Logistic Regression

I. Probability LR

Logistic Regression is an approach to learning function of the form $f: X \to Y$, or $P(Y|X)$ in the case where Y is discreate valued and $X = \langle x_1 \ldots x_n \rangle$ is any vector containing discreate or continous variables. LR = Gaussian Naive Bayes + Bernolli

Parametric model assumed by Logistic Regression in case Y is boolean is :

$$P(Y=1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)} \quad \text{--- (1)}$$

and

$$P(Y=0|X) = \frac{\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)} \quad \text{--- (2)}$$

Sum of probabilities (1) & (2) must be 1.

To classify any given X we generally want to assign the value y_k that maximizes $P(Y=y_k|X)$.

Example ~~tag~~ label $Y=0$ is assigned $Y=0$ if following condition holds

$$1 < \frac{P(Y=0|X)}{P(Y=1|X)} \quad \text{--- (3)}$$

$$P(Y=1|X) < P(Y=0|X)$$

Substituting (1) & (2) in (3)

$$1 < \exp\left(w_0 + \sum_{i=1}^{n} w_i X_i\right) \quad \text{--- (4)}$$

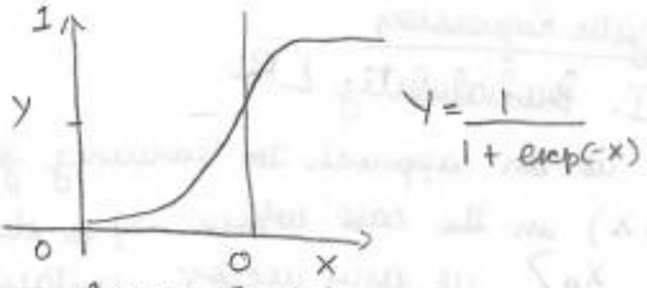


Figure : logistic Function

$$\ln(1) < \ln\left(\exp\left(w_0 + \sum_{i=1}^{n} w_i x_i\right)\right)$$

$$0 < w_0 + \sum_{i=1}^{n} w_i x_i$$

Derivations

We now derive the parametric form of $P(Y|X)$

$$P(Y=1|X) = \frac{P(Y=1)\,P(X|Y=1)}{P(Y=1)\,P(X|Y=1) + P(Y=0)\,P(X|Y=0)}$$

Dividing both the numerator and denominator by the numerator yields:

$$P(Y=1|X) = \frac{1}{1 + \frac{P(Y=0)\,P(X|Y=0)}{P(Y=1)\,P(X|Y=1)}}$$

or equivalently

$$P(Y=1|X) = \frac{1}{1 + \exp\left(\ln \frac{P(Y=0)\,P(X|Y=0)}{P(Y=1)\,P(X|Y=1)}\right)}$$

because of our conditional independence assumption we can write this

$$P(Y=1|X) = \frac{1}{1 + \exp\left(\ln \frac{P(Y=0)}{P(Y=1)} + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)}\right)}$$

$$= \frac{1}{1 + \exp\left(\ln \frac{1-\pi}{\pi} + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)}\right)} \quad \text{—(5)}$$

Note the final step expresses $P(Y=0)$ and $P(Y=1)$ in terms of the binomial parameter π.

Now consider just the summation in denominator of equation (5). Given our assumption that $P(X_i|Y=y_k)$ is Gaussian, we can expand this term as follows:

$$\sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)} = \sum_i \ln \frac{\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(\frac{-(X_i-\mu_{i0})^2}{2\sigma_i^2}\right)}{\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(\frac{-(X_i-\mu_{i1})^2}{2\sigma_i^2}\right)}$$

$$= \sum_i \ln \exp\left(\frac{(X_i-\mu_{i1})^2-(X_i-\mu_{i0})^2}{2\sigma_i^2}\right)$$

$$= \sum_i \left(\frac{(X_i-\mu_{i1})^2-(X_i-\mu_{i0})^2}{2\sigma_i^2}\right)$$

$$= \sum_i \left(\frac{(X_i^2-2X_i\mu_{i1}+\mu_{i1}^2)-(X_i^2-2X_i\mu_{i0}+\mu_{i0}^2)}{2\sigma_i^2}\right)$$

$$= \sum_i \left(\frac{2X_i(\mu_{i0}-\mu_{i1})+\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}\right)$$

$$= \sum_i \left(\frac{\mu_{i0}-\mu_{i1}}{\cancel{2}\sigma_i^2} X_i + \frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}\right) \quad \text{—(6)}$$

Subs. (6) in (5)

$$P(Y=1|X) = \frac{1}{1+\exp\left(\ln\frac{1-\pi}{\pi}+\sum_i\left(\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}X_i+\frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}\right)\right)}$$

Or equivalently,

$$P(Y=1|X) = \frac{1}{1+\exp\left(w_0+\sum_{i=1}^{n} w_i X_i\right)}$$

where $\quad w_i = \dfrac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}$

and where

$$\omega_0 = \ln \frac{1-\pi}{\pi} + \sum_i \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2}$$

$$P(Y=0|X) = 1 - P(Y=1|X) = \frac{\exp\left(\omega_0 + \sum_{i=1}^{n} \omega_i X_i\right)}{1 + \exp\left(\omega_0 + \sum_{i=1}^{n} \omega_i X_i\right)}$$

One reasonable approach to training logistic Regression is to choose parameter values that maximizes the conditional data likelihood.

$$W \leftarrow \arg\max_{W} \prod_{l} P(Y^l | X^l, W)$$

Y^l denotes the observed value of Y in the l^{th} training example

X^l denotes the observed value of X in the l^{th} training example

This conditional data likelihood → log likelihood.

$$l(\omega) = \sum_l Y^l \ln P(Y^l = 1 | X^l, \omega) + (1 - Y^l) \ln P(Y^l = 0 | X^l, \omega)$$

Note that Y can take values 0 & 1.

$$= \sum_l Y^l \ln \frac{P(Y^l = 1 | X^l, \omega)}{P(Y^l = 0 | X^l, \omega)} + \ln P(Y^l = 0 | X^l, \omega)$$

$$= \sum_l Y^l \left(\omega_0 + \sum_i^n \omega_i X_i^l\right) - \ln\left(1 + \exp\left(\omega_0 + \sum_i^n \omega_i X_i^l\right)\right)$$

Gradient ascent,

$$\frac{\partial l(\omega)}{\partial \omega_i} = \sum_l X_i^l \left(Y^l - \hat{P}(Y^l = 1 | X^l, \omega)\right)$$

Optimize weights ω.

$$\omega_i \leftarrow \omega_i + \eta \sum_l X_i^l \left(Y^l - \hat{P}(Y^l = 1 | X^l, \omega)\right)$$

step size / learning rate

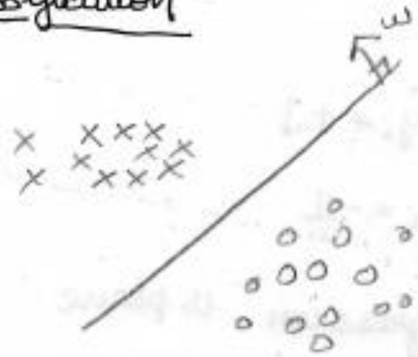
Regularization in Logistic Regression

Overfitting the training data is problem that can cause in logistic Regression especially when data is very high dimensional and training data is sparse. One approach to reduce overfitting is regularization

$$w \leftarrow \arg\max_{w} \sum_{e} \ln P(y^e | x^e, w) - \frac{\lambda}{2}\|w\|^2$$

◇ II Logistic regression (Geometric Intuition)

Used for classification



× → +ve class pt
O → -ve class pt

if my data is (linearly seperable)
almost linearly seperable

2D : line } linear
nD : hyperplane } surface

2D : line $y = mx + c$

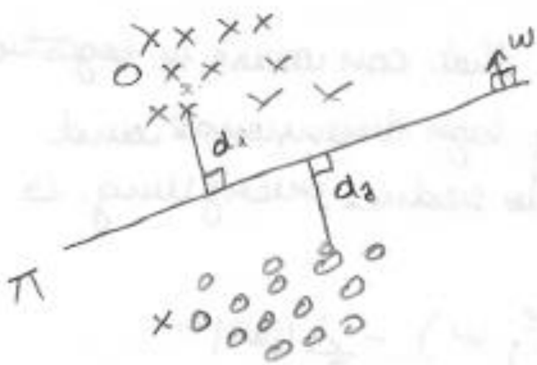
plane $w^T x + b = 0$

Assumption of LR: classes are almost / perfectly linear seperable.

Given : $\mathcal{D}_n = \{+ve, -ve\}$.

Task: find : $(w \& b)$

π that best seperates +ve pts from -ve pts



$y_i = +1$: +ve pt

-1 : -ve pts

$d_i = \dfrac{w^T x_i}{\|w\|}$; w is the normal to the plan.

$\|w\| = 1 \Rightarrow$ unit vector

$d_i = w^T x_i > 0$

$d_j = w^T x_j < 0$

classifier.

if $w^T x_i > 0$ then $y_i = +1$

if $w^T x_i < 0$ then $y_i = -1$

decision surface logistic Regression is plane

Case 1: $y_i * w^T x_i > 0$

$\hookrightarrow y_i = +1$

w is correctly classified pt

case 2: $y_i = -1$: -ve pt

$y_i * w^T x_i > 0$

$y_i = -1 \searrow -1$ is correctly classifying pt

Case 3 ÷ $y_i = +1$ (+ve pt)

$w^T x_i < 0 \Rightarrow$ LR is saying x_i is -ve class

$\boxed{y_i w^T x_i < 0} \Rightarrow y_i = +1$ misclassified pt.

LR $= -1$

Case 4 ÷ $y_i = -1$ (-ve pt)

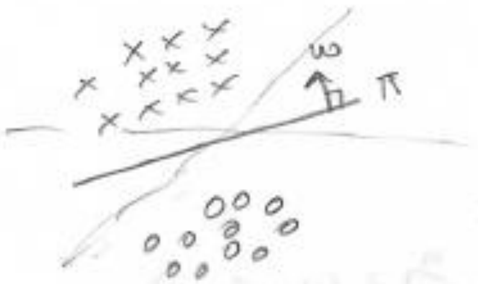
$w^T x_i > 0$

$y_i w^T x_i < 0 \Rightarrow y_i = -1$ misclassified pt.

Objective → minimum number of misclassification
maximum # correctly classified

$$w^* = \underset{w}{\text{argmax}} \left(\sum_{i=1}^{n} y_i w^T x_i \right)$$

Sigmoid function



$$w^* = \text{argmax} \sum_{i=1}^{n} \overbrace{y_i w^T x_i}^{\text{signed distance}}$$

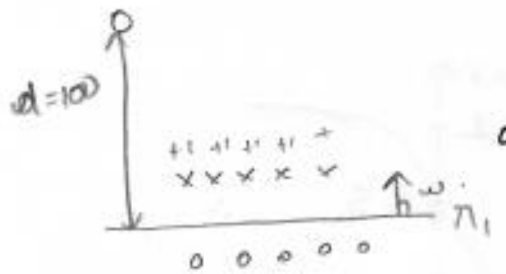
$w^T x_i$: dist. from x_i to π

signed distance

$y_i w^T x_i \rightarrow$ +ve correctly classified

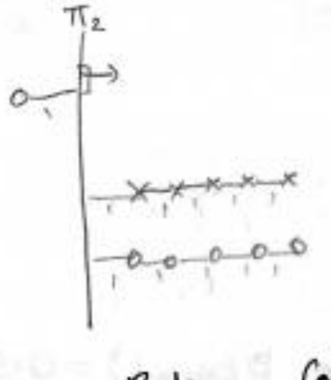
$y_i w^T x_i \rightarrow$ -ve incorrectly classified

Case 1:



$$\text{argmax} \sum_{i=1}^{n} y_i w_i^T x_i = 1+1+1+1+1 -100 + 1+1+1+1+1 = -90$$

Case 2



$$\text{argmax} \sum_{i=1}^{n} y_i w^T x_i = 1+2+3+4+5 -1-2-3-4-5 + 1 \rightarrow \text{outlier} = 1$$

~~Among~~ Between Case 1 and Case 2. Case 2 will be chosen as our objective is to find $\text{argmax} \sum_{i=1}^{n} y_i w^T x_i$. π_2 is better classifier. But it is terrible classifier

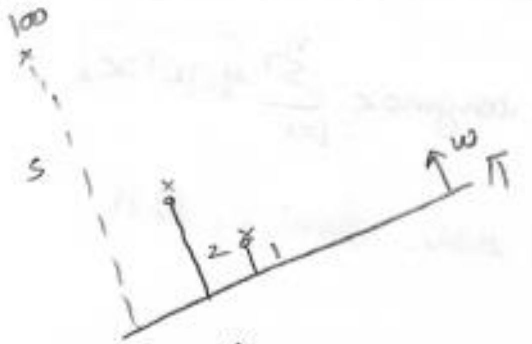
Because of single outlier point hyperplane changed.

So we introduce Sigmoid function

idea - instead of using signed distance

if signed distance is small :- use it.

______ is large : make it small



$$\underset{w}{\text{argmax}} \sum_{i=1}^{n} y_i w^T x_i \quad \Big| \quad \underset{w}{\text{argmax}} \sum_{i=1}^{n} f(\underbrace{y_i w^T x_i}_{\text{signed dist}})$$

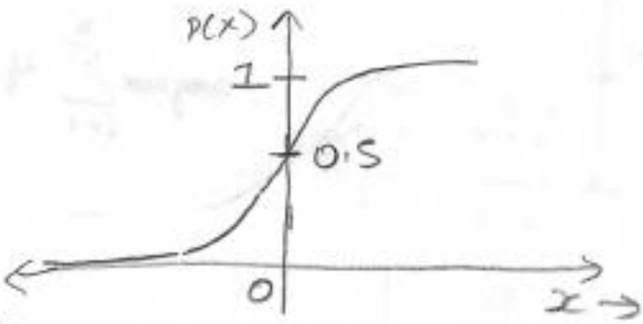
Sigmoid function $\sigma(x)$

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$\downarrow$ $\sigma(0) = 0.5$

max : 1

min : 0



$$\text{argmax} \sum_{i=1}^{n} \sigma(y_i w^T x_i)$$

Reason to choose sigma

1) Nice probabilistic interpretation

point lie on hyperplane $P(y_i = 1) = 0.5$

max. sum of signed dist → outlier

↓

$\sigma(x)$ → sigmoid

$$w^* = \underset{w}{\arg\max} \sum_{i=1}^{n} \sigma(y_i w^T x_i)$$

$$w^* = \underset{w}{\arg\max} \sum_{i=1}^{n} \frac{1}{1 + \exp(-y_i w^T x_i)}$$

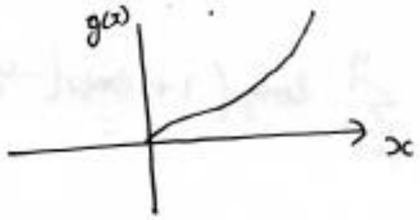
less impacted by outlier

Mathematical formulation of Objective function

☆ monotonic functions :- $g(x)$

$x \uparrow \quad g(x) \uparrow$ → monotonically increasing fn.

if $x_1 > x_2$ then $g(x_1) > g(x_2)$



$\log(x)$ → monotonic function

Theorem 1 :- if $g(x)$ is a monotonic fn.

$$\underset{x}{\arg\min}\ f(x) = \arg\min g(f(x))$$

$$\underset{x}{\arg\max}\ f(x) = \underset{x}{\arg\max}\ g(f(x))$$

Using Theorem 1

$$w^* = \text{argmax} \sum_{i=1}^{n} \frac{1}{1+\exp(-y_i w^T x_i)}$$

$g(x)$: $\log(x)$: montonic fn.

$$w^* = \text{argmax} \sum_{i=1}^{n} \log(\sigma(y_i \cdot w^T x_i))$$

$$w^* = \text{argmax} \sum_{i=1}^{n} \log \left[\frac{1}{1+\exp(-y_i \cdot w^T x_i)} \right]$$

Using $\Rightarrow \log \frac{1}{x} = -\log(x)$

$$w^* = \text{argmax} \sum_{i=1}^{n} -\log(1+\exp(-y_i w^T x_i))$$

$$\boxed{w^* = \underset{w}{\text{argmin}} \sum_{i=1}^{n} \log(1+\exp(-y_i w^T x_i))}$$

$\Leftarrow ||\ \text{argmax} f(-y) = \text{argmin} f(y)$

◇ <u>Weight-Vector</u>

$$w^* = \underset{w}{\text{argmin}} \sum_{i=1}^{n} \log(1+\exp(-y_i w^T x_i))$$

↓

weightvector

$$w = \langle w_1, w_2, w_3, \dots\ w_d \rangle$$

$$f_1, f_2, f_3, \dots\dots f_d \rightarrow \text{features}$$

Decision $x_q \rightarrow y_q$

If $w^T x_q > 0$ then $y_q = +1$

If $w^T x_q < 0$ then $y_q = -1$

Interpretation of w :

① If $w_i = +ve$, $x_{qi} \uparrow \rightarrow (w_i, x_{qi}) \uparrow$

f_i

$\Rightarrow \sigma(w^T x_q) \uparrow$

$\Rightarrow P(y_q = +1) \uparrow$

② If $w_i = -ve$, $x_{qi} \uparrow \rightarrow (w_i x_{qi}) \downarrow$

f_i

$\Rightarrow \left(\sum_{i=1}^{n} w_i x_{qi}\right) \downarrow$

$\Rightarrow \sigma(w^T x_q) \downarrow$

$\Rightarrow P(y_q = +1) \downarrow$

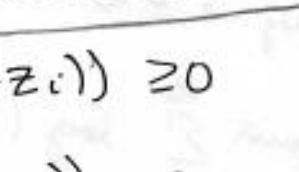
$\Rightarrow P(y_q = -1) \uparrow$

☆ L2 Regularization : Overfitting and Underfitting

$$w^* = \text{argmin} \sum \log(1 + \exp(-y_i w^T x_i))$$

$$\text{let } z_i = y_i w^T x_i$$

$$w^* = \text{argmin} \sum_{i=1}^{n} \log(1 + \underbrace{\exp(-z_i)})$$



is always +ve

$$\boxed{\exp(-z_i) \geq 0}$$

$$\log(1 + \exp(-z_i)) \geq 0$$

$$w^* = \text{argmin} \sum_{i=1}^{n} \log(1 + \exp(-z_i)) \geq 0$$

it occurs when $z_i \to \infty \ \forall i$

$z_i \to \infty$

↳ w modify my w in such a way that $z_i \to \infty$

1. $z_i = +ve$; x_i is correctly classified by w.

$z_i = +\infty$;

if I pick w s.t

a) all training pts are correctly classified → overfitting (perfect job on training data which doesnot tell perfect job on test data)

b.) $z_i \to \infty$

then we get best w

2. $w_i \to +\infty$ or $w_i \to -\infty$

one key aspect w_i is a normal to hyperplane

$$\boxed{w^T w = 1}$$

To get rid of this problem. regularization is used

$$w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{\infty} \underbrace{\log\left(1 + \exp(-y_i w^T x_i)\right)}_{\text{loss term}} + \underbrace{\lambda w^T w}_{\text{L1: Regularization term}}$$

$w_j \to \infty$, $w_j \to -\infty$] → regulizer will not allow it

Case 1 ⟶

$$W^* = \underbrace{\underset{w}{\operatorname{argmin}} \sum_{i=1}^{\infty} \log\left(1 + \exp(-y_i w^T x_i)\right)}_{0} + \underbrace{\lambda w^T w}_{\text{vv. large}}$$

$\lambda \to$ hyper parameter

$\lambda = 0 \Rightarrow$ overfit or high variance

$\lambda = \infty \Rightarrow$ underfitting as influence of regulizer increase.
or high bias

L1 Regularization and sparsity

$$w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} \log\left(1+\exp(-\overbrace{y_i w^T x_i}^{z_i})\right) + \underbrace{\lambda \|\cancel{w}\, w\|_2^2}_{L2\ \text{regulizer}}$$

hyperparameter

$$L1:\quad w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} \log\left(1+\exp(-y_i w^T x_i)\right) + \underbrace{\lambda\|w\|_1}_{L1\ \text{reg}}$$

will avoid $w \to \pm\infty$

~~Advantages~~ $\Rightarrow$

Sparsity $\rightarrow$ $w = \langle w_1, w_2, \ldots, w_d \rangle$
Solution to LR is said tobe sparse if many w_i'S are zero.

If we use L_1 reg in LR, all the less important features becomes zero.

less impor

$f_1, f_2, f_3 \ldots, f_i \quad f_d$

$w = \langle w, w_2, w_i, \ldots w_d \rangle$

we use L1 regularization $\Rightarrow$ creates sparsity.

Comparing Geometric and probability L.R

geom : $w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} \log(1+\exp(-y_i w^T x_i)) + \text{reg}$

y_i: $+1$, -1

probability : $w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} -y_i \log P_i - (1-y_i)\log(1-P_i)$

y_i: $+1$, 0 where $P_i = \sigma(w^T x_i)$

Case 1 y_i: +ve

geom $y_i = +1$
prob: $y_i = +1$

geom : $w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} \log(1+\exp(-w^T x_i))$ —(1)

prob : $w^* = \operatorname{argmin} \sum_{i=1}^{n} - \log\left(\frac{1}{1+e^{-w^T x_i}}\right) =$

$= \operatorname{argmin} \sum_{i=1}^{n} \log(1+e^{-w^T x_i})$ —(2)

(1) & (2) are same

Case 2 : y_i: -ve

geom : $y_i = -1$
prob : $y_i = 0$

geom $w^* = \operatorname{argmin} \sum_{i=1}^{n} \log(1+\exp(w^T x_i))$ —(3)

prob $w^* = \operatorname{argmin} \sum_{i=1}^{n} - \left(\log 1 - \frac{1}{1+e^{w^T x_i}}\right)$

$= \operatorname{argmin} \sum - \log\left(\frac{1+e^{w^T x_i}-1}{1+e^{w^T x_i}}\right)$ —(4)

$$w^* = \text{argmin} \sum - \log \frac{e^{w^T x_i}}{1 + e^{w^T x_i}}$$

$$= \text{argmin} \sum + \log \frac{1 + e^{-w^T x_i}}{e^{w^T x_i}}$$

$\log(-x) = \frac{1}{\log x}$

$$= \text{argmin} \sum \log \frac{(1 + e^{-w^T x_i})}{e^{-w^T x_i}}$$

$$= \sum \log 1 + \frac{1}{\exp^{-w^T x'}}$$

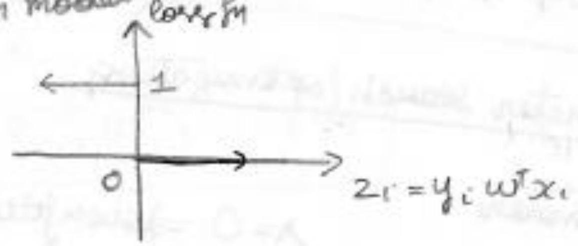
$$= \sum \log (1 + \exp^{w^T x_i})$$

(3) and (4) are same

III Loss minimization interpretation of LR.

Ideal optimization model

0-1 loss function



$z_i \rightarrow +ve$ loss fn should be 0 (for correct classification)

$z_i \rightarrow -ve$ (misclassification) loss fn should be 1

$$0\text{-}1 \text{ loss } (z_i) = \begin{cases} 1 & \text{if } z_i < 0 \\ 0 & \text{if } z_i > 0 \end{cases}$$

Solve optimization problem in ML

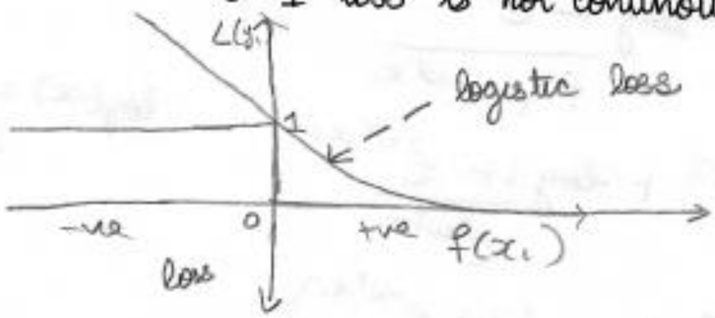
↳ differentiation in Calculus

Column / feature standardization

f_1 f_2 f_3 f_d

$x_i \in \mathbb{R}^d$

0 - 1 loss is not continous ⇒ not differentiable



◇ logistic regression is approximation of 0-1 loss.

1) In 0-1 loss $z_i > 0$ value is 0 but in logistic loss it is not zero but tends to become 0.

2) On the other hand negative side 0-1 gives value of 1 but logistic loss gives value greater than 1

Loss - minimization interpretation

↳ loss fn 1) logistic loss → LR

2) hinge loss → SVM.

3) SQ loss → linear loss

Hyperparameter search / optimization

λ: hyperparameter

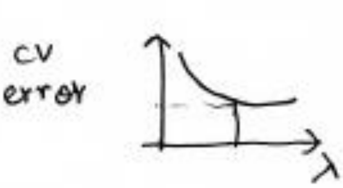
$\lambda = 0 \Rightarrow$ overfitting

$\lambda = \infty \Rightarrow$ underfitting

Q] How to determine the best λ ?

(λ) in LR is a real number.

1) Grid Search (Brute force)

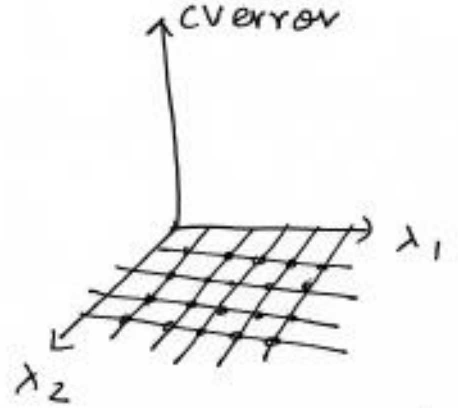


① $\lambda = [0.001, 0.01, 0.1, 1, 10, 100, 1000]$

② $\lambda = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

elastic net :-

$$\underbrace{\lambda_1 \|w\|_1}_{\text{L1 regularization}} + \underbrace{\lambda_2 \|w\|_2^2}_{\text{L2 regularization}}$$



$$\lambda_1 = \underbrace{[10^{-3}, 10^{-2}, 10^{-1}, \ldots, 10^{3}]}_{m_1}$$

$$\lambda_2 : \underbrace{[10^{-3}, 10^{-2}, \ldots, 10^{3}]}_{m_2}$$

λ_1 : 1 hyperparameter m_1 times m^1

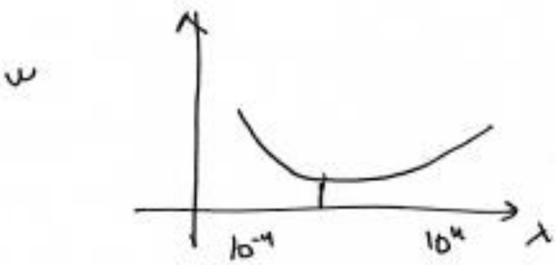
λ_1, λ_2 : 2 hyperpar $m_1 \times m_2$ m^2

$\lambda_1 \lambda_2 \lambda_3$: $m_1 \times m_2 \times m_3$ m^3

Grid search not good when we have many hyper parameter

2) Random Search :-

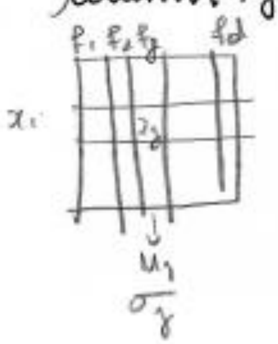
$\lambda \in [10^{-4}, 10^{4}]$ ← randomly pick values in the given interval



λ :→ hyperparameter search/optm

- → Grid search.
- → Random search.

Column / feature Standardization



$x_i \in \mathbb{R}^d$

$$x_{ij}' = \frac{x_{ij} - \mu_j}{\sigma_j} \text{ : standardization}$$