Logistic Regression

24.1 Geometric intution of Logistic Regression.

-> classification Technique

-> Simple & Elegant model

NB : Probabilistic model / Tech

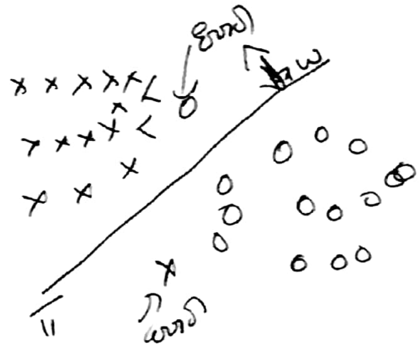
LR : geometric intuition.

Logistic Regression can be interpreted using below Technique

↳ Geometry

↳ Probability

↳ Loss function.



x -> +ve

o -> -ve

2D : line

nD : hyperplane } linear surface.

If my data is linear separable

If the π passes through origin = b = 0

$$w^Tx + b = 0 \Rightarrow w^Tx = 0$$

Assumption of Log Reg is class are almost / perfectly linearly separable

$$\pi = w^Tx + b$$

y_n = {+ve, −ve} -> given to us

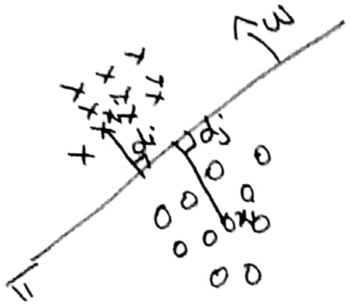
Task to Find : w & b

such that the line separates both +ve & −ve pts

Assumptions

NB: conditional indepence of features

K-NN: Neighbhdhood.



$y_i = +1$: +ve pts

-1 : -ve pts

$y_i \in \{-1, +1\}$

d_i = distance of point from plane

$= \frac{w^T x_i}{||w||}$; w is normal to the plane

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

$d_j = w^T x_j$

Since w & x_i are on the same side $d_i = w^T x_i > 0$

Since w & x_j are not on the same side $d_j = w^T x_j < 0$

Classifier says is

If $w^T x_i > 0$ then $y_i = +1$
$w^T x_j < 0$ then $y_i = -1$ } line passes through origin.

→ Decision surface in LR is a plane

→ Classifier to be v.good

↳ min # misclassificatn

or

↳ max # correctly classified pts

←

as many pts as possible to have $y_i * w^T x_i > 0$

Scanned by CamScanner

$\max \sum_{i=1} y_i w^T x_i$ optimal w^* means best hyperplane.

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

2.4.2 Sigmoid function & Squashing:

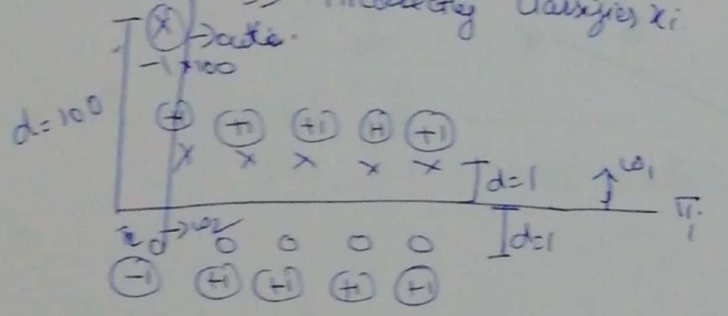
$$\text{argmax} \sum_{i=1}^{n} \underbrace{y_i w^T x_i}$$

↳ Signed distance.

$w^T x_i$ distance from x_i to π (w is a unit vector)

$y_i w^T x_i$: +ve ⇒ π as defined by w correctly classifies x_i

↳ : -ve ⇒ incorrectly classifies x_i

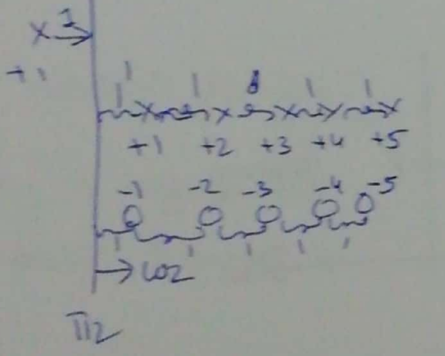


5 ✓ (correct)
5 ✓ (~~misclassified~~ correct)
1 × (misclassified)

Case 1: π_1 is my separator $\sum_i y_i w^T x_i = 1+1+1+1+1+1+1+1+1+1-100$

$\boxed{\therefore -90}$

Case 2:



$\sum_{i=1}^{n} y_i w_2^T x_i = 1+2+3+4+5-1-2-3-4-5 +1$ (outlier)

$= +1$

5 ✓
5 ×
1 ✓

One single extreme/outlier pt is changing my model (hyperplane) in Case 1 which is very bad.

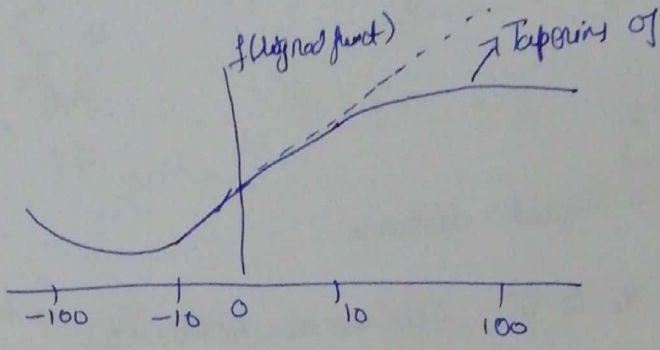
Max. sum of signed distances not outlier prone.

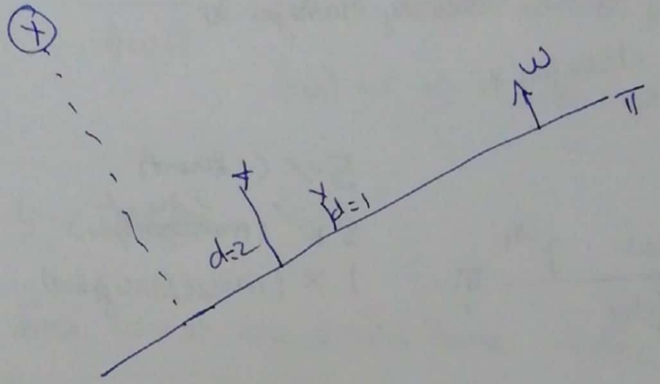
Squashing

idea: Instead of using signed distance.

If signed distance is small → use it as is

" " large → make it smaller as possible.

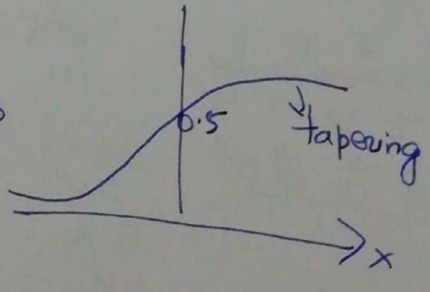




$$\underset{w}{\text{argmax}} \sum_{i=1}^{n} f(\underbrace{y_i w^T x_i}_{\text{signed distance.}})$$

Sigmoid function

$\sigma(x) = \dfrac{1}{1+e^{-x}}$ →



max: 1

min: 0

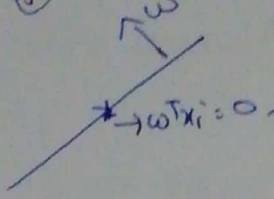
$\sigma(0) = 0.5$

we will change our problem to

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

Scanned by CamScanner

(1) → $w^T x_i$ is very large → $P(y=1) : 0.9999$



→ $P(y=1) : 0.5$

} Problistic Interpretation.

max. sum of signed dist → outlier problem.

↓

$\sigma(x)$ → Sigmoid

↳ tapperin linear

↳ problistic model.

max. sum of transformed signed interpretatio.

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

↓

$$w^* = \underset{(w)}{\operatorname{arg\,max}} \sum_{i=1}^{n} \frac{1}{1+\exp(-y_i w^T x_i)}$$ ← less impacted by outlier.

distance : $(-\infty, \infty)$

(squashing using σ function

↓

d (0 to 1)

why sigmoid function?

→ Easy to differentiate

→ problistic interpretation.

24.3 Mathematical formulation of Objective function

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

→ monotonic functions : $g(x)$

$x\uparrow$; $g(x)\uparrow$ monotonically increasing fn.

If $x_1 > x_2$ then $g(x_1) > g(x_2)$ then it is called monotonical function

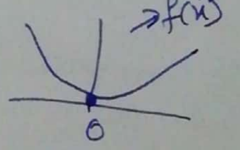
Eg $\log(x) > 0$; should be >0

Optimation problem:-

$$x = \underset{x}{\text{argmin}} (x^2) = 0$$

best(x)

x^2



min=0

x^2 is monotonic increasing when $x>0$

x^2 is " decrea when $x<0$

$g(x) = \log(x)$

$x^* = \text{argmin}(f(x))$; $f(x) = x^2$

$$x' = \underset{x}{\text{argmin}}\ g(f(x))$$

$$x' = \text{argmin} \log(x^2)$$

Claim $x^* = x'$

If $g(x)$ is a monotonic function

$$\underset{x}{\text{argmin}}\ f(x) = \underset{x}{\text{argmin}}\ g(f(x))$$

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

$x\uparrow$ $g(x)\uparrow$

$x\uparrow$ $g(x)\downarrow$

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

$g(x) = \log(x)$: monotonic fn.

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

$$\log(1/x) = -\log(x)$$

$$\left.w^* = \underset{w}{\operatorname{argmax}} \sum_{i=1}^{\omega} -\log(1+\exp(-y_i w^T x_i))\right\} \text{geometry}$$

$$\boxed{\operatorname{argmax} f(x) = \operatorname{argmin} -f(x)}$$

$$w^* = \operatorname{argmin} \sum_{i=1}^{n} \log(1+\exp(\underbrace{-\overset{\{-1 \text{ or } +1\}}{y_i} w^T x_i}_{\text{Singed dist}}))$$

$$\boxed{\log(e^x) = x}$$

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

$$\operatorname{argmin} \sum_{i=1}^{n} -y_i w^T x_i$$

Probabilistic method

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

$$\boxed{p_i = \sigma(w^T x_i)}$$

24.4 Weight Vector

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

$\downarrow$

weight vector $(w) = \langle w_1, w_2, w_3, w_4 \cdots w_d \rangle$

$\in \mathbb{R}^d \rightarrow$ d features of weight vector w

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

$\langle f_1, f_2, f_3 \cdots f_d \rangle$

decision If $x_q \rightarrow y_q$

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

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

Problistic functn

$\sigma(w^T x_q) = P(y_q = +1)$

$(0 \text{ to } 1)$

Interpretation of w :-

If $\underset{(f_i)}{\underset{\uparrow}{w_i}} = +ve$, $x_{qi} \uparrow \Rightarrow (w_i x_{qi}) \uparrow$

$\sum_{i=1}^{d} (w_i x_{qi}) \uparrow$

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

$P(y_q = +1) \uparrow$

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

$= \sum_{i=1}^{d} w_i x_{qi} \downarrow$

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

$= P(~~y~~y_q = +1) \downarrow$ $P(y_q = -1) \uparrow$

Scanned by CamScanner

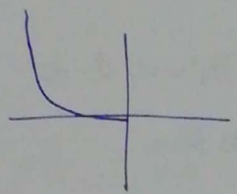
24.5 L2 Regularization: Overfitting & underfitting

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

Let $z_i = y_i w^T x_i \rightarrow$ Signed distance.

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

Plot $(\exp(-z))$ is always > 0



$\sum_{i=1}^{n} \log(1+ \exp(-z_i)) \geq 0$ $\Rightarrow \log(1) = 0$

$\log(2) > \log(1)$

$\log(1+\delta) > \log(1)$

$\delta \geq 0$

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

minimal value of $\sum_{i=1}^{n} \log(1+ \exp(-z_i)$ is zero

If $z_i = +ve$, $z_i \rightarrow +\infty$

Then $\exp(-z_i) \rightarrow 0$

$\log(1+ \exp(-z_i)) = 0$ since $\log(1) = 0$

If I pick my w such that

(a) all training points are correctly classified
$\downarrow$
overfit

(b) $z_i \rightarrow \infty$

Then that is called best w.

If we make $w_i \rightarrow \infty$ or $-\infty$ we will reach minima $= 0$

Regularization (To avoid w^T to become $+\infty$ or $-\infty$)

$$w^* = \underset{(w)}{\operatorname{argmin}} \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i)) + \lambda w^T w = \lambda \sum_{j=1}^{d} w_j^2$$

loss term

regularization term

$\lambda \|w\|_2^2$

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

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

When $\lambda = 0$, overfit → high variance, by making $z_i \to \infty$ & $-\infty$

λ = v large; underfit → high bias, we are ignoring loss term

we have to find the right λ using CV

24.6 L1 regularization and sparsity

$z_i \to +\infty$

$$w^* = \operatorname{argmin} \underbrace{\sum_{i=1}^{n} \log(1 + \exp(-y_i \underbrace{w^T x_i}_{z_i})}_{\text{logistic class}} + \underbrace{\lambda \|w\|_2}_{L_2\text{-reg}}$$

Alternative to L2 reg is L1

$\|w\|_2^2$ for reg
↓
$\|w\|_1$ for reg: $\|w\|_1 = \sum_{i=1}^{d} |w_i|$

$$w^* = \underset{(w)}{\operatorname{argmin}} \left(\text{Logistic loss for training data}\right) + \underbrace{\lambda \|w\|_1}_{L_1\text{-reg}}$$

→ hyper parameter

Will avoid $w_i \to +\infty$
$w_i \to -\infty$

Sparsity :- $w = \langle w_1, w_2, \dots w_d \rangle$

Solution to LR is said to be sparse if many w_i's are zero

If we use L_1 reg in LR, all the unimportant (or) less important become zero

$$\begin{array}{l} \qquad\qquad f_1, f_2 \dots \overset{\text{Less important}}{f_i} \dots f_d \\ w = \langle w_1, w_2 \dots w_i \dots w_d \rangle \end{array}$$

$\downarrow$

zero if L_1 is used.

If L_2 reg is used; w_i becomes a small value but not necessarily zero.

(Q) why does L_1 reg create sparsity in w as compared to L2 reg

ppl generally use L1 than L2

→ Elastic net :- Either L1 & L2

$$w^* = \underset{w}{\text{argmin}} \sum_{i=1}^{n} \log(1 + \exp(-z_i)) + \lambda_1 \|w\|_1 + \lambda_2 \|w\|_2^2$$

we have to find two hyper parameters λ_1 & λ_2

24.7 Probabilistic Interpretation: Gaussian Naive Bayes

cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf

LR ⇒ GNB + Bernoulli

$\downarrow \qquad\qquad \downarrow$

$P(x_i|y_i) \qquad y_i \sim$ Bernoulli

24.8 Loss minimization interpretation

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

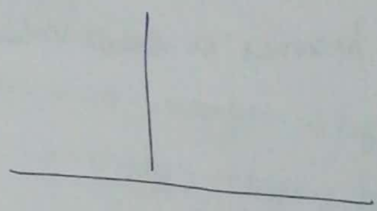
$$z_i = y_i w^T x_i = y_i f(x_i)$$

If we build a ideal optimization model.

$$w^* = \underset{w}{\text{argmin}} \text{ (num. of incorrectly classified pts)}$$

function
+1 : incorrectly classified
0 : correctly classified

min : loss
max : profit



$z_i = y_i w^T x_i$
$y \cdot f(x_i)$

24.9 hyperparameter & random search

λ : hyper paramater

λ = 0 ⇒ overfitting

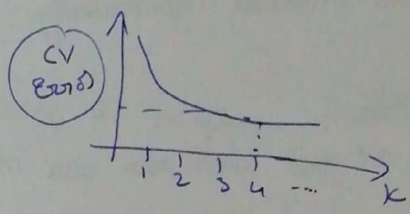
λ = ∞ ⇒ underfitting

Q How to find the best 'λ'

λ = LR

K = KNN

α = NB (Laplace Smoothing)



K in KNN is an integer, which takes value {1, 2, 3 .... n}

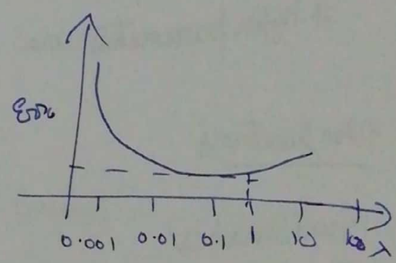
λ in LR is a real number.

$\lambda \in \mathbb{R}$ $\begin{cases} \lambda = 0.1234 \\ \lambda = 0.2386 \end{cases}$

⇒ One technique to find λ is GRID Search.

Case 1 : λ = [0.001, 0.01, 0.1, 1, 10, 100, 1000]

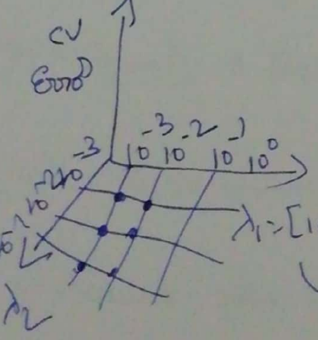
Case 2 : λ = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]



→ Generally ppl select a large window.

$\lambda = [10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10, 10^{1}, 10^{2}, 10^{3}, 10^{4}]$

Elasticnet : $\lambda_1 \|w\|_1 + \lambda_2 \|w\|_2^2$



$\lambda_1 = \underbrace{[10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10, 10^{1}, 10^{2}, 10^{3}]}_{m_1}$

$\underbrace{[10^{-3}, 10^{-2}, \dots 10^{3}]}_{m_2}$

$(m_1 \times m_2)$

Grid search

λ_1 : 1 hyperparameter : (m_1)

$\lambda_1 \lambda_2$: 2 " : $m_1 \times m_2$

$\lambda_1, \lambda_2, \lambda_3$: 3 " : $m_1 \times m_2 \times m_3$

as # hyper parameters increase, the # times model needs to be trained increases exponentially

Grid Search

is not good when there are more hyper parameters

To overcome this issue we have another technique called

Random Search

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

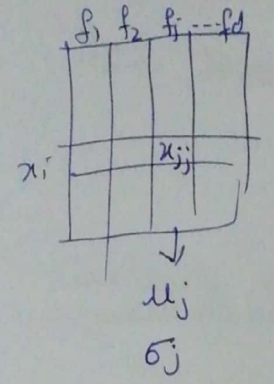
→ Random Search is almost as good as grid search especially when # hyper parameters are large.

Other functions

GridSearchCV

RandomizedSearchCV

24.10 Column Standardization (Z-score)



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

$$x_{ij}' = \frac{x_{ij} - \mu_j}{\sigma_j}$$ = Standardization

Even in Logistic Regression its mandatory to perform feature standardization before training

mean-Centering & Scaling } Standardization.

24.11 Feature Importance & model interpretability

f_1 f_2 f_j f_d

$w \rightarrow$ w_1 w_2 w_j w_d.

assume If all features are independent (Naive Bayes)

feature Importance can be acheived based on the weights

In K-NN: feature imp $\rightarrow$ forward feature Selection.
↳ we cannot get directly.

NB: $P(x_i | y=+1)$ $\rightarrow$ features which are Important.

LR: w_j's $\rightarrow$ to determine feature Importance.

$|w_j|$ = absolute value of weigh corresponding to f_j

$|w_j| \uparrow$; $(w^T x_i) \uparrow$

Case 1 $w_j = +ve$ & large ; $\sum_{j=1}^{d} w_j \cdot x_{qj} \Rightarrow w^T x_q$

$\hookrightarrow P(y_q = +1) \Uparrow$

Case 2 : $w_j : -ve$ & large ; $\sum_{j=1}^{d} w_j x_{qj} \Rightarrow P(y_q = -1) \Uparrow$

We can determine the important features in LR based on the weights

E.g.: Predict the gender : male & female
(+1) (-1)

1 feature: hair_length = $|w_{hl}|$ is large

$w_{hl} : -ve$

$w_{hl} \Uparrow$; $P(y_q = -1) \Uparrow$

2 feature: height $\uparrow$; $P(y_q = +1) \Uparrow$
$\uparrow$ male

$w_h = +ve.$

model interpretability

$x_q \rightarrow y_q = +1 \rightarrow$ Top features
$\downarrow$ hair_length, height$\uparrow$

$x_q \rightarrow y_q = -1$

24.12

Collinearity of features

Feature Importance : features are independant

$|w_j|$ as F.I values.

Collinearity (or) multicollinearity

collinearity :- f_i, f_j

s.t if $f_i = \alpha f_j + \beta$

Then f_i & f_j are collinear.

multicollinearity

If f_1, f_2, f_3 & f_4 such that

$f_1 = \alpha_1 + \alpha_2 f_2 + \alpha_3 f_3 + \alpha_4 f_4$

Then f_1, f_2, f_3 & f_4 are said to be multicollinearity

(Q) why does $|w_j|$ not be useful as f.I if features are collinear?

$D = \langle x_i, y_i \rangle_{i=1}^{n}$

$w^* = \langle 1, 2, 3 \rangle$; $x_q = \langle x_{q1}, x_{q2}, x_{q3} \rangle$

$f_1, f_2\ f_3$

$w^{*T} x_q = x_{q1} + 2x_{q2} + 3x_{q3}$

If $f_2 = 1.5 f_1 \Rightarrow f_1$ & f_2 are collenian.

$w^T x_q = x_{q1} + 3x_{q2} + 3x_{q3} = 4x_{q1} + 3x_{q3}$

$\langle 4, 0, 3 \rangle$

$x_{q1}\ x_{q2}\ x_{q3}$

$\rightarrow f_3$ is imp

$w^* = \langle 1, 2, 3 \rangle$

$\tilde{w} = \langle 4, 0, 3 \rangle$ $\therefore$ assumption are completely changing

f_1 is imp

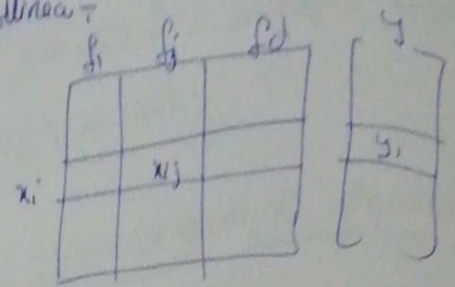
if features are collinear $\Rightarrow$ weight vector can can change arbitrarily $\Rightarrow$ $|w_j|$ can be used for feature importance

$|w_j|$ as F.I

determine if features are multicollinear ÷

Perturbation technique

means to share the value a little by adding a ϵ



Standardized $\rightarrow x_{ij} + \epsilon$

small noise

$N(0, 0.01)$

Before Perturbation ÷ $w = \langle w_1, w_2, \ldots w_j \ldots w_d \rangle$

after " ÷ $\tilde{w} = \langle \tilde{w}_1, \tilde{w}_2 \ldots \tilde{w}_j \ldots \tilde{w}_d \rangle$

If w_i & $\tilde{w}_i$ differ significantly than your features are collinear

$|w_j|$'s as F.I cannot be used

24.13 Test/Run time space & time Complexity

Train LR ÷ solving Logistic Regression problem.

Training time of LR is $O(nd)$

After this we get $w^* = \langle w_1, w_2, w_3 \cdots w_d \rangle$

$w^{*T} x_q > 0 \rightarrow +ve$

$< 0 \rightarrow -ve.$

At runtime we have to store w^T

Space $= O(d)$ $\rightarrow$ memory Efficient as well

Time $\div O(d)$

If d is small ~~for~~, LR is VVgood for low latency application

$x_q \rightarrow 1ms \rightarrow y_q$

If d is large $d \approx 1000$

$w^T x_q$ ÷ 1000 multi & addition.

→ L1 reg ÷ Sparsity (w_j's corresponding to less important features = 0)

λ ÷ reasonability

λ↑ ; Sparsity ↑

more of $w_j = 0$ { 50 mult & 50 additions

↑ latency

Bias vs Latency

λ↑ ; Bias ↑, latency ↓

24.14 Real world cases

Decision surface ÷ Linear / hyperplane. { +

{ -

assumption ÷ data is linearly separable or almost linearly separable.

Imbalanced data : upsampling or down sampling.

Outliers : less impact ∵ of $\sigma(x)$

→ $D_{train} \rightarrow w^*$

→ $x_i \rightarrow w^{*T} x_i$ :: distance from plane π to point x_i

→ remove points which are ~~are~~ very far away from π

from $D_{train} \rightarrow D'_{train}$.

→ $D'_{train} \rightarrow \tilde{w}^*$

↑ final solution.

missing ÷ Standard imputation.

multiclass : one Vs Rest ← typically

{ maxent model ← Extension to LR
softmax Classifier ← deeplearni
multinomial LR.

Similarity matrix : Extension to LR → Kernal LR

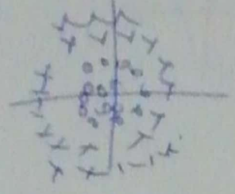
Best & worst cases

→ almost lr separable
→ low-latency requirement (L1 reg)
→ very fast to train.

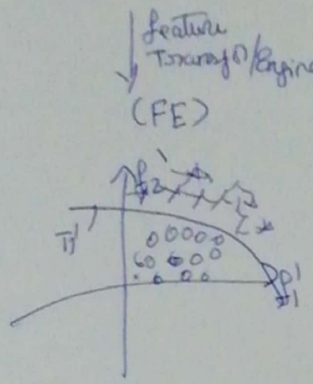
Large dimensionality

→ d is large, chance data is linearly separable is high
↓
low latency → L1 regularization.

24.15 Non-linearly separable data & feature Engineering



(Q) Can we use LR to separate the classes



$$\{ f_1' = f_1^2 \,;\; f_2' = f_2^2 \} \rightarrow FE$$

$x_q = \langle x_{q1}, x_{q2} \rangle$

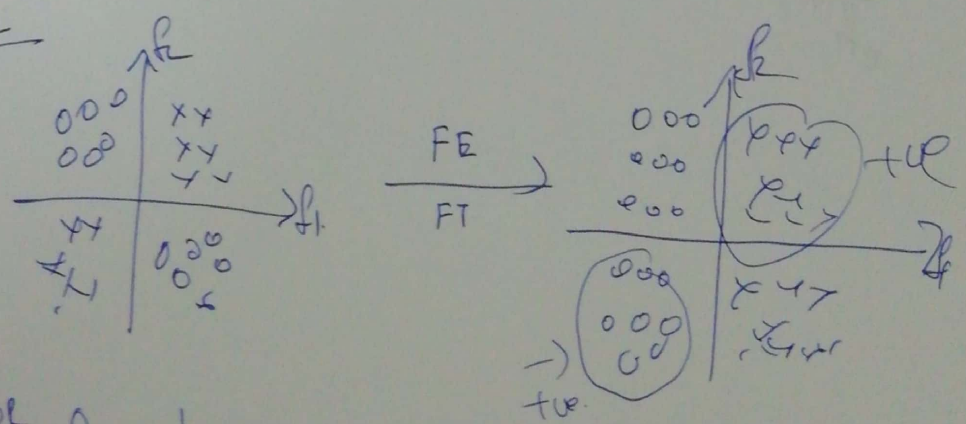
$\downarrow$ FT or FE

$x_q' = \langle x_{q1}', x_{q2}' \rangle \rightarrow \boxed{LR} \rightarrow y_q$

(Q) how to know which transform to apply

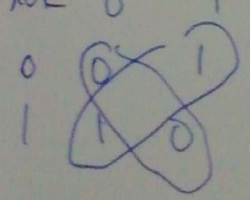
↳ By experience.

Case 2

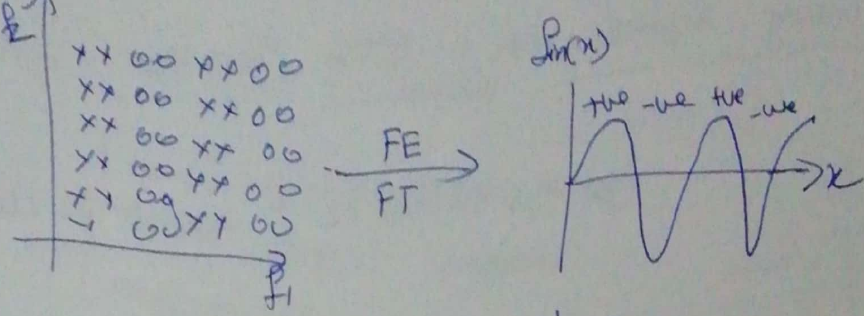


$f_1' = f_1 * f_2$

$f_2' = f_2$

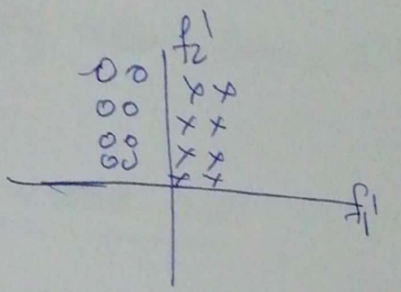


(3)



$f_1' = \sin(f_1)$

$f_2' = f_2$



Typical transform for real value features:

(1) $f_1 * f_2$, $f_1^2, f_2^2, f_2^3, f_1^3$ } polynomial features

(2) Trignomatric feature

$\sin(f_1)$; $\cos(f_1)$

$\sin(f_1) * \cos(f_2)$

$\sin(f_1^2)$

(3) boolean featur : OR, AND, XOR

(4) Other

$\log(f_1)$

e^{f_1}