Regularized Linear Regression

- Shortcomings of Ordinary Linear Regression
- Ridge Regression
- Lasso Regression
- When to use each!

Why Regularized Linear Regression?

$$\sum_{i=1}^{N}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2$$

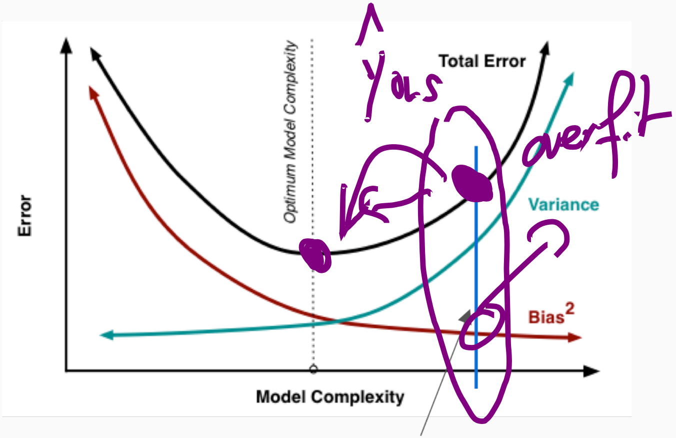
Bias $= E[f - \hat{f}]$

Unbiased $E[f - \hat{f}] = 0 = E[f] - E[\hat{f}]$

$E[\hat{f}] = f$

linear regression is an unbiased estimator

$\Rightarrow$ minimize bias for $\hat{f}$



Linear regression in high dimensions

Linear Regression (another review)

We model the world as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$$

We estimate the model parameters by minimizing:

$$RSS = \underbrace{\sum_{i=1}^{N}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2} = \text{loss function}$$

Ridge Regression

(Linear Regression w/ Ridge (L2) Regularization)

complexity of model $\|\beta\|_2^2$ L2 norm

We model the world as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon$$

(same as before)

We estimate the model parameters by minimizing:

(the "regularization" parameter)

loss function $= \underbrace{\sum_{i=1}^{N} (y_i - \hat{\beta}_0 - \sum_{j=1}^{p} x_{ij} \hat{\beta}_j)^2}_{\text{RSS}} + \lambda \underline{\sum_{i=1}^{p} \hat{\beta}_i^2}$

penalty for complexity

(new term!)

Did we see this before?

Yes: Subset Selection

RSS

complexity penalty

$$C_p = \frac{1}{n}(RSS + 2p\hat{\sigma}^2)$$

Mallow's Cp
p is the total # of parameters
$\hat{\sigma}^2$ is an estimate of the variance of the error, ε

$$AIC = -2logL + 2 \cdot p$$

L is the maximized value of the likelihood function for the model estimated

goodness of fit

$$BIC = \frac{1}{n}(RSS + log(n)p\hat{\sigma}^2)$$

This is Cp, except 2 is replaced by log(n). log(n) > 2 for n>7, so BIC generally exacts a heavier penalty for more variables

$$Adjusted\ R^2 = 1 - \frac{RSS/(n-p-1)}{TSS/(n-1)}$$

Similar to R^2, but pays price for more variables

Side Note: Can show AIC and Mallow's Cp are equivalent for linear case

Ridge Regression

$$\sum_{i=1}^{N}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2 + \lambda \sum_{i=1}^{p} \hat{\beta}_i^2$$

What if we set the lambda equal to zero? $\lambda = 0 \Rightarrow$ RSS/ OLS

What does the new term accomplish? built-in feature selection

What happens to a features whose corresponding coefficient value (beta) is zero? they "disappear"

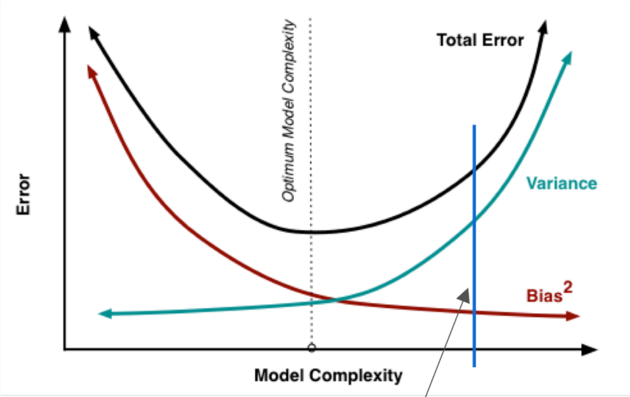
Ridge Regression

$$\sum_{i=1}^{N}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2 + \lambda \sum_{i=1}^{p} \hat{\beta}_i^2$$

Notice, we do not penalize B_0.

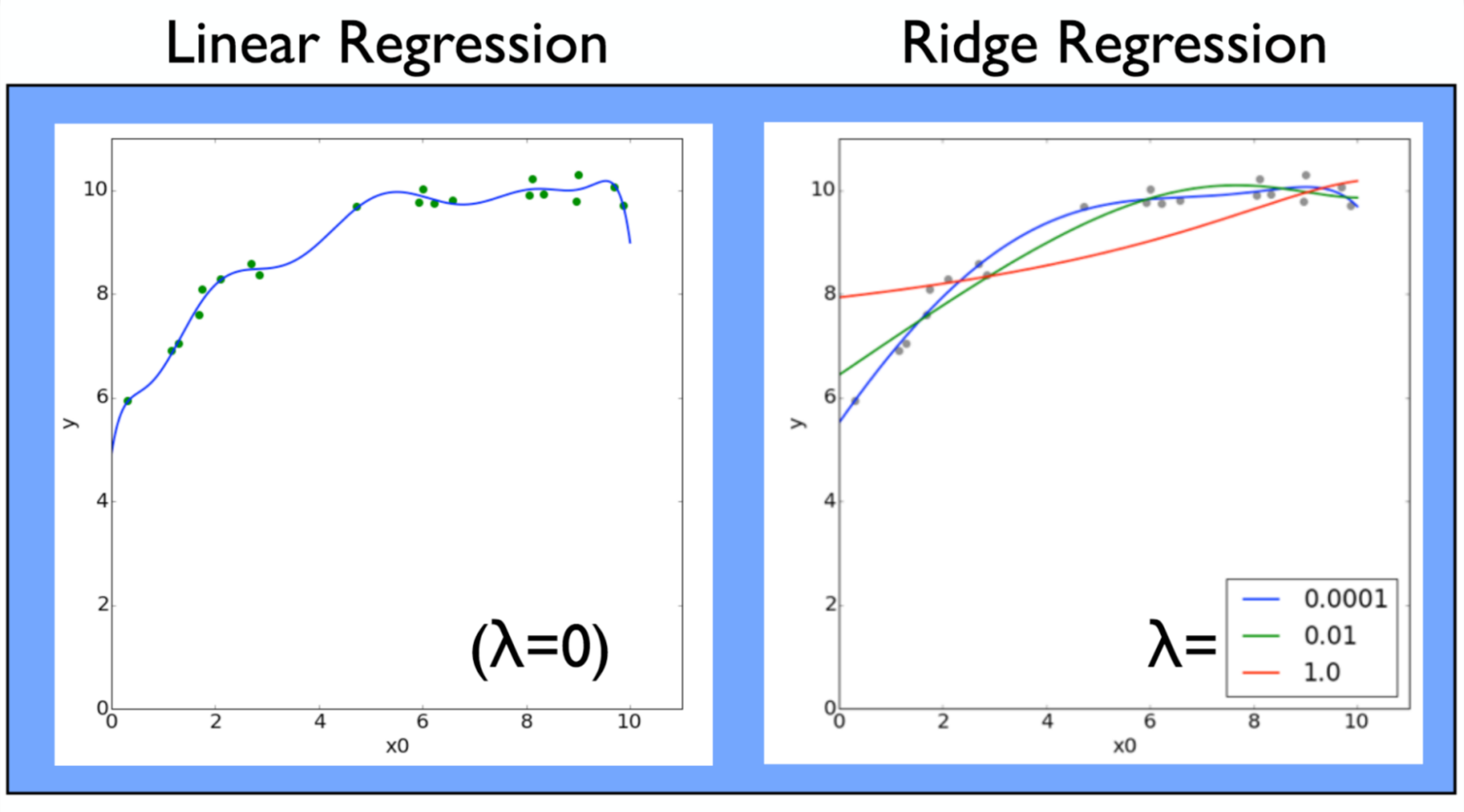
Changing lambda changes the amount that large coefficients are penalized.

Increasing lambda increases the model's bias and decreases its variance.



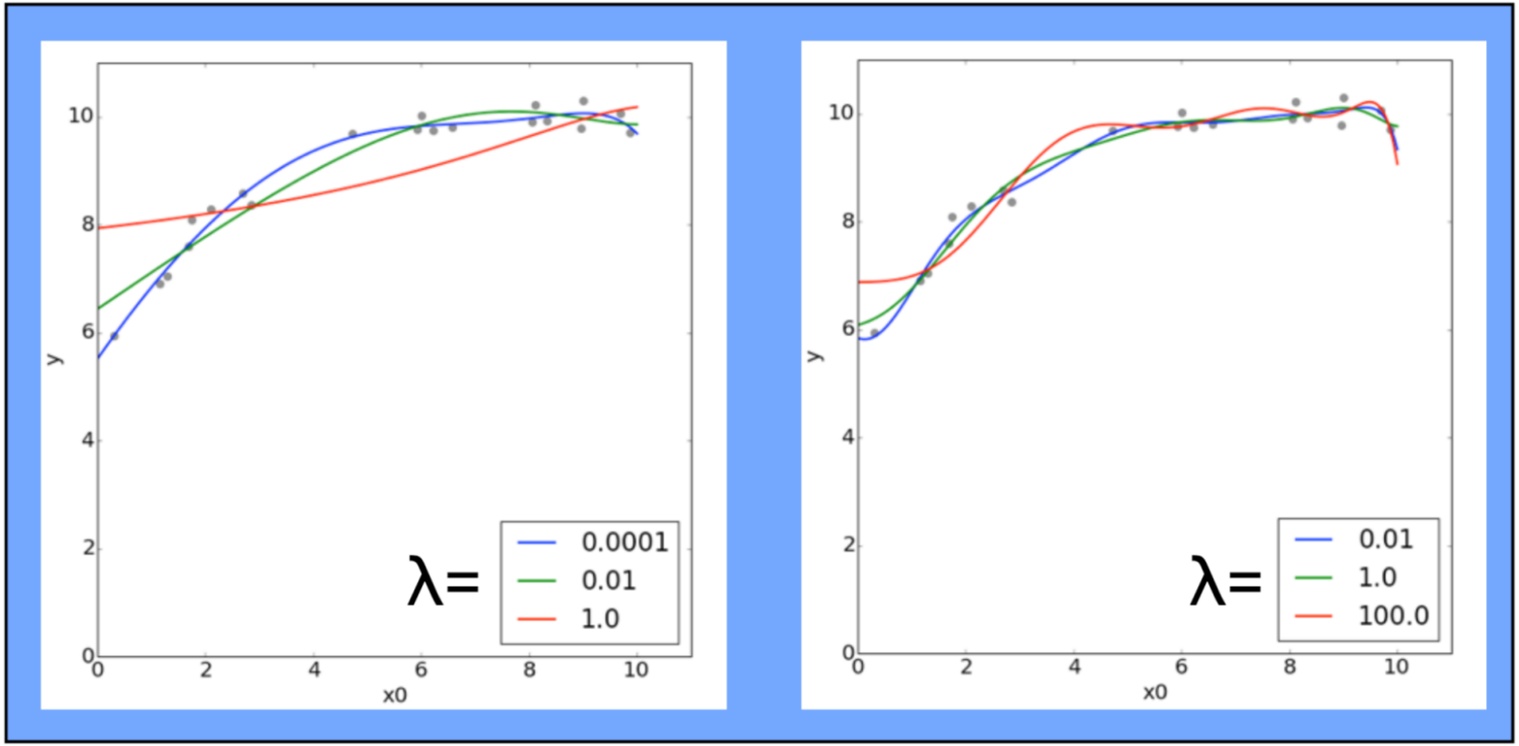
Linear regression in high dimensions

Ridge Regression



Ridge Regression

Normalized Data — Non-Normalized Data



Single value for λ assumes features are on the same scale!!

Lasso Regression

LASSO

(Linear Regression w/ Lasso (L1) Regularization)

We model the world as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$$

least absolute shrinkage and selection parameter

(same as before)

We estimate the model parameters to minimizing:

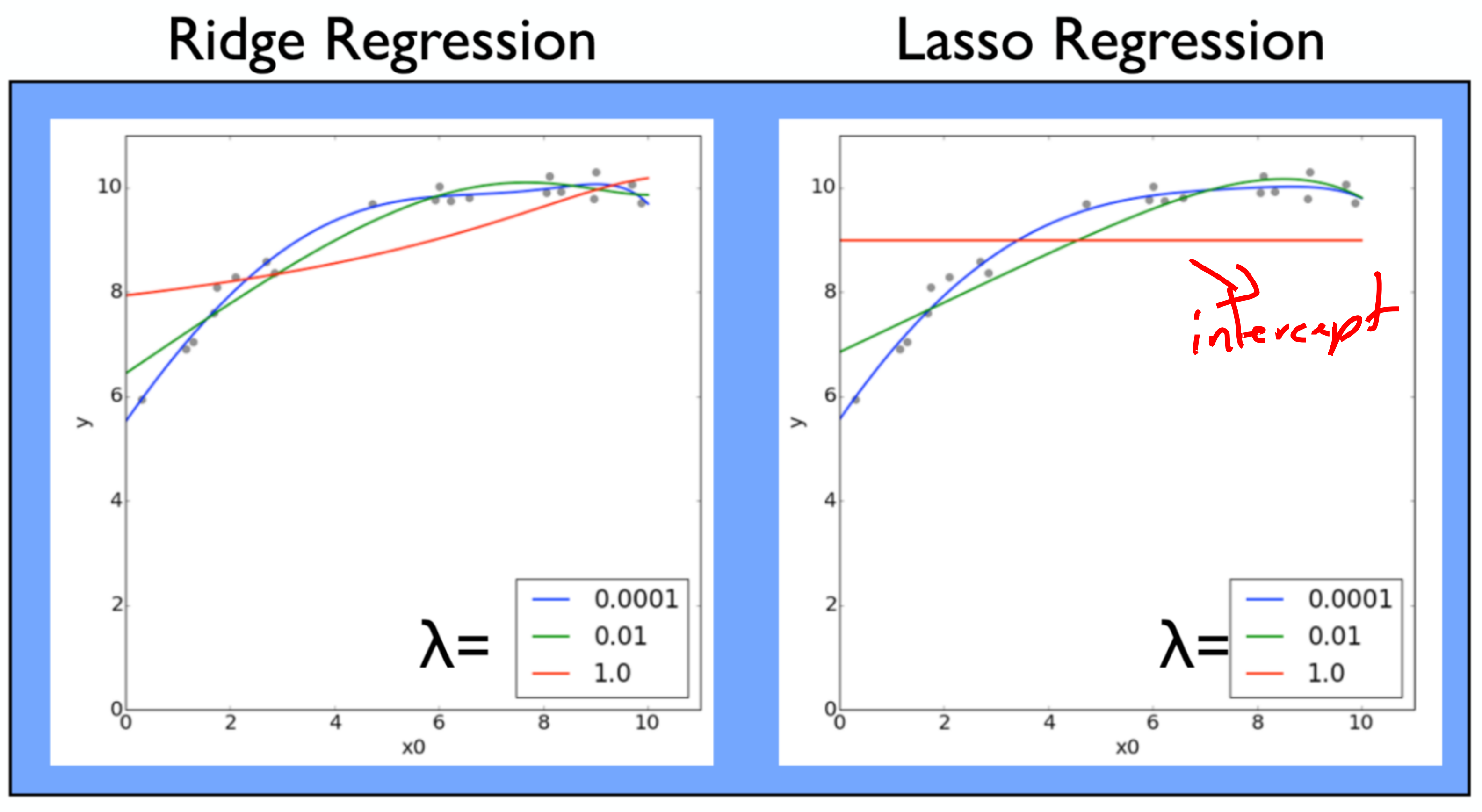
(the "regularization" parameter)

$$\sum_{i=1}^{N}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2 + \lambda \sum_{i=1}^{p} |\hat{\beta}_i|$$

(absolute value instead of squared)

$\|\beta\|_1$, norm L1

Lasso Regression



Ridge vs Lasso

Which is better depends on your dataset!

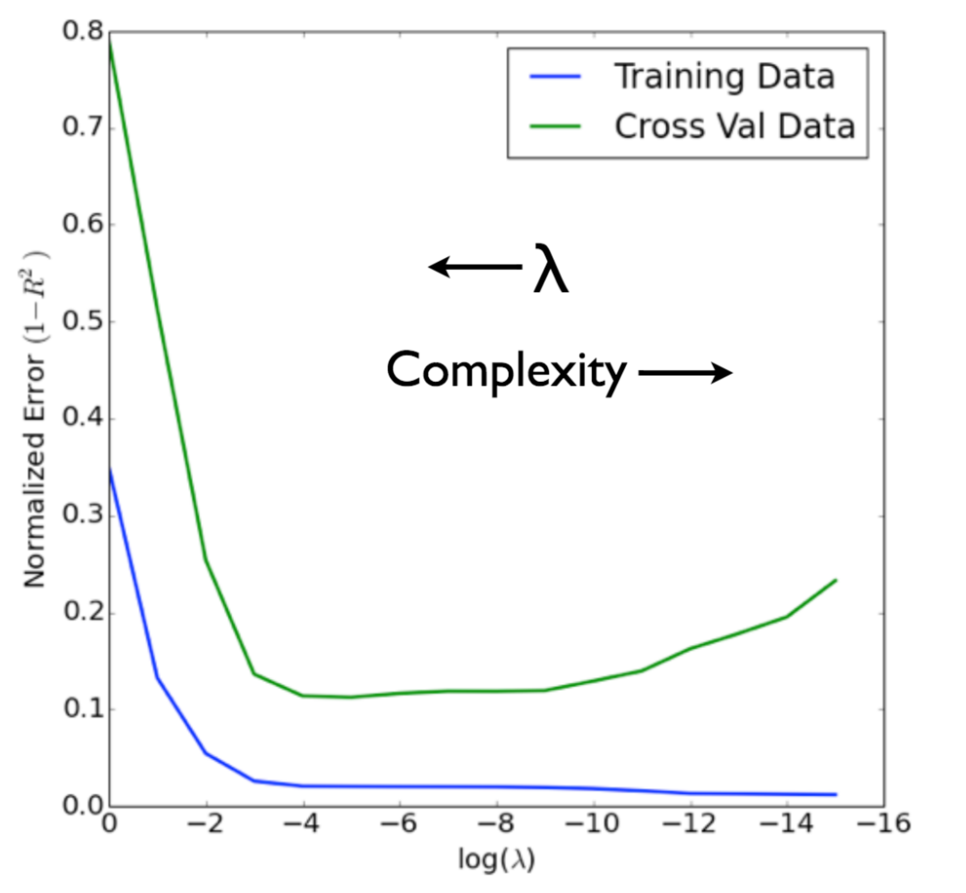
True sparse models will benefit from lasso; true dense models will benefit from ridge.

Ridge forces parameters to be small + Ridge is computationally easier because it is differentiable

Lasso tends to set coefficients exactly equal to zero

- This is useful as a sort-of "automatic feature selection" mechanism,
- leads to "sparse" models
- serves a similar purpose to stepwise features selection

Chose lambda via Cross-Validation



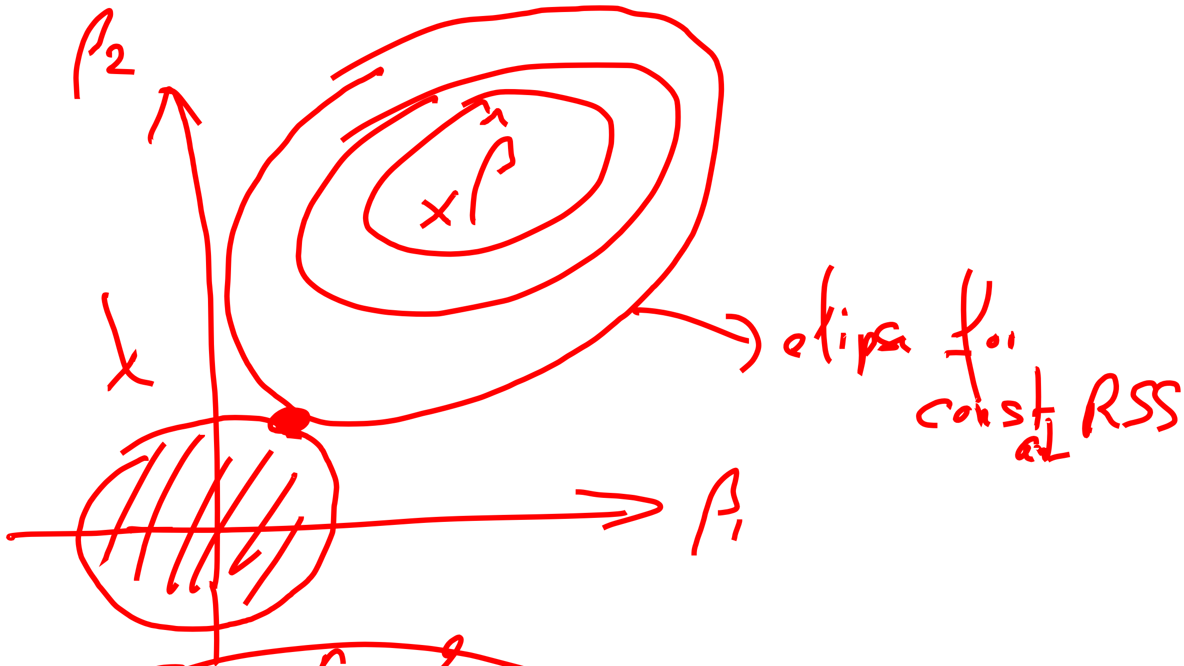
scikit-learn

Classes:

- sklearn.linear_model.**LinearRegression**(...)
- sklearn.linear_model.**Ridge**(alpha=my_alpha, …)
- sklearn.linear_model.**Lasso**(alpha=my_alpha, …)

All have these methods:

- fit(X, y)
- predict(X)
- score(X, y)



Ridge $\|\beta\|_2 = \sum_{i=1}^{n} \beta_i^2$

LASSO $\|\beta\|_1 = \sum_{i=1}^{n} |\beta_i| = |\beta_1| + |\beta_2|$