if the π passes through origin:- $b = 0$

$$w^T x = 0$$

$$\pi: \underline{w}^T \underline{x} + b = 0$$

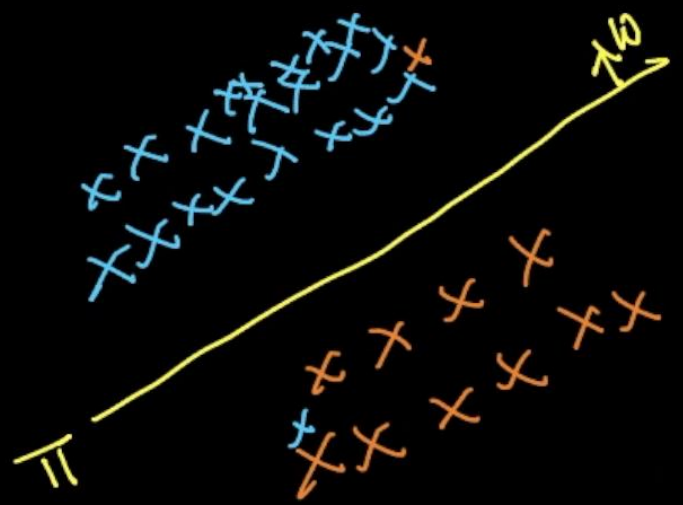
$x \in \mathbb{R}^d$; $w \in \mathbb{R}^d$

x ← vec, w → vec

$b \in \mathbb{R}^1$ ← scalar

Assumption of LR

classes are almost/perfectly linearly seperable } (X)



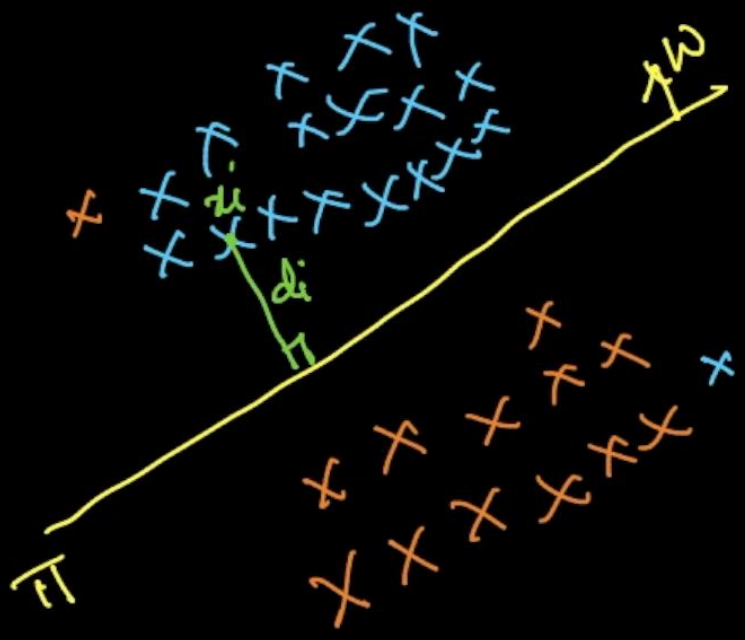
$\pi : w^T x + b$

given: $D_n = \{+ve, -ve\}$

Task: find: (w & b)

{ NB: Condition indep of features
K-NN: Neighborhood

Task: π that best seperates +ve pts from -ve pts



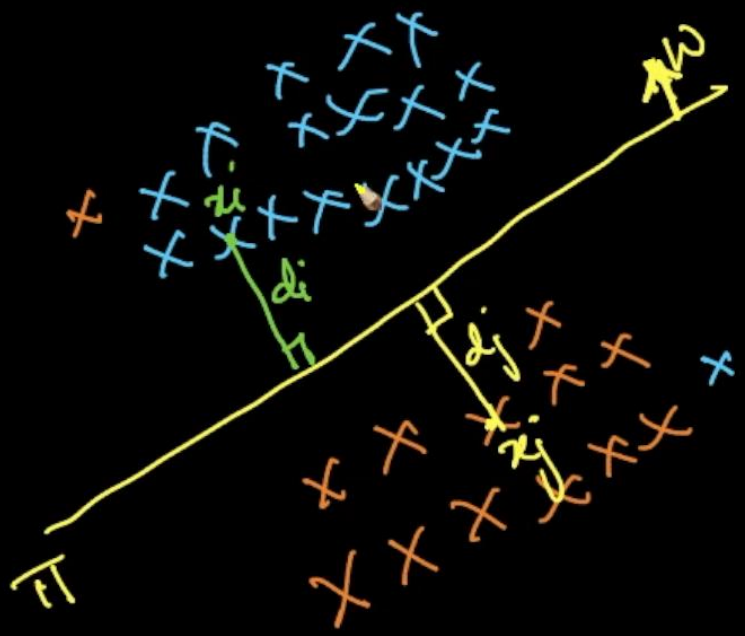
$y_i = +1$: +ve pts
-1 : -ve pts ✓

1 : +ve pts
0 : -ve pts ✗

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

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

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



$y_i = +1$: +ve pts
-1 : -ve pts ✓

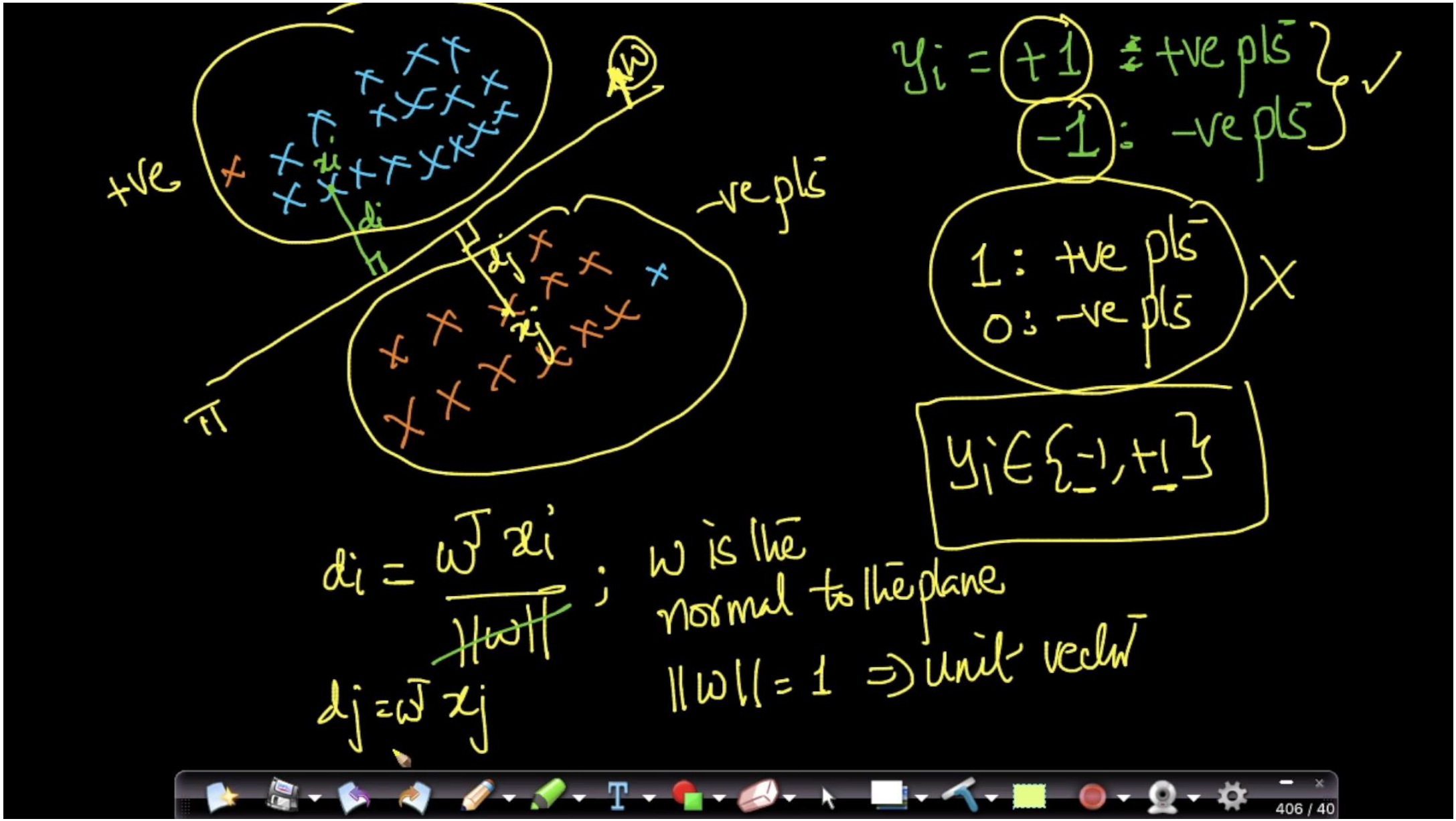
1 : +ve pts
0 : -ve pts ✗

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

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

$d_j = w^T x_j$

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


+ve
-ve pts
w
π
d_i
d_j
x_i
$y_i = +1$: +ve pts
-1 : -ve pts
1 : +ve pts
0 : -ve pts
X
$y_i \in \{-1, +1\}$
$d_i = \frac{w^T x_i}{\|w\|}$; w is the normal to the plane
$d_j = w^T x_j$
$\|w\| = 1 \Rightarrow$ unit vector
406 / 40

$d_i = w^T x_i > 0$

$d_j = w^T x_j < 0$

classifier

$$\begin{cases} \text{if } w^T x_i > 0 \text{ then } y_i = +1 \\ \text{if } w^T x_i < 0 \text{ then } y_i = -1 \end{cases}$$

decision surface in LR : line / plane

Case 1: +ve pt

$y_i * w^T x_i > 0$

$y_i = +1$

$w^T x_i > 0$

w is correctly classifying the pt

$\rightarrow$ if $y_i = +1 \rightarrow$ +ve pt

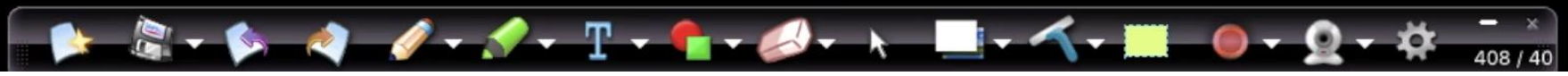
$\Rightarrow$ classifier is saying it's +ve pt

Case 2: -ve pt

$y_i = -1$: -ve pt's

$w^T x_i < 0 \Rightarrow$ LR concluding that x_i is a -ve pt

$$y_i w^T x_i > 0$$



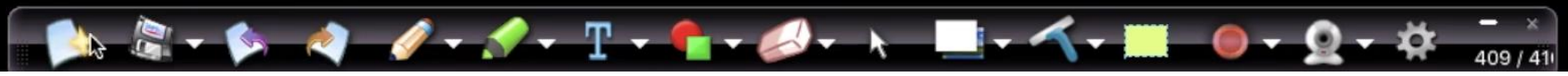
both +ve & -ve pts

$y_i \underline{w^T x_i} > 0$ $\Rightarrow$ the LR model is Correctly classifying the pt x_i

Case 3: $y_i = \underline{+1}$ (+ve pt)

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

$y_i w^T x_i < 0$ $\Rightarrow$ $y_i = +1$, LR : -1 — misclassified



Case 4: $y_i = -1 \Rightarrow$ -ve class

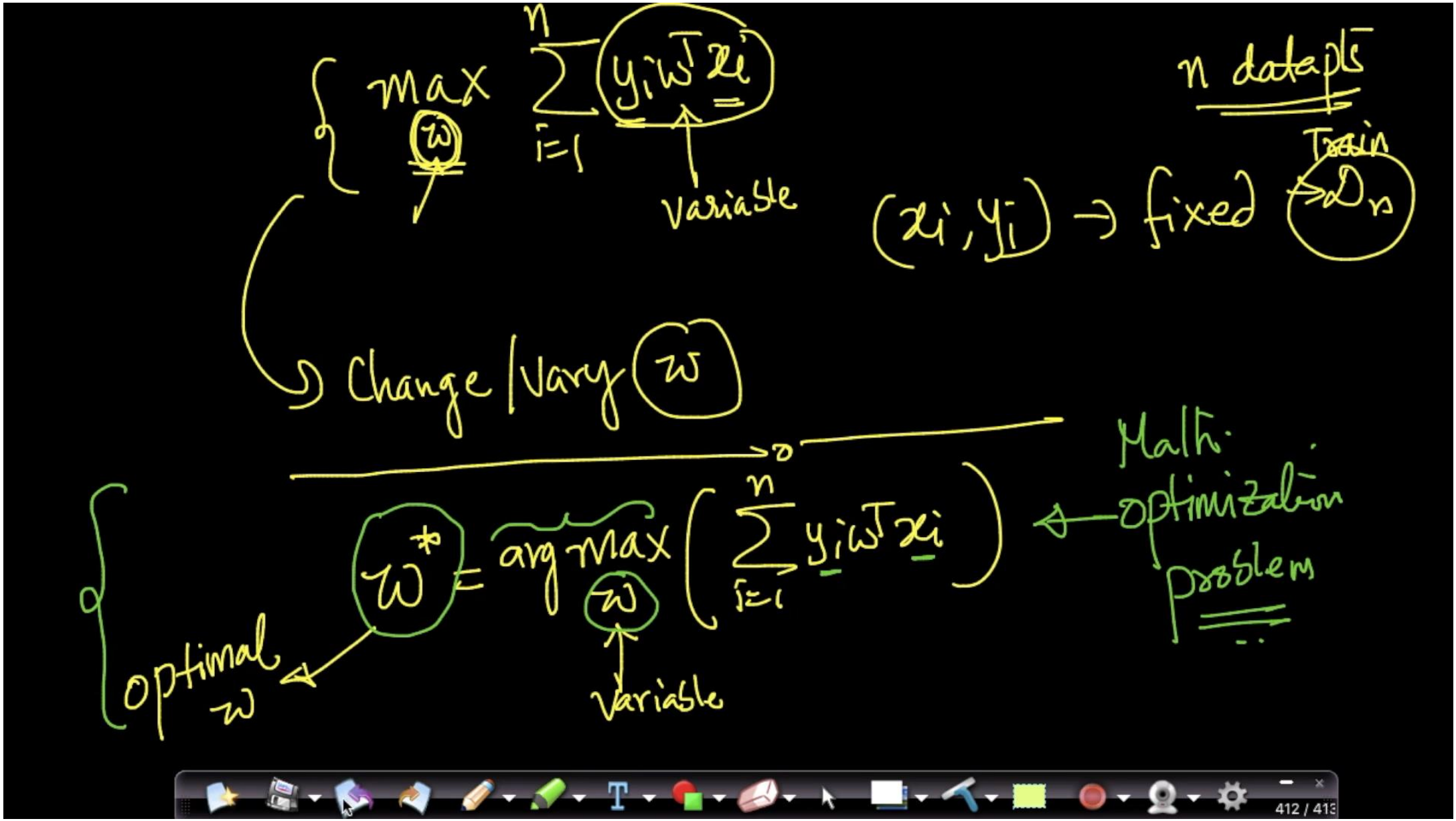
$w^T x_i > 0 \Rightarrow$ LR is saying x_i is +ve pt

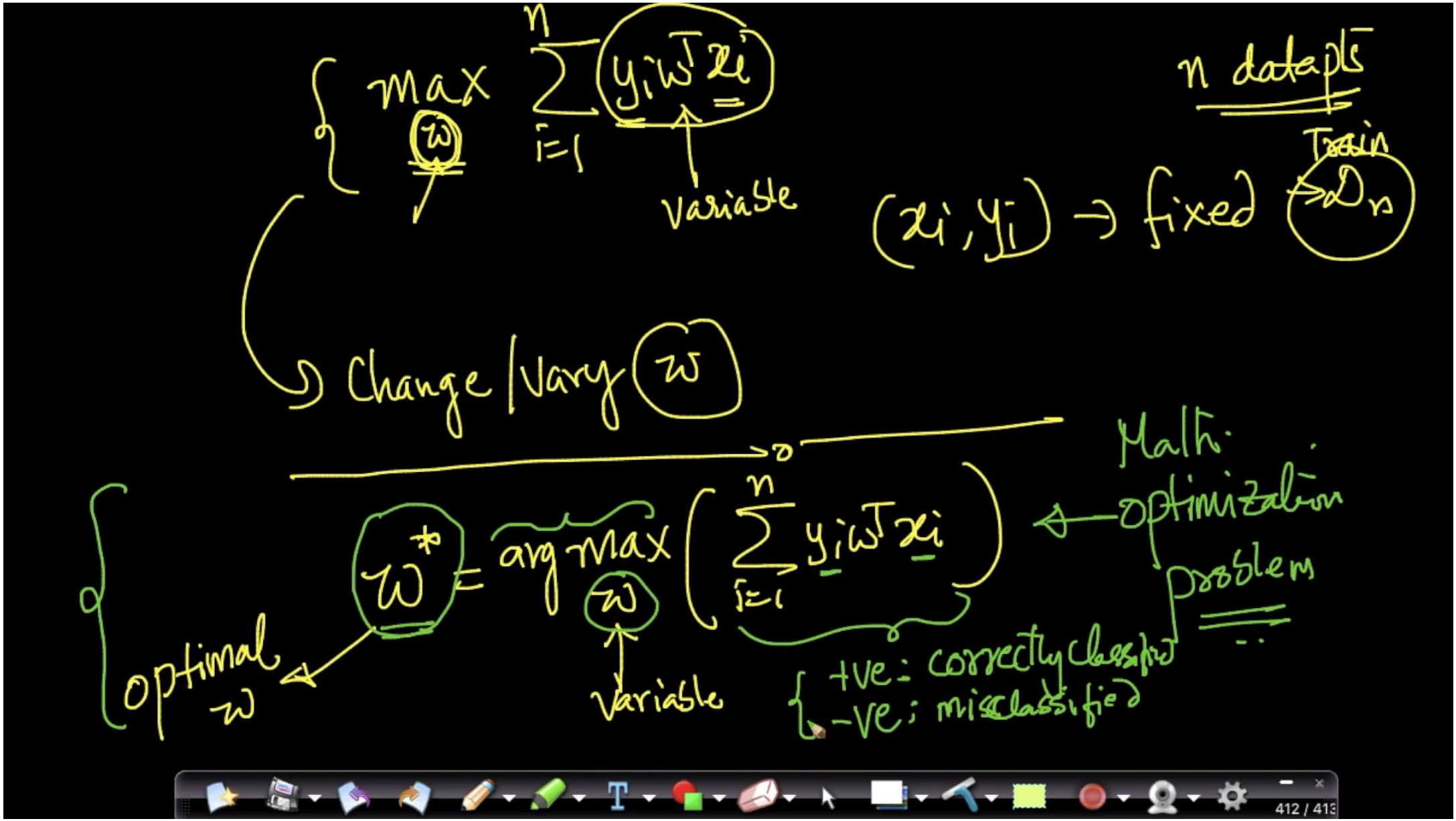
↳ Misclassified

$y_i w^T x_i < 0$

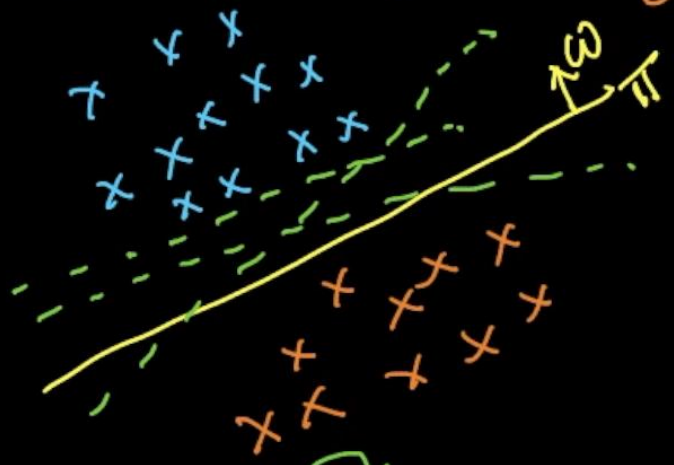
$$\max \sum_{i=1}^{n} y_i w^T x_i$$

n datapts


max
w
$\sum_{i=1}^{n} y_i w^T x_i$
variable
n datapts
Train
(x_i, y_i) ⇒ fixed
D_n
Change/Vary w
$w^* = \arg\max_w \left(\sum_{i=1}^{n} y_i w^T x_i \right)$
Math optimization problem
optimal w
variable
412 / 413


max
w
$\sum_{i=1}^{n} y_i w^T x_i$
variable
n datapts
Train
(x_i, y_i) ⇒ fixed
D_n
Change/Vary w
$w^* = \arg\max_w \left(\sum_{i=1}^{n} y_i w^T x_i \right)$
Math: optimization problems
optimal w
variable
+ve: correctly classified
-ve: misclassified
412 / 413

Squashing & sigmoid function:



optimization problem

$$w^* = \arg\max_{w} \sum_{i=1}^{n} y_i w^T x_i$$

optimal w

π_i (w_i) : best/optimal $w \rightarrow w^*$

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

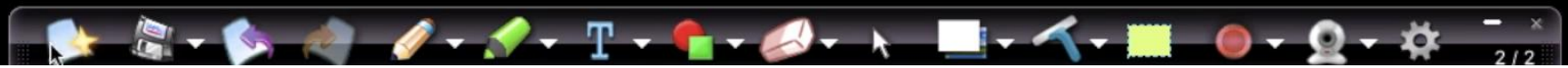
$y_i w^T x_i$ → signed distance

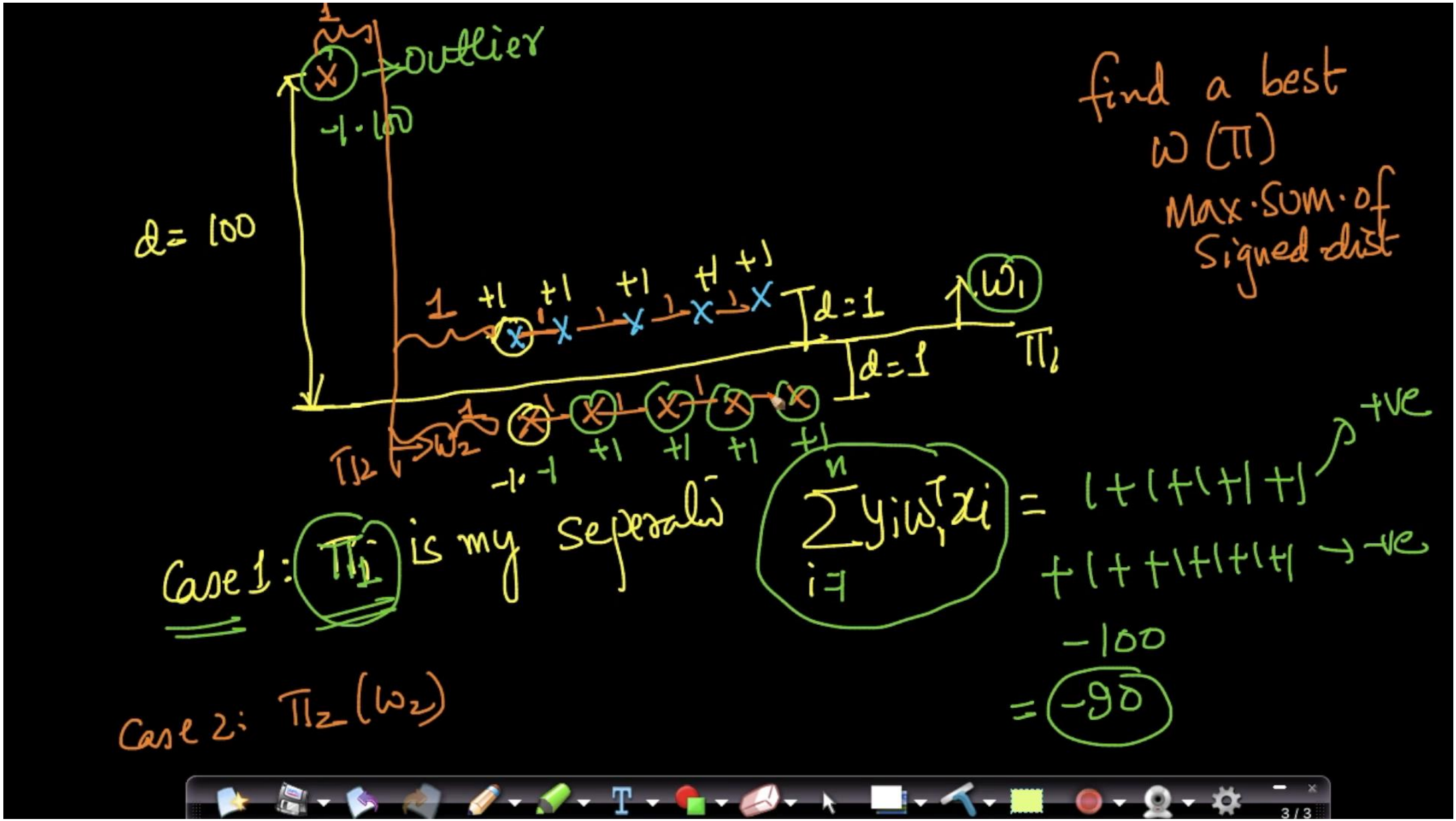
$y_i \in +1 \text{ or } -1$

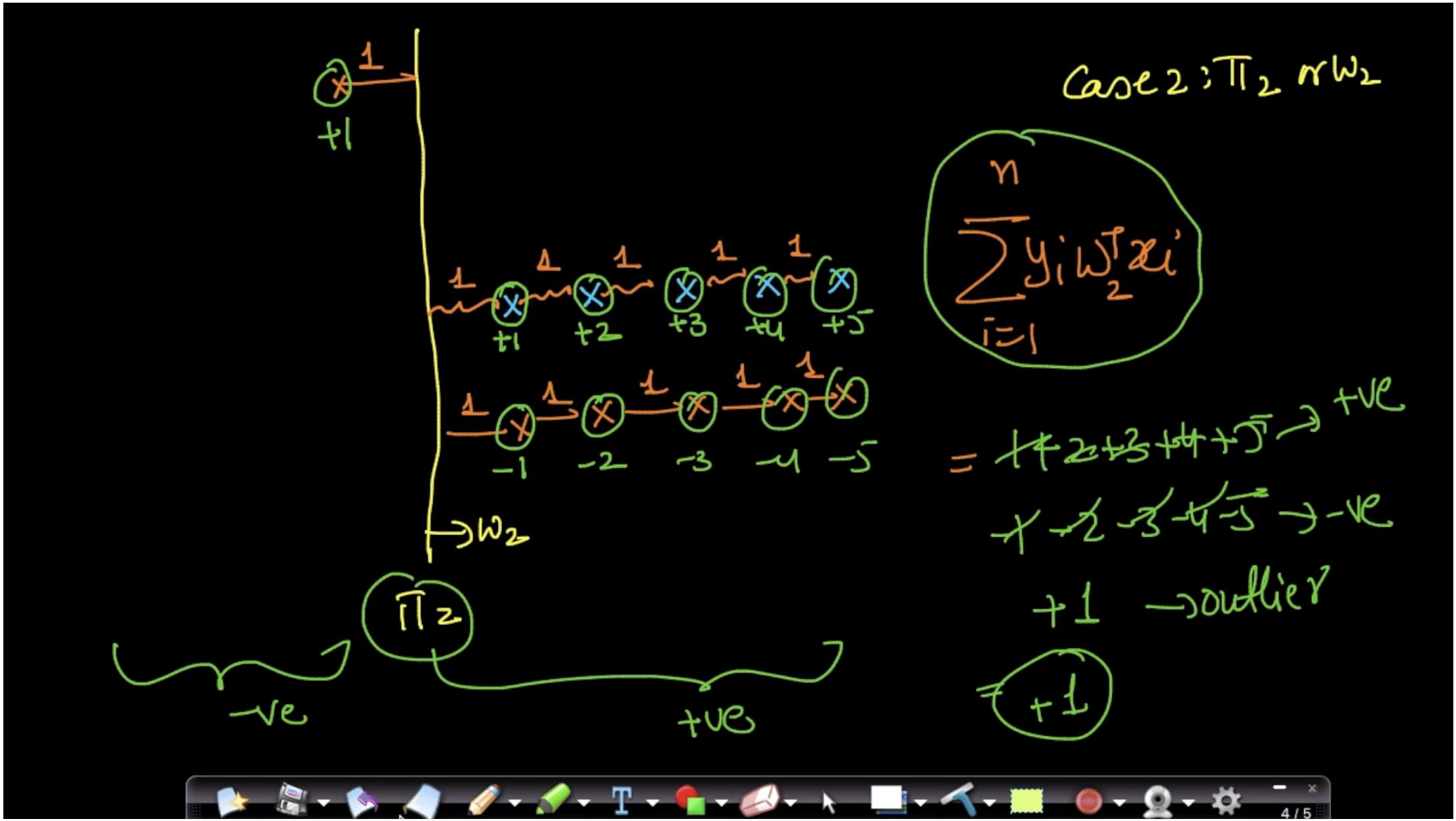
$w^T x_i$: dist from x_i to Π (w is a unit vector)

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

-ve ⇒ incorrectly classifies x_i




outlier
-1.100
d= 100
find a best
w (π)
Max. Sum. of
Signed dist
+1 +1 +1 +1 +1
d=1
d=1
w_1
π_1
π_2
w_2
-1 -1 +1 +1 +1 +1
Case 1: π_1 is my seperator
$\sum_{i=1}^{n} y_i w_1^T x_i$ = 1+1+1+1+1 → +ve
+1+1+1+1+1+1 → +ve
-100
= -90
Case 2: π_2 (w_2)

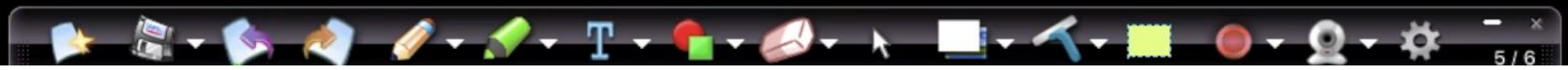

Case 2 : Π_2 or w_2
$\sum_{i=1}^{n} y_i w_2^T x_i$
+1
1
+1 +2 +3 +4 +5
-1 -2 -3 -4 -5
w_2
Π_2
-ve
+ve
= +1+2+3+4+5 → +ve
+1 -2 -3 -4 -5 → -ve
+1 → outlier
+1

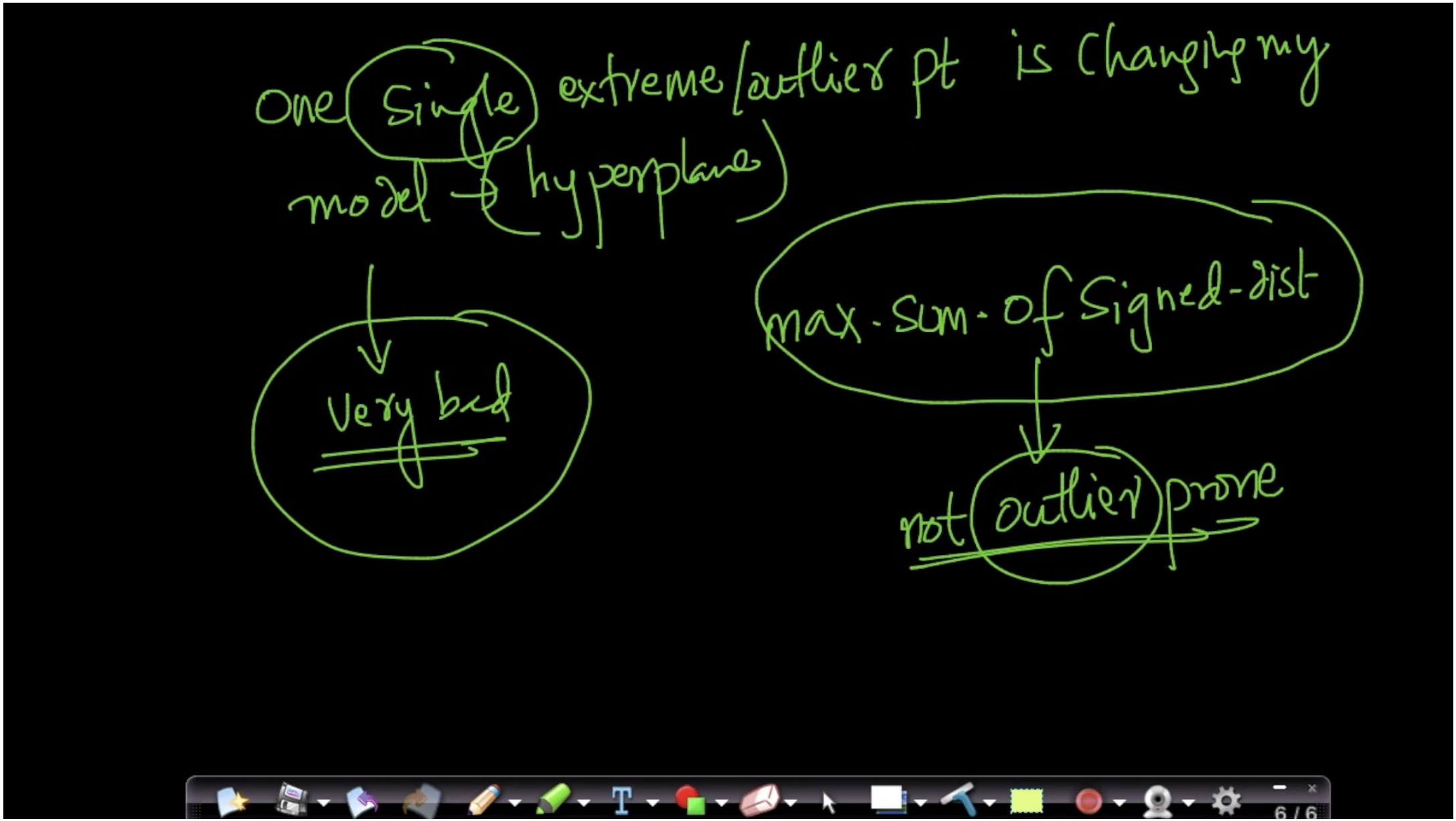
obj :- find w that Maximizes sum of Signed dist

+1 ≥ -90

✓ Π_2 as my classifier

intuitively :- Π_1 is better than Π_2



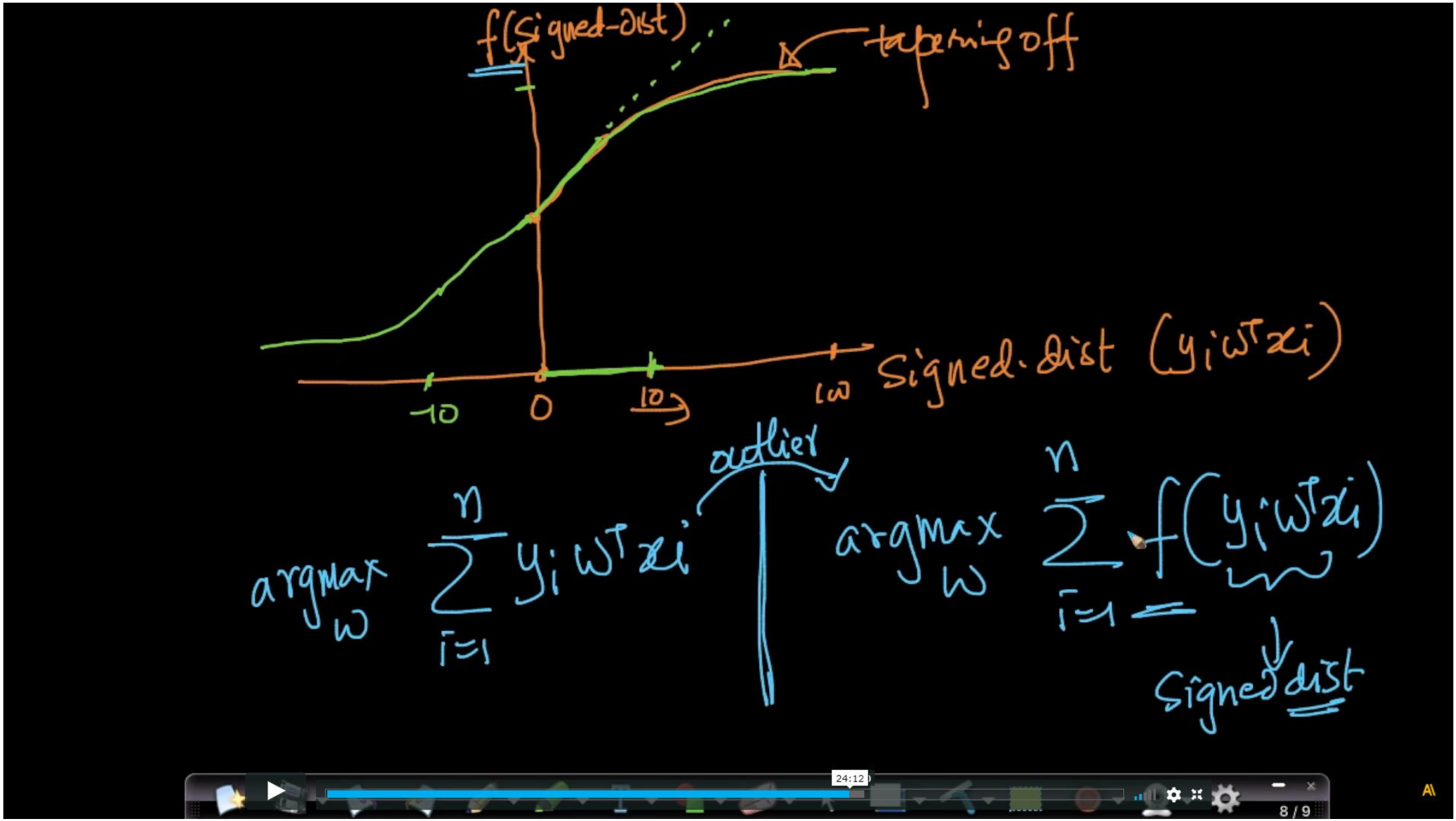

one Single extreme/outlier pt is Changing my model → (hyperplane)
Very bad
max-sum-of Signed-dist
not outlier prone
6 / 6

Squashing:

Idea:- instead of using signed-dist

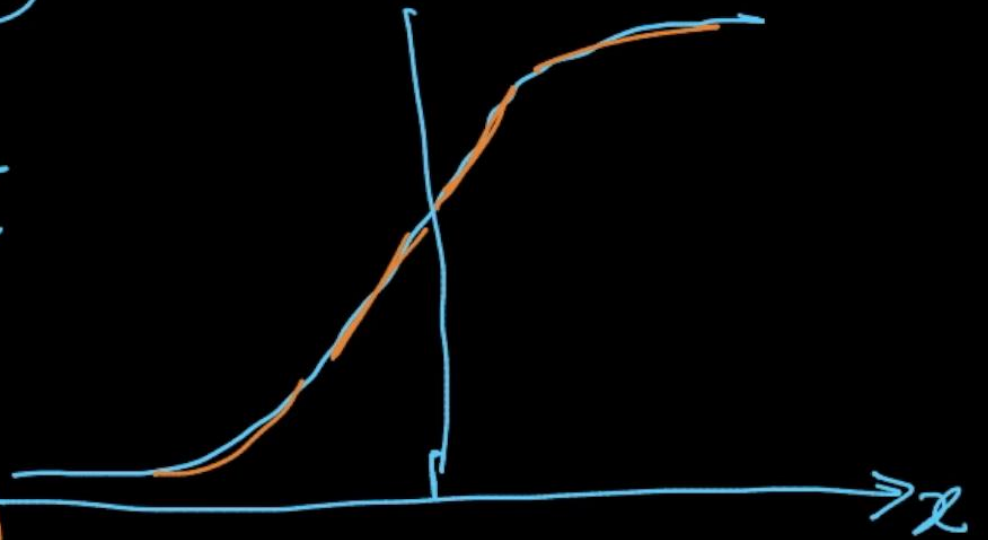
if signed-dist is small :- use it as is

" " is large : make it a smaller value


f(signed-dist)
tapering off
-10
0
10
ω
signed. dist $(y_i w^T x_i)$
outlier
$\underset{w}{\text{argmax}} \sum_{i=1}^{n} y_i w^T x_i$
$\underset{w}{\text{argmax}} \sum_{i=1}^{n} f(y_i w^T x_i)$
signed dist
24:12
8 / 9

Sigmoid fn:- $\sigma(x)$

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



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

probabilistic interpretation

$$p(y_i = 1)$$

Contents - Google Docs | NBayesLogReg.pdf | plot(1/(1+e^-x)) - Google Search | Chekuri Srikan...

Secure | https://www.google.co.in/search?ei=p7QoWtf6PMTqvASm5JiwCg&q=plot%281%2F%281%2Be%5E-x%29%29&oq=plot%281%2F%28...

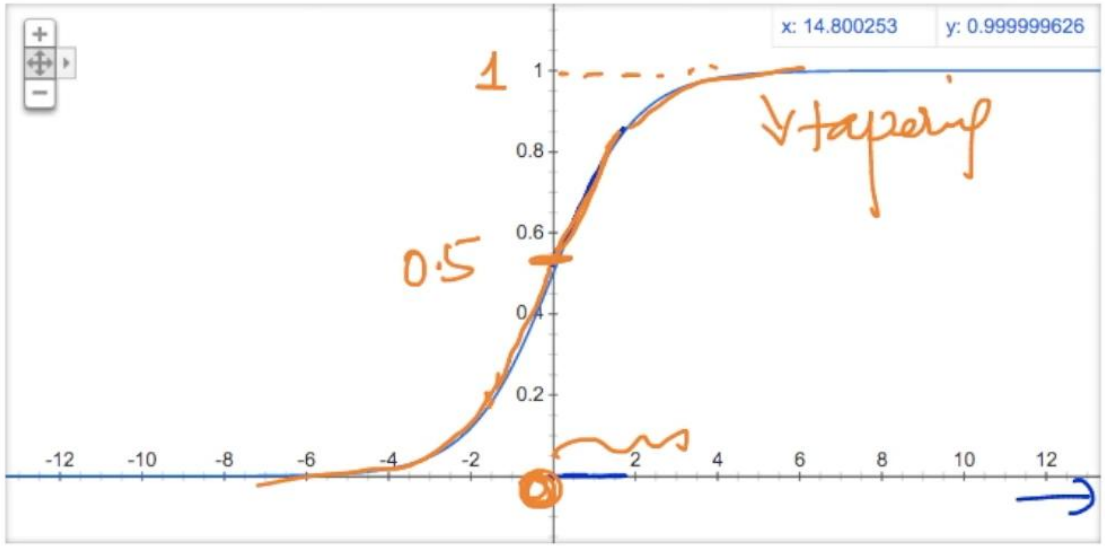
Google

plot(1/(1+e^-x))

All | Maps | Images | Videos | News | More | Settings | Tools

About 16,10,00,000 results (0.35 seconds)

Graph for 1/(1+e^-x)



More info

plot x e^-x, x^2 e^-x, x=0 to 8 - Wolfram|Alpha

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

✓ Max : 1

✓ min : 0

$\sigma(0) = 0.5$

x = Signed dist

1/1

Contents - Google Docs | NBayesLogReg.pdf | plot(1/(1+e^-x)) - Google Search | Chekuri Srikan...

Secure | https://www.google.co.in/search?ei=p7QoWtf6PMTqvASm5JiwCg&q=plot%281%2F%281%2Be%5E-x%29%29&oq=plot%281%2F%28...

Google

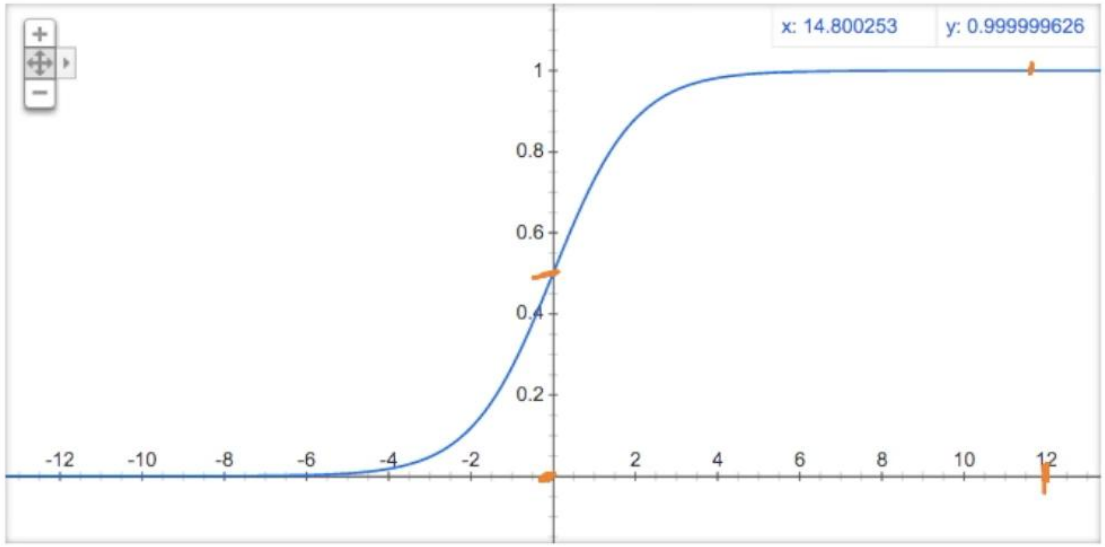
plot(1/(1+e^-x))

All | Maps | Images | Videos | News | More | Settings | Tools

About 16,10,00,000 results (0.35 seconds)

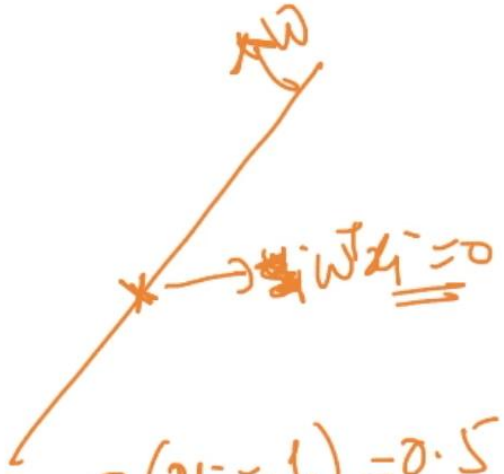
$\sigma(0) = 0.5$

Graph for 1/(1+e^-x)



More info

plot x e^-x, x^2 e^-x, x=0 to 8 - Wolfram|Alpha



$p(y_i = 1) = 0.5$

$w^T x_i$ is v. large

$p(y_i = 1) = 1.0$

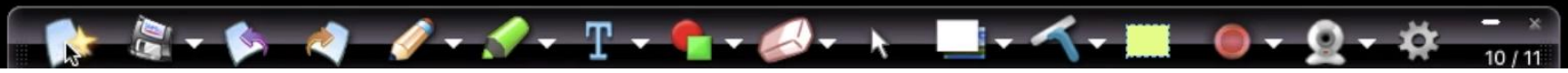
2 / 2

max. sum. of signed dist ⟶ outlier

↓

$\sigma(\alpha x)$ → sigmoid
↳ tapering, linear
↳ prob.

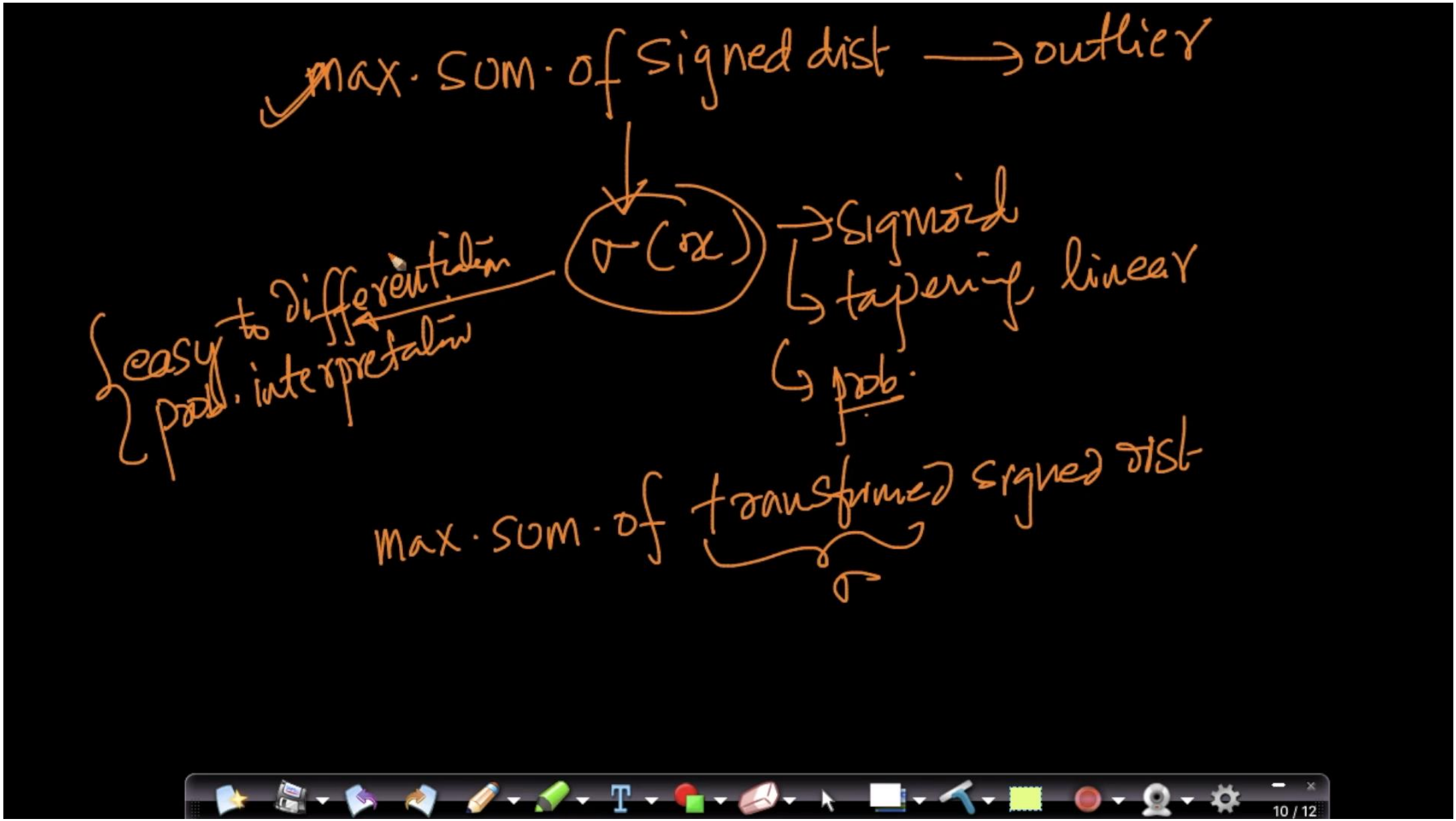
max. sum. of transformed signed dist

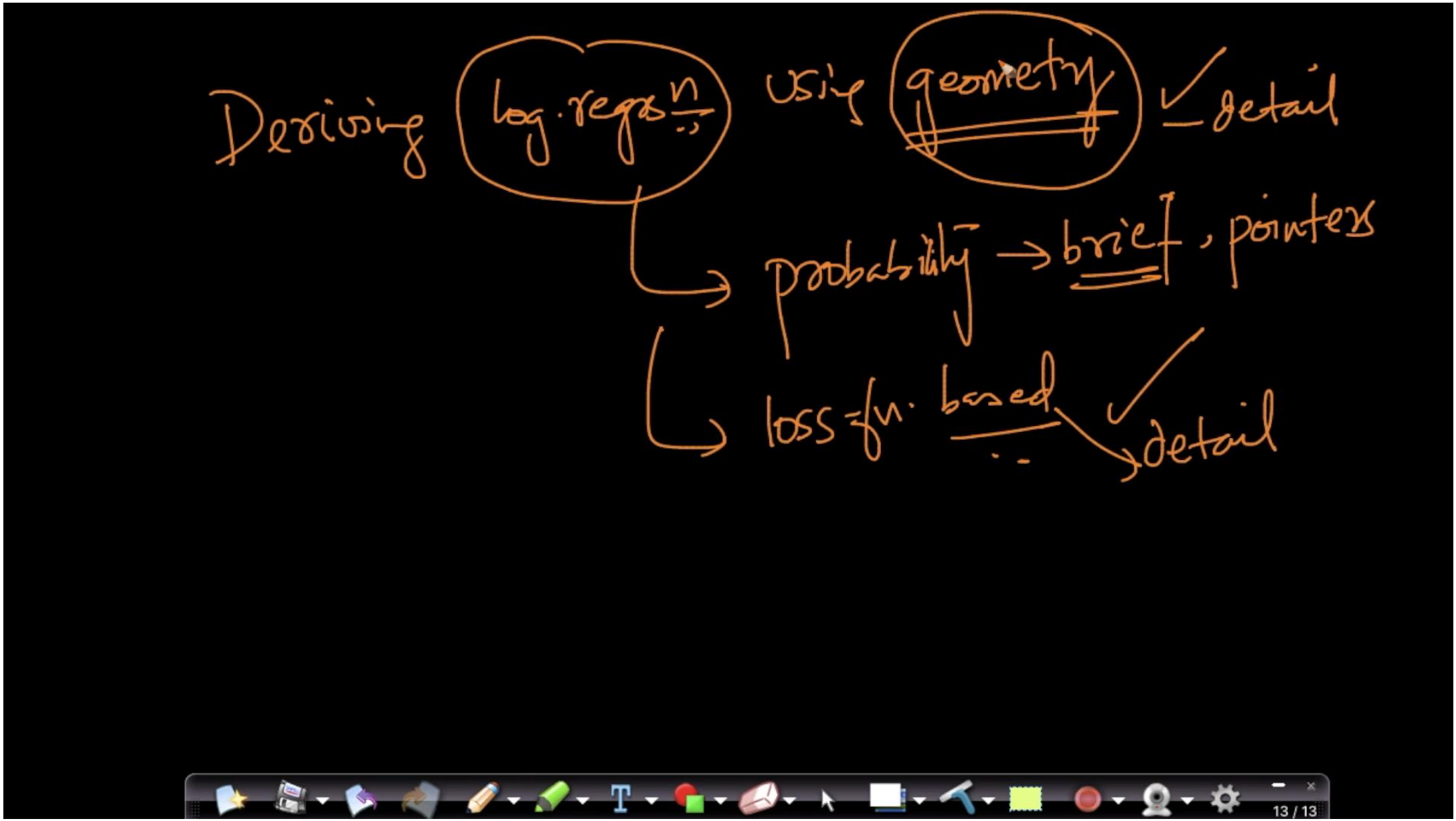


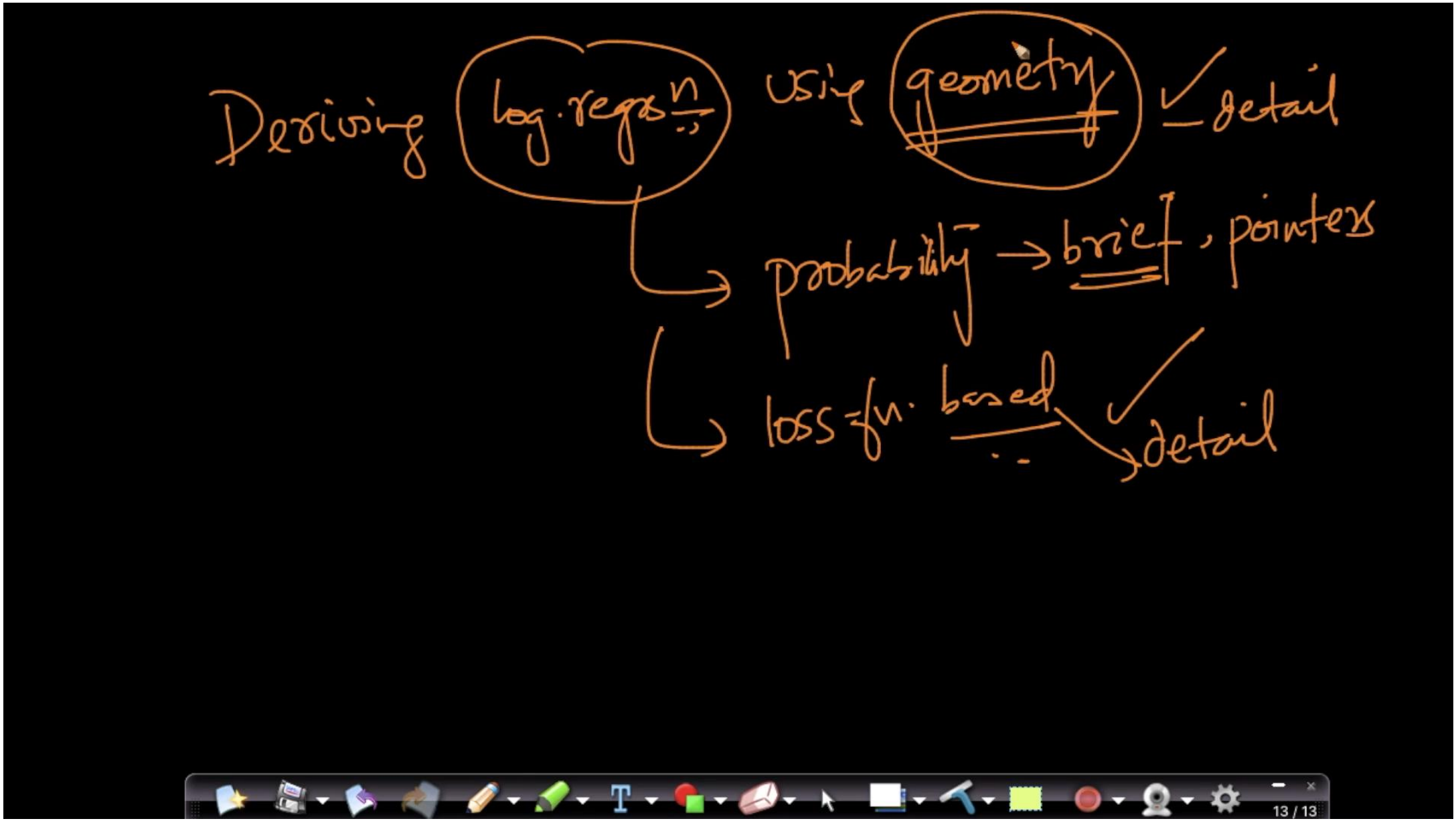
$$w^* = \arg\max_w \sum_{i=1}^{n} \sigma(y_i w^T x_i) \qquad \sigma(x) = \frac{1}{1+e^{-x}}$$

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

less impacted by outliers


max. sum. of signed dist → outlier
σ(αx)
→ Sigmoid
↳ tapering linear
↳ prob.
easy to differentiate
poss. interpretation
max. sum. of transformed signed dist


Deriving
log. regression
using
geometry
detail
probability → brief, pointers
loss fn. based
detail


Deriving
log. regression
using
geometry
detail
probability → brief, pointers
loss fn. based
detail
13 / 13

Monotonic fn:- $g(x)$

$x \uparrow$; $g(x) \uparrow$ $\longrightarrow$ monotonically incr-fn

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

then $g(x)$ is said to be mon. incr. fn

Contents - Google Docs | NBayesLogReg.pdf | plot(x^2) - Google Search | plot(log(x)) - Google Search | plot(log(x^2)) - Google Search | Chekuri Srikan...

Secure | https://www.google.co.in/search?q=plot(log(x))&oq=plot(log(x)&aqs=chrome.0.69i59j69i57j0l4.3944j0j7&sourceid=chrome&ie=UTF...

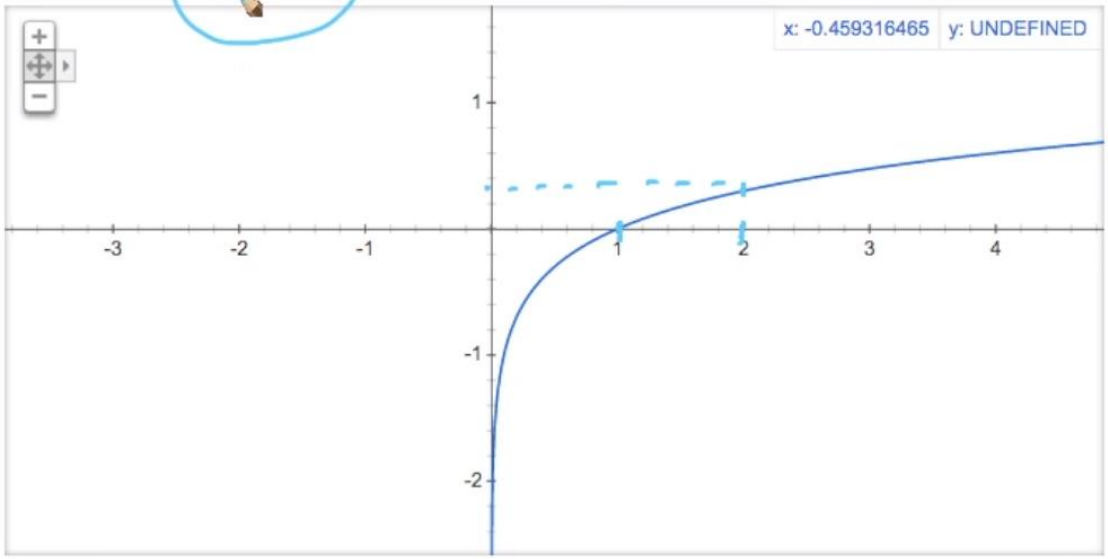
Google

plot(log(x))

All | Images | Maps | Videos | News | More | Settings | Tools

About 44,10,00,000 results (0.44 seconds)

Graph for log(x)



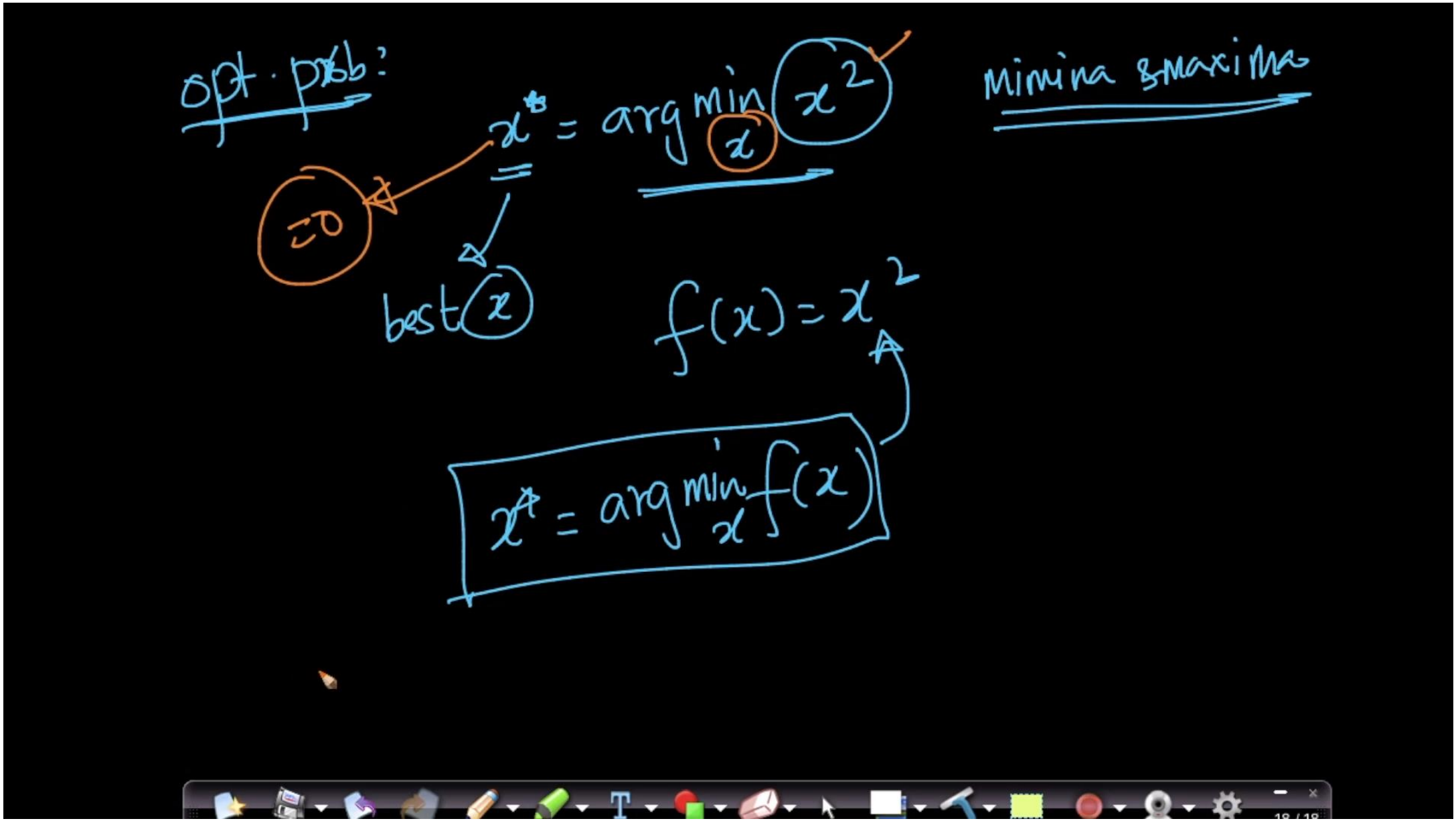
More info

log(x) ; x > 0

x↑ ; log(x)↑

x↓ ; log(x)↓

Semilogarithmic plot - MATLAB semilogx - MathWorks


opt. prbb:
$x^* = \arg\min_x x^2$
x^*
$= 0$
best x
Minima & Maxima
$f(x) = x^2$
$x^* = \arg\min_x f(x)$

mon. incr

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

optimization

1 → 0 = $x^* = \underset{x}{\operatorname{argmin}}\, f(x)$; $f(x) = x^2$

2 → $x' = \underset{x}{\operatorname{argmin}}\, g(f(x))$

$\log(x^2)$

claim: $x^* = x'$ ∵ $g(x)$ is a monotonic fn

if $g(x)$ is a monotonic fn:

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

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

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

$$\underset{x}{\operatorname{argmax}} f(x) = \underset{x}{\operatorname{argmax}} g(f(x))$$

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

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

$$\checkmark\ w^* = \arg\max_w \sum_{i=1}^{n} -\log\left(1+\exp(-y_i w^T x_i)\right)$$

$$\arg\max_x f(x) = \arg\min_x -f(x)$$

$$\log(e^x) = x$$

$$\arg\max_x -f(x) = \arg\min_x f(x)$$

$+1$ or -1

opt

LR

$$w^* = \arg\min_w \sum_{i=1}^{n} \log\left(1 + \exp(-y_i w^T x_i)\right)$$

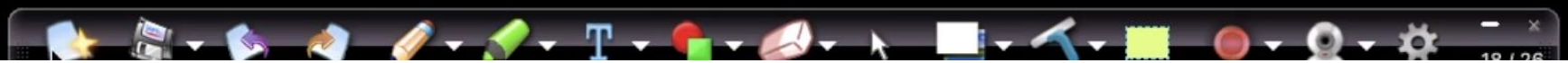
signed-dist

$$\checkmark \quad \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} \log\left(1 + \exp(-y_i w^T x_i)\right) \longleftarrow \text{opt. LR}$$

$$\underset{w}{\operatorname{argmin}} \sum -y_i w^T x_i$$

$$\underset{w}{\operatorname{argmax}} \sum y_i w^T x_i \longleftarrow \text{Sum of signed dist}$$

$\rightarrow$ huge outlier problem



geometry

$$w^* = \arg\min_w \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i w^T x_i\right)\right)$$

y_i: $+1$ or -1

Prob method

$$w^* = \arg\min_w \sum_{i=1}^{n} -y_i \log p_i - (1-y_i)\log(1-p_i)$$

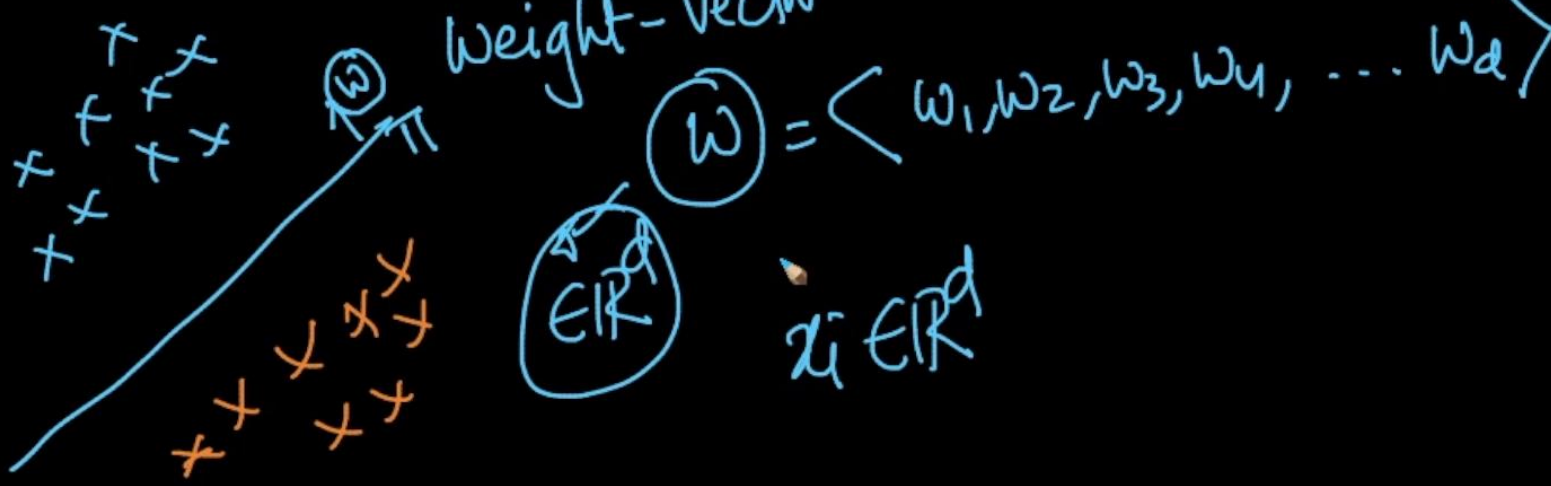
y_i: $+1$ or 0

$$p_i = \sigma(w^T x_i)$$

loss fn

Weight-Vector

$$w^* = \arg\min_w \sum_{i=1}^{n} \log\left(1+\exp\left(-y_i(w^T x_i)\right)\right)$$



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

$f_1, f_2, f_3, \ldots f_d$

decision: $x_q \rightarrow y_q$ $\begin{cases} \text{if } w^T x_q \geq 0 & \text{then } y_q = +1 \\ \text{if } w^T x_q < 0 & \text{then } y_q = -1 \end{cases}$

Prob. interp: $\sigma(w^T x_q) = P(y_q = +1)$

0 to 1

interpretation of w:

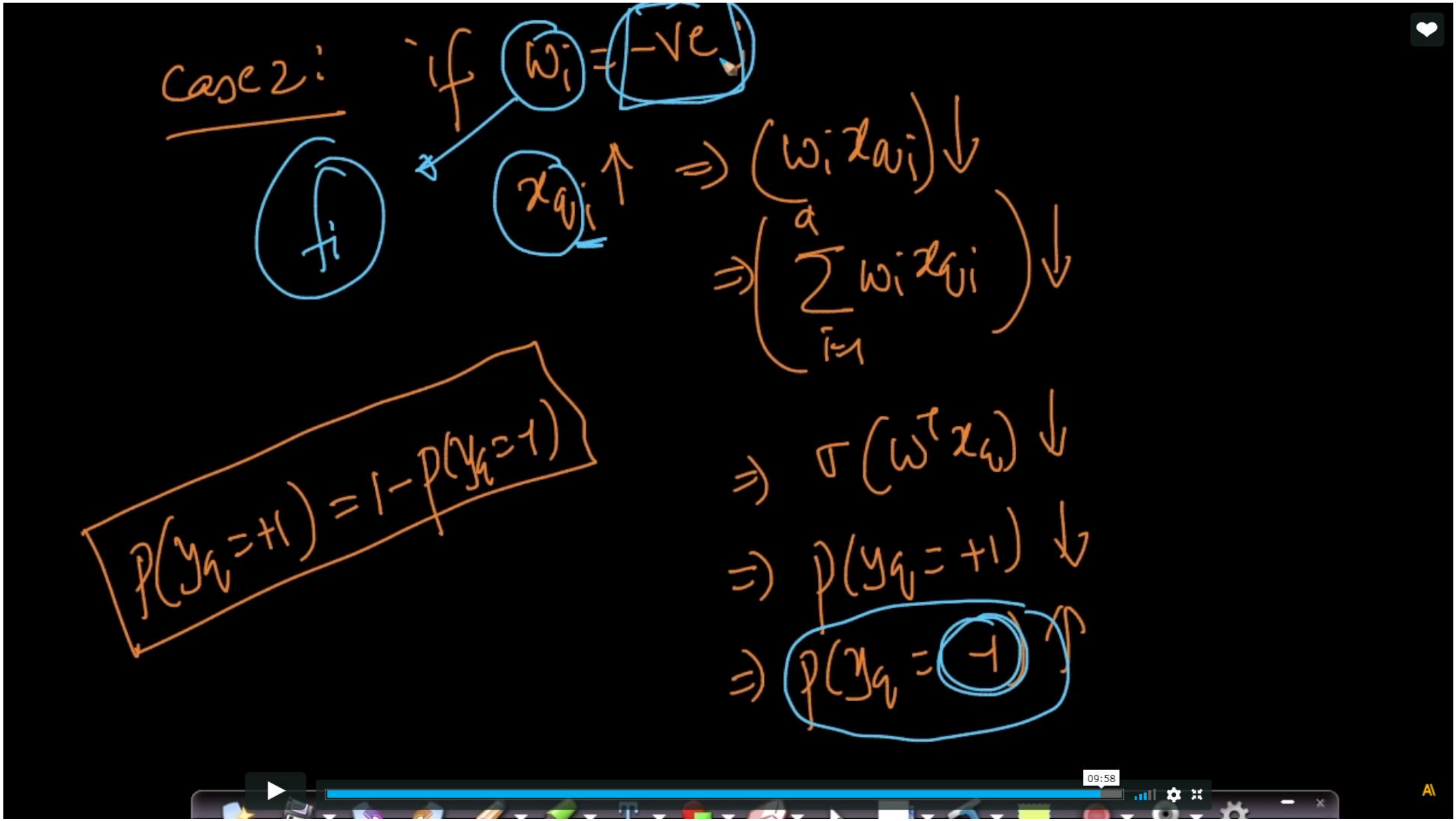
Case (1) if w_i = +ve, f_i

$x_{qi} \uparrow \Rightarrow (w_i x_{qi}) \uparrow$

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

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

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


Case 2:
if w_i = -ve
fi
x_{qi} ↑ ⇒ $(w_i x_{qi})$ ↓
⇒ $\left(\sum_{i=1}^{a} w_i x_{qi}\right)$ ↓
$P(y_q=+1) = 1 - P(y_q=-1)$
⇒ $\sigma(w^T x_q)$ ↓
⇒ $P(y_q=+1)$ ↓
⇒ $P(y_q=-1)$ ↑
09:58

Sathvikchiramana — 6 Votes

How can we say that if sigmoid(W^T*Xq)increases then p(y=+1) increases....because y*W^T*Xq is high for all the correctly classified points so is the sigmoid(y*W^T*Xq) .....so for all correctly classified points sigmoid(y*W^T*Xq) is high i.e for both +ve and -ve points..how can we say that
if sigmoid(y*W^T*Xq) is high then only p(y=+1) is high but not p(y=-1)?

Reply | Jun 02, 2019 18:37 PM

Applied AI Tech Admin

It is because in the problem formulation we have assumed that If the value of product **y_i*w_T*x_i** is positive, then the point is correctly classified. Otherwise it is not correctly classified. But here along with checking whether the data points are classified correctly or not, we also have to ensure our model predicts a label for those data points. Hence we need a function to estimate the labels. For this purpose we are taking the sigmoid function into consideration.The possible range of values for result of sigmoid function are [0,1]. Hence we are using a threshold of 0.5. If sigmoid (y_q*W^T*Xq) >0.5, then we predict the label as 1. Otherwise we predict the label as 0. Also we have chosen sigmoid as it has got nice probabilistic interpretation and also it is differentiable easily.
If sigmoid(y_q*W^T*Xq) = 0.6, then we can say y_q = 1 with a low confidence as 0.6 is nearer to the threshold 0.5. Whereas if sigmoid(y_q*W^T*Xq) = 1, we can say y_q = 1 with high confidence as 1 is far away from the threshold 0.5 and on the positive side.
If sigmoid(y_q*W^T*Xq) = 0.4, then we cansay y_q = 0 with a low confidence as 0.4 is nearer to the threshold 0.5.
Whereas if sigmoid(y_q*W^T*Xq) = 0 we can say y_q = 0 with high confidence as 0 is far away from the threshold 0.5 and on the negative side.
On the basis of these two reasons, we can say that
a) As the value of sigmoid(y_q*W^T*Xq) increases, the value of P(y_q=1|x_q) increases and P(y_q=0|x_q) decreases

Solving Optimization Problems

Interview Questions on Logistic Regression and Linear Regression

Applied AI Tech Admin

It is because in the problem formulation we have assumed that If the value of product **y_i*w_T*x_i** is positive, then the point is correctly classified. Otherwise it is not correctly classified. But here along with checking whether the data points are classified correctly or not, we also have to ensure our model predicts a label for those data points. Hence we need a function to estimate the labels. For this purpose we are taking the sigmoid function into consideration.The possible range of values for result of sigmoid function are [0,1]. Hence we are using a threshold of 0.5. If sigmoid (y_q*W^T*Xq) >0.5, then we predict the label as 1. Otherwise we predict the label as 0. Also we have chosen sigmoid as it has got nice probabilistic interpretation and also it is differentiable easily.

If sigmoid(y_q*W^T*Xq) = 0.6, then we can say y_q = 1 with a low confidence as 0.6 is nearer to the threshold 0.5. Whereas if sigmoid(y_q*W^T*Xq) = 1, we can say y_q = 1 with high confidence as 1 is far away from the threshold 0.5 and on the positive side.

If sigmoid(y_q*W^T*Xq) = 0.4, then we cansay y_q = 0 with a low confidence as 0.4 is nearer to the threshold 0.5.

Whereas if sigmoid(y_q*W^T*Xq) = 0 we can say y_q = 0 with high confidence as 0 is far away from the threshold 0.5 and on the negative side.

On the basis of these two reasons, we can say that

a) As the value of sigmoid(y_q*W^T*Xq) increases, the value of P(y_q=1|x_q) increases and P(y_q=0|x_q) decreases.

b) As the value of sigmoid(y_q*W^T*Xq) decreases, the value of P(y_q=0|x_q) increases and P(y_q=1|x_q) decreases.

Reply

Jun 02, 2019 21:27 PM

Mr. Chauhan

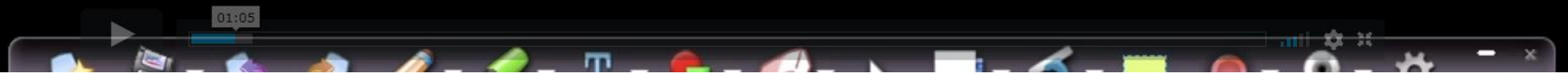
Nice explanation thanx!!

L_2 regularization : Overfitting vs Underfitting:

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

Let $\boxed{z_i = y_i w^T x_i}$

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



$$\sum_{i=1}^{n} \log\left(1+\exp(-z_i)\right)$$

$\exp(-z_i) \geq 0$ (≥ 0)

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

$\log(1) = 0$

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

$\{\log(1+\delta) \geq \log(1)$, $\delta \geq 0\}$

Contents - Google D | NBayesLogReg.pdf | cs229-notes1.dvi | Multicollinearity - Wi | plot(exp(-x)) - Googl | plot(log(1+x)) - Goog | Chekuri Srikan...

Secure | https://www.google.co.in/search?q=plot(exp(-x))&oq=plot(exp(-x))&aqs=chrome..69i57j0l5.4686j0j4&sourceid=chrome&ie=UTF-8

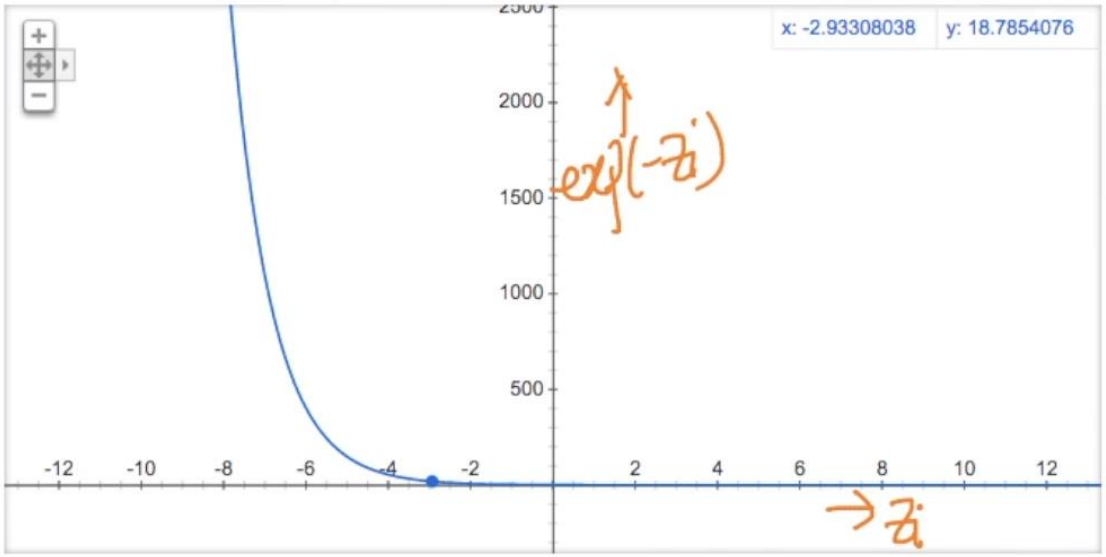
Google

plot(exp(-x))

All Maps Images Videos News More Settings Tools

About 48,80,00,000 results (0.46 seconds)

Graph for exp(-x)



$z \to \infty$

$\exp(-z) \to 0$

More info

Graphing Exponential Functions: More Examples - Purplemath

$$w^* = \arg\min_{w} \left[\sum_{i=1}^{n} \left(\log\left(1 + \exp(-z_i)\right) \right) \right] \geq 0$$

✓ minimal value of $\sum_{i=1}^{n} \log\left(1 + \exp(-z_i)\right)$ is "0"

[it occurs when $z_i \Rightarrow \infty$ for all i

$$w^* = \arg\min_w \sum_{i=1}^{n} \log(1+\exp(-z_i))$$

if z_i = +ve, $z_i \to +\infty$

then $\exp(-z_i) \to 0$

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

$z_i = y_i (w^T x_i)$ → +ve

$D_{train} = \{ \langle x_i, y_i \rangle : i = 1 \text{ to } n \}$

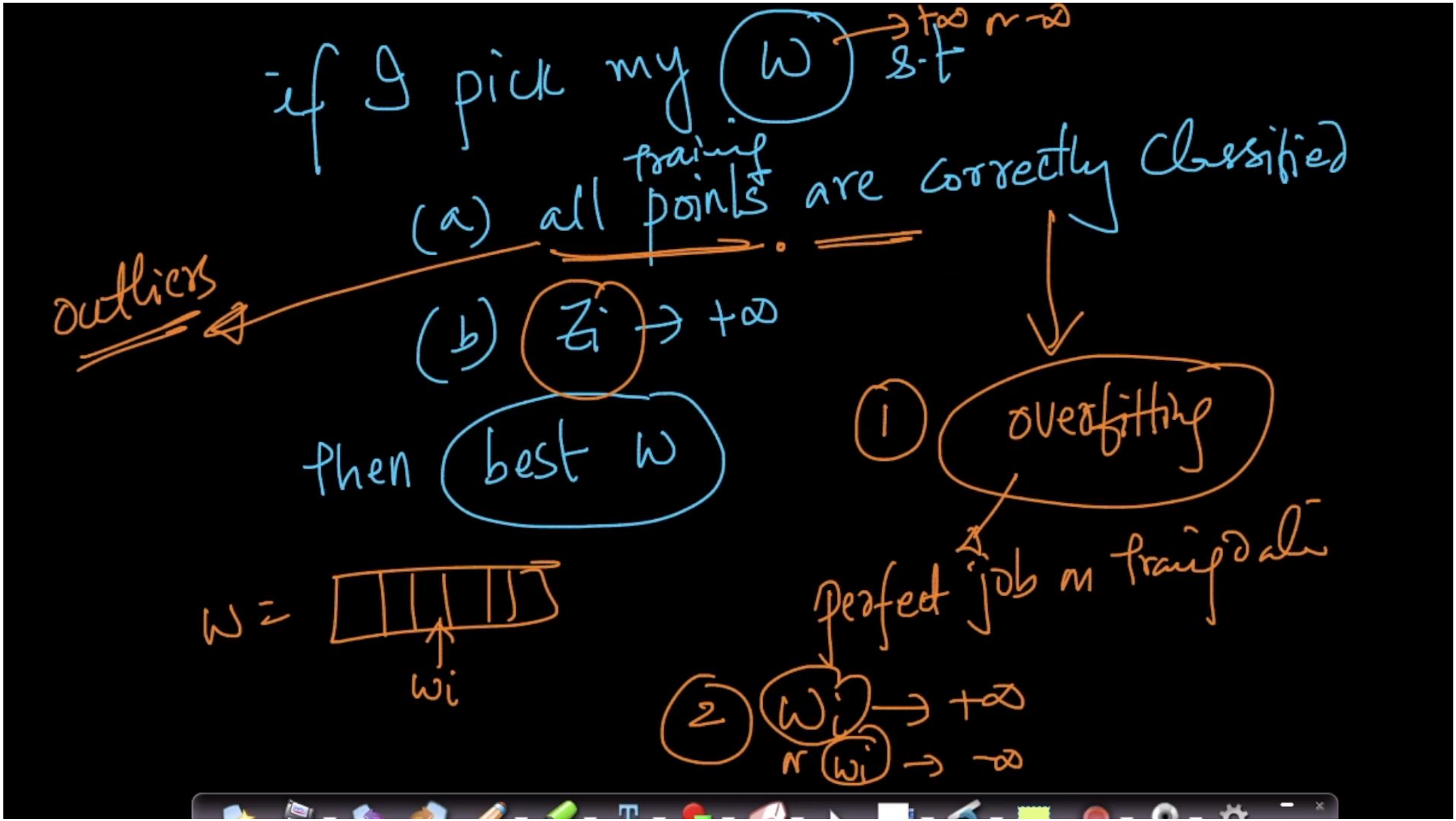
Correctly-classified

$z_i \to +\infty$

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

(1) z_i = +ve; x_i is correctly classified by w

(2) $z_i \to +\infty$;


if I pick my w → +∞ or −∞ s.t
(a) all training points are correctly classified
outliers
(b) $z_i \to +\infty$
then best w
1 overfitting
perfect job on training data
$w =$
w_i
2
$w_i \to +\infty$
or $w_i \to -\infty$

w_i's $\rightarrow \infty$

or

w_i's $\rightarrow -\infty$

minima $\rightarrow 0$

One key aspect

w is a normal

$$w^T w = 1$$

Regularization

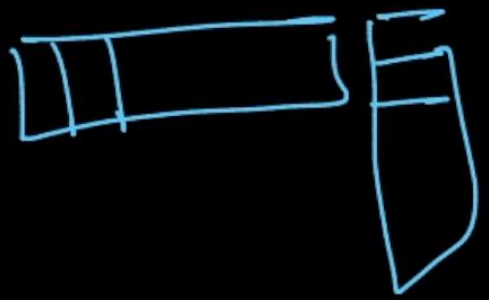
$$w^* = \arg\min_w \left[\sum_{i=1}^{n} \log\left(1+\exp(-y_i w^T x_i)\right) + \lambda\, w^T w \right]$$

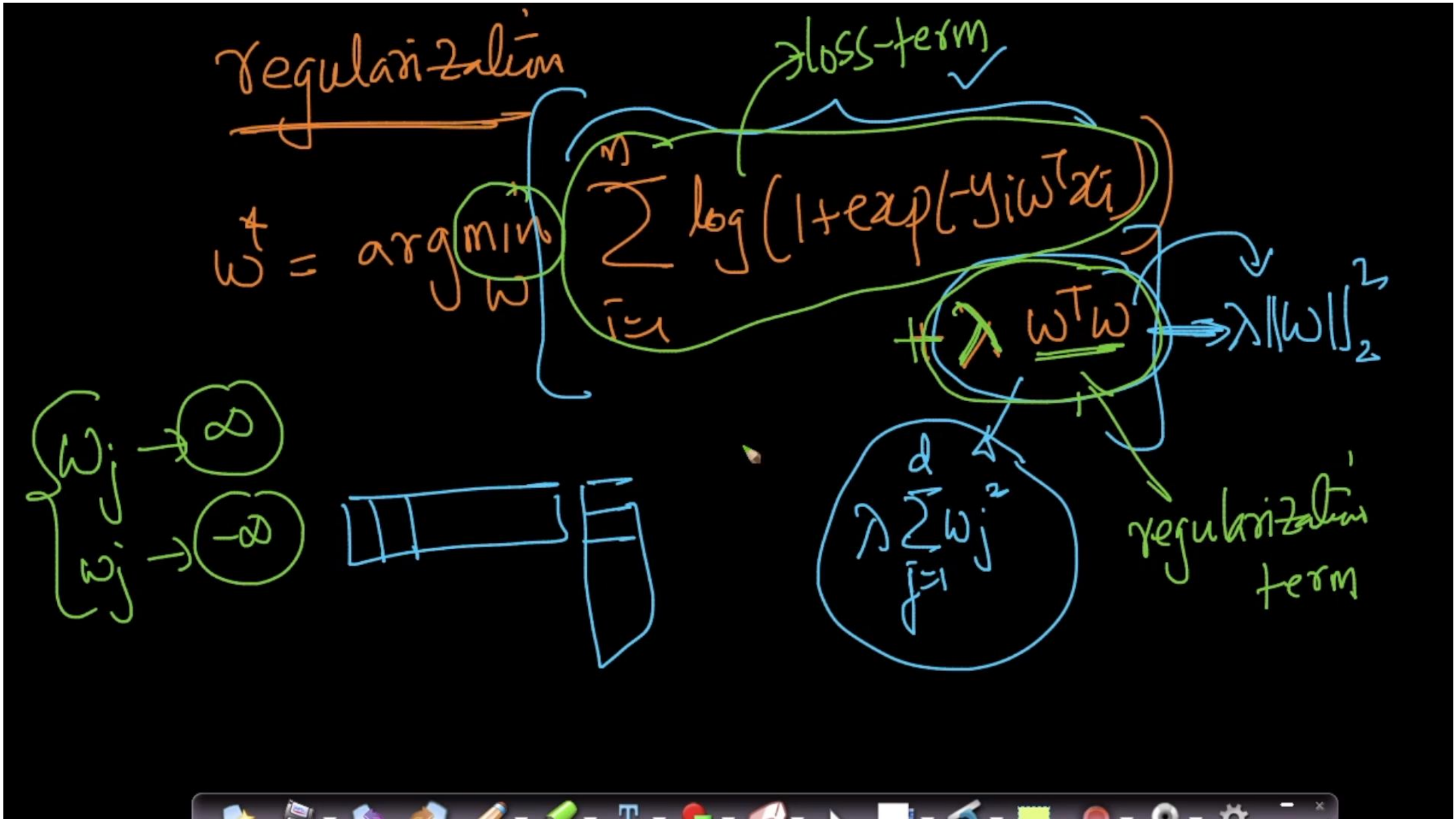
$\lambda\, w^T w \rightarrow \lambda \|w\|_2^2$

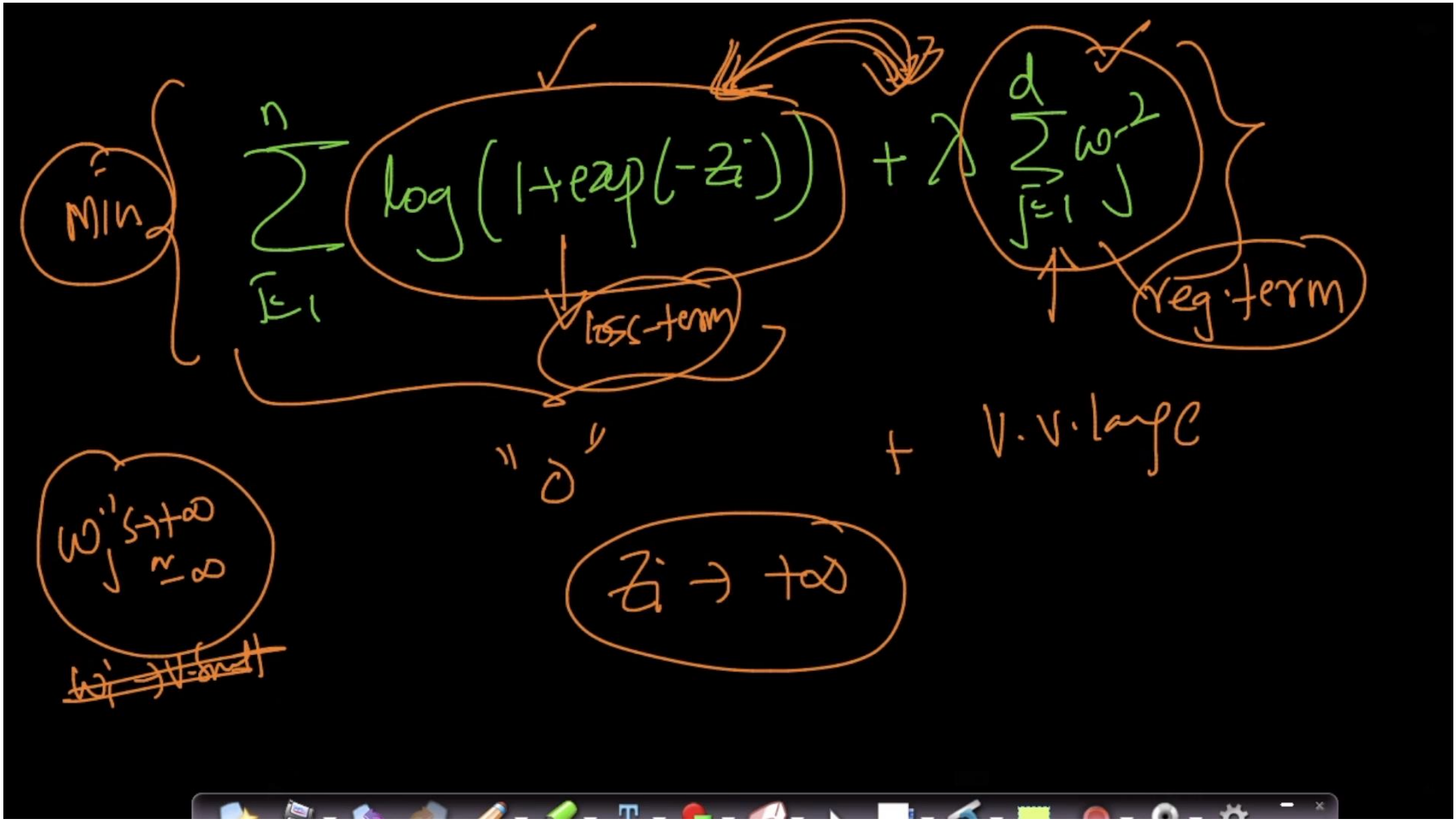
$\lambda \sum_{j=1}^{d} w_j^2$

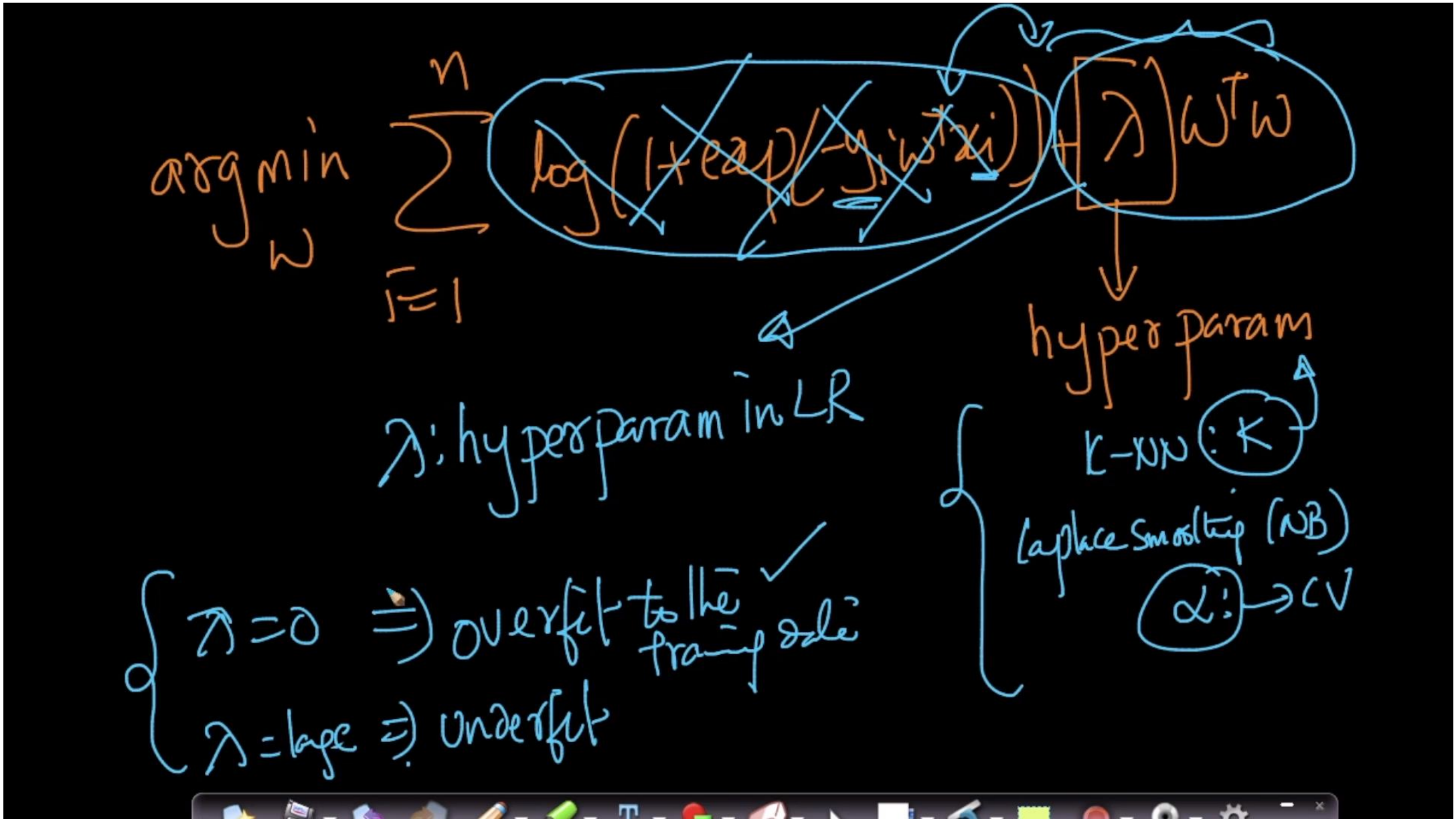
$w_j \rightarrow \infty$

$w_j \rightarrow -\infty$




Regularization
loss-term
$w^* = \arg\min_w \sum_{i=1}^{n} \log(1+\exp(-y_i w^T x_i))$
$+ \lambda w^T w$
$\lambda \|w\|_2^2$
$w_j \to \infty$
$w_j \to -\infty$
$\lambda \sum_{j=1}^{d} w_j^2$
regularization term


Min
$\sum_{i=1}^{n}$
$\log(1+\exp(-z_i))$
loss-term
$+\lambda \sum_{j=1}^{d} w_j^{-2}$
reg-term
"0"
+ V.V.Imp
w_j^{-1} s.t. $+\infty$ ≈ $-\infty$
$z_i \to +\infty$


argmin
w
n
∑
i=1
log(1+exp(-y_i w^T x_i)) + λ w^T w
λ: hyperparam in LR
hyperparam
K-NN : K
Laplace Smoothing (NB)
α: → CV
λ=0 ⇒ overfit to the training data
λ=large ⇒ underfit

$$\min \left(\text{loss-fn over train data} + \text{reg. fn.} \right)$$

→ Cross-validation

- $\lambda = 0 \Rightarrow$ overfit → high variance
- $\lambda =$ v. large $\Rightarrow$ underfit → high bias

L_1 regularization and Sparsity

$$w^* = \arg\min_w \underbrace{\sum_{i=1}^{n} \log\left(1+\exp\left(\overbrace{-y_i w^T x_i}^{z_i}\right)\right)}_{\text{logistic-loss}} + \underbrace{\lambda \|w\|_2^2}_{L_2\text{-reg.}}$$

$z_i \to +\infty \Rightarrow$ { $w_i \to +\infty$ or $w_i \to -\infty$ } {overfit

Question: alternatives to L_2-reg.

popular:- **L_1-reg**

$\|w\|_2^2$ for reg

↓

$\|w\|_1$ for reg:-

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

$$\|w\|_1 = \sum_{i=1}^{d} \underbrace{|w_i|}_{\text{abs}}$$

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

↳ hyperparam

will avoid $w_i \to +\infty$ or $-\infty$

Sparsity:

Sparse $w = \langle w_1, w_2, \ldots w_d \rangle$

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

w^+

If we use L1 reg in LR, all the unimportant or less-important features (w_i's) became zero

less-important

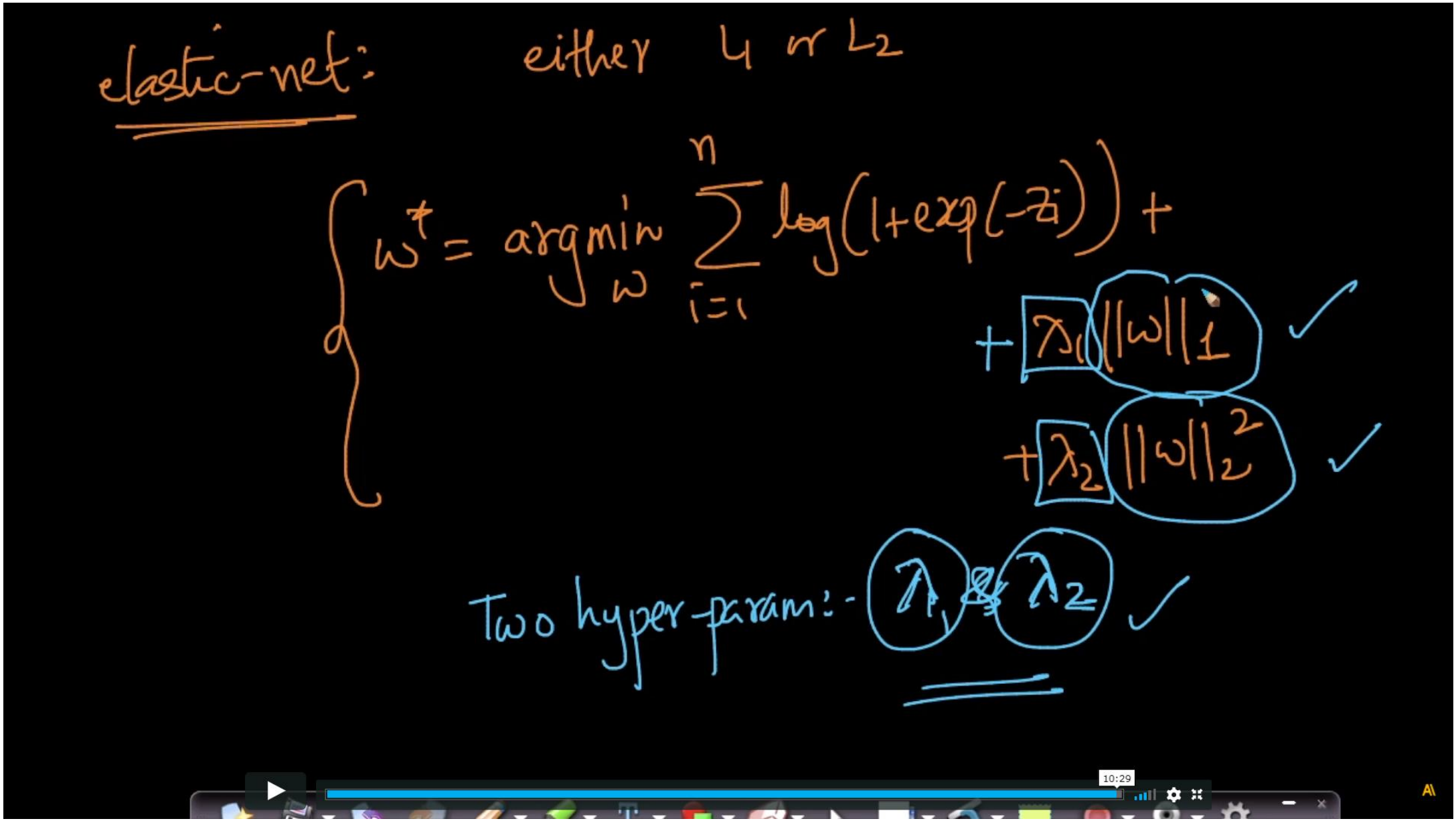
$f_1, f_2, \ldots$ ~~f_i~~ fd

$w = \langle w_1, w_2, \ldots$ ~~w_i~~ $\ldots$ $w_d \rangle$

↓

Zero if L_1 is used

if L_2 reg is used; w_i becomes a small values but not necessarily Zero


elastic-net: either L_1 or L_2
$w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} \log(1+\exp(-z_i)) +$
$+ \lambda_1 \|w\|_1$
$+ \lambda_2 \|w\|_2^2$
Two hyper-param:- λ_1 & λ_2
10:29

Probabilistic interpretation of Logistic regression

(Logistic regression) ① : Geometry & simple algebra ✓

↳ ② Probability

↳ ③ loss minimization

(Textbook)

prob. derivation of LR

Naive Bayes

→ (1) features are real valued :-

↳ Gaussian disb

$P(x_i | y_i) \sim N(\mu_i, \sigma_i)$

→ (2) $y_i = +1$ or 0

Bernoulli r.v (Coin-toss)

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

y_i: +1 or −1

prob: $w^t = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} -y_i \log p_i - (1-y_i)\log(1-p_i) + \text{reg}$

where $p_i = \sigma(w^T x_i)$

y_i: +1 or 0

Case 1: y_i : +ve

geom:- $y_i = +1$

prob:- $y_i = +1$

geom:- $\log(1+\exp(-w^T x_i))$

prob:- $(-1)\log\left(\frac{1}{1+\exp(-w^T x_i)}\right)$

$= \log(1+\exp(-w^T x_i))$

$p_i = \sigma(w^T x_i)$

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

Case 2: $y_i = -ve$ j $y_i = -1$ ← geom
$y_i = 0$ ← prob.

geom: $\log(1+\exp(w^T x_i))$

prob: $-1 \cdot \log\left(1 - \frac{1}{1+\exp(-w^T x_i)}\right)$

$-1 \cdot \log\left(\frac{\exp(-w^T x_i)}{1+\exp(-w^T x_i)}\right)$

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

$$= \log\left(\frac{1+\exp(-w^T x_i)}{\exp(-w^T x_i)}\right)$$

divide numerator & den by $(\exp(-w^T x_i))$

$$= \log\left(1+\frac{1}{\exp(-w^T x_i)}\right)$$

$$\frac{1}{e^{-x}} = e^{x}$$

$$= \log\left(1+\exp(w^T x_i)\right)$$

Loss minimization interpretation of LR

✓ geometric & probabilistic

Loss-minimization ✓

$$\begin{cases} f(x_i) = w^T x_i \\ w^* = \arg\min_w \sum_{i=1}^{n} \log\left(1 + \exp(-y_i w^T x_i)\right) \\ z_i = y_i w^T x_i = y_i \cdot f(x_i) \end{cases}$$

ideal - optmzn. model:

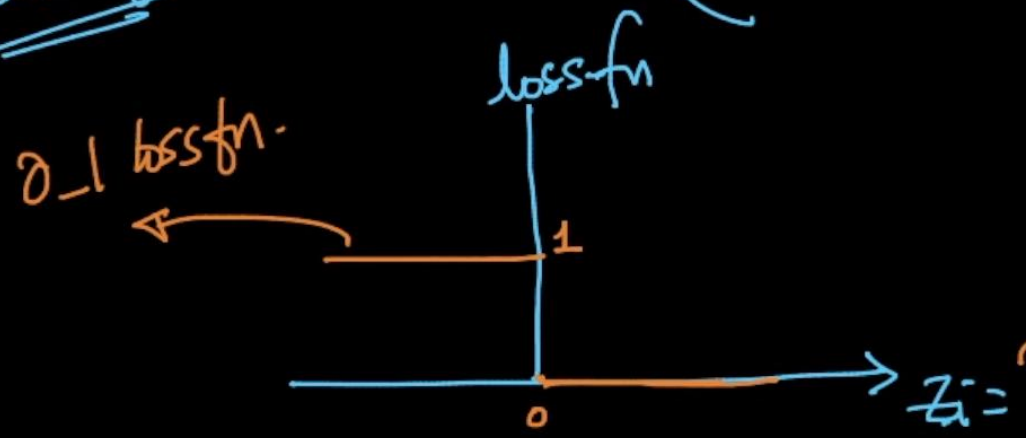
Classfn:

$$w^* = \arg\min_{w} \left(\text{number of incorrectly classified pts} \right)$$

ideal loss-function (0_1 loss-fn)

+1 : incorrectly classified

0 : correctly classified



$z_i = y_i \cdot w^T x_i$

$y_i : f(x_i)$

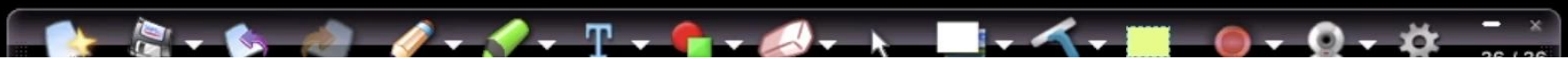
min. loss ✓

Max. profit ✓

ideal

$$w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} \underbrace{0\_1 \text{ loss}(x_i, y_i, w)}$$

$$0\_1 \text{ loss}(z_i) = \begin{cases} 1 & \text{if } z_i < 0 \\ 0 & \text{if } z_i \geq 0 \end{cases}$$



Solve optimization problems in ML

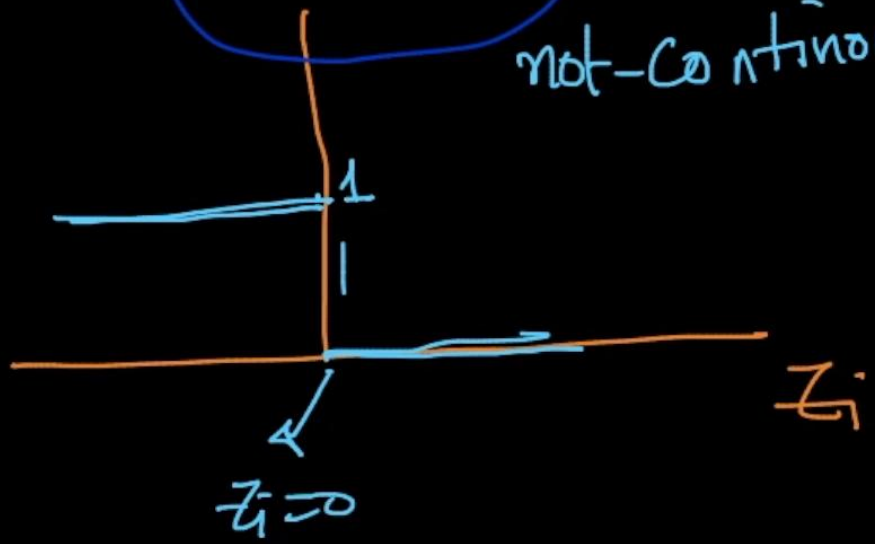
↳ "differentiation" in calculus.

↳ next-chapter

$f(x)$ is differentiable

↳ "Continous"

not-Continous ⇒ not-Differentible

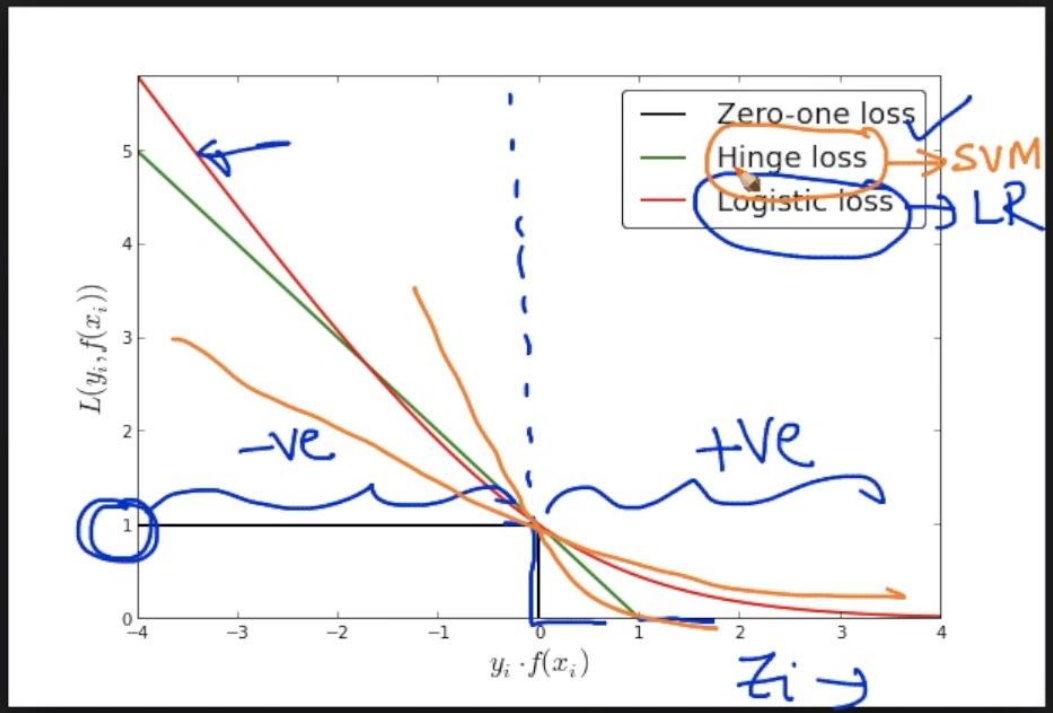


Contents - Google Docs | NBayesLogReg.pdf | logistic loss function - Google | plot(log(1+exp(-x)) - Google S | Chekuri Srikan...

Secure | https://www.google.co.in/search?q=logistic+loss+function&biw=1164&bih=635&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjli8Lx...

(in this set up, $y_i = 0$ or 1)

logistic loss ~ 0-1 loss



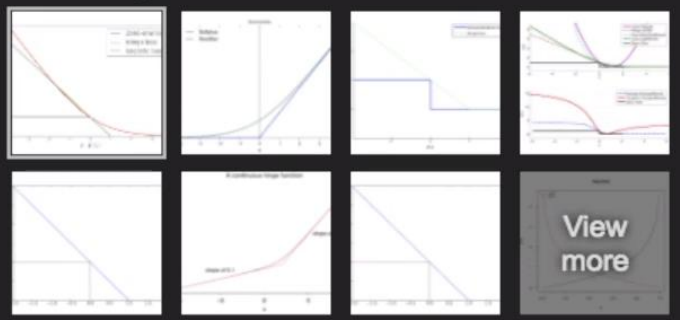
How can logistic loss return 1 for x = ...

Cross Validated - Stack Exchange - 612 × 432 - Search by image

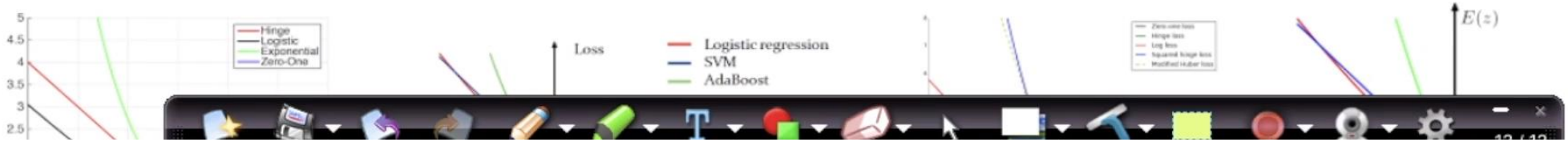
enter image description here

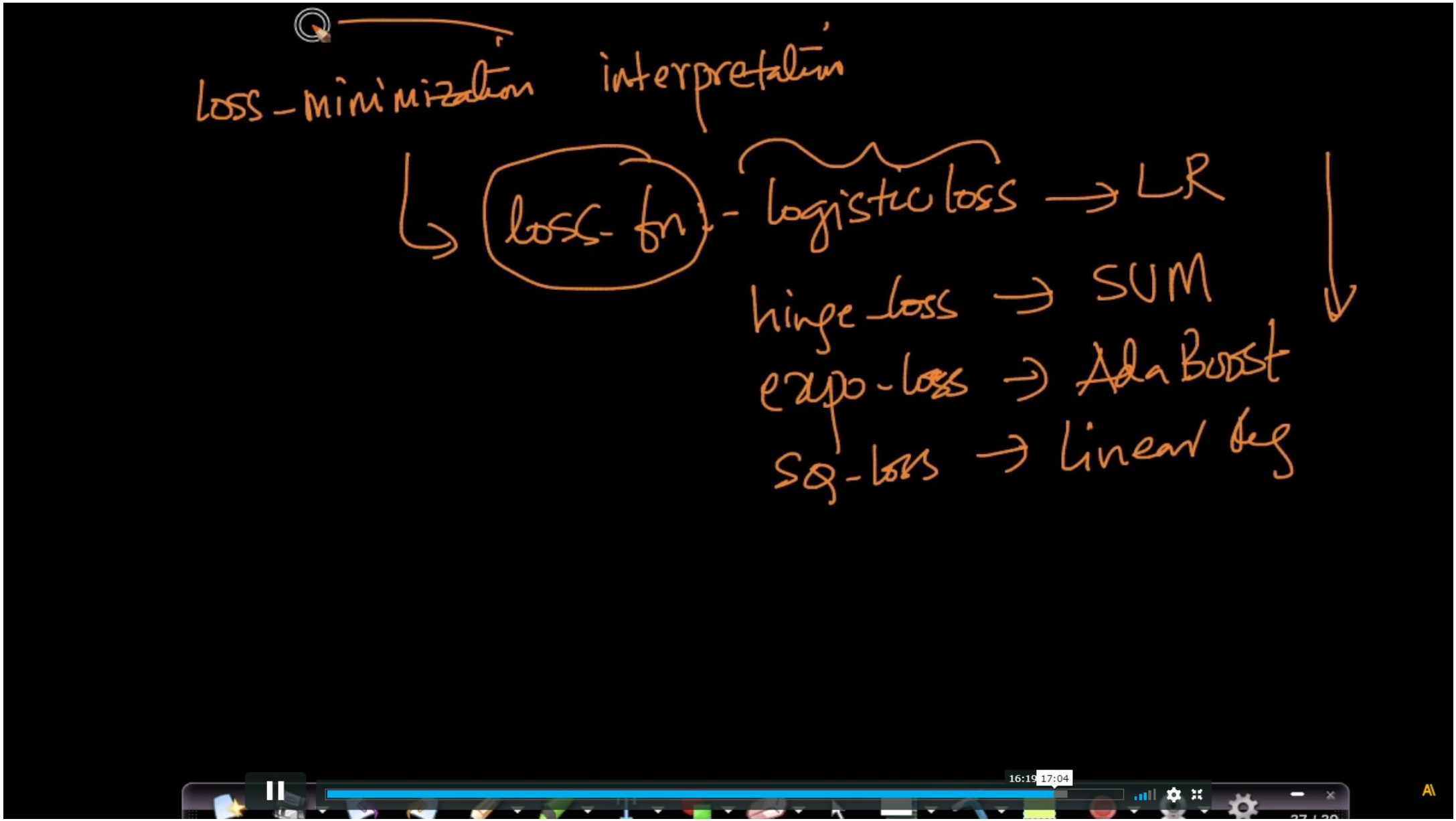
Visit | View image | Save | View saved | Share

Related images:



Images may be subject to copyright. - Get help - Send feedback




loss - minimization
interpretation
loss_fn - logistic loss → LR
hinge loss → SVM
expo - loss → AdaBoost
sq - loss → Linear Reg
16:19
17:04

Hyperparameter search/optimization

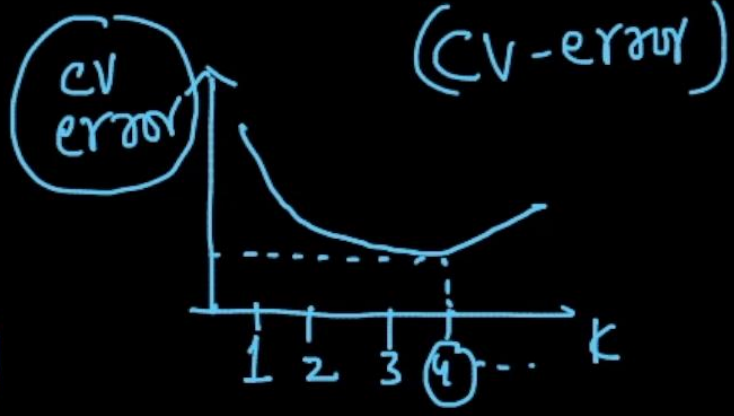
λ : hyperparam

$\lambda = 0 \Rightarrow$ overfitting

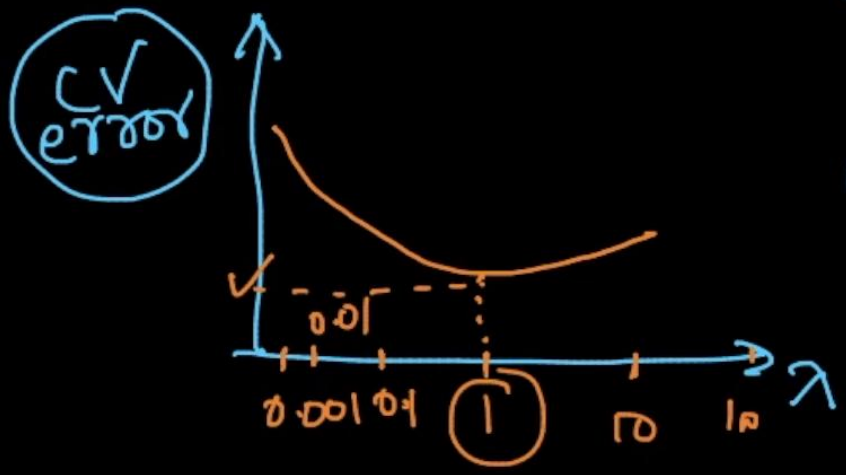
$\lambda = \infty \Rightarrow$ underfitting.

(Q) how the determine the best λ

- λ : LR
- (K) : K-NN
- α : NB (Laplace Smoothing)



✓ Grid Search (Brute force)



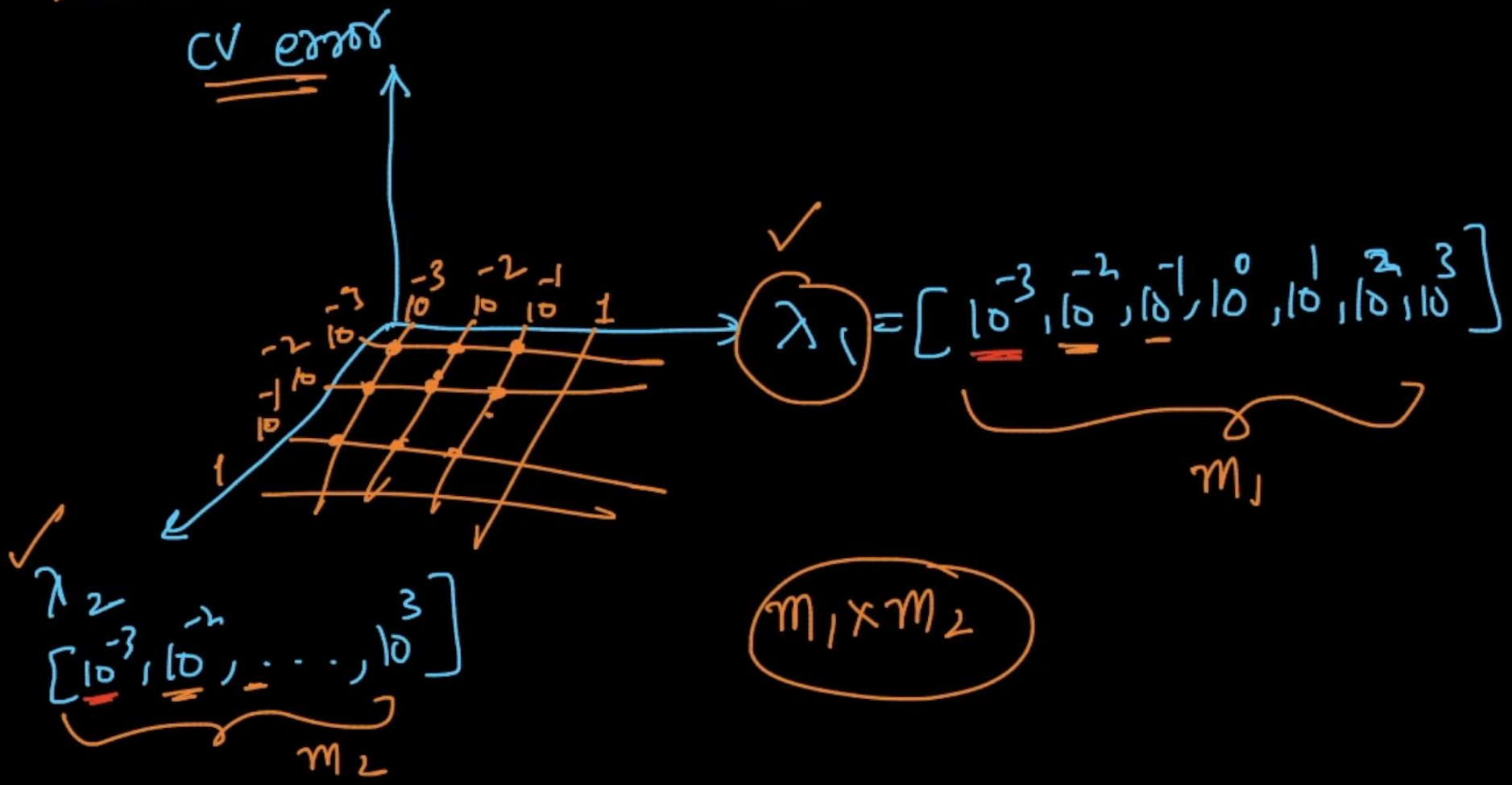
① $\lambda = [0.001, 0.01, 0.1, 1, 10, 100, 1000]$ (10, 10, 10, 10)

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

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

$2 \times 2 \times 10^{2}$

elastic net :- $\lambda_1 \|w\|_1 + \lambda_2 \|w\|_2^2$



Grid Search

$m_1 = m_2 = m_3 = \dots = m$

→ deeplearning

λ_1 : 1 hyperparam :- m_1 times train m

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

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

K-hyperpara :- m^k

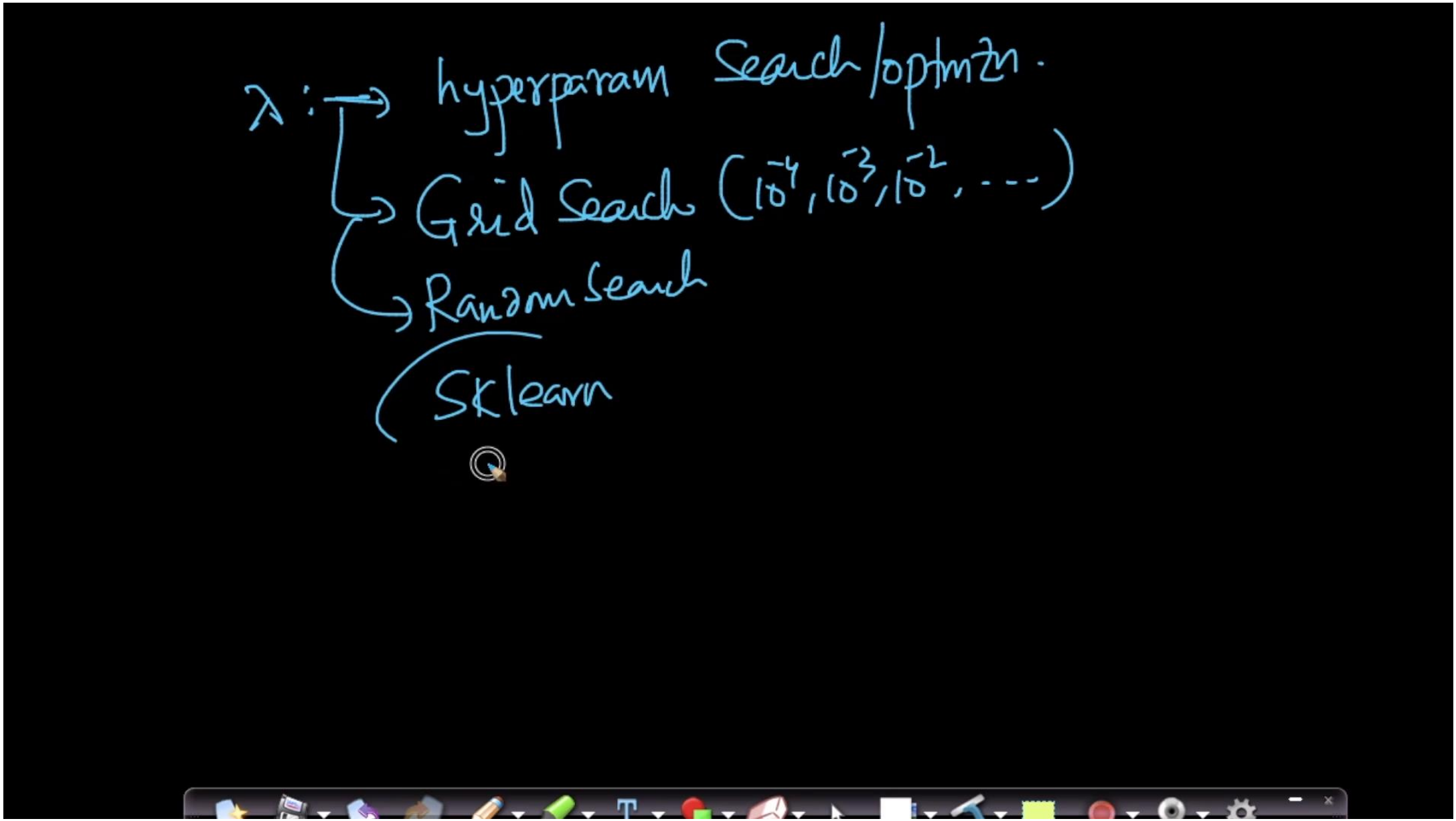
as # hyperparam increase, the # times the model need to be trained increases exponentially.

$k \uparrow$; $m^k \uparrow$

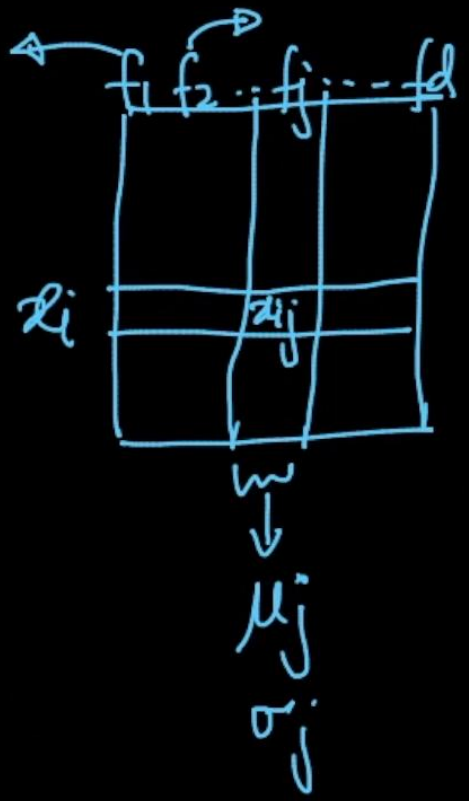
"Random" Search's → almost is as good as grid_search

#hyperparam is large

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


λ :
hyperparam Search/optmzn.
Grid Search ($10^{-4}, 10^{-3}, 10^{-2}, \ldots$)
Random Search
Sklearn

Column / feature standardization



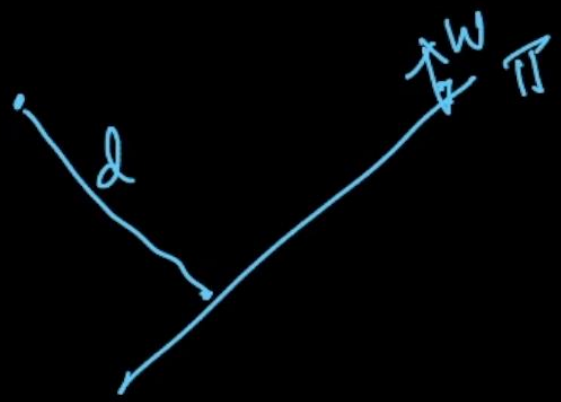
$x_i \in R^d$

K-NN :- distances

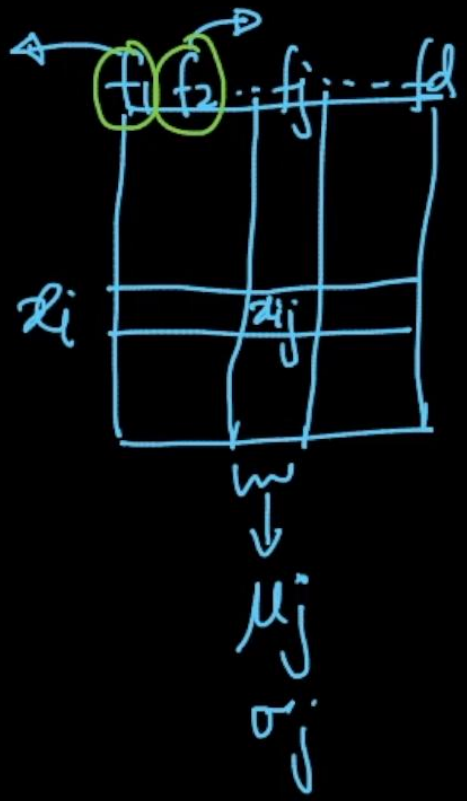
(Standardize our feature)

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

Logistic regression :- "mandatory" to perform feature standardization before training on your 2D



Column / feature standardization



✓ K-NN :- distances

Standardize our feature

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

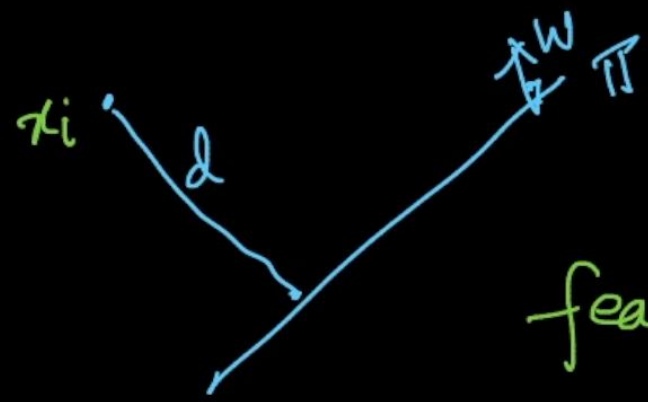
centering

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

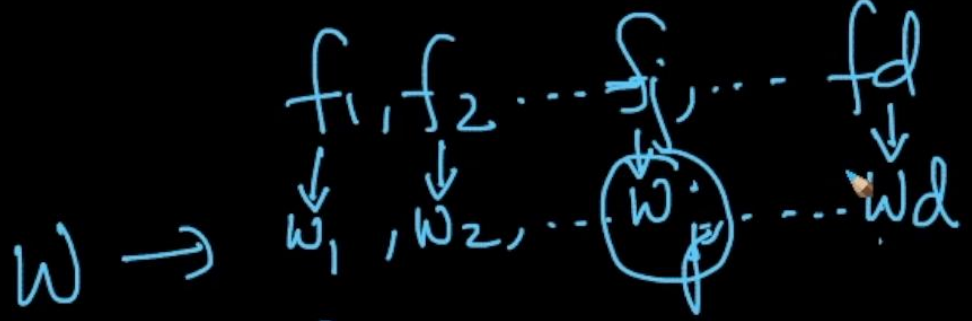
scaling

standardization = mean-centering & scaling

Logistic regression :- "mandatory" to perform feature standardization before training on your 2D feature scales



Feature importance & Model Interpretability



LR : GNB + Bernoulli

assume: if all features are independent (NaiveBayes)

feature-imp: w_i's

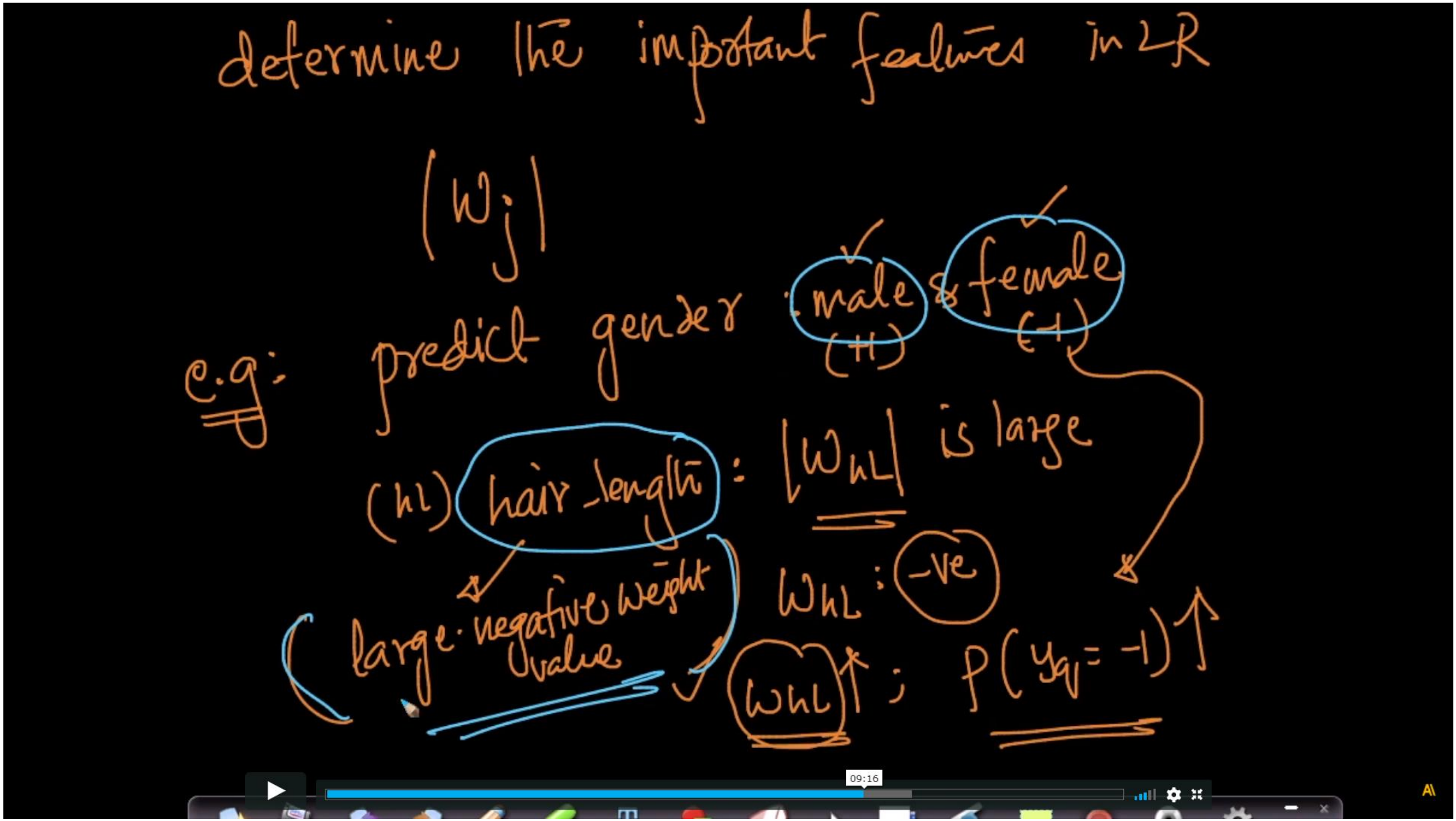
$|w_j|$ = absolute value of weight corresponding to f_j

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

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

↳ $P(y_q = +1) \uparrow$

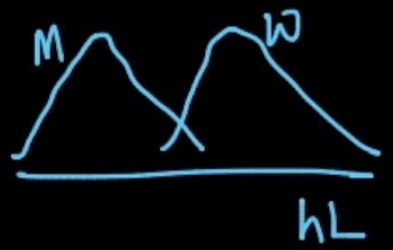
Case 2: w_j : -ve & large ; $\sum_{j=1}^{d} w_j x_{qj}$ $P(y_q = -1) \uparrow$


determine the important features in LR
$|w_j|$
e.g: predict gender : male (+1) & female (-1)
(hl) hair_length : $|w_{hl}|$ is large
large negative weight value
w_{hl} : -ve
$w_{hl}\uparrow$; $P(y_q = -1)\uparrow$
09:16

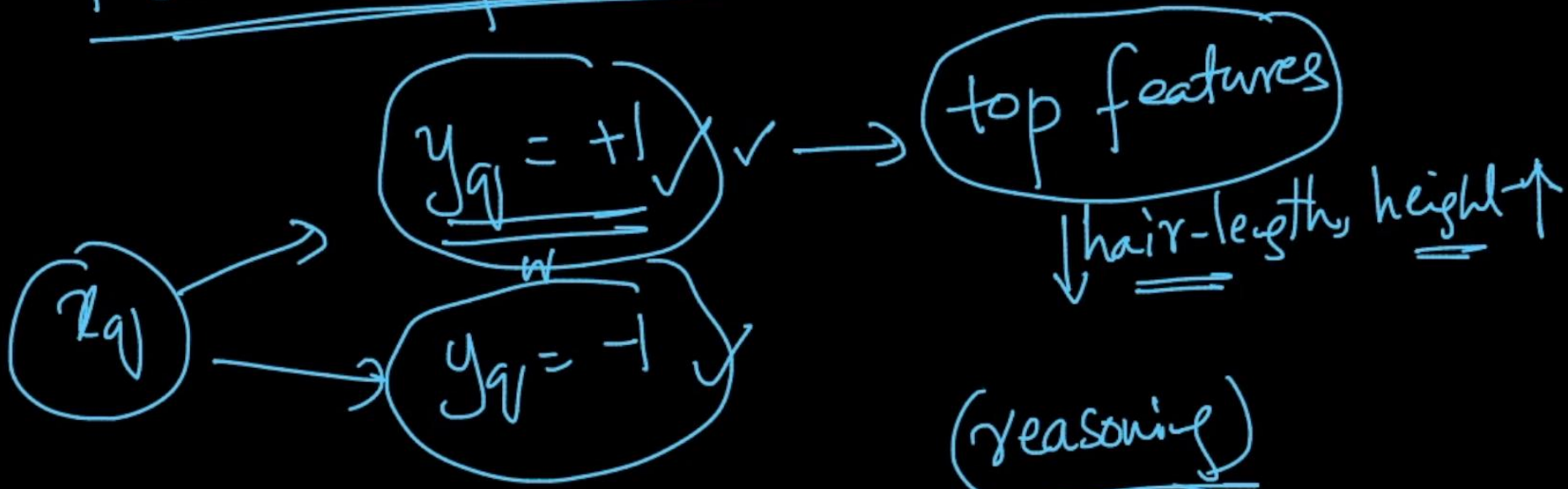
[h3] height ↑ ; $p(y_q = +1) \uparrow$ (↑ male)

w_h : +ve

↳ medium. positive weight



Model interpretability



(reasoning)

100 features:- Top 5-features

"Collinearity" (or) Multicollinearity

f.I :- features are indep; $|w_j|$ as f.I values

Collinearity: $\tilde{f}_i, \tilde{f}_j$

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 s.t

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

then f_1, f_2, f_3 & f_4 are said to be multicollinear

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

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

$$w^* = \langle 1, 2, 3 \rangle \quad ; \quad x_q = \langle x_{q1}, x_{q2}, x_{q3} \rangle$$
$$f_1 \; f_2 \; f_3$$

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

if $f_2 = 1.5 f_1 \Rightarrow f_1$ & f_2 are collinear

$$w^T x_q = x_{q1} + 3 x_{q1} + 3 x_{q3} = 4 x_{q1} + 3 x_{q3}$$
$$= \langle 4, 0, 3 \rangle$$

$w^* = \langle 1, 2, 3 \rangle$ — f_3 is the most imp

$\tilde{w} = \langle 4, 0, 3 \rangle$ — not all imp

$\therefore f_2 = 1.5 f_1$ — same classifier

f_1 is the most imp

$x_q \rightarrow w^T x_q$

$|w_j|$ as f.I

determine. If features are multicollinear:

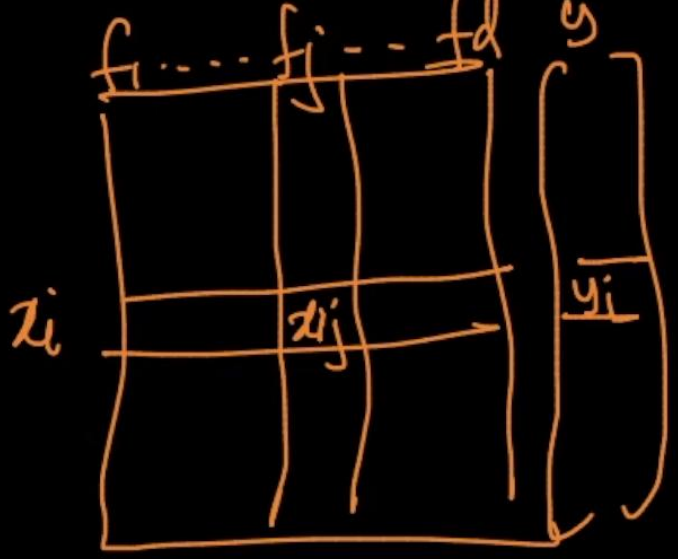
perturbation technique:

$x_{ij} + \epsilon$

Small noise

$N(0, 0.01)$

$\mu \quad \sigma$



before perturbation: $w = \langle w_1, w_2 \cdots w_j \cdots w_d \rangle$

after perturbation: $\tilde{w} = \langle \tilde{w}_1, \tilde{w}_2 \cdots \tilde{w}_j \cdots w_d \rangle$

---

if w_i & $\tilde{w}_i$ differ significantly
then your features are collinear

$\Downarrow$

Cannot use $|w_j|$'s as f: I

Train & Run time Space & time Complexity

n = # pts in D_{train}
d = dim

$O(nd)$

Train LR :-

SGD

$w^* = \underset{w}{\text{argmin}}$ (logistic loss + reg)

↳ next-chapter optimizn

$w^t = \langle w_1, w_2, \ldots w_d \rangle$

Run Time (LR)

Space: $O(d)$

Time : $O(d)$

$w^{*T} x_q \begin{cases} > 0 \rightarrow +ve \\ < 0 \rightarrow -ve \end{cases}$

$\sum_{j=1}^{d} w_j x_{qj}$

if d is small (d=30)

→ LR is v.v-good fw low-latency applicatin

x_q → (1ms) y_q ✓

(30 multiplications + 29 additions)

favourite algo internet Companies

→ memory efficient

(w^T)

x_q → [ ] → y_q

if d is large $d \approx 1000$

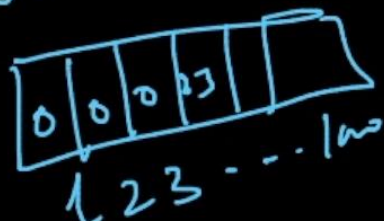
$\lambda \uparrow$; underfit

Bias-Var ✓

$w^T x_q$:- 1000 mult & additions

→ L_1 reg: sparsity (w_j's corresponding to less important features $= 0$

w^t:



λ : reasonably

1ms

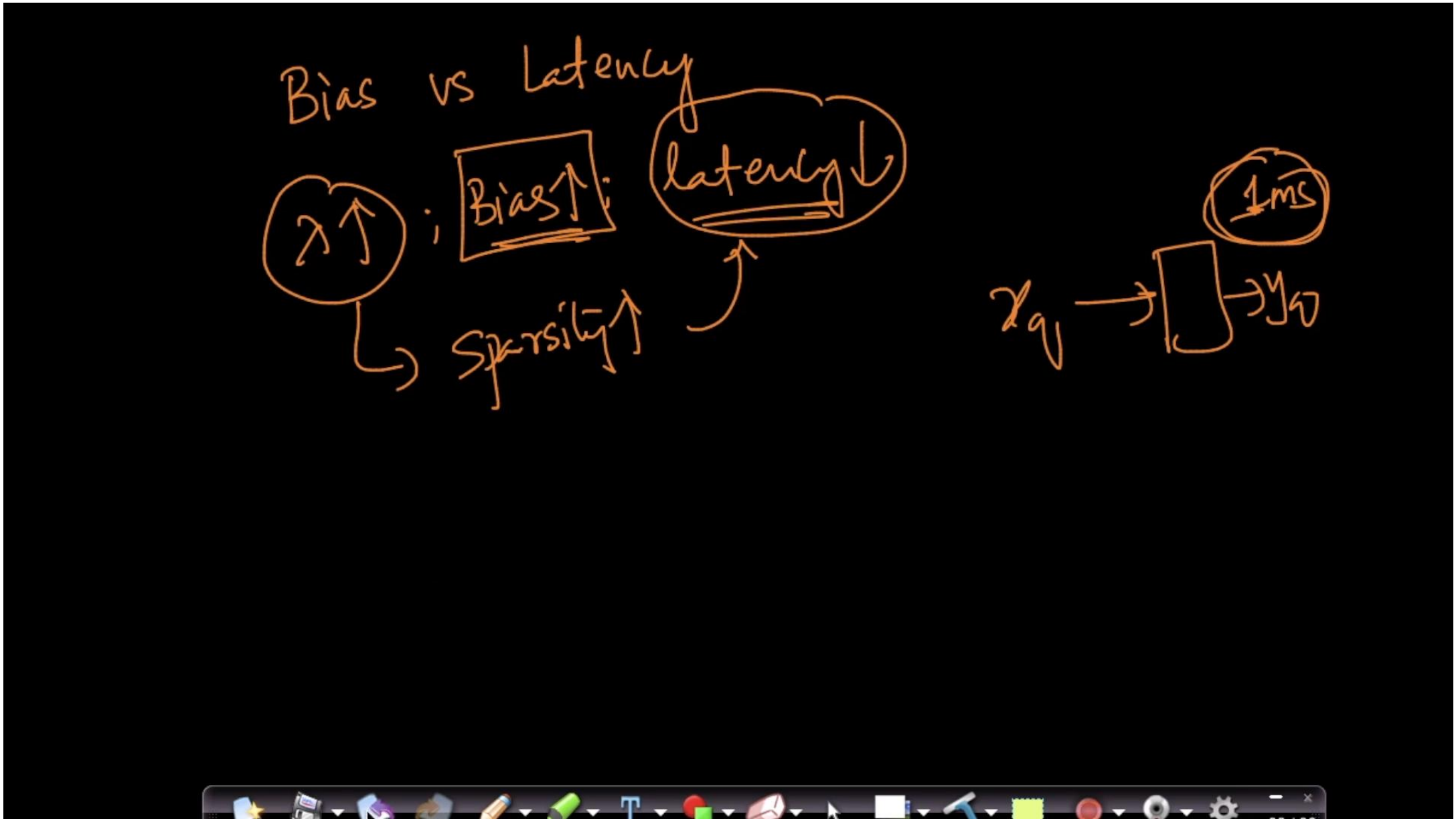
50 mult & 50 adds

latency req

$\lambda \uparrow$; Sparsity ↑

more of $w_j = 0$

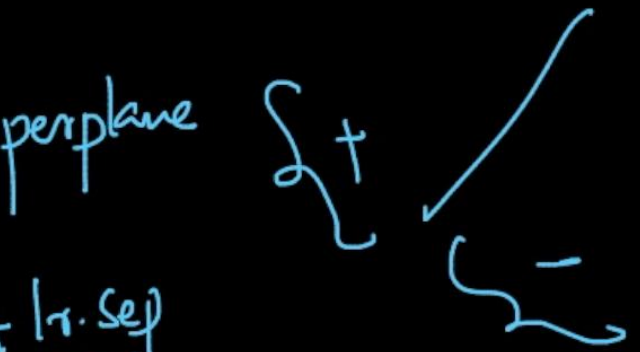
Bias, Var, Latency

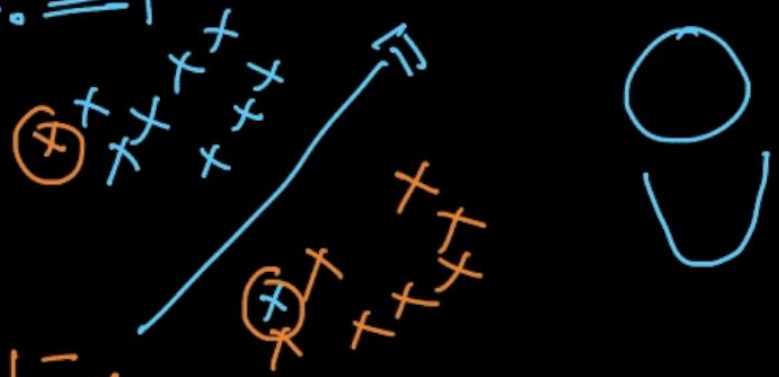

Bias vs Latency
λ↑
Bias↑
latency↓
Sparsity↑
1ms
x_q
y_q

Cases:

decision surface:- Linear / hyperplane

assumption :- data is linearly
Seperable (n) almost ln. sep





feature Imp & Interpretability

$|w_j|$ if features are not multicollinear

$x_{qj} \rightarrow y_{qp} = +1$

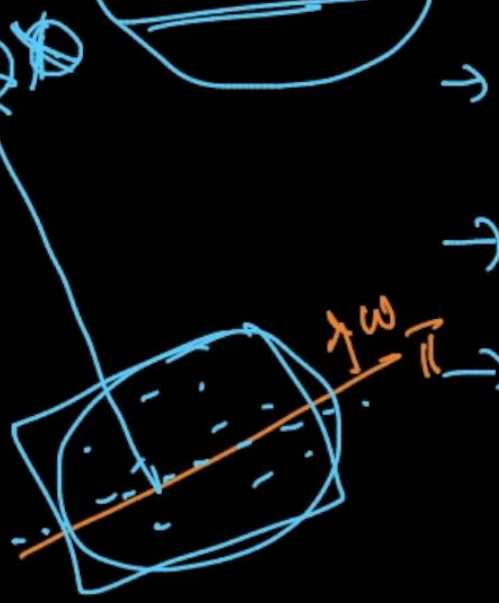
Imbalanced data:- upsampling n downsampling

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

→ $D_{train} \rightarrow w^*$

→ $x_i \rightarrow w^{*T} x_i$: dist fm π to x_i

→ remove pts which are very fw away fm π from $D_{train} \rightarrow D'_{train}$



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

*

Missing values:- imputation → mean, median
↳ model

Multi-class: One Vs Rest ← Typically extensions to LR

- MaxEnt models
- Softmax classifier → deep learning
- Multinomial LR

Similarity matrix: extension to LR → {Kernel LR}

SVM

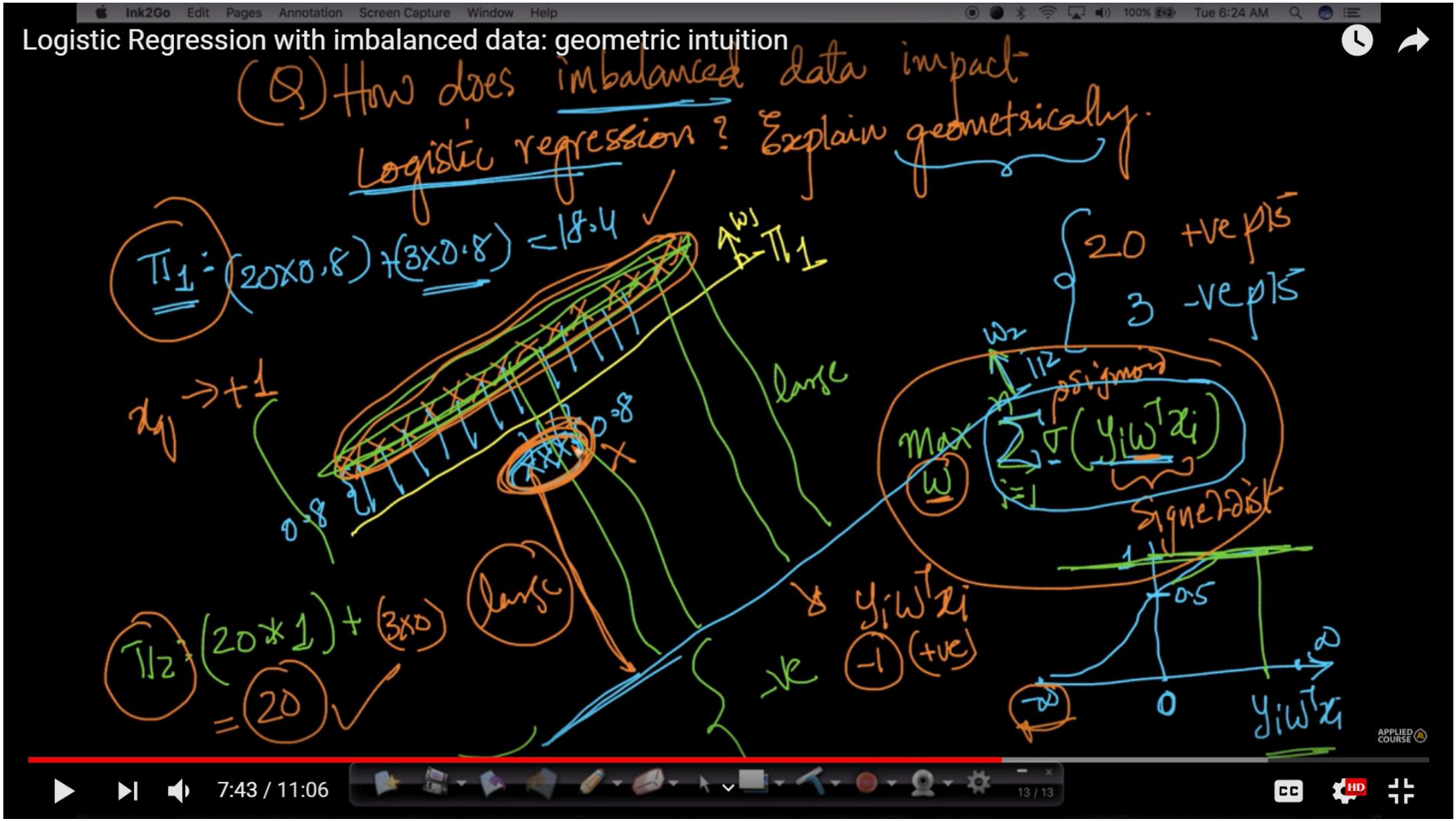
Best & Worst cases:- → almost Lr. seperable
→ low-latency requirement (L1-reg)
→ very fast to train

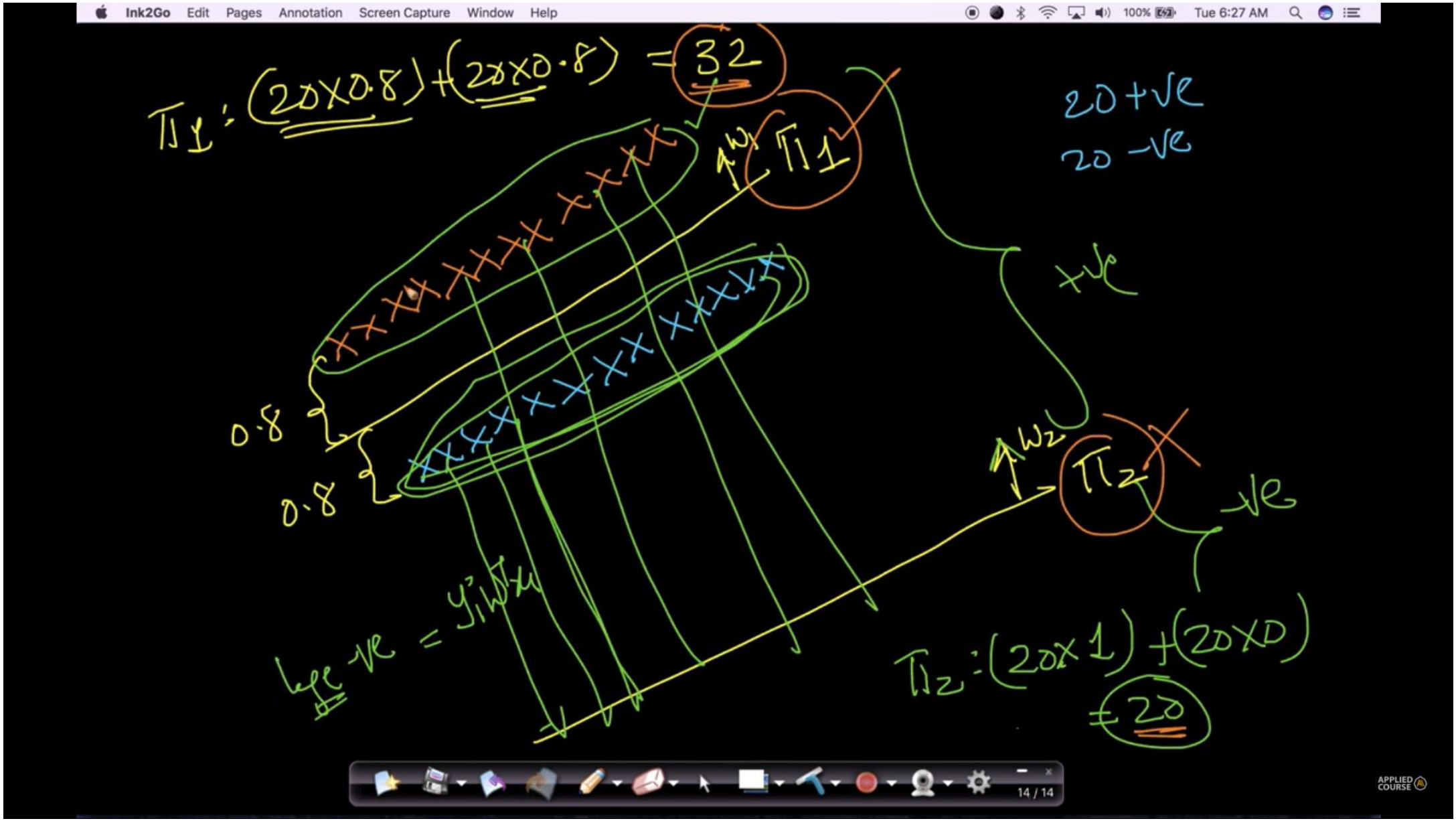
Large 'd':

d is large: { chance data is lr-seperable is high

↓

low-latency → L1 reg


Logistic Regression with imbalanced data: geometric intuition
(Q) How does imbalanced data impact Logistic regression? Explain geometrically.
π_1 : (20x0.8) + (3x0.8) = 18.4
π_2 : (20*1) + (3x0)
= 20
large
large
$x_q \to +1$
0.8
0.8
w_1
w_2
π_1
π_2
20 +ve pts
3 -ve pts
sigmoid
$\max_w \sum_{i=1}^{n} \sigma(y_i w^T x_i)$
Signed dist
$y_i w^T x_i$
-ve
(-1)
(+ve)
1
0.5
0
∞
$-\infty$
$y_i w^T x_i$
APPLIED COURSE
7:43 / 11:06
13 / 13
CC
HD


Ink2Go Edit Pages Annotation Screen Capture Window Help
100% Tue 6:27 AM
Π_1 : (20x0.8) + (20x0.8) = 32
W_1
Π_1
20 +ve
20 -ve
+ve
0.8
0.8
W_2
Π_2
-ve
Let -ve = $y_i W_{xi}$
Π_2 : (20x1) + (20x0)
= 20
14 / 14
APPLIED COURSE