Regularization

Constraint it

Constraint Model ⇝ Solve Overfitting

- for polynomial reg ⇒ constraint degree
- for linear reg ⇒ actually constraint weights

Table of Contacts :

1. Ridge Regression
2. lasso Regression
3. Elastic Net

⇒ Three different way to constraint weights

① Ridge Regression

(Tikhonov Regularization)

- Regression + Regularizations

with L_2 norm

- let $\vec{w} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{bmatrix}$ $\quad \|\vec{w}\|_2^2 = \theta_1^2 + \theta_2^2 + \cdots$

- Cost function after Regularization

$$J(\vec{\theta}) = MSE(\theta) + \frac{\alpha}{2} \sum_{i=1}^{n} \theta_i^2$$

$$\sum_{i=1}^{n} \theta_i^2 \rightarrow \|w\|_2^2$$

- **Notes** : 1- it penalty large weights of features (fit train set with small weights as possible ↳ it a)

  - large α will lead model to be ⇒ flat line around average
  - small α will lead model ≃ Normal linear Regress.

2- Bias term not added to regularization

3- Use only Regularized performance measure in training

NOTE

It is quite common for the cost function used during training to be different from the performance measure used for testing. Apart from regularization, another reason they might be different is that a good training cost function should have optimization-friendly derivatives, while the performance measure used for testing should be as close as possible to the final objective. For example, classifiers are often trained using a cost function such as the log loss (discussed in a moment) but evaluated using precision/recall.

For training :

1- Closed form

before : $\hat{\theta} = (X^T X)^{-1} X^T y$

after : $\hat{\theta} = (X^T X + \alpha A)^{-1} X^T y$

$A = \begin{bmatrix} 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}_{(n+1)(n+1)}$

2- Gradient descent

$J(\theta) = MSE(\theta) + \frac{\alpha}{2} \sum_{i=1}^{n} \theta_i^2$

$\nabla J = \nabla MSE(\theta) + \alpha \vec{w}$

Just derive $\frac{\alpha}{2} \frac{d}{d\theta_2} (\theta_2^2) = \alpha \theta_2$

Observation

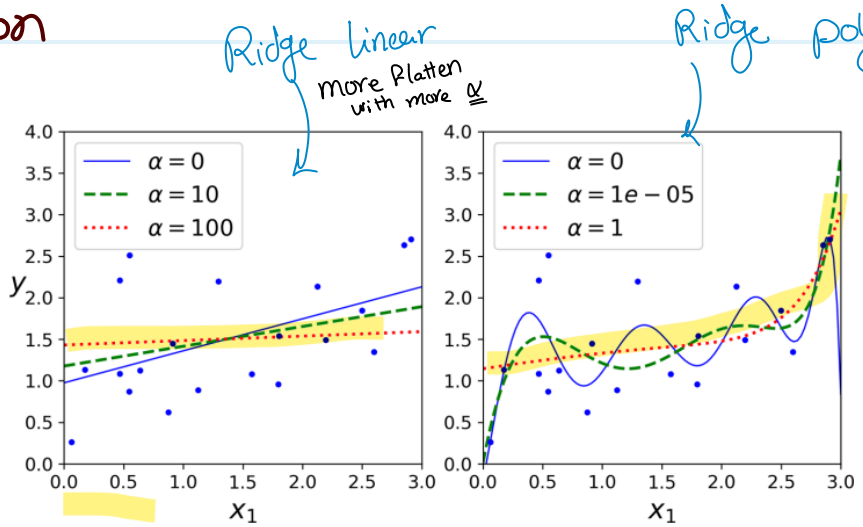


Figure 4-17. A linear model (left) and a polynomial model (right), both with various levels of Ridge regularization

With "α large" ⇝ underfit

"α small" ⇝ overfit

Must ReScale // implementation in Note Book

② Lasso Regression

① — least absolute shrinkage & selection operator

② — add l_1 norm instead of $\frac{1}{2}$ l_2 norm

③ — need less "α" values than ridge

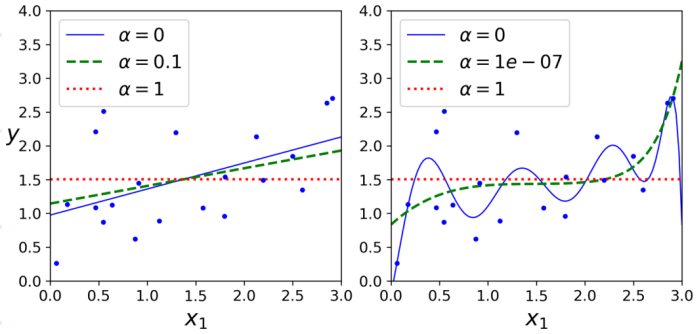


Figure 4-18. A linear model (left) and a polynomial model (right), both using various levels of Lasso regularization

④ $J(\theta) = \overrightarrow{MSE} + \alpha \sum_{i=1}^{n} \theta_i$

$\vec{g}$ instead of ∇J as it's not differentiable when "$\theta_i = 0$"

$\vec{g} = \nabla_\theta MSE(\theta) + \alpha \begin{pmatrix} sign(\theta_1) \\ sign(\theta_2) \\ sign(\theta_n) \end{pmatrix}$

$\vec{g}$ = sub gradient vector

$sign(x) = \begin{cases} +1 \\ 0 \\ -1 \end{cases}$

⑤ tend to eliminate ⇒ the weights of least important features

↳ output sparse model "few non zero features"

↳ automatically feature selection

how? or Why! ①

⇒ implementation in Note book

⑥ Why & How?

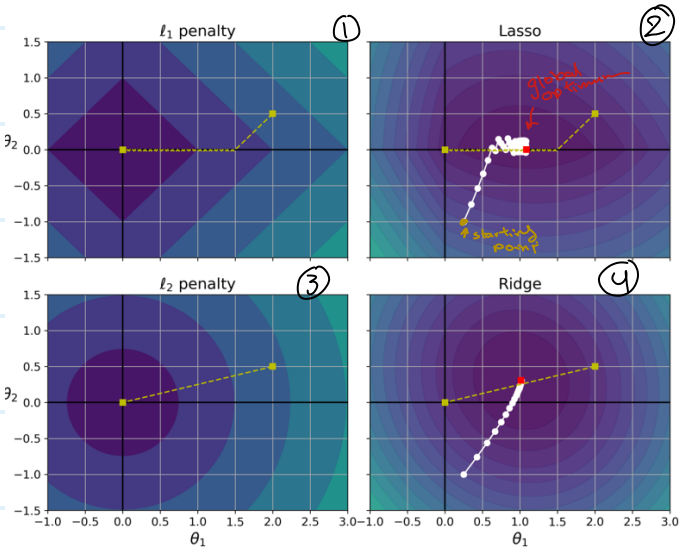


Figure 4-19. Lasso versus Ridge regularization

① L_1 penalty

$$\text{Loss} = J(\vec{\theta}) = \alpha \sum_{i=1}^{n} |\theta_i|$$

$$\nabla_\theta J(\theta) = \begin{pmatrix} \text{sign}(\theta_1) \\ \vdots \end{pmatrix}$$

$$\theta_i = \theta_i - \eta \ \text{sign}(\theta_i) \Leftarrow \text{gradient descent}$$

→ it decrease all features linearly till reach all zero (or close as it update $(\pm\eta, \text{zero})$)

→ it start @ $\vec{\theta} = \begin{pmatrix} 2.0 \\ 0.5 \end{pmatrix}$ same as optimal parameters for unregularized gradient "MSE"

② Lasso

$$\vec{J}(\vec{\theta}) = MSE(\vec{\theta}) + \alpha \sum_{i=1}^{n} |\theta_i|$$

$$\nabla J(\vec{\theta}) = \nabla MSE(\vec{\theta}) + \alpha \begin{pmatrix} \text{sign}(\theta_1) \\ \vdots \\ \text{sign}(\theta_n) \end{pmatrix}$$

$$\theta_i = \theta_i - \eta \left(\nabla_\theta MSE(\vec{\theta}) + \alpha \begin{pmatrix} ! \\ \vdots \end{pmatrix} \right)$$

$$= \theta_i - \eta \left(\underset{(MSE)}{\text{unregularized}} + \alpha \ L_1 \right)$$

if we decrease "α" red point move Right to unregularized global optimum

if we increase "α" red point move Left to regularized " "

Note : it keep Bouncing around global optimum in Both ①, ② images

→ as it update By $\pm\eta\alpha$ or zero

→ One solution's to gradually decrease η

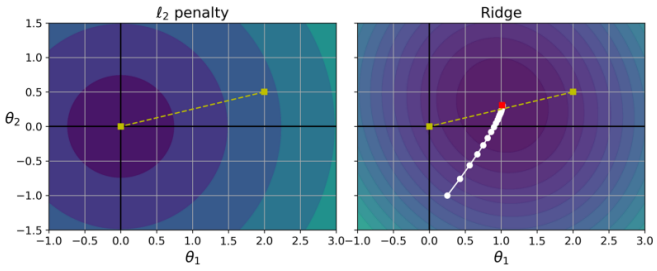
→ it will bounce around But Steps will get smaller till Converge

③ l_2 penalty , ④ Ridge

$$J_{L2} = \frac{\alpha}{2} \sum \theta_i^2$$

$$= \frac{\alpha}{2} \cdot \text{Squared distance of parameters from origin}$$

$$= \text{like circle}$$



→ gradient descent just take straight path to origin

Main two Difference

→ gradients get smaller as it approach global minimum

→ making it converge ⇒ no Bouncing around

→ the optimal parameters "red point" come close to zero as we increase "α" But never get elimenated (true zero)

③ Elastic Net

- middle ground Between (Ridge, Lasso)

$$J(\theta) = MSE(\theta) + r\alpha\sum_{i=1}^{n}|\theta_i| + \frac{(1-r)\alpha}{2}\sum_{i=1}^{n}\theta_i^2$$

$$r = \begin{cases} 0 \rightarrow \text{Ridge} \\ 1 \rightarrow \text{Lasso} \end{cases}$$

→ implementation in Notebook

✳ Frame Work to Choose From

① Ridge is Good Default

② Elastic Net
- → When Susbecting only Few Features are useful
- → prefered over "Lasso" → as Lasso perform Randomly when
  - ① #Features > #samples
  - ② Several Features are Strong correlated

③ Lasso

④ plain Linear Regression
- → it always perfer to have some ↑ Regularization or little bit of

(4) Early - Stopping

- Stop training when Validation Reach it's minimum & Before starting to go up again "Before overfit"

- called "Beautiful Free lunch"

- For "SGD", "Mini-Batch GD" the curve will be Bumbier & will be hard to determine "min point"
  ↳ Solution is to wait some time tell be sure then Roll-Back

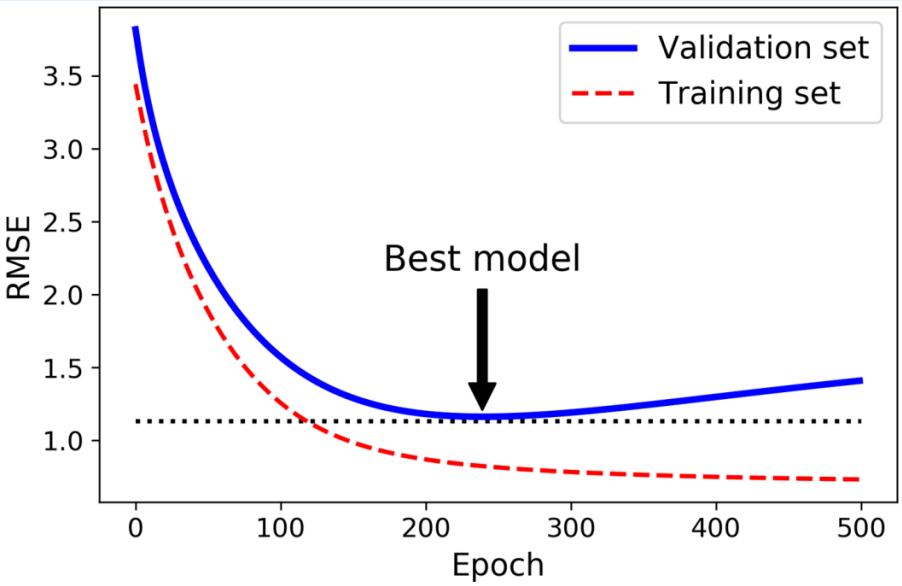


Figure 4-20. Early stopping regularization

⟹ implementation in NoteBook