① Bias Variance Tradeoff
② Regularization
③ Hyper-Parameter tuning
④ Data Normalization

Matrix Notation.

Overfit Underfit tradeoff

$y = w_0 + w_1 x_1 + w_2 x_2^2 + w_3 x_3^3 + \dots + w_d x_d^d$

output.

Cost:

$\{w_0, w_1, \dots, w_d\} \in \mathbb{O}$

$w_0 + w_1 \neq 0$

$\boxed{w_0 + w_1 = w_2 + w_3}$

[underfit]

Cons: $w_1 + w_2 \neq 0$

$w_1 + w_2 + w_3 = 0$

$w_1 = w_2 + w_3 = w_4 + w_5$

(Purpose model)

Cost: $w_1 + w_2 + w_3 + w_4 \neq 0$

[overfit]

Bias Variance Tradeoff

model

Bias Variance

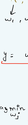
Bias & low variance, model

Low Bias High variance

Low Bias

[good]

High Bias

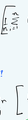


[underfit]

High variance

[overfit]

(high Bias, low w_1) $\rightarrow$ w_1 $\rightarrow$ underfit



(low Bias, low w_1) $\rightarrow$ (good model)

(low Bias, high w_1) $\rightarrow$ w_1 $\rightarrow$ overfit

Regularization

w_1, w_2, w_3, w_4, w_5

$w_1 - w_5$

Cost:

$y = w_0 + w_1 x_1 + w_2 x_2^2 + w_3 x_3^3 + w_4 x_4^4$

∞

Cost:

$\min_{w_j} \left[\frac{1}{2n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \frac{\lambda}{2} \sum_{j=1}^{d} w_j^2 \right]$

x_1, x_2, x_3, x_4, x_5

Regularization

Regularization term

actual weight in the regularization term.

Price: Increasing $(\lambda \uparrow)$ + Increase values of w_j

$w_j \downarrow$ good

How to control regularization.

Cost:

$\min_{w_j} \left[\frac{1}{2n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{d} w_j^2 \right]$

x_1, x_2, x_3, x_4, x_5

Hyper-Parameter

λ $\rightarrow$ [0, ∞]

$\lambda \rightarrow$ const. ≥ 0

Tuning!

$\arg\min_{w_j} \left[f(w_j) + \lambda \sum_{j=1}^{d} w_j^2 \right]$

λ^{Tune}

Tune Cost

$\lambda \rightarrow$ const.

λ_1	λ_2	λ_3	λ_4	λ_5

$\hat{y} = w_0 + w_1 x_1 + w_2 x_2^2 + \dots + w_d x_d^d$

weight of unused features will be 0 ?

in Lasso

Significance of λ : What is actually such is the weight

$\lambda \rightarrow [0, \infty)$

Cons: $\lambda = 0$. : "overfit"

Cost: "Compared with Nominal"

Cons: if $\lambda \geq 10^8$: "underfit"

$\rightarrow$ Bigger λ diminishing, & all weights are bigger to be close to 0. $\rightarrow$ [underfit]

$\lambda = 10^8$

$\left[\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{d} w_j^2 \right]$

$\hat{y} = w_0$

Why Square of w_j for reg.

$\frac{\partial L}{\partial w_j}$ $\quad$ $\underset{w_j}{\operatorname{argmin}} \sum_{i=1}^{n} \text{Loss}_i + \lambda \sum_{j=1}^{d} |w_j|$

$\lambda \sum_{j=1}^{d} |w_j|$

not differentiable at $|w_j| = 0$ (kink)

L1 kink

L2-Reg.

$\lambda \sum_{j=1}^{d} w_j^2$

$\|w\|_2^2$

$\|w\|_2^2 = (w_1, w_2)^2 = (w_1, w_2)^2$

L2-Regularization

Ridge Regularization.

$w_j \rightarrow$ ∞, $-\infty$.

$|w_j|$ $\rightarrow$ ∞ diff.

$w_j^2 \rightarrow$ ∞, $-\infty$.

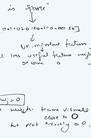
w_j^2 $\rightarrow$ ∞ or more times

L1 Regularization / Lasso Regularization.

Lasso

$+ \lambda \sum_{j=1}^{d} |w_j|$

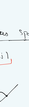
$\frac{\partial L}{\partial w_j}\Big|_{w_j=0}$ is not defined



$|w_j|$

L1 Lasso

$f(x) = |x|$



$\frac{d|w_j|}{dw_j} \Rightarrow \begin{cases} 1 & ; w_j > 0 \\ -1 & ; w_j < 0 \\ 0 & ; w_j = 0 \end{cases}$

$+ w_j \quad w_j > 0$

$-w_j \quad w_j < 0$

Interesting Property of L1-Reg.

L1-Reg

Lasso "Sparsity"

$\rightarrow$ w is sparse

[0.01, 0.0, 0.1, 0.0, 0.00, 0.5]

L1: unimportant features

[not 100% useful features weight set to 0]

L2-Reg

$\rightarrow$ why L1 creates sparsity?

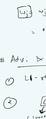
$w_j = 0$

all low useful features weight. because it tends close to 0

but not exactly $w_j = 0$.

why L1 creates sparsity

$\min \; + \lambda \sum_{j=1}^{d} |w_j|$



$\frac{\partial L}{\partial w_j} \Rightarrow$ 2

$\eta = 0.5$

$w_j = w_j - \eta \frac{\partial L}{\partial w_j}$

$\eta \cdot 2 = 0.5 \cdot 2 = 1$

$\min \; + \lambda \sum_{j=1}^{d} w_j^2$

$\frac{\partial L}{\partial w_j} \Rightarrow \boxed{\partial w_j}$

$w_j = w_j - \eta \frac{\partial L}{\partial w_j}$

Adv. & Disadv.

① L1 - reg. : $\rightarrow$ Lasso Sparsity $\rightarrow$ "feature selection"

② $\boxed{\lambda > 10}$ (Tune) (Tune) (general)

L1 will remove most of the non-important features

Elastic Net (L1 + L2)

$+ \lambda_1 \sum_{j=1}^{d} |w_j| + \lambda_2 \sum_{j=1}^{d} w_j^2$

$\lambda_1 \rightarrow$

$\lambda_2 \rightarrow$

Hyper-parameter

Tune

Cons

Model Hyper-Parameter.

Cons

Learning parameter, Hyper... (weights)

$\rightarrow$ Parameter, is not fixed, or not sensitive

$\eta \rightarrow 0.1$

$\lambda_1, \lambda_2 \rightarrow 1, 5$

Hyper-Parameter Tuning

$\lambda \rightarrow 1$ $\rightarrow$ Train + Predict. (w_j) $\rightarrow$ P_1

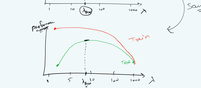
$\lambda \rightarrow 10$ $\rightarrow$ Train + Predict (w_j) $\rightarrow$ P_2

$\lambda \rightarrow 100$ $\rightarrow$ Train + Predict (w_j) $\rightarrow$ P_3

$\vdots$

P_r

Compare $P_1, P_2, P_3 \dots P_r$

Choose high P_r $\rightarrow$ [Tune] hyper-parameter

Tuning