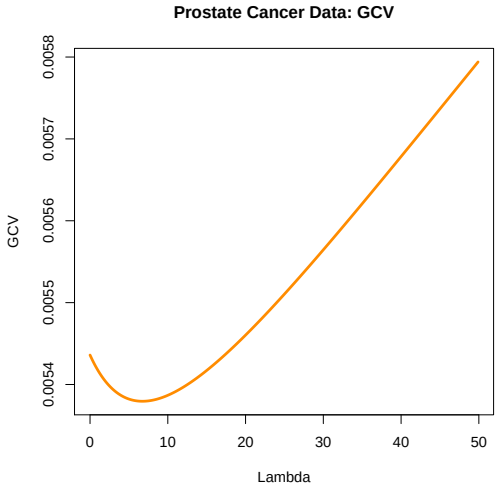
$$\text{GCV}(\lambda) = \frac{n^{-1}\|(\mathbf{I} - \mathbf{S}_\lambda)\mathbf{y}\|^2}{\left(n^{-1}\text{Trace}(\mathbf{I} - \mathbf{S}_\lambda)\right)^2}$$

- GCV is motivated from the leave-one-out cross-validation. This is implemented in lm.ridge .

Prostate Cancer Example



Prostate Cancer Example

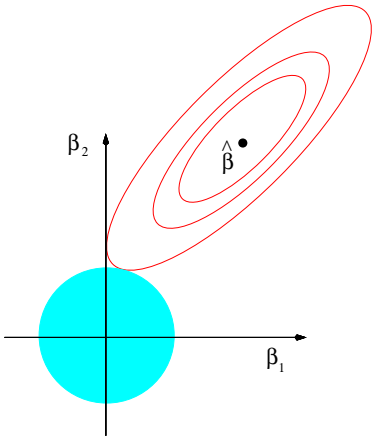


Alternative View

- An equivalent formulation is given by

$$\begin{aligned} \underset{\boldsymbol{\beta}}{\text{minimize}} \quad & \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \\ \text{subject to} \quad & \sum_{j=1}^{p} \beta_j^2 \leq s \end{aligned}$$

- There is a one-to-one correspondence between the parameters λ and s, but we can't find the explicit formula.



Ridge constrained solution

- Although $\widehat{\beta}^{\text{ridge}}$ is p-dimensional, it does not use the full potential of all p covariates due to the shrinkage.
- For example, if λ is very large, all the parameter estimates are 0. Then intuitively, the df should be close to 0. If λ is 0, then we reduce to the OLS with p df.
- The df of a Ridge regression is given by

$$\text{df}(\lambda) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}$$

which is always between 0 and p.

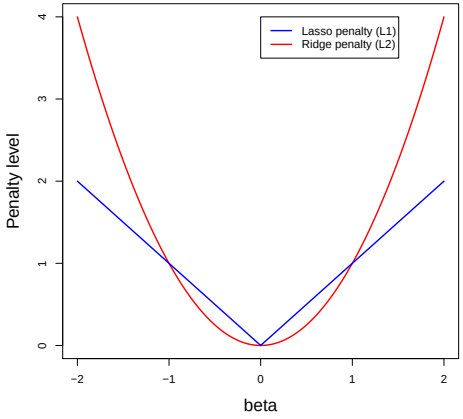
- The Ridge regression shrinks the coefficients towards 0, however, they are not exactly zero. Hence, we haven't achieve any "selection" of variables.
- Parsimony: we would like to select a small subset of predictions. Stepwise regression does not guarantee the global solution.
- Lasso provides a continuous process. We will discuss:
  - The formulation and convexity
  - The solution when $\mathbf{X}$ is orthogonal
  - Some examples

Lasso

Least absolute shrinkage and selection operator (Tibshirani 1996)

$$\widehat{\boldsymbol{\beta}}^{\text{lasso}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1$$

- Shrinkage of the ℓ_1 norm of the parameters
- Property: some will be exactly 0, hence achieves selection of parameters



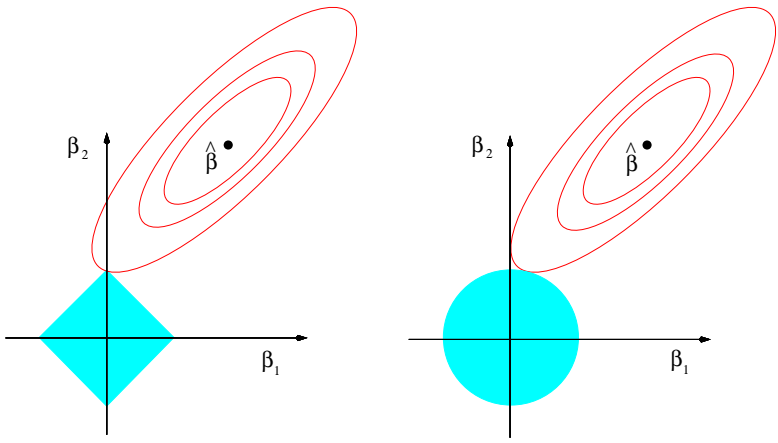
Lasso

Equivalent Formulation

- The Lasso optimization problem is equivalent to

$$\begin{aligned} \underset{\boldsymbol{\beta}}{\text{minimize}} \quad & \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \\ \text{subject to} \quad & \sum_{j=1}^{p} |\beta_j| \leq s \end{aligned}$$

- Each value of λ corresponds to an unique value of s.
- Compare Ridge and Lasso?



Comparing Lasso and Ridge solutions

Lasso Under Orthogonal Design

- Again, it will be helpful to view Lasso assuming orthogonal design, i.e., $\mathbf{X}^T\mathbf{X} = \mathbf{I}_{p\times p}$.
- We first analyze the loss part:

$$\begin{aligned}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 =& \|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} + \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}\|^2 \\ =& \|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}}\|^2 + \|\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}\|^2\end{aligned}$$

- The cross-product term is

$$2(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}})^T(\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}) = 2\mathbf{r}^T(\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}) = 0,$$

since the second term is in the column space of $\mathbf{X}$, while $\mathbf{r}$ is orthogonal to that space.

Lasso Under Orthogonal Design

- Our Lasso problem can be rewritten as

$$\begin{aligned}\widehat{\boldsymbol{\beta}}^{\text{lasso}} &= \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1 \\ &= \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}}\|^2 + \|\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1\end{aligned}$$

- Since $\|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}}\|^2$ is not a function of $\boldsymbol{\beta}$, this problem is reduced to

$$\widehat{\boldsymbol{\beta}}^{\text{lasso}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1$$

Lasso Under Orthogonal Design

- Then, since $\mathbf{X}^{\mathsf{T}}\mathbf{X} = \mathbf{I}_{p\times p}$, we have

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}^{\text{lasso}} &= \arg\min_{\boldsymbol{\beta}} \ \|\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1 \\
&= \arg\min_{\boldsymbol{\beta}} \ (\widehat{\boldsymbol{\beta}}^{\text{ols}} - \boldsymbol{\beta})^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{X}(\widehat{\boldsymbol{\beta}}^{\text{ols}} - \boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1 \\
&= \arg\min_{\boldsymbol{\beta}} \ (\widehat{\boldsymbol{\beta}}^{\text{ols}} - \boldsymbol{\beta})^{\mathsf{T}}(\widehat{\boldsymbol{\beta}}^{\text{ols}} - \boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1 \\
&= \arg\min_{\boldsymbol{\beta}} \ \sum_{j=1}^{p}(\widehat{\beta}_j^{\text{ols}} - \beta_j)^2 + \lambda|\beta_j|.
\end{aligned}$$

- Note that each β_j is involved in a separate term, we can solve the lasso estimators individually from the OLS estimators.

Lasso Under Orthogonal Design

- Each of the β_j's is essentially solving for an optimization problem

$$\arg\min_{\beta} (\beta - a)^2 + \lambda|\beta|, \quad \lambda > 0$$

- The solution is simply

$$\begin{aligned}\widehat{\beta}_j^{\text{lasso}} &= \begin{cases} \widehat{\beta}_j^{\text{ols}} - \lambda/2 & \text{if} \quad \widehat{\beta}_j^{\text{ols}} > \lambda/2 \\ 0 & \text{if} \quad |\widehat{\beta}_j^{\text{ols}}| \le \lambda/2 \\ \widehat{\beta}_j^{\text{ols}} + \lambda/2 & \text{if} \quad \widehat{\beta}_j^{\text{ols}} < -\lambda/2 \end{cases} \\ &= \text{sign}\left(\widehat{\beta}_j^{\text{ols}}\right)\left(|\widehat{\beta}_j^{\text{ols}}| - \lambda/2\right)_+ \end{aligned}$$

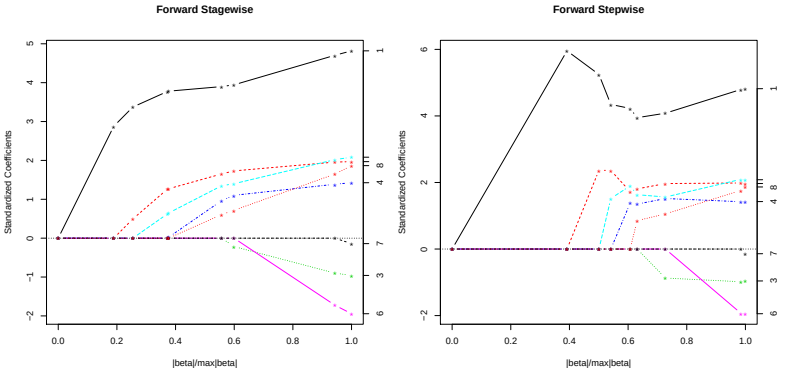
- A large λ will shrink some of the coefficients to exactly zero, which achieves "variable selection".

Computation of Lasso Solution

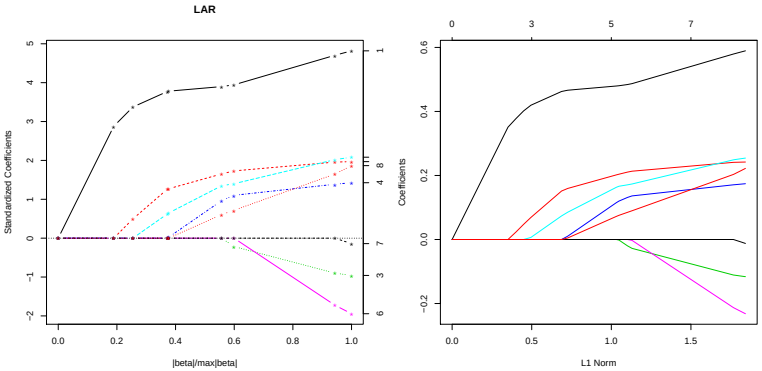
- When the covariates are not orthogonal, we will not be able to write down the explicit solution
- The Lasso problem is convex, although it may not be strictly convex in β when p is large
- The solution is a global minimum, but may not be unique

Computation of Lasso Solution

- There are algorithms that will produce equivalent solutions, although their computational costs are not the same
- Stage-wise regression (what is this?) Read ESL 3.3.3.
- Least angle regression (Efron et al. 2004) Read ESL 3.4.4.
- Coordinate descent (Friedman et al 2010): The most popular and fastest implementation, glmnet package
  - Also provides the solution path for an entire sequence of λ values
  - Start with the largest λ, use the previous estimation of $\boldsymbol{\beta}$ as a warm start for the solution of smaller λ



Comparing stagewise regression with stepwise regression



Comparing least angle regression with coordinate descent

ℓ_q Penalties

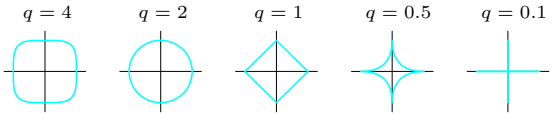


FIGURE 3.12. *Contours of constant value of* $\sum_j |\beta_j|^q$ *for given values of* q.

- Ridge is ℓ_2 penalty
- Lasso is ℓ_1 penalty
- Best subset is ℓ_0 penalty
- Elastic-net is a combination of Lasso and Ridge:

$$\lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_2^2$$

Table of Contents

Bayesian Linear regression models

- Consider the following liner regression model:

$$y|x, \beta, \sigma^2 \sim N(x\beta, \sigma^2 I_n)$$

- y is a column vector of n outcome observations, x is an $n \times (p+1)$ matrix of predictors with its first column being all 1's.
- β is a column vector with $p+1$ elements $(\beta_0, \beta_1, ..., \beta_p)$ where β_0 is the intercept and β_j is the effect of the j^{th} predictor x_j on y.
- In Bayesian analysis, a common prior for parameters are

$$\sigma^2 \sim \text{Inv-}\chi^2(\nu_0, \sigma_0^2)$$
$$\beta|\mu_0, \Lambda_0 \sim N_{p+1}(\mu_0, \Lambda_0)$$

where $\mu_0 = (\mu_{00}, \mu_{01}, ..., \mu_{0p})$ typically set to zero (unless we believe otherwise), and $\Lambda_0 = \text{diag}(\tau_0^2, \tau_1^2, ..., \tau_p^2)$ should be sufficiently broad.

Posterior distributions of β

- The posterior distributions of β has the following closed form:

$$\begin{aligned} \beta|x,y,\sigma^2 &\sim N(\mu_n,\Lambda_n) \\ \mu_n &= (x_*'\Sigma_*^{-1}x_*)^{-1}x_*'\Sigma_*^{-1}y_* \\ \Lambda_n &= (x_*'\Sigma_*^{-1}x_*)^{-1} \\ x_* &= \begin{pmatrix} x \\ I_{p+1} \end{pmatrix} \qquad y_* = \begin{pmatrix} y \\ \mu_0 \end{pmatrix} \qquad \Sigma_* = \begin{pmatrix} \sigma^2 I_n & 0 \\ 0 & \Lambda_0 \end{pmatrix} \end{aligned}$$

- Looking at it this way, the prior plays the role of extra data with $x_{\beta=I_{p+1}}$, $y_\beta = \mu_0$ and the covariance Λ_0.
- That's why Bayesian models do not break down when $p > n$.

Connection to ridge regression!

- Let's take a closer look at the maximum a posterior (MAP)

$$\begin{aligned} \mu_n &= (x_*'\Sigma_*^{-1}x_*)^{-1}x_*'\Sigma_*^{-1}y_* \\ &= (\sigma^{-2}x'x + \Lambda_0^{-1})^{-1}(\sigma^{-2}x'y + \Lambda_0^{-1}\mu_0) \end{aligned}$$

- Let $\mu_0 = 0$, $\sigma^2\Lambda_0^{-1} = \lambda I$, then we have

$$\mu_n = \hat{\beta}^{\text{ridge}} = (x'x + \lambda I)^{-1}x'y$$

- This is exactly the solution to ridge regression!
- Indeed, if we write down the negative logarithm of posterior density of β we have

$$\begin{aligned} -\log P(\beta|x, y, \sigma^2) &= -\log P(y|x, \beta, \sigma^2) - \log P(\beta) \\ &= \frac{1}{2}\sigma^{-2}\|y - x\beta\|_2^2 + \frac{1}{2}\beta'\Lambda_0^{-1}\beta \\ &= \frac{1}{2}\sigma^{-2}(\|y - x\beta\|_2^2 + \lambda\|\beta\|_2^2) \end{aligned}$$

Posterior distributions of σ^2

- Now, we want to obtain the posterior distribution of σ^2
- Given β, again we have a simple normal model with observations y_i with known mean $(x\beta)$, unknown variance σ^2, and conditionally conjugate prior Inv-$\chi^2(\nu_0, \sigma_0^2)$.
- As we saw before, the posterior distribution of $\sigma^2|x, y, \beta$ is also scaled Inv-χ^2

$$\begin{aligned} \sigma^2|x, y, \beta &\sim \text{Inv-}\chi^2(\nu_0 + n, \frac{\nu_0\sigma_0^2 + n\nu}{\nu_0 + n}) \\ \nu &= \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i\beta)^2 \end{aligned}$$

Improper priors

- If we do not have an informative priors, we can instead use the following prior:

$$p(\beta, \sigma^2|x) \propto \sigma^{-2}$$

- For β this is equivalent (in limit) to taking all $\tau_j^2 \to \infty$.
- The posterior distribution therefore becomes

$$\begin{aligned} \beta|y, \sigma^2 &\sim N(\hat{\beta}, V_\beta \sigma^2) \\ \hat{\beta} &= (x'x)^{-1}x'y, \qquad V_\beta = (x'x)^{-1} \end{aligned}$$

- $\hat{\beta}$ is exactly the OLS solution!
- The posterior distribution of σ^2 also has a closed form

$$\begin{aligned} \sigma^2|x, y, \hat{\beta} &\sim \text{Inv-}\chi^2(n-p-1, s^2) \\ s^2 &= \frac{1}{n-p-1}\sum_{i=1}^{n}(y_i - x_i\hat{\beta})^2 \end{aligned}$$

Example: Children's test score

- Consider the children's test score example discussed by Gelman and Hill (2007).
- In this example, we are interested in the effect of mother's education (mhsg) and her IQ (miq) on the cognitive test score of 3 to 4 year old children.
- For our Bayesian model, we use the following broad priors

$$\sigma^2 \sim \text{Inv-}\chi^2(1, 0.5)$$
$$\beta \sim N_{p+1}(0, 100^2 I)$$

- We used the Gibbs sampler to obtain 10000 samples and discarded the first 1000.

Example: Children's test score

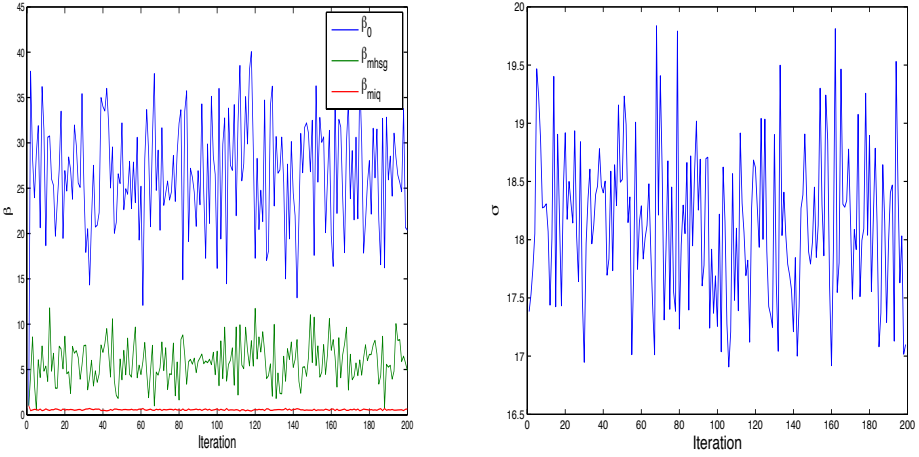


Figure: The trace plots of posterior samples for β's (left) and σ (right).

Example: Children's test score

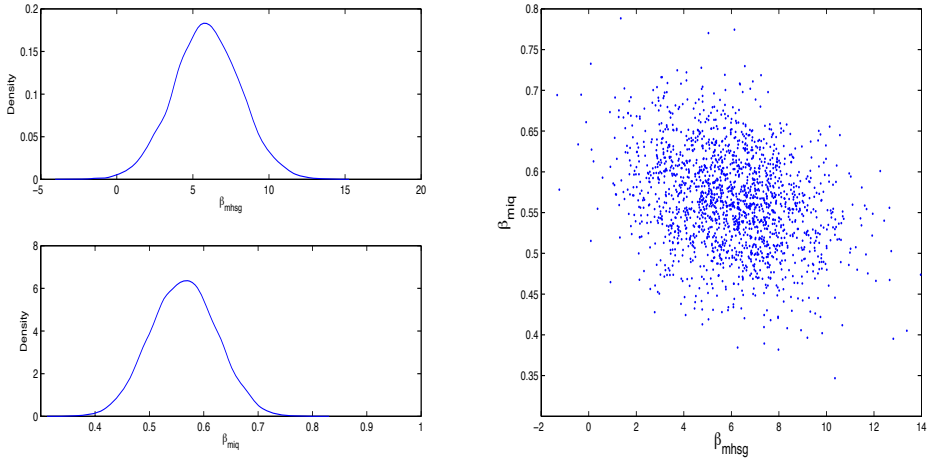


Figure: Marginal posterior distributions for β's (left) and the scatter plot of posterior samples for β_{mhsg} and β_{miq} (right).

Example: Children's test score

Table: The posterior estimates and 95% intervals for the regression parameters in the children's test score example.

Parameter	Posterior expectation	95% Probability Interval
β_0	25.7939	[14.4, 37.2]
β_{mhsg}	5.9278	[1.6, 10.3]
β_{miq}	0.5633	[0.4, 0.7]
σ	18.2	[16.9, 19.4]

- How about Lasso?

$$\underset{\beta}{\text{argmin}} \ \|y - x\beta\|_2^2 + \lambda\|\beta\|_1$$

- Can we come up the similar Bayesian version as ridge regression?
- That is, can we have some prior for β, such that the ℓ_1-penalization corresponds to the log-prior?

$$\begin{aligned} -\log P(\beta|x, y, \sigma^2) &= -\log P(y|x, \beta, \sigma^2) - \log P(\beta) \\ &\propto \|y - x\beta\|_2^2 + \lambda\|\beta\|_1 \end{aligned}$$

- Actually, this is called *Laplace* distribution $P(\beta) \propto \exp(-\lambda\|\beta\|_1)$.

Bayesian Lasso

- More generally, we use the following (conditional) Laplace prior

$$P(\beta|\sigma^2) = \prod_{j=0}^{p} \frac{\lambda}{2\sqrt{\sigma^2}} e^{-\lambda|\beta_j|/\sqrt{\sigma^2}}$$

  where σ^2 can be given some non-informative prior $1/\sigma^2$.
- This distribution has the following representation as a scale mixture of normals with an exponential mixing density

$$\frac{a}{2} e^{-a|z|} = \int_0^\infty \frac{1}{\sqrt{2\pi s}} e^{-z^2/(2s)} \frac{a^2}{2} e^{-a^2 s/2} ds, \quad a > 0$$

- Denote $\Lambda_0 = \mathrm{diag}(\tau_0^2, \tau_1^2, ..., \tau_p^2)$. Then we use the following priors

$$\begin{aligned} \beta|\sigma^2, \Lambda_0 &\sim N_{p+1}(0, \sigma^2\Lambda_0) \\ \tau_j &\overset{iid}{\sim} \mathrm{Exp}(\lambda^2/2), \quad j = 0, 1, \cdots, p \end{aligned}$$

- Then we can have conditional conjugacy and the full conditional posteriors are

$$\begin{aligned}
\beta|x,y,\sigma^2,\Lambda_0^2 &\sim N(\mu_n,\Lambda_n)\\
\mu_n &= (x'x+\Lambda_0^{-1})^{-1}x'y\\
\Lambda_n &= \sigma^2(x'x+\Lambda_0^{-1})^{-1}\\
1/\tau_j^2 &\overset{iid}{\sim} \mathrm{IG}(\mu',\lambda'), \quad j=0,\cdots,p\\
\mu' = \sqrt{\frac{\lambda^2\sigma^2}{\beta_j^2}} &, \quad \lambda'=\lambda^2
\end{aligned}$$

where the inverse-Gaussian distribution $\mathrm{IG}(\mu',\lambda')$ has the following density

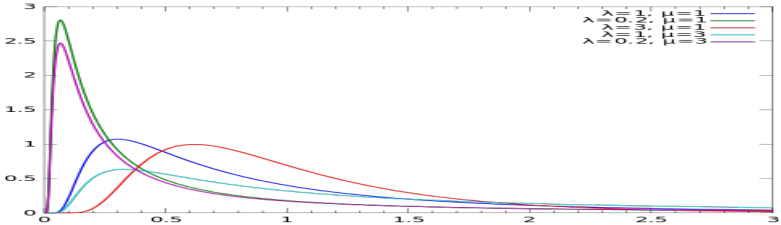
$$f(x) = \sqrt{\frac{\lambda'}{2\pi}}x^{-3/2}\exp\left\{-\frac{\lambda'(x-\mu')^2}{2(\mu')^2x}\right\}, \quad x>0$$

Connection to OLS and Ridge?

- Let's take a closer look at the conditional MAP

$$\begin{aligned} \mu_n &= (x'x + \Lambda_0^{-1})^{-1}x'y \\ 1/\tau_j^2 &\overset{iid}{\sim} \mathrm{IG}(\mu', \lambda'), \quad j = 0, \cdots, p \end{aligned}$$

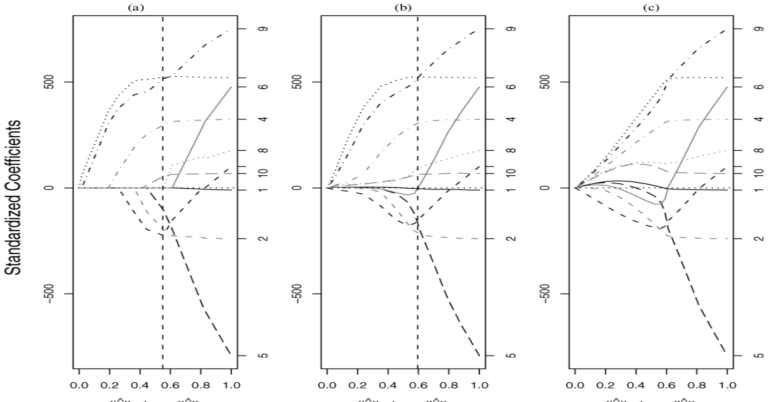
$$\mu' = \sqrt{\frac{\lambda^2\sigma^2}{\beta_j^2}} \quad , \quad \lambda' = \lambda^2$$



- Let $\lambda \to 0$, $1/\tau_j \to 0$, then $\mu_n \to (x'x)^{-1}x'y$, which is OLS!
- Let $\lambda \to \infty$, $1/\tau_j \sim \lambda$, then $\mu_n \sim (x'x + \lambda I)^{-1}x'y$ which behaves similarly as ridge solution!

Diabetes Data

- Now we consider for the diabetes data of Efron et. al (2004) which has $n = 442$ and $p = 10$.
- We compare Bayesian Lasso with Frequentist Lasso and ridge regression for the entire solution path for λ.



Bayesian bridge regression

- We can consider the following more general 'bridge' regression

$$\underset{\beta}{\operatorname{argmin}} \|y - x\beta\|_2^2 + \lambda\|\beta\|_q^q, \qquad \|\beta\|_q := \left(\sum_{j=0}^{p} |\beta_j|^q\right)^{1/q}$$

- One can consider the following (conditional) prior

$$P(\beta|\sigma^2) \propto \prod_{j=0}^{p} e^{-\lambda(|\beta_j|/\sqrt{\sigma^2})^q}$$

- And construct a similar mixture representation which is much more involved.
- Read *The Bayesian Lasso* (2008) by Park and Casella.