Ridge Regression

Before diving into these methods, taking a look at the **Norms** will help understanding and intuition since they are derived from.

Norms

When thinking of geometric vectors intuitively the direction and length of the vector are first that comes to mind, Simply **Norm** is a function that assigns each vector x it's **length** $\|x\|$ or **magnitude**

- $\|\lambda x\| = |\lambda| \|x\|$
- $\|x + y\| \leq \|x\| + \|y\|$
- $\|x\| \geq 0$ and if $\|x\| = 0 \iff x = 0$

The L_p Norm

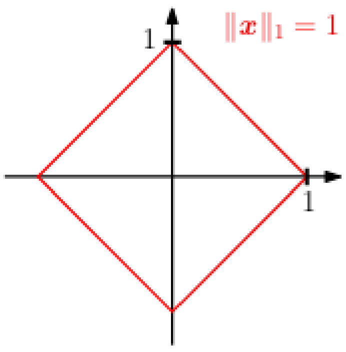
Also written as $\|x\|_p$, is defined as:

$$\|x\|_p = \sqrt[p]{\sum_{i=1}^{n} |x_i|^p}$$

with : $p > 0$ and x_i the **components** of x

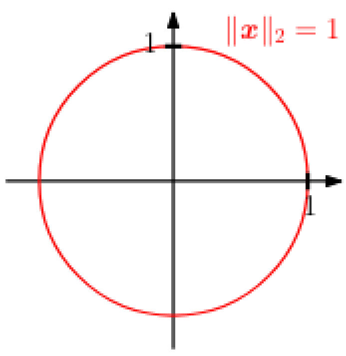
The L_1 Norm (Manhattan Norm)

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$



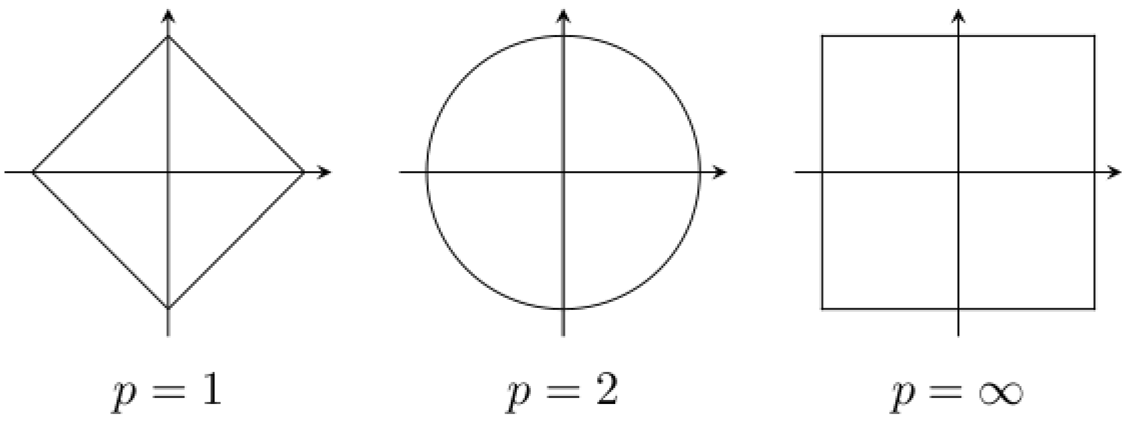
The L_2 Norm (Euclidean Norm)

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{x^T x}$$



The L_∞ Norm

$$\|x\|_\infty = \max_i(|x_i|)$$



- Which results in a **square**

Ridge Regression

The Ridge Regression originally proposed to deal with the **Multicollinearity** in the predictors, The Ordinary Least Squares results in a **Best Linear Unbiased Estimators** β, Since highly correlated variables may cause the model to become **unstable** (abnormal high variances in $\hat{\beta}$) and accompanied by large values of the **estimates**.

The **Ridge** Solution suggest that we introduce **bias** into the coefficients estimates which lowers the **variance** introduced by the **collinearity** following the Bias-Variance Trade-Off

There is many cases where the number of **predictors** p exceed the number of observations or samples n, the **Design matrix** X is called high-dimensional which using Multiple Linear Regression yields no unique solutions, Since the number of Unknown p is larger than the number of equations n, and often high-dimensional data can lead to **Multicollinearity**

Why Ridge Regression is Used ?

- High Multicollinearity
- High Dimensionality
- Prediction Accuracy

Ridge Regression Estimator

It's was proven in Ordinary Least Squares that's the estimated value of β is given by :

$$\hat{\beta} = (X^\intercal X)^{-1} X^\intercal Y$$

- This estimator is only defined if the **Gram Matrix** is invertible
- When the **Design matrix** is high dimensional it's impossible to yield unique solutions
- When the Predictors of the **Design matrix** are highly correlated results in **unstable large estimates**
- Often overfits the data and picks noise

There is two ways to solve this invertibility problem :

- Moore-Penrose inverse : It's provides an **Unbiased** best linear estimator but suffers from overfitting and poor prediction capabilities since it yield a sensitive model (Higher variance)
- Ridge Regression estimator : It's **Biased** and shrunken toward zero with low variance

The Ridge Regression Estimator simply replace $X^\intercal X$:

$$X^\intercal X + \lambda I_{pp}$$

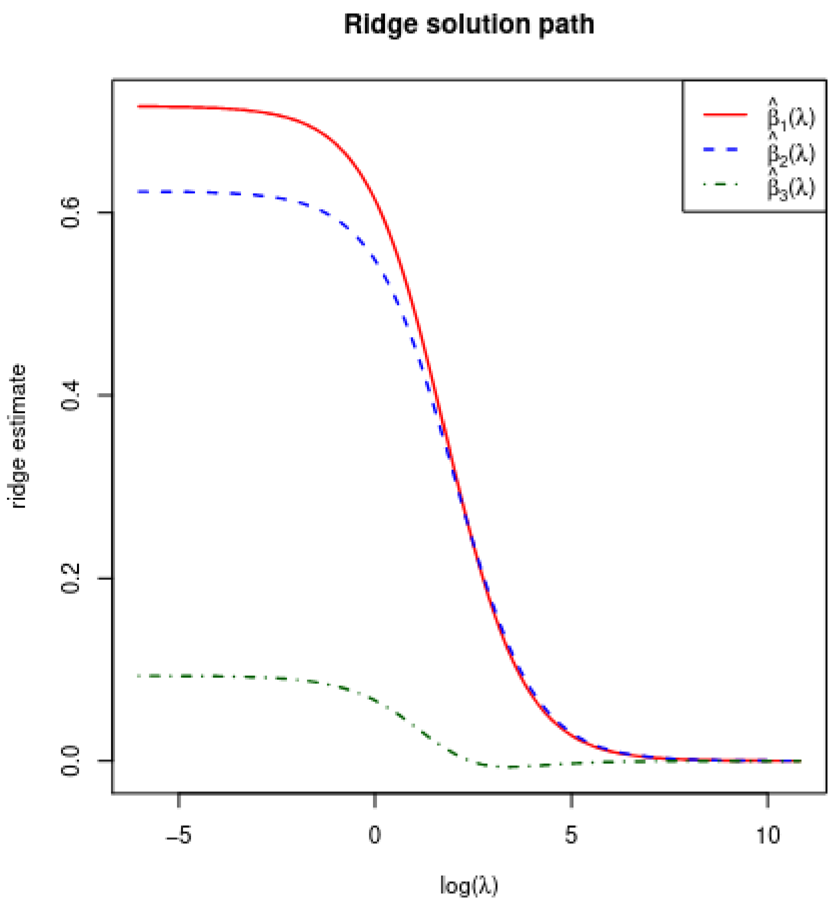
With :

- $\lambda \in [0, \infty)$ considered as a tuning parameter or **penalty parameter**, which solves the singularity by adding a positive matrix λI_{pp}

Results in the ridge regression estimator (coefficient estimate) **closed-form** :

$$\hat{\beta}(\lambda) = (X^\intercal X + \lambda I_{pp})^{-1} X^\intercal Y$$

Each value of the tuning parameter results in a different ridge regression estimator and the set of these estimates are called **Solution Path** or **Regularization Path**



Expectation

It's was mentioned that the ridge regression introduce **bias** to the estimators for better **variance**, Shown by calculating the expected value of $\hat{\beta}(\lambda)$:

$$\mathbb{E}[\hat{\beta}(\lambda)] = \mathbb{E}[(X^\intercal X + \lambda I_{pp})^{-1} X^\intercal Y] = (X^\intercal X + \lambda I_{pp})^{-1} X^\intercal \mathbb{E}[Y]$$

$$\mathbb{E}[\hat{\beta}(\lambda)] = (X^\intercal X + \lambda I_{pp})^{-1} X^\intercal X \beta$$

To show the **Bias** term let's substitute $(X^\intercal X) = (X^\intercal X + \lambda I) - \lambda I$

$$\mathbb{E}[\hat{\beta}(\lambda)] = (X^\intercal X + \lambda I_{pp})^{-1} [(X^\intercal X + \lambda I_{pp}) - \lambda I_{pp}] \beta$$

$$\mathbb{E}[\hat{\beta}(\lambda)] = \beta - (X^\intercal X + \lambda I_{pp})^{-1} \lambda I_{pp} \beta$$

$$\mathrm{Bias}(\hat{\beta}(\lambda)) = \mathbb{E}[\hat{\beta}(\lambda)] - \beta = -\lambda (X^\intercal X + \lambda I_{pp}) \beta$$

Relation Between the OLS and Ridge Estimators

In low-dimensionality the ridge regression estimator is related to it's maximum likelihood(OLS) solution :

$$W_\lambda = (X^\intercal X + \lambda I_{pp})^{-1} X^\intercal X$$

$$W_\lambda \hat{\beta} = \hat{\beta}(\lambda)$$

- In high-dimension there is no such linear relation between the **ridge** and the **OLS**

Variance

The variance of the Ridge regression estimator is obtained :

$$\mathrm{Var}(\hat{\beta}(\lambda)) = \mathrm{Var}(W_\lambda \hat{\beta}) = W_\lambda \mathrm{Var}(\hat{\beta}) W_\lambda^\intercal$$

With $\mathrm{Var}(\hat{\beta}) = \sigma^2 (X^\intercal X)^{-1}$

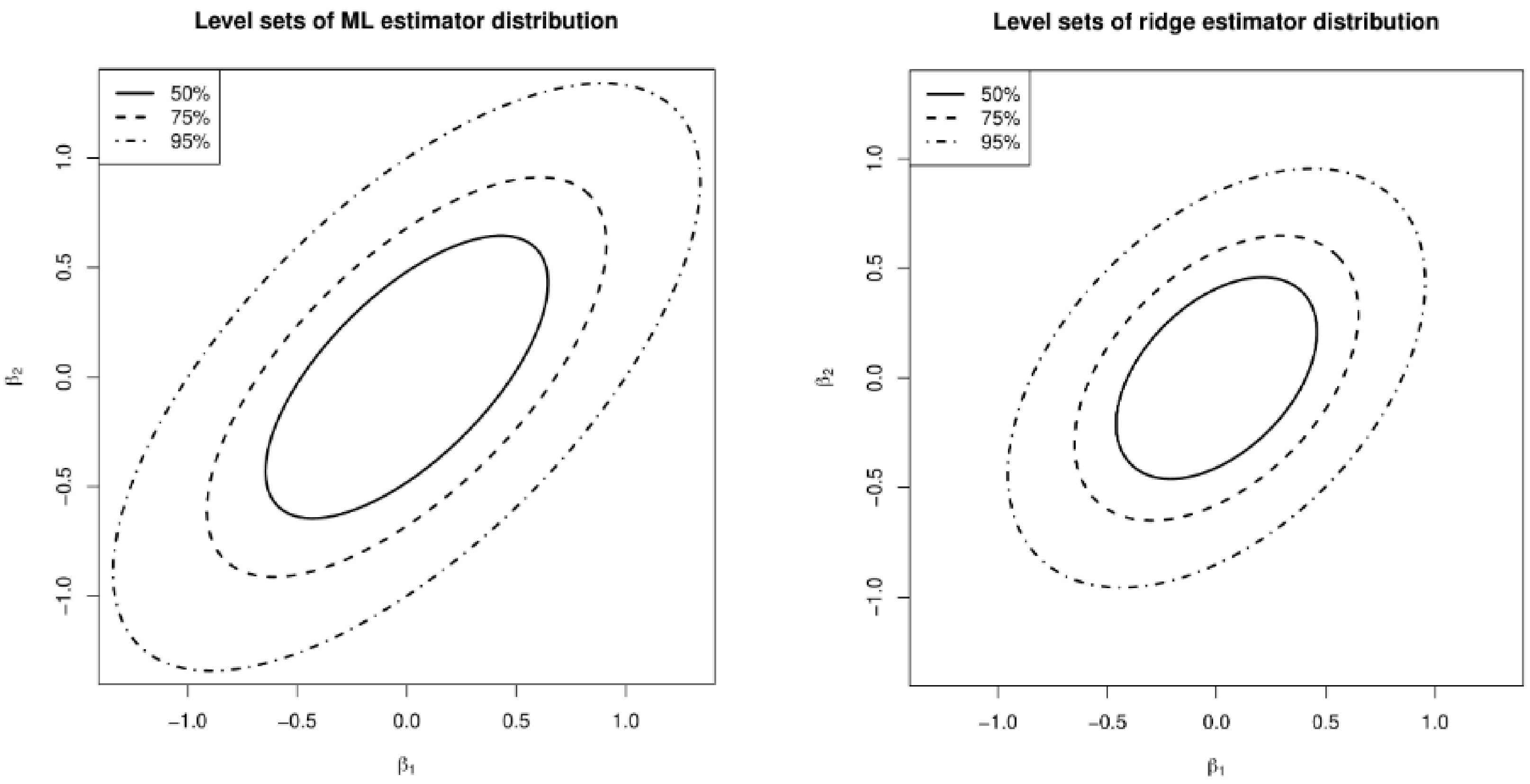
$$\mathrm{Var}(\hat{\beta}(\lambda)) = \sigma^2 W_\lambda (X^\intercal X)^{-1} W_\lambda^\intercal$$

- As the $\lambda \to \infty$ the $\mathrm{Var}(\hat{\beta}(\lambda)) = 0_{pp}$

It should be clear that the $\mathrm{Var}(\hat{\beta}) \geq \mathrm{Var}(\hat{\beta}(\lambda))$, Shown by :

$$\mathrm{Var}(\hat{\beta}) - \mathrm{Var}(\hat{\beta}(\lambda)) = \sigma^2 (X^\intercal X + \lambda I_{pp})^{-1} [2\lambda I_{pp} + \lambda^2 (X^\intercal X)^{-1} (X^\intercal X + \lambda I_{pp})^{-1}]^\intercal$$

- the difference is non-negative which aligns with the decrease of variance the **ridge** estimator have



- The **ridge estimator distribution** have less variance compared to the maximum likelihood estimator **OLS**

Mean Squared Error

For a way to choice the suitable λ that's can outperform the **OLS** MSE, The ridge MSE is given by:

$$\mathrm{MSE}(\hat{\beta}(\lambda)) = \sigma^2 \mathrm{tr}[W_\lambda (X^\intercal X)^{-1} W_\lambda^\intercal] + \beta^\intercal (W_\lambda - I_{pp})^\intercal (W_\lambda - I_{pp})\beta$$

And $\mathrm{MSE}(\hat{\beta}(\lambda)) < \mathrm{MSE}(\hat{\beta})$.

Ridge Loss Function

the ridge regression estimator minimized the **Ridge Loss Function** which is defined as :

$$\mathcal{L}_{ridge}(\beta; \lambda) = \|Y - X\beta\|_2^2 + \lambda\|\beta\|_2^2$$

- This is the traditional Residual Sum of Squares augmented with a penalty
- $\lambda\|\beta\|_2^2$ is the **Ridge penalty** or **Ridge Regularization Term**
- λ is the **Penalty Parameter**

The β that minimizes the $\mathcal{L}_{ridge}(\beta; \lambda)$ balances out the **sum of squares** and the **penalty**, the role of the **penalty** is to shrink the coefficients towards zero.

By solving Ridge Loss function for β, we arrive at the close solution :

$$\frac{\partial}{\partial \beta}\mathcal{L}_{ridge}(\beta; \lambda) = -2X^\intercal (Y - X\beta) + 2\lambda I_{pp}\beta = -2X^\intercal Y + 2(X^\intercal X + \lambda I)\beta$$

$$(X^\intercal X + \lambda I)\beta = X^\intercal Y$$

$$\hat{\beta}_{ridge} = (X^\intercal X + \lambda I)^{-1} X^\intercal Y$$

Motivation Behind the Ridge Penalty $\lambda\|\beta\|_2^2$

The First motivation as mentioned earlier was to deal with the Multicollinearity and High- dimensionality

The Ridge Penalty or the Ridge Regularization Term was added as a **Stabilizer** which penalize the addition of large coefficients which **OLS** tend to do in Highly correlated predictors, the ridge regression shrinks the coefficients toward zero making them much more stable

Another important motivation is reducing **high variance** that's the **OLS** yield, Which the Ridge regression reduce by introducing **Bias** on the ridge estimator as proven above by the **Bias Term** being :

$$\mathbb{E}[\hat{\beta}(\lambda)] - \beta = \mathrm{Bias}[\hat{\beta}(\lambda)] = -\lambda(X^\intercal X + \lambda I)^{-1}\beta$$

Penalty Parameter Selection λ

Cross-Validation

Detailed in Cross-Validation the procedure to select the best λ for the predictive ridge regression model :

1. Split the data into **Training and Test** sets
2. Define a Grid or range of values for λ the penalty parameter
3. Perform the Cross-validation loop for each λ value
    1. Fit the model using the **training set**
    2. Validate using the **test set**
    3. calculate the MSE
    4. Calculate the average performance across all the **K-Folds**
4. identify the **Penalty Parameter** λ that results in the lowest Test error

Generalized Cross-Validation