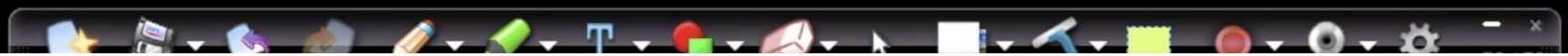
Support Vector Machines (SVM)

✓ (SVM) → popular ML → classifn
                      → regrsn

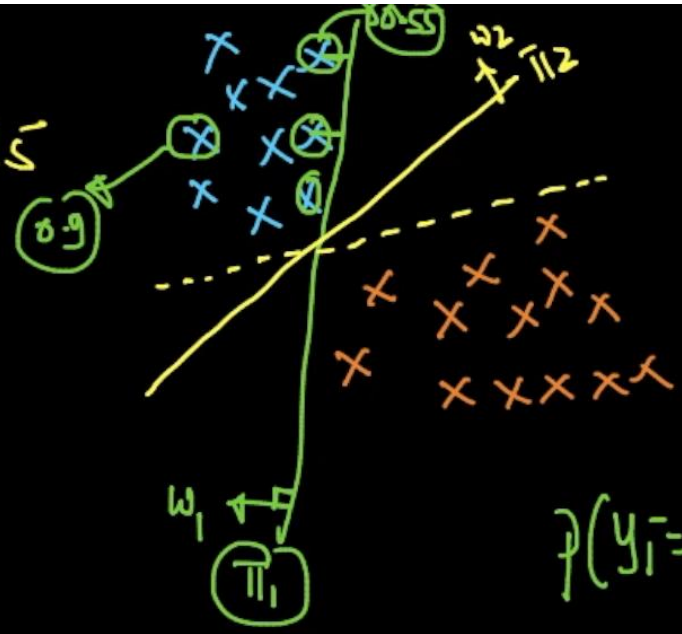
→ 1990's

geom-intuition



→ many Πs that seperate +ve from -ve pts

→ Key idea of SVM:-
Π that seperates +ve from -ve pts as 'widely' as possible

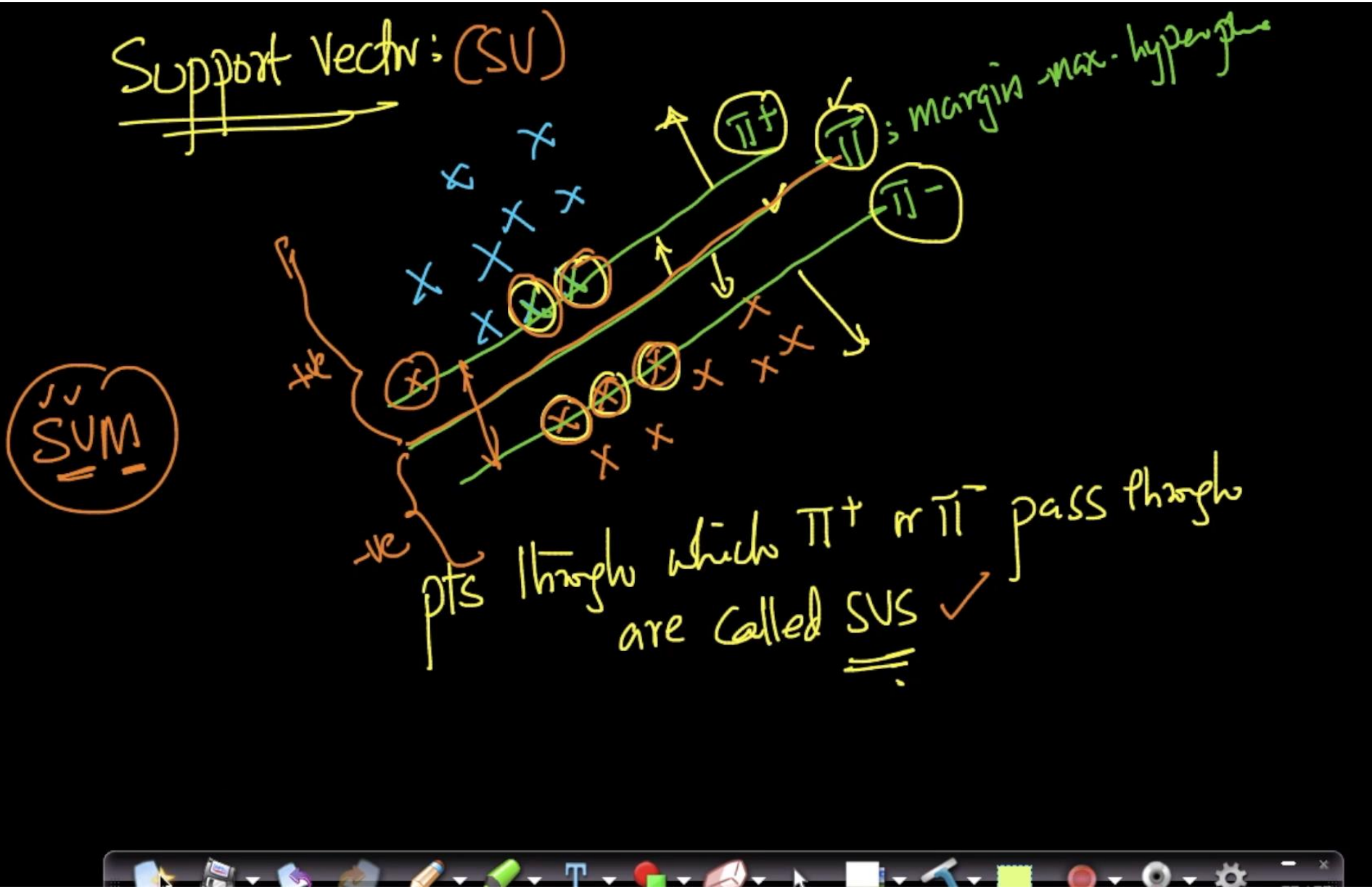


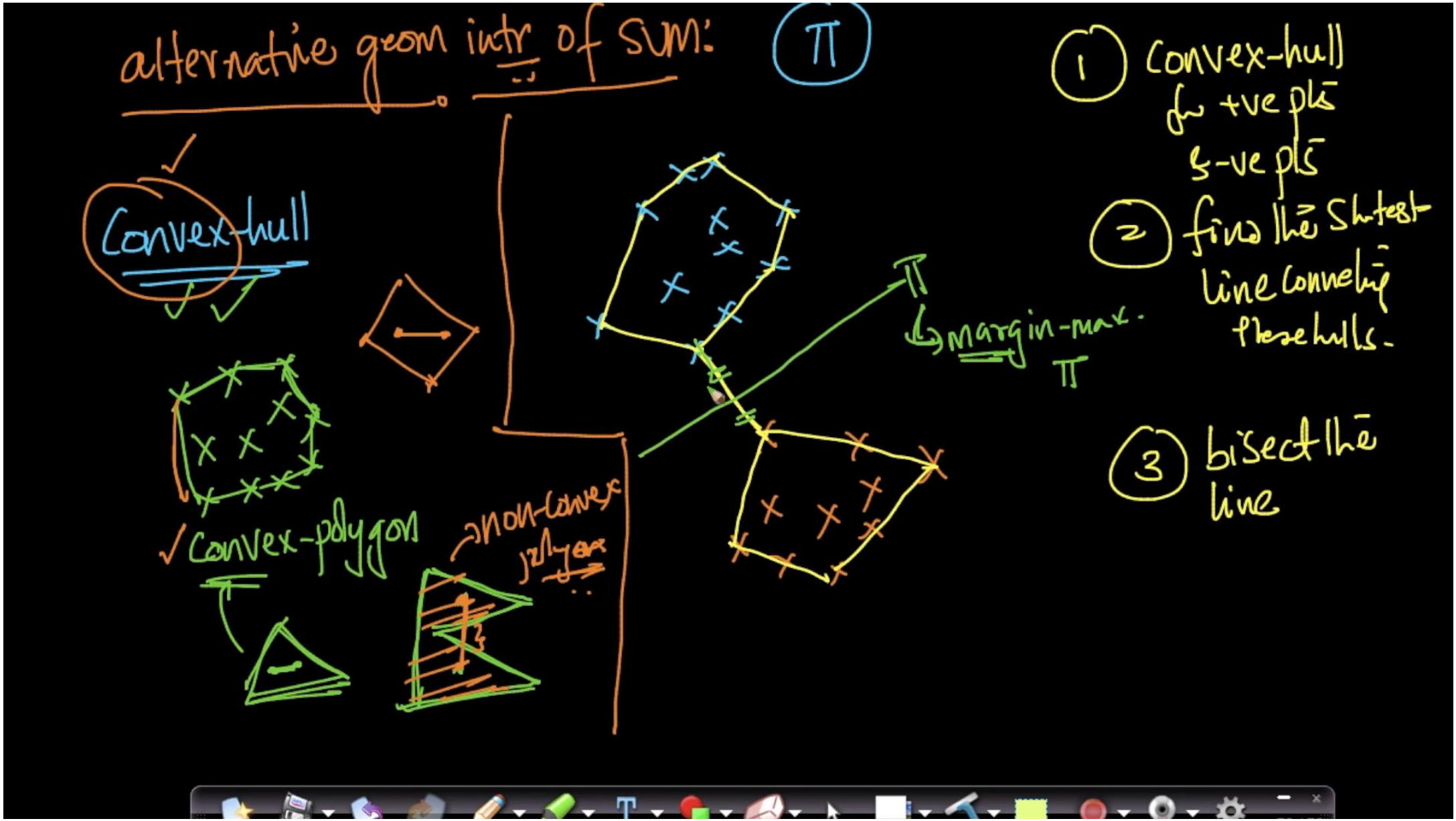
$$P(y_i = 1) = \sigma(w^T x_i)$$

SUM :- Try to find a Π that maximizes the margin = dist (Π^+, Π^-)

generalization acc ↑

Support Vectrs: (SV)




alternative geom intr of SVM:
π
Convex-hull
Convex-polygon
non-convex polygon
margin-max. π
1 Convex-hull for +ve pts & -ve pts
2 find the shortest line connecting these hulls.
3 bisect the line

Mathematical formulation of SVM:

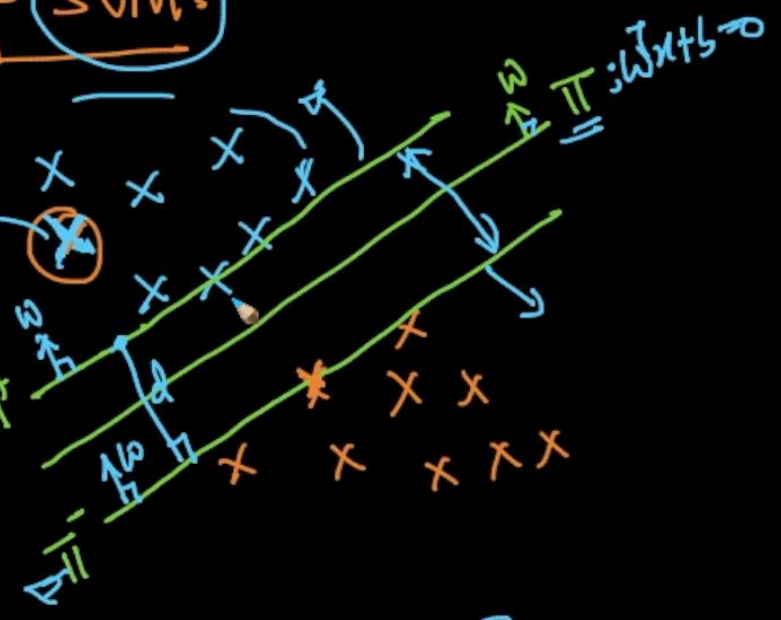
π :- margin - maximization

Let $\pi : w^T x + b = 0$

if $\pi^+ : w^T x + b = 1$

$\pi^- : w^T x + b = -1$

$y_i (w^T x_i + b) \geq 1$

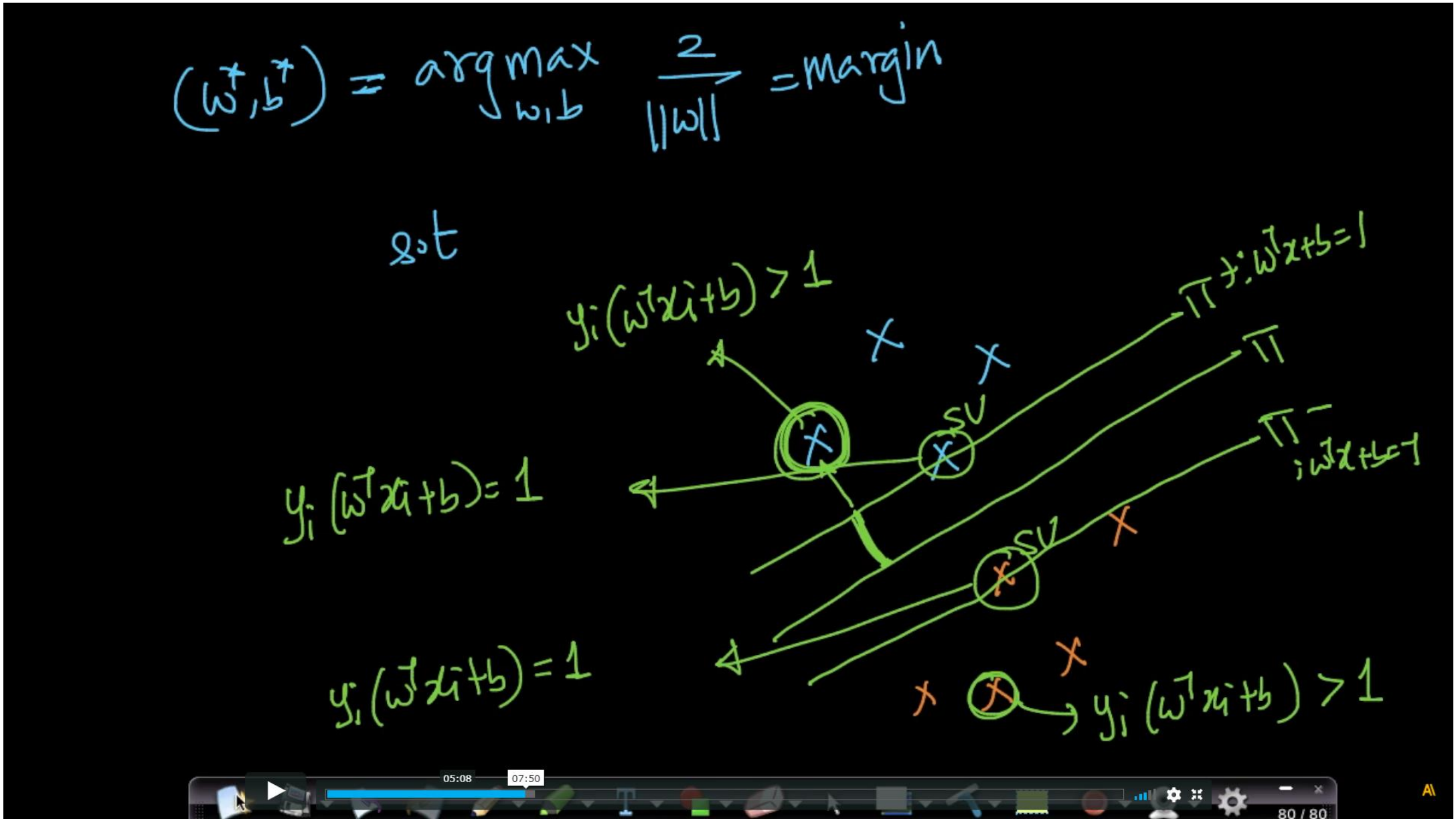


Margin: $d = \frac{2}{\|w\|}$ ✓

$(w^T x + b = -1)$

$(w, b) = \underset{w,b}{\text{argmax}} \frac{2}{\|w\|}$

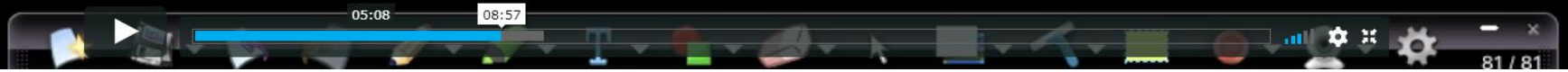
s.t (constr)


$(w^*, b^*) = \underset{w,b}{\arg\max} \frac{2}{\|w\|}$ = margin
s.t
$y_i(w^T x_i + b) > 1$
$\pi^+ : w^T x + b = 1$
π
$\pi^- : w^T x + b = -1$
SV
SV
$y_i(w^T x_i + b) = 1$
$y_i(w^T x_i + b) = 1$
$y_i(w^T x_i + b) > 1$
05:08
07:50
80 / 80

$$(w^*, b^*) = \underset{w,b}{\operatorname{argmax}} \quad \frac{2}{\|w\|} \leftarrow \text{Margin}$$

$$\text{s.t} \quad y_i (w^T x_i + b) \geq 1 \text{ for all } x_i$$

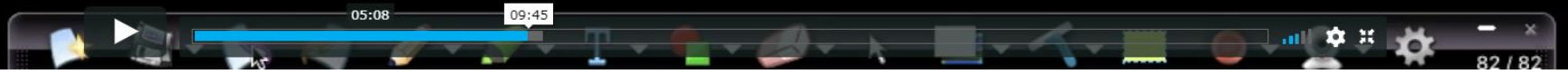
n-constr.



Constr. optmzn. pool. of SVM

$$\vec{w}, b^* = \underset{w,b}{\operatorname{argmax}} \frac{2}{\|w\|}$$

$$\text{s.t} \quad \forall i, \; y_i(\vec{w}^T x_i + b) \geq 1$$

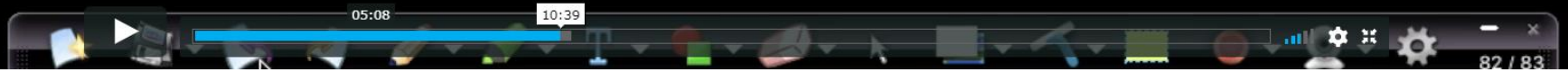


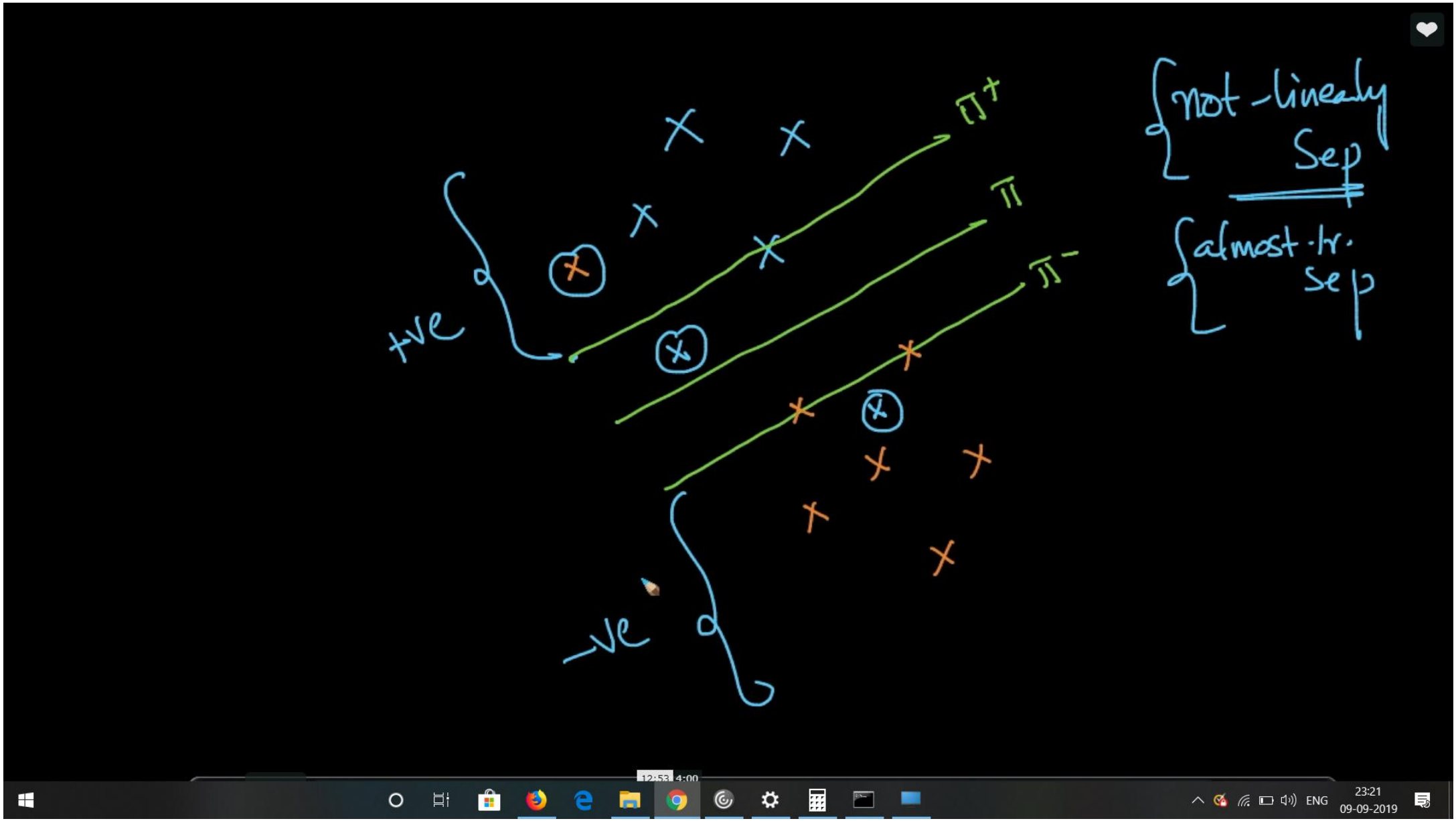
Constr. optmzn. problem of SVM

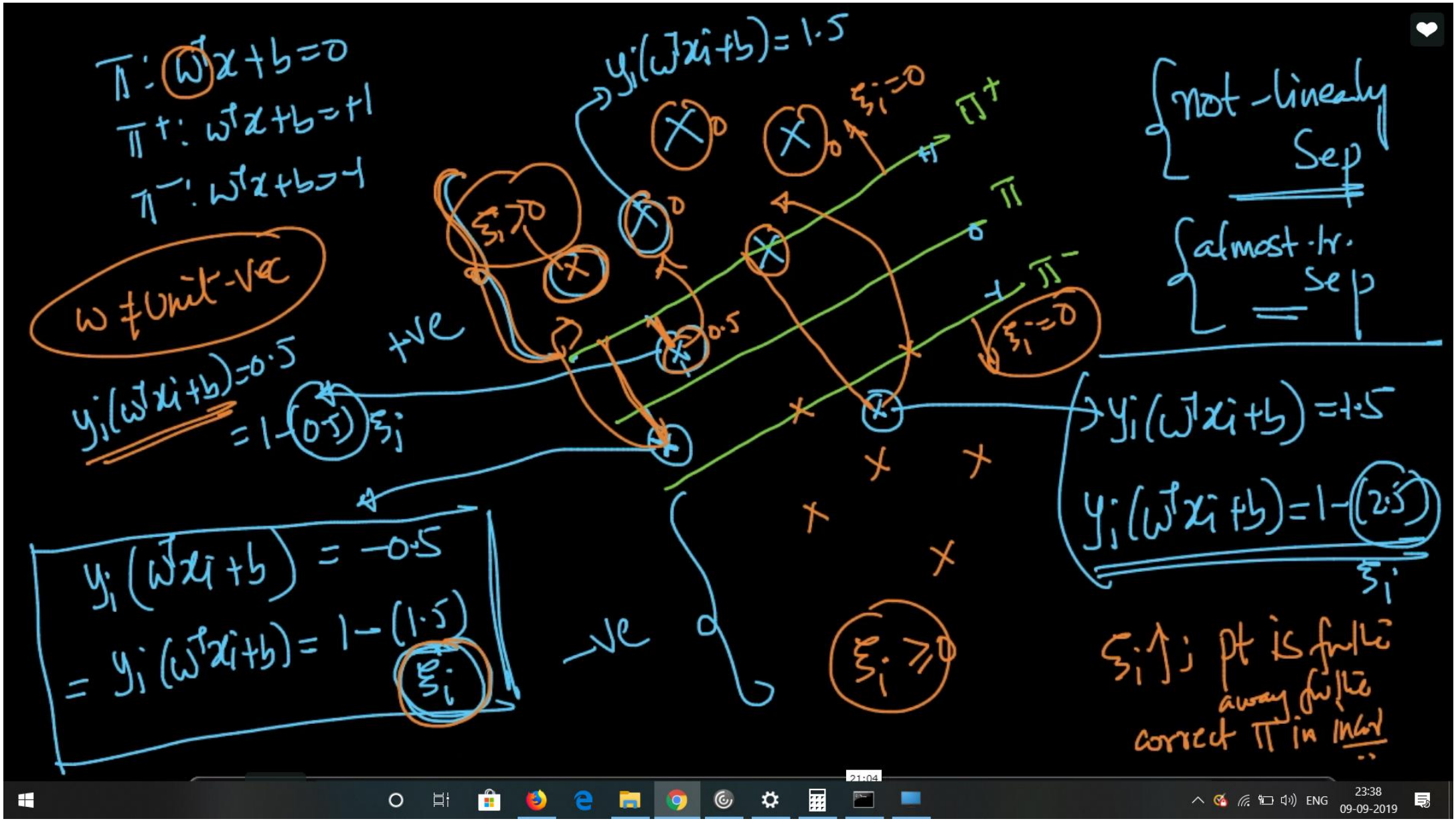
$$\vec{w}, b^* = \underset{w,b}{\operatorname{argmax}} \frac{2}{\|w\|}$$

$$\text{s.t} \quad \forall i, \quad y_i(\vec{w}^T x_i + b) \geq 1$$

data is linearly sep.




not-linearly Sep
almost-lr. sep
+ve
-ve
π^+
$\bar{\pi}$
π^-
$\bar{x}$


$\pi: \vec{w}^T x + b = 0$
$\pi^+: w^T x + b = +1$
$\pi^-: w^T x + b = -1$
$w \neq$ unit-vec
$y_i(w^T x_i + b) = 1.5$
$\xi_i = 0$
π^+
π
π^-
$\xi_i > 0$
+ve
$y_i(w^T x_i + b) = 0.5$
$= 1 - (0.5)\ \xi_i$
0.5
$\xi_i = 0$
$y_i(w^T x_i + b) = -0.5$
$= y_i(w^T x_i + b) = 1 - (1.5)\ \xi_i$
-ve
$\xi_i \geq 0$
$y_i(w^T x_i + b) = +1.5$
$y_i(w^T x_i + b) = 1 - (2.5)\ \xi_i$
$\xi_i \uparrow$; pt is further away from correct π in ingen..
not-linearly Sep
almost-lr. Sep

$$(w^*, b^*) = \arg\max_{w,b} \frac{2}{||w||} = \arg\min_{w,b} \frac{||w||}{2}$$

$$\max_x f(x) = \min_x \frac{1}{f(x)}$$

$$(w^*, b^*) = \underset{w,b}{\operatorname{argmin}} \left(\frac{\|w\|}{2} + C \cdot \frac{1}{n} \sum_{i=1}^{n} \xi_i \right)$$

1/margin

avg. dist of miss. pts from correct TLs

hyperparam

$$\text{s.t} \begin{cases} y_i (w^T x_i + b) \geq 1 - \xi_i & \forall i \\ \xi_i \geq 0 \end{cases}$$

corr. classif. pts $\xi_i = 0$

incorr. class. pts $\xi_i > 0$

Minimize errors $\equiv$ min misclassif.

$\equiv$

$$\min \sum \xi_i$$

$$(\vec{w}, b) = \underset{w,b}{\operatorname{argmin}} \frac{\|w\|}{2} + C \cdot \frac{1}{n} \sum_{i=1}^{n} \xi_i$$

$\frac{\|w\|}{2}$ → margin, reg

C → hyperparam

$\frac{1}{n} \sum_{i=1}^{n} \xi_i$ → loss → avg. dist fr miss pts

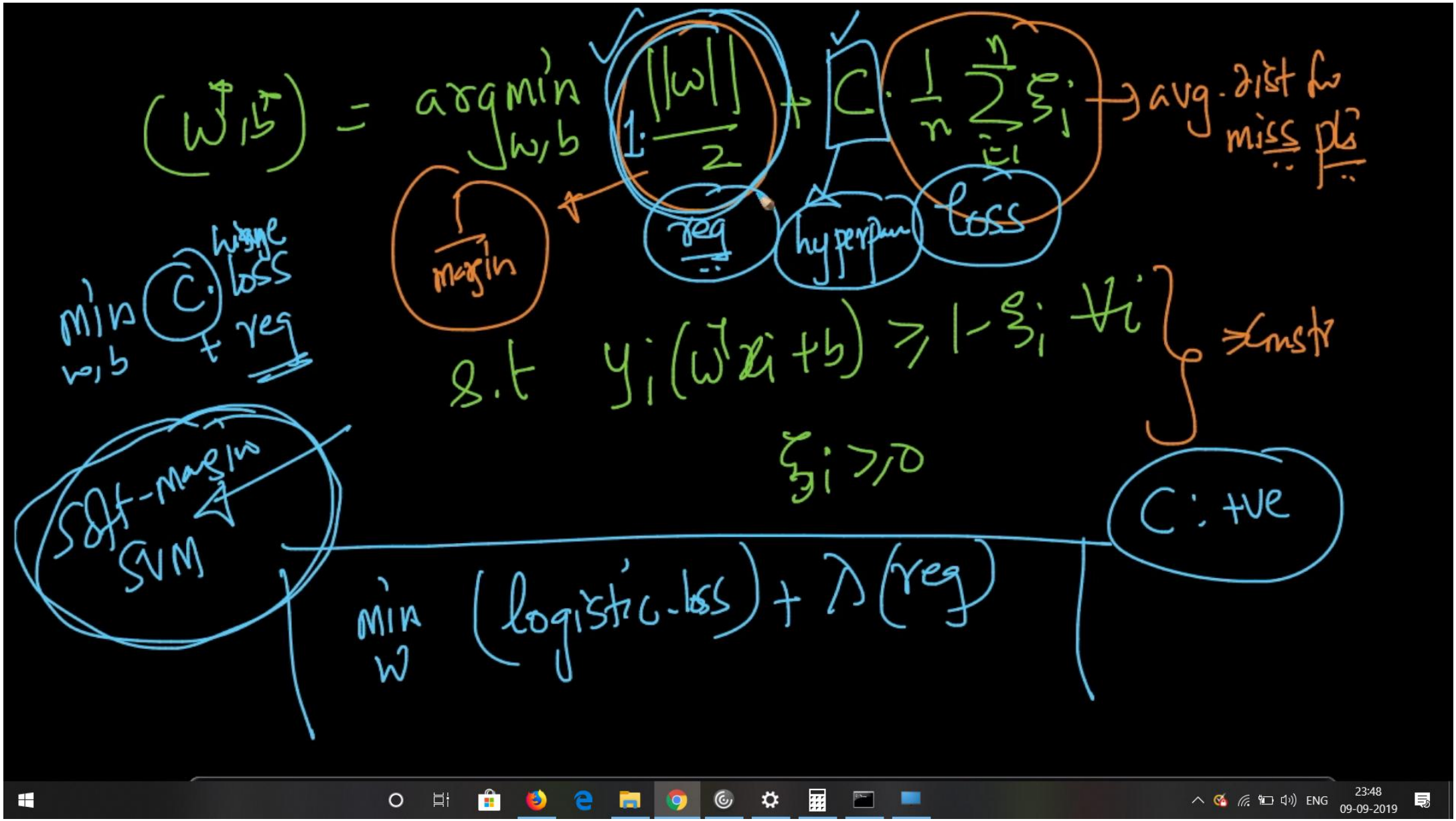
$$\min_{w,b} \; C \cdot \text{hinge loss} + \text{reg}$$

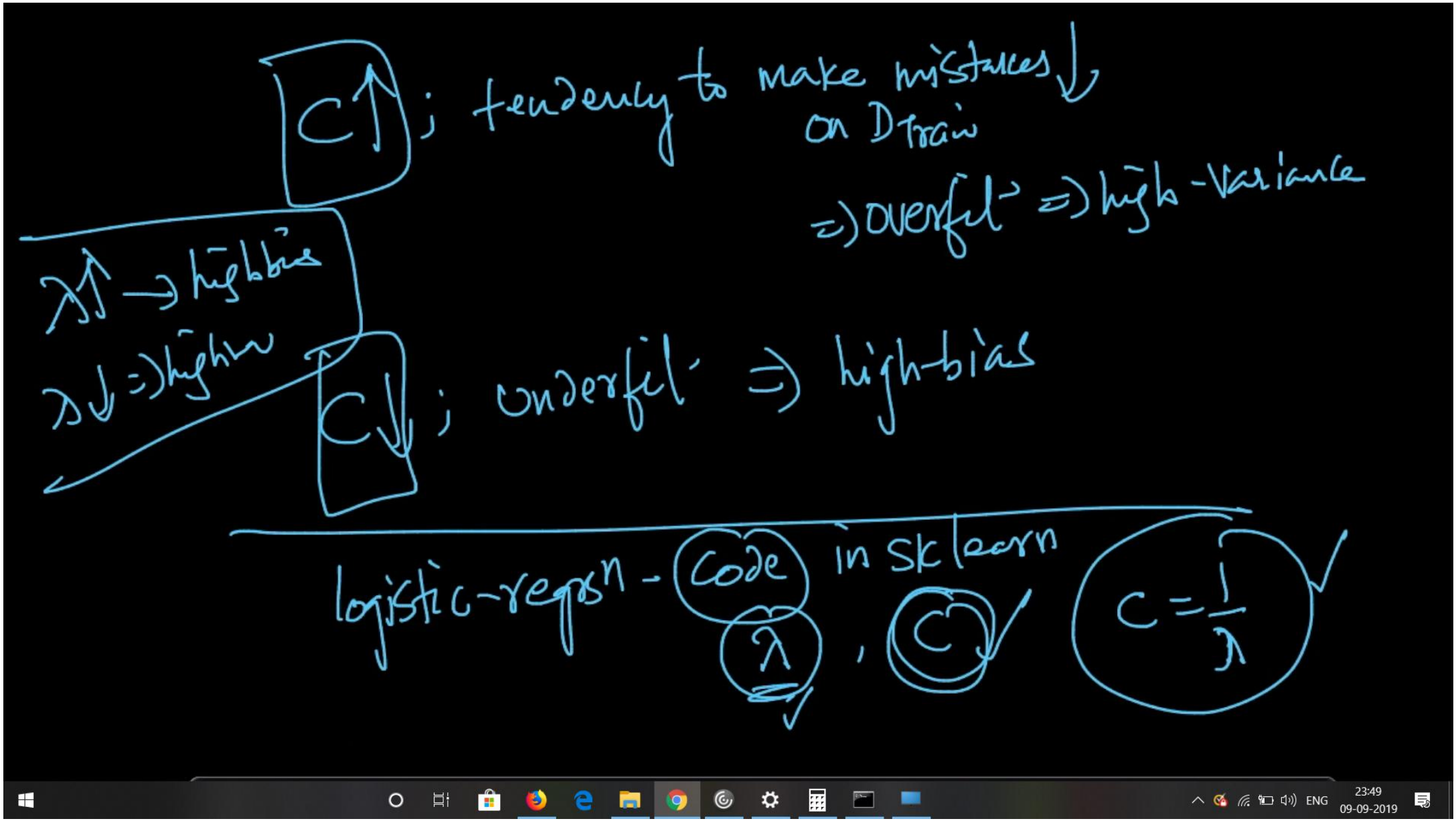
$$\text{s.t} \quad y_i(\vec{w}^T \vec{x}_i + b) \geq 1 - \xi_i \;\; \forall i$$

$$\xi_i \geq 0$$

C: +ve

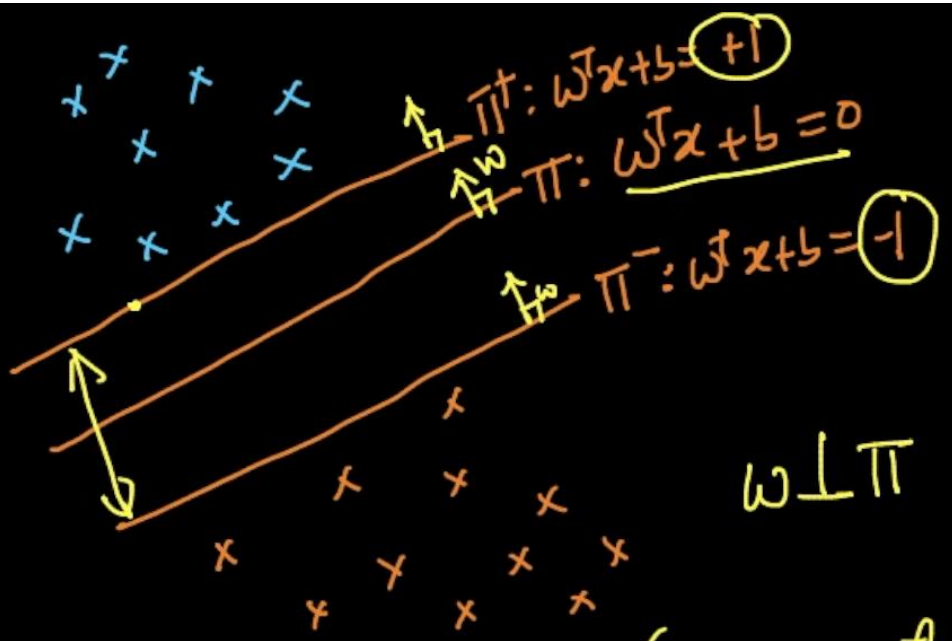
$$\min_{w} \; (\text{logistic-loss}) + \lambda(\text{reg})$$


$(\vec{w}, b) = \underset{w,b}{\operatorname{argmin}} \; 1:\frac{\|w\|}{2} + C \cdot \frac{1}{n}\sum_{i=1}^{n} \xi_i$
avg. dist for miss pts
margin
reg
hyperpm
loss
$\min_{w,b}$ C · hinge loss + reg
s.t $y_i(\vec{w}^T \vec{x}_i + b) \geq 1 - \xi_i \; \forall i$
constr
$\xi_i \geq 0$
Soft-margin SVM
C: +ve
$\min_{w}$ (logistic-loss) + λ (reg)


C↑ ; tendency to make mistakes ↓ on D_train
=) overfit =) high-variance
λ↑ → high bias
λ↓ =) high var
C↓ ; underfit ⇒ high-bias
logistic-regn - code in sklearn
λ , C
C = 1/λ

SVM:

(Q) Why +1 & -1 on the RHS of $\pi^+ \& \pi^-$



Margin :- $\frac{2}{\|w\|}$

$w^*, b^* = \underset{w,b}{\arg\max} \frac{2}{\|w\|}$

constants

$\{ \|w\| \neq 1$ (any vectors need not be unit-vectors)

(1) $\pi^+ :- \quad w^T x + b = k$

$\pi^- :- \quad w^T x + b = -k$

$k > 0$

k : Constant

let $k = 4$

margin :- $\frac{2k}{\|w\|}$

$\underset{w,b}{\text{argmax}} \; \frac{2}{\|w\|} = \underset{w,b}{\text{argmax}} \; \frac{2k}{\|w\|} = \frac{8}{\|w\|}$

(2) π :- $w^T x + b = k$

$w \perp \pi$

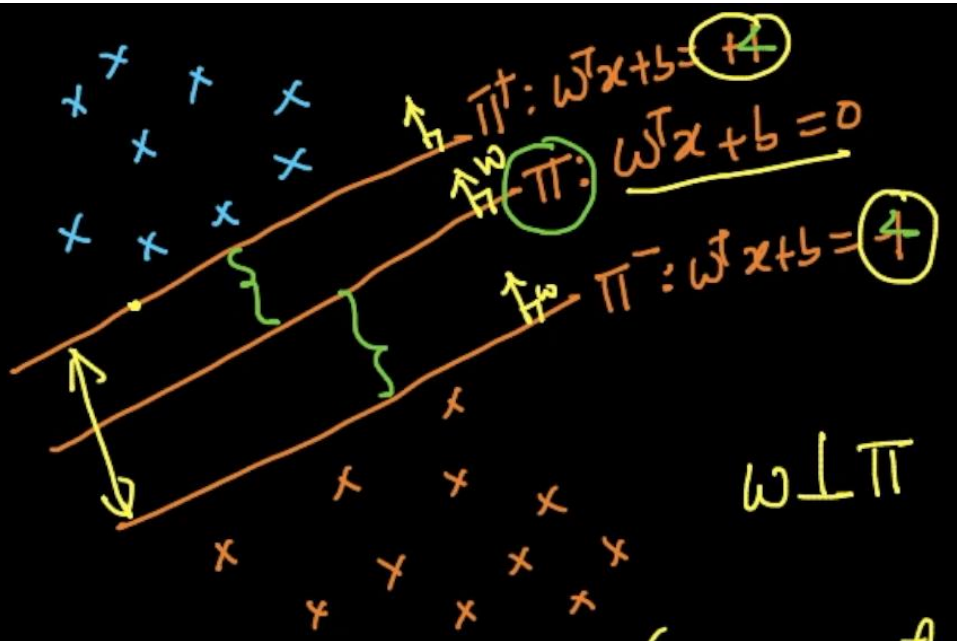
$\left(\frac{w}{k}\right)^T x + \left(\frac{b}{k}\right) = 1$

$||w||$ need not be 1

$(w')^T x + b' = 1$

SVM:

(Q) Why $+1$ & -1 on the RHS of $\pi^+ \& \pi^-$? (convinience)



$w \perp \pi$

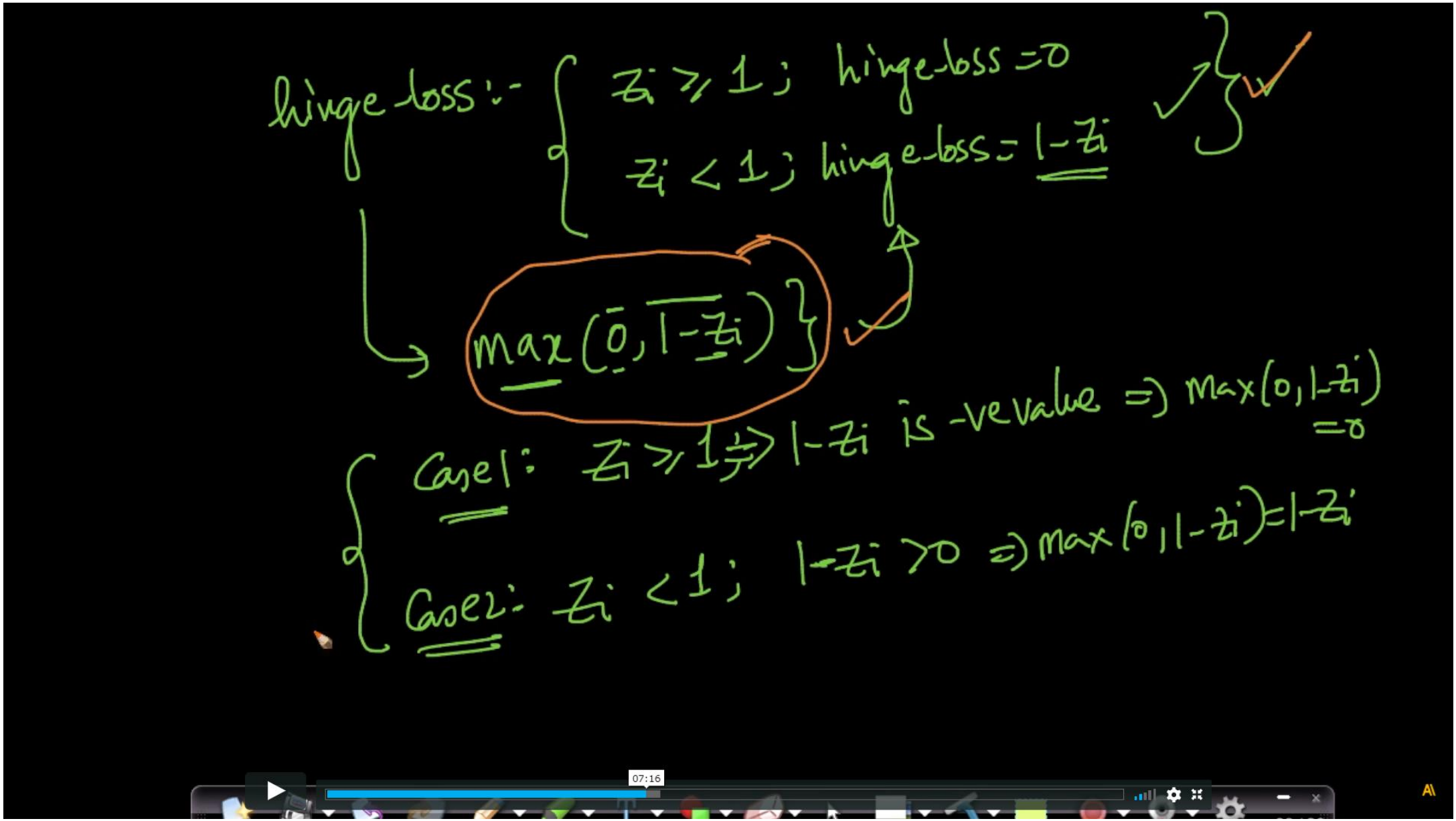
Margin :- $\frac{2}{\|w\|}$

$w^*, b^* = \underset{w,b}{\arg\max} \frac{2}{\|w\|}$

constants

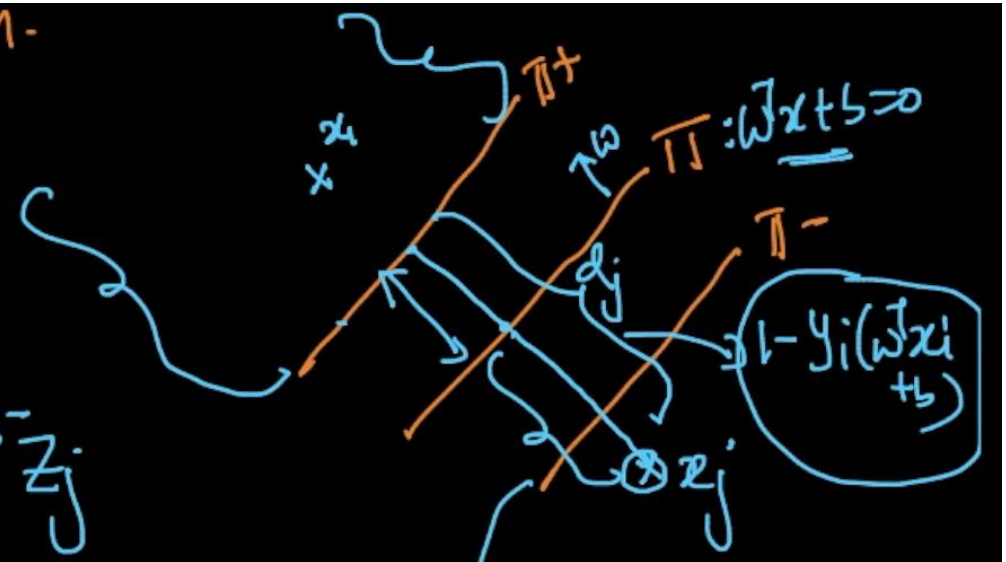
$\underset{w}{\text{argmax}} \frac{2}{\|w\|}$ $\Rightarrow \|w\| \neq 1$ (any vector need not be unit-vector)

Can be, anything


hinge-loss :-
zi ≥ 1 ; hinge-loss = 0
zi < 1 ; hinge-loss = 1 - zi
max(0, 1 - zi)
Case1: zi ≥ 1 ⇒ 1 - zi is -ve value ⇒ max(0, 1 - zi) = 0
Case2: zi < 1 ; 1 - zi > 0 ⇒ max(0, 1 - zi) = 1 - zi
07:16

geom - formulation & loss - min.

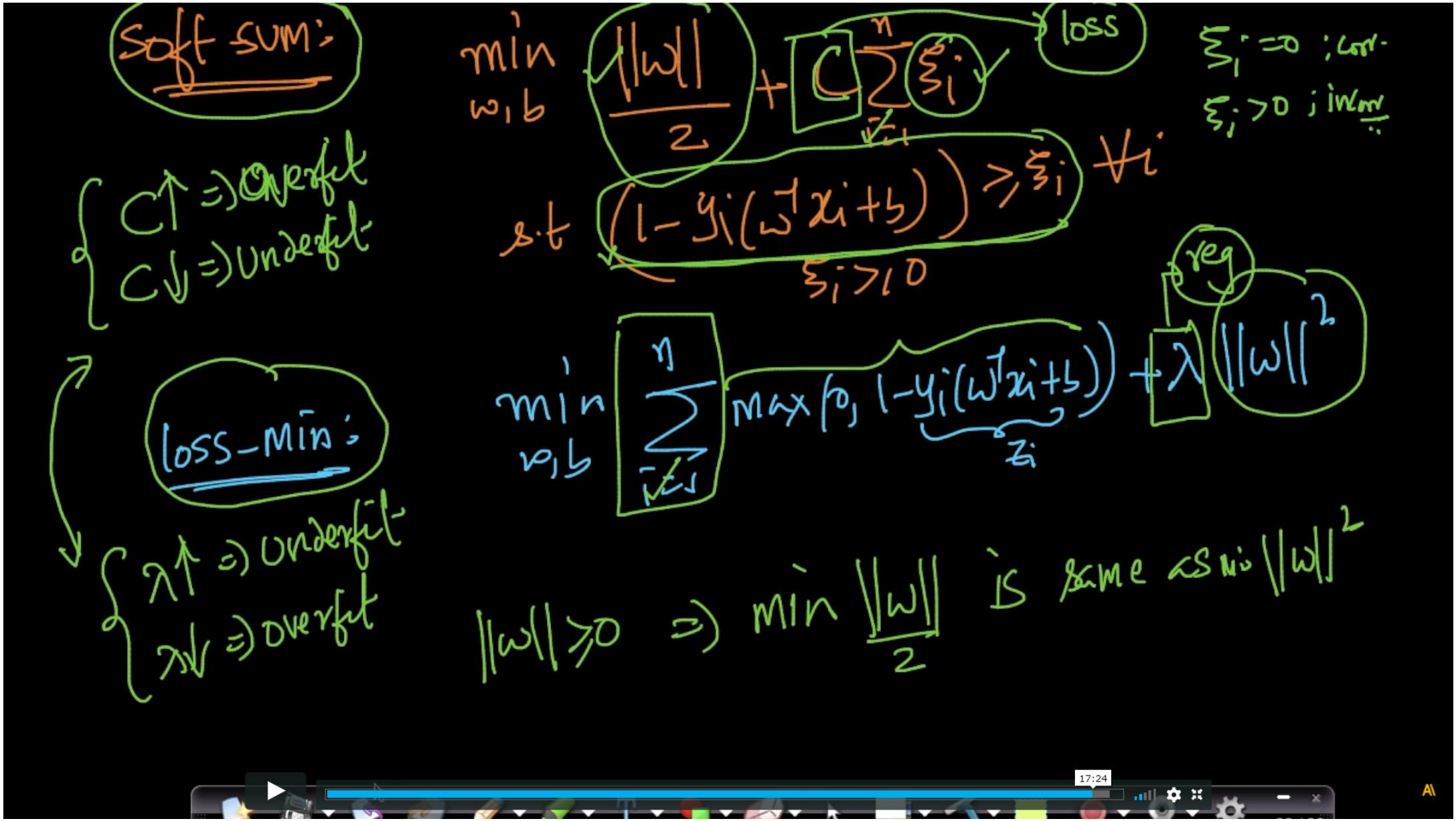
Corr classifi pt → $\xi_i = 0$ ← x_i



$$d_j = 1 - \underbrace{y_i(\vec{w}^T x_i + b)}_{z_j} = 1 - z_j$$

ξ_j = dist fr x_j to the $\pi^+ = d_j = 1 - z_j$

$\xi_j = 1 - z_j$ → when x_j is misclassified


Soft sum:
min w,b ||w||/2 + C Σ ξi
loss
ξi = 0 ; corr.
ξi > 0 ; incorr.
s.t (1 − yi(wᵀxi + b)) ≥ ξi ∀i
ξi ≥ 0
C↑ ⇒ overfit
C↓ ⇒ underfit
loss-Min:
min w,b Σ max(0, 1 − yi(wᵀxi + b)) + λ ||w||²
zi
reg
λ↑ ⇒ underfit
λ↓ ⇒ overfit
||w|| ≥ 0 ⇒ min ||w||/2 is same as min ||w||²
17:24

Dual form of SVM:

soft-margin SVM

$$\min_{w,b} \ \frac{1}{2}\|w\| + C\sum_{i=1}^{n}\xi_i$$

$$\text{s.t } y_i(w^T x_i + b) \geq 1-\xi_i \ \forall i$$

$$\xi_i \geq 0$$

Equivalent

Dual

$$\max_{\alpha_i} \ \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j (x_i^T x_j)$$

$$\text{s.t } \alpha_i \geq 0$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

④ $\alpha_i > 0$ only for SVs

$\alpha_i = 0$ for non-SVs

Primal of SVM

$$\checkmark \ z_q := \underbrace{\text{sign}(w^T x_q + b)}_{y_q} = f(x_q)$$

① $x_i \rightarrow \alpha_i$

② x_i's only occur in the form of $x_i^T x_j$

③ $f(x_q) = \sum_{i=1}^{n}\alpha_i y_i x_i^T x_q + b$

→ The most imp. idea in SVM is kernel-trick

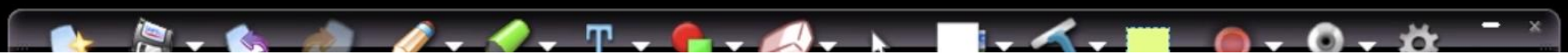
Soft-SVM-hyperplanes ≈ log-reg

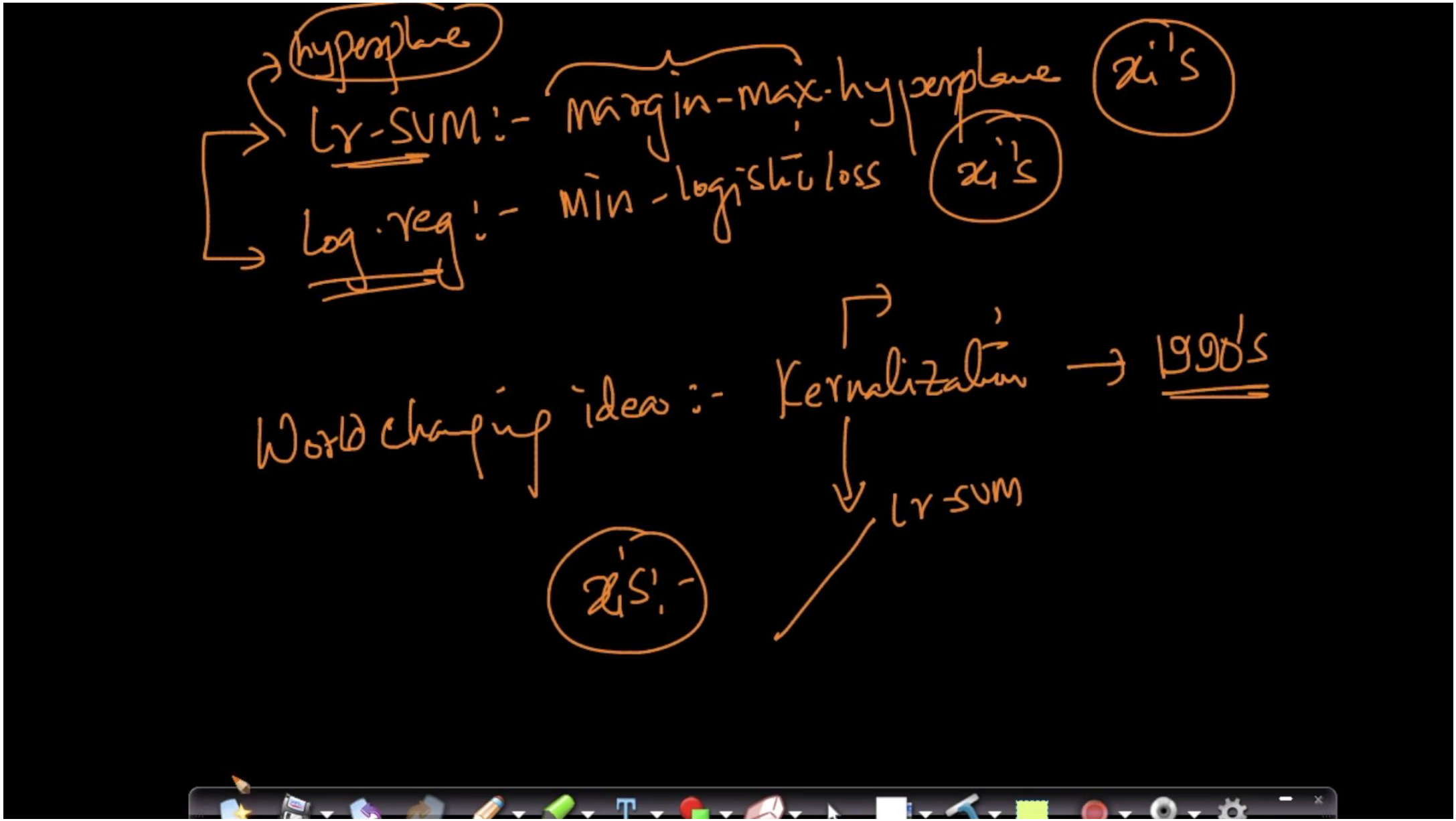
↳ margin-max.

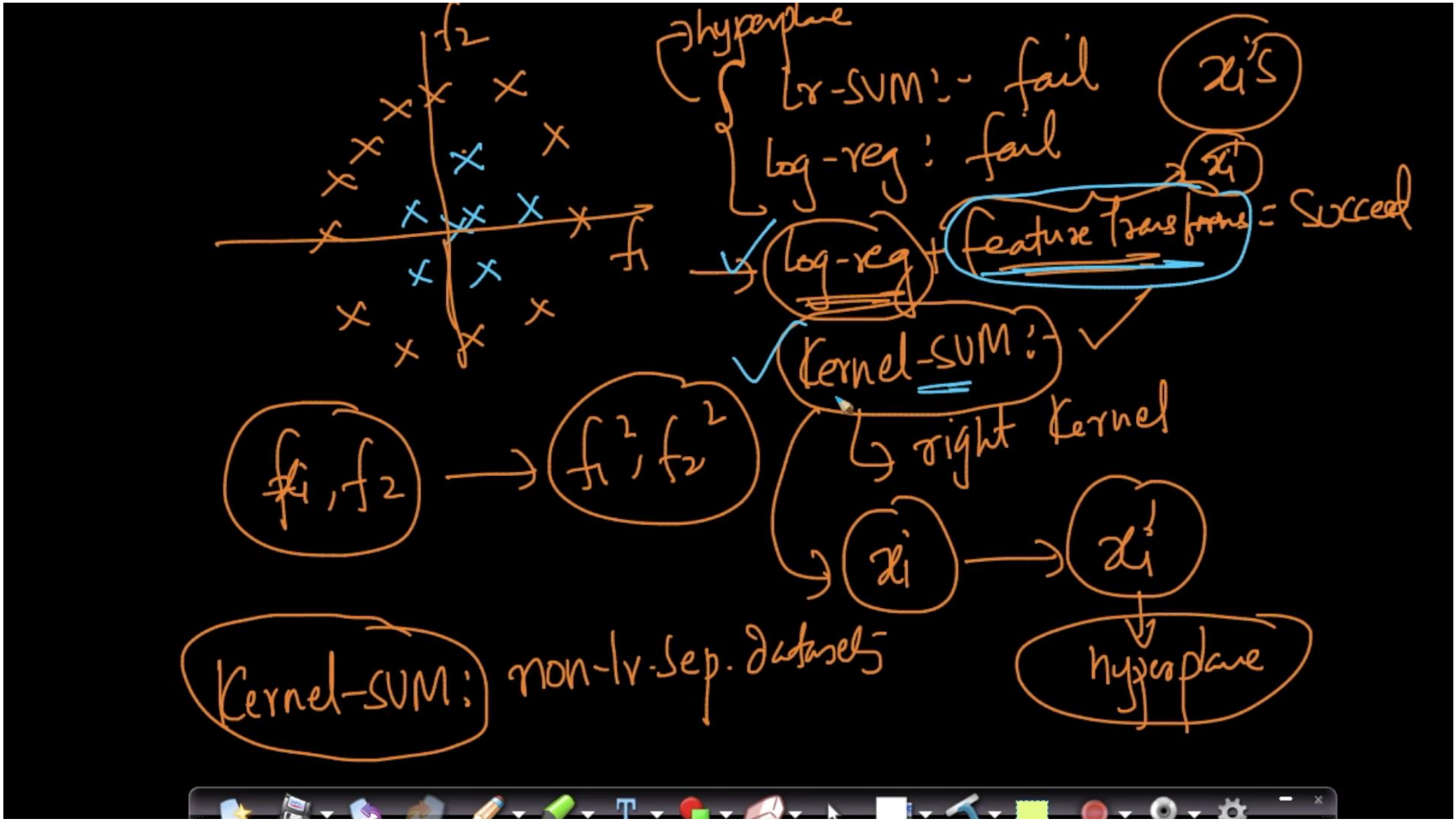
Lr-SVM:- $x_i^T x_j$

Kernel-SVM:- $K(x_i, x_j)$

$K(x_i, x_j) = x_i^T x_j$



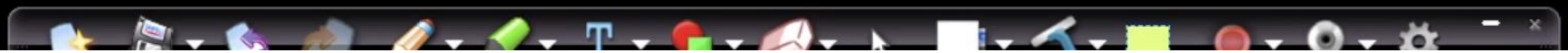

hyperplane
Lr-SVM :- margin-max-hyperplane
xi's
Log.reg :- min-logistic loss
xi's
World changing idea :- Kernalization → 1990's
Lr-SVM
xi's :-


f2
f1
hyperplane
lr-SVM :- fail
log-reg : fail
xi's
xi'
log-reg
feature Transformation = Succeed
Kernel-SVM :
right Kernel
f1, f2
f1², f2²
xi
xi'
Kernel-SVM : non-lr-sep. datasets
hyperplane

Kernalization :- SUM handle non-lr.sep. datasets

Kernel + logistic - regrsn

$x_i^T x_j \rightarrow K(x_i, x_j)$

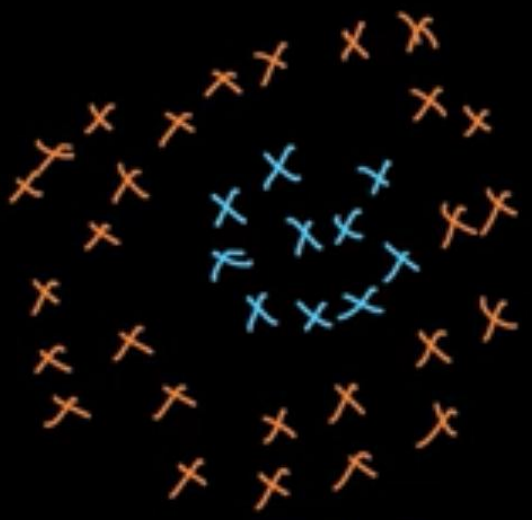


Polynomial Kernel

→ Kernelization

$$K(x_1, x_2) = (x_1^T x_2 + c)^d$$

(e.g) $K(x_1, x_2) = \underbrace{(1 + x_1^T x_2)^2}_{\text{Quadratic Kernel}}$



$(f_1, f_2) \xrightarrow{f_1} (f_1^2, f_2^2)$

↓

log-reg

$$K(x_1, x_2) = (1 + x_1^T x_2)^2$$

$$= (1 + x_{11}x_{21} + x_{12}x_{22})^2$$

$x_1 = \langle x_{11}, x_{12} \rangle$ 2D

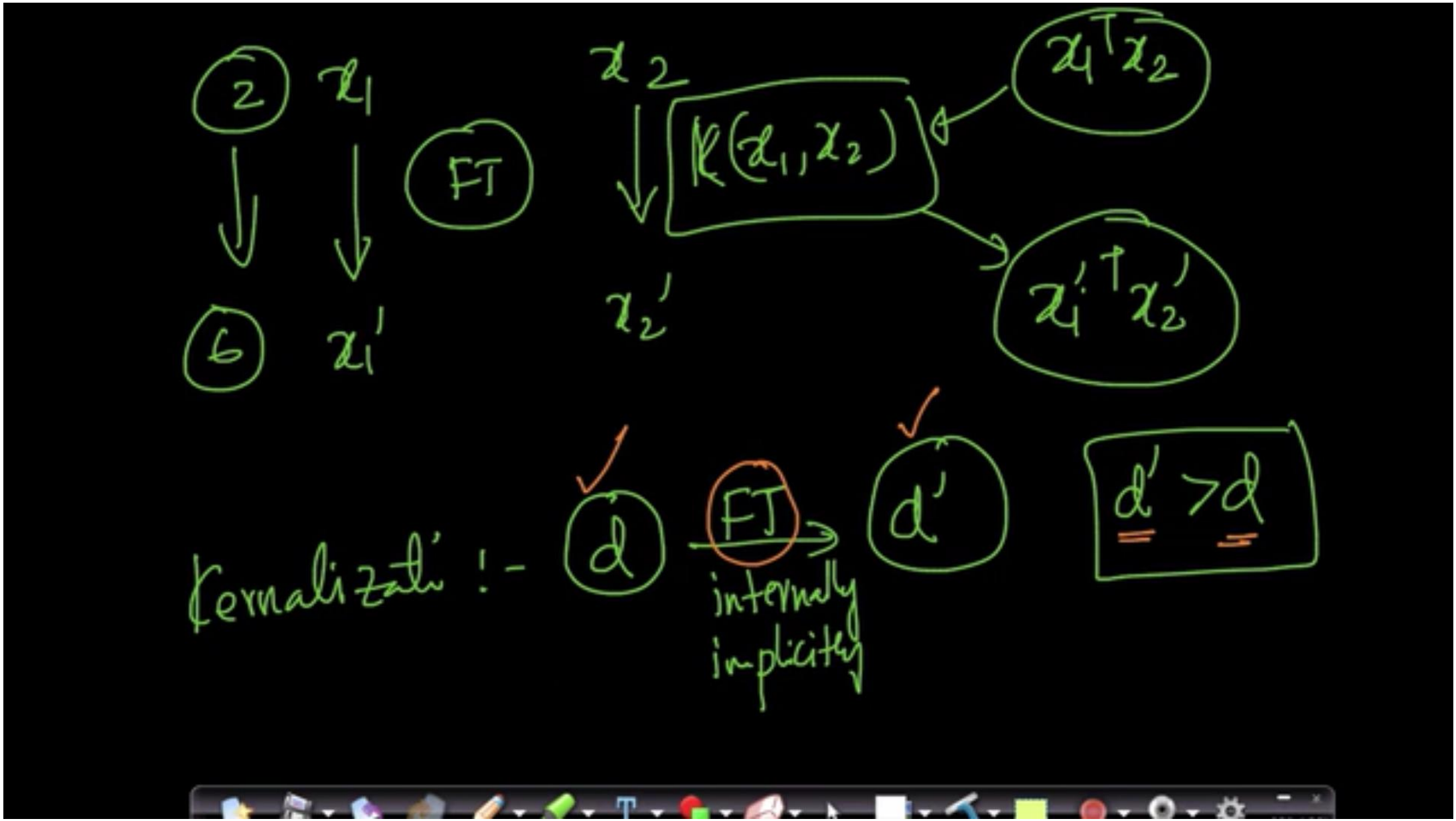
$x_2 = \langle x_{21}, x_{22} \rangle$ 2D

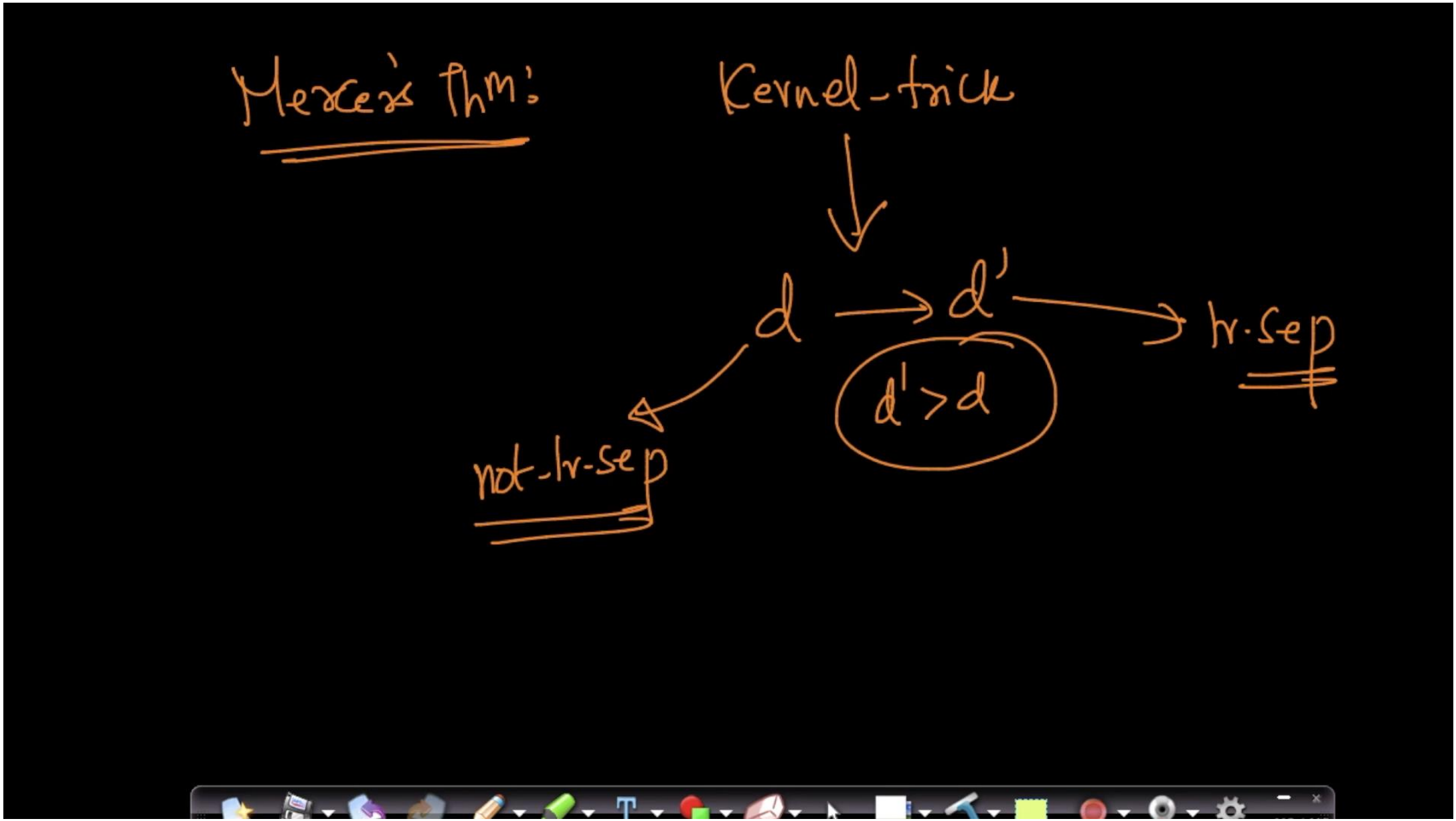
$$= 1 + x_{11}^2 x_{21}^2 + x_{12}^2 x_{22}^2 + 2x_{11}x_{21} + 2x_{12}x_{22} + 2x_{11}x_{21}x_{12}x_{22}$$

Let $[1, x_{11}^2, x_{12}^2, \sqrt{2}x_{11}, \sqrt{2}x_{12}, \sqrt{2}x_{11}x_{12}] : x_1'$

$[1, x_{21}^2, x_{22}^2, \sqrt{2}x_{21}, \sqrt{2}x_{22}, \sqrt{2}x_{21}x_{22}] : x_2'$

$$= (x_1')^T (x_2')$$


2
6
FT
x_1
x_1'
x_2
x_2'
$K(x_1, x_2)$
$x_1^T x_2$
$x_1'^T x_2'$
Kernalization :-
d
FT
internally
implicitly
d'
d' > d


Mercer's thm:
Kernel-trick
d
d'
d'>d
hr.sep
not_hr-sep

Polynomial Kernel

Kernelization → implicit FT

$K(x_1, x_2) = (x_1^T x_2 + c)^d$

(e.g) $K(x_1, x_2) = (1 + x_1^T x_2)^2$

Quadratic Kernel

bias-Var ↔ C

$(f_1, f_2) \xrightarrow{FT} (f_1^2, f_2^2)$

explicit

log-reg

Polynomial Kernel

Kernelization → implicit FT

$K(x_1, x_2) = (x_1^T x_2 + c)^d$

(e.g) $K(x_1, x_2) = (1 + x_1^T x_2)^2$

Quadratic Kernel

$(f_1, f_2) \xrightarrow{FT} (f_1^2, f_2^2)$

explicit

log-reg

Radial Basis Function (RBF)

SVM:- most popular / general-purpose : RBF

(x_1, x_2)

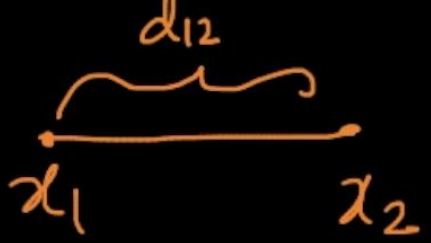
$$K_{RBF}(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right)$$

σ^2 ← hyperparam

Soft-margin Kernel SVM

RBF Kernel

→ C : hyperparam



$$\|x_1 - x_2\|^2 = d_{12}^2$$

$$K(x_1, x_2) = \exp\left(-\frac{d_{12}^2}{2\sigma^2}\right) \qquad d_{12} = \|x_1 - x_2\|_2$$

$$\frac{1}{e^{d^2/2\sigma^2}}$$

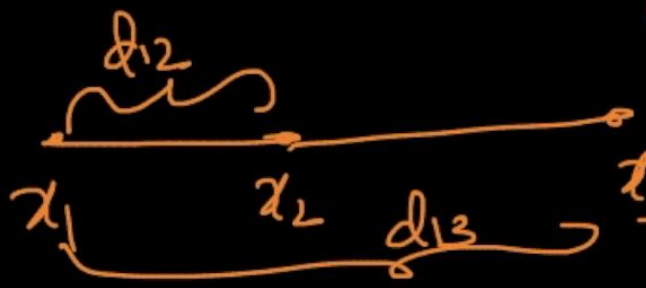
$d \uparrow ; d^2 \uparrow$

$e^{d^2} \uparrow$

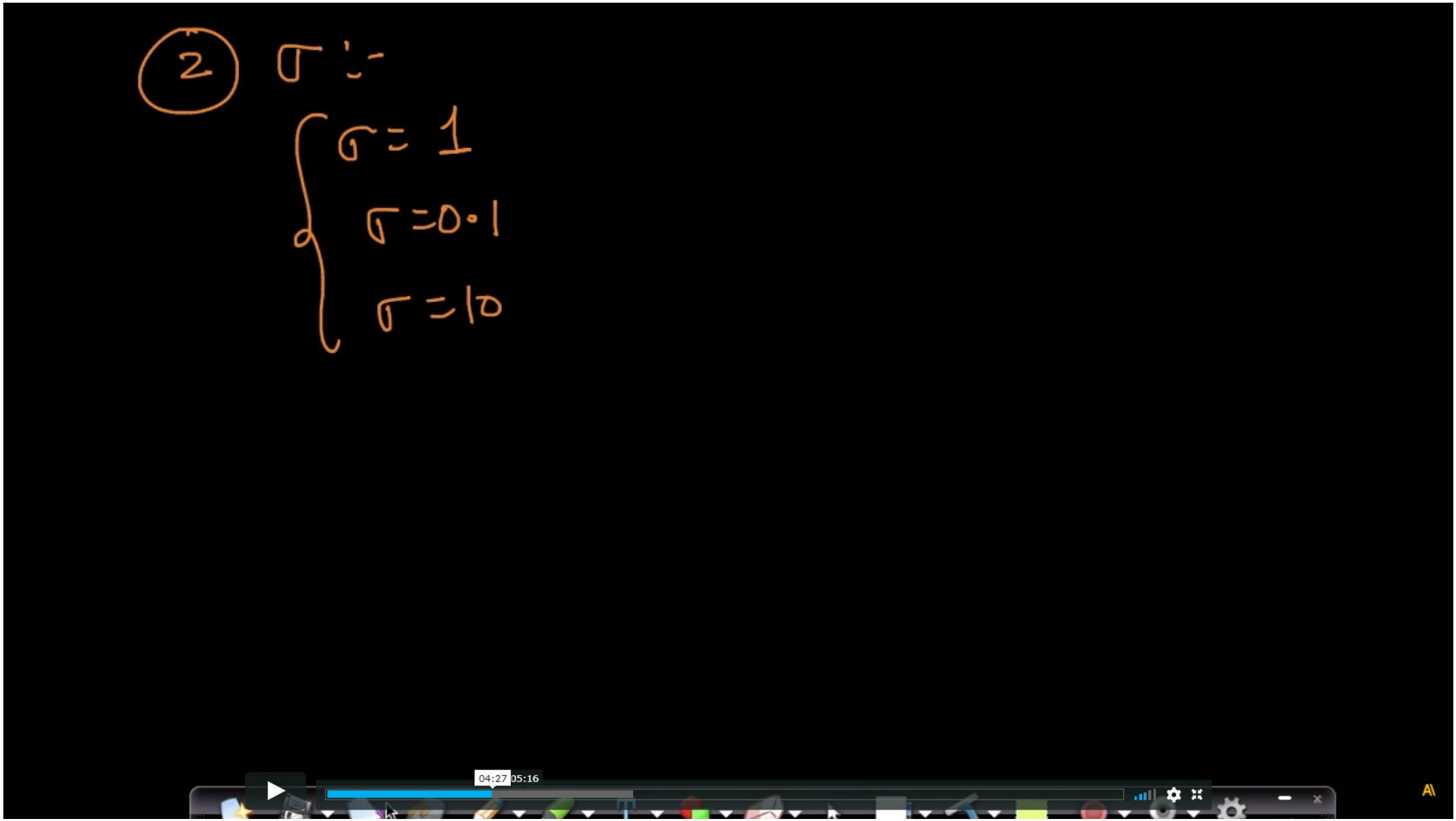
$\frac{1}{e^{d^2}} \downarrow$

① $d_{12} \uparrow$; $K(x_1, x_2) \downarrow$

Similarity



$K(x_1, x_2) > K(x_1, x_3)$


2
σ :-
σ = 1
σ = 0.1
σ = 10
04:27
05:16

Contents - Google Docs | plot(exp(-x^2/(2*1)) - Google S | plot(exp(-x^2/(2*0.01)) - Goog | plot(exp(-x^2/(2*100)) - Googl | Chekuri Srikan...

Secure | https://www.google.co.in/search?ei=Jw03WuWqGcmLvQTi1oSwBg&q=plot%28exp%28-x%5E2%2F%282*1%29%29&oq=plot%28exp...

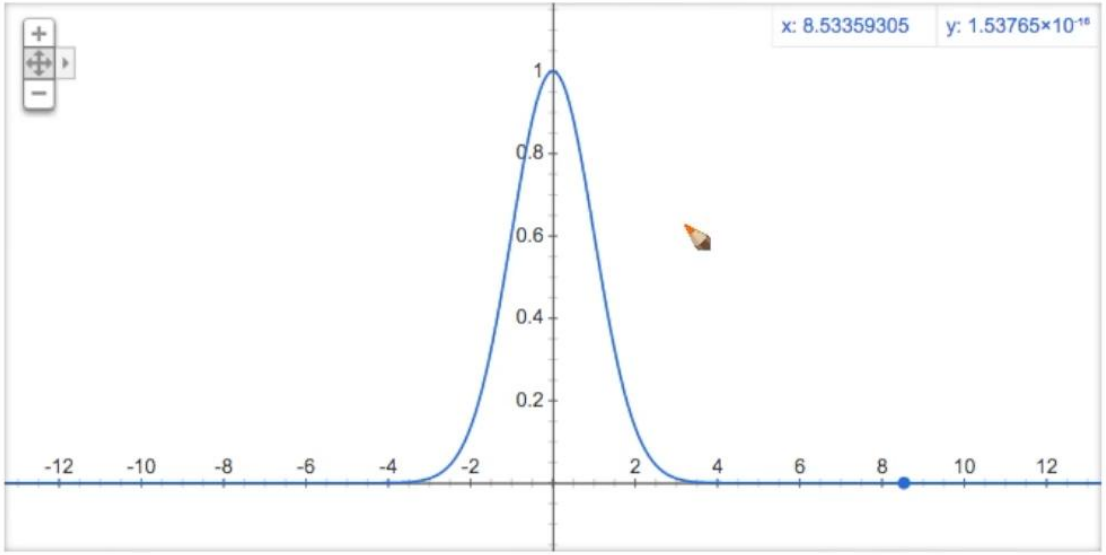
GOOGLE

plot(exp(-x^2/(2*1))

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

About 7,26,00,000 results (0.86 seconds)

Graph for exp((-(x^2))/(2*1))



More info

plot x e^-x, x^2 e^-x, x=0 to 8 - WolframAlpha

07:44

APPLIED COURSE

Contents - Google Docs | plot(exp(-x^2/(2*1)) - Google S | plot(exp(-x^2/(2*0.01)) - Goog | plot(exp(-x^2/(2*100)) - Googl | Chekuri Srikan...

Secure | https://www.google.co.in/search?ei=Jw03WuWqGcmLvQTi1oSwBg&q=plot%28exp%28-x%5E2%2F%282*1%29%29&oq=plot%28exp...

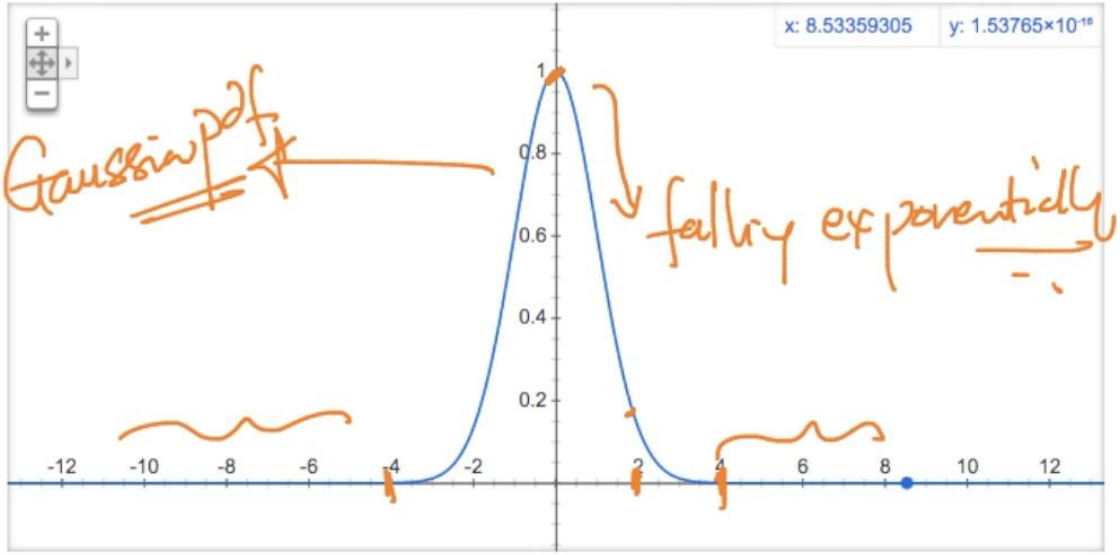
GOOGLE

plot(exp(-x^2/(2*1))

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

About 7,26,00,000 results (0.86 seconds)

Graph for exp((-(x^2))/(2*1))



More info

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

$\sigma = 1$;

RBF ~ gaussian PDF

Contents - Google Docs | plot(exp(-x^2/(2*1)) - Google S | plot(exp(-x^2/(2*0.01)) - Goog | plot(exp(-x^2/(2*100)) - Googl | Chekuri Srikan...

Secure | https://www.google.co.in/search?ei=MQ03Ws2jBoH6vgSBoI3YCw&q=plot%28exp%28-x%5E2%2F%282*0.01%29%29&oq=plot%28e...

GOOGLE

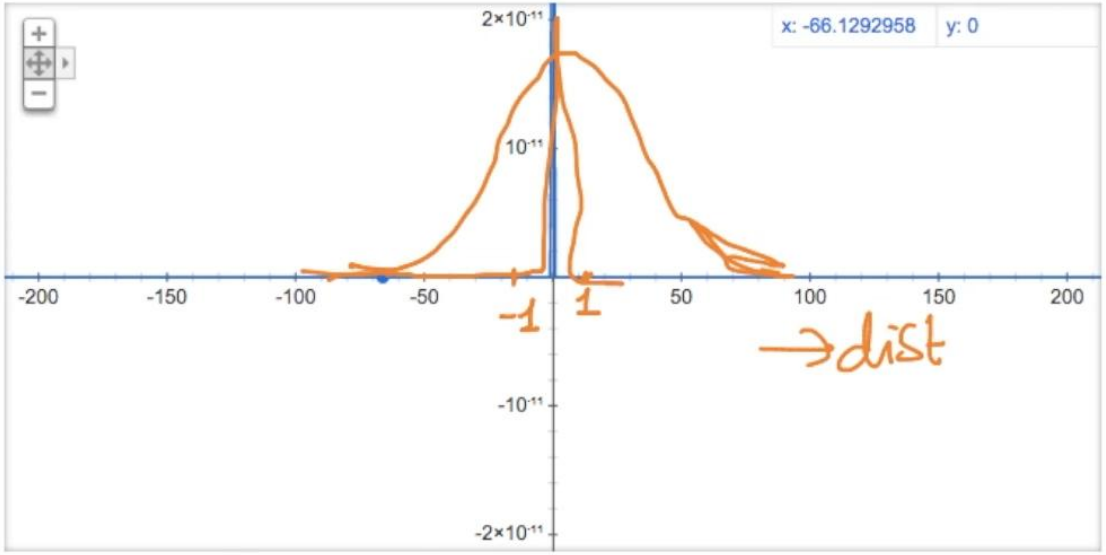
plot(exp(-x^2/(2*0.01))

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

$\sigma = 1$ to $\sigma = 0.1$

About 2,30,00,000 results (0.51 seconds)

Graph for exp((-(x^2))/(2*0.01))



More info

$\sigma = 0.1$

$\sigma^2 = 0.01$

dist > 1; k = 0

y = e^(-x^2) - MATLAB Answers - MATLAB Central - MathWorks

$$K(x_1, x_2) = \exp\left(-\frac{d_{12}^2}{2\sigma^2}\right) \qquad d_{12} = \|x_1 - x_2\|_2$$

$$\frac{1}{e^{d^2/2\sigma^2}}$$

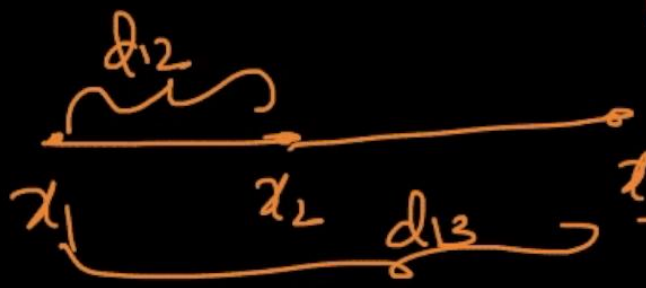
$d \uparrow ; d^2 \uparrow$

$e^{d^2} \uparrow$

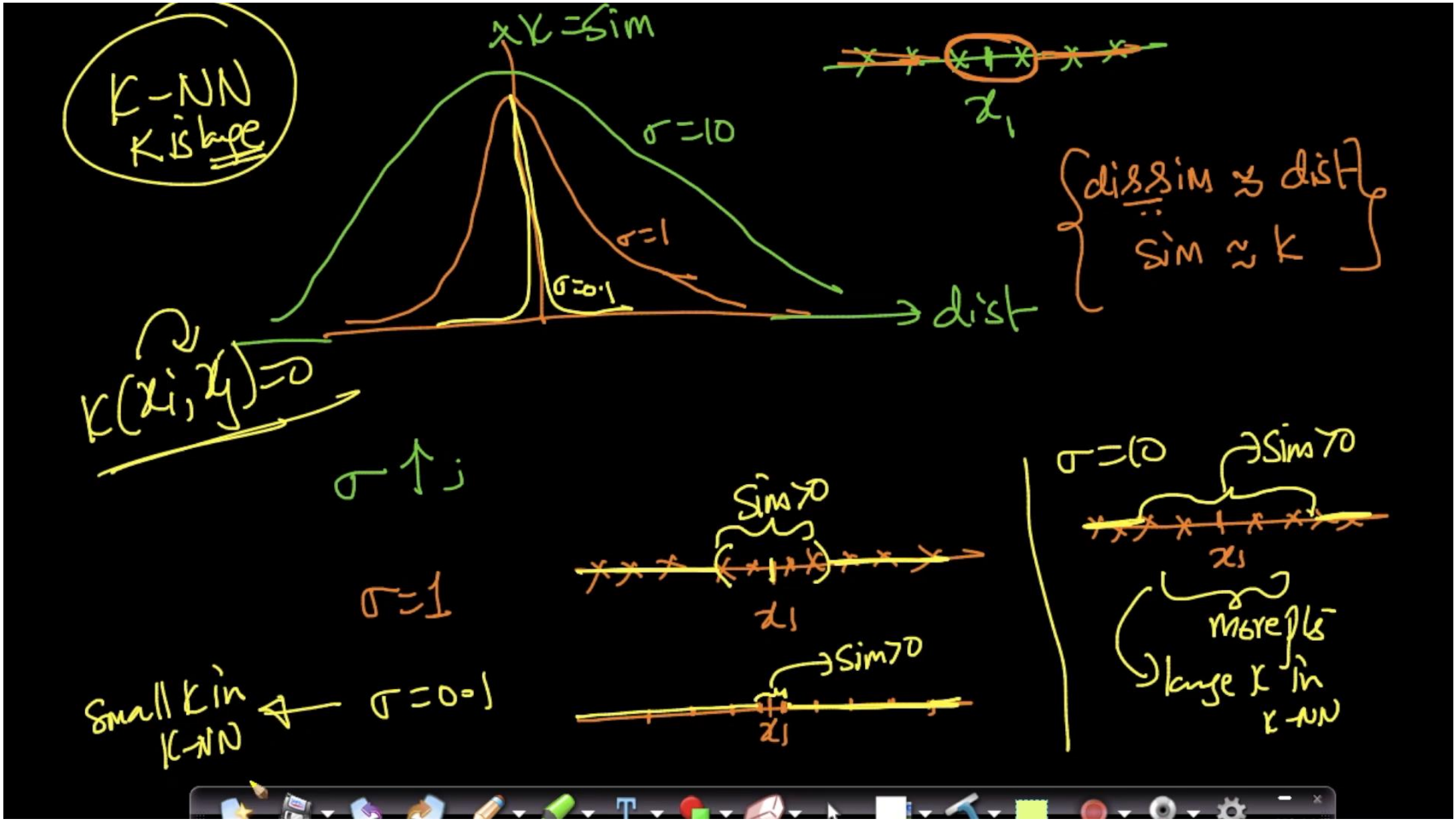
$\frac{1}{e^{d^2}} \downarrow$

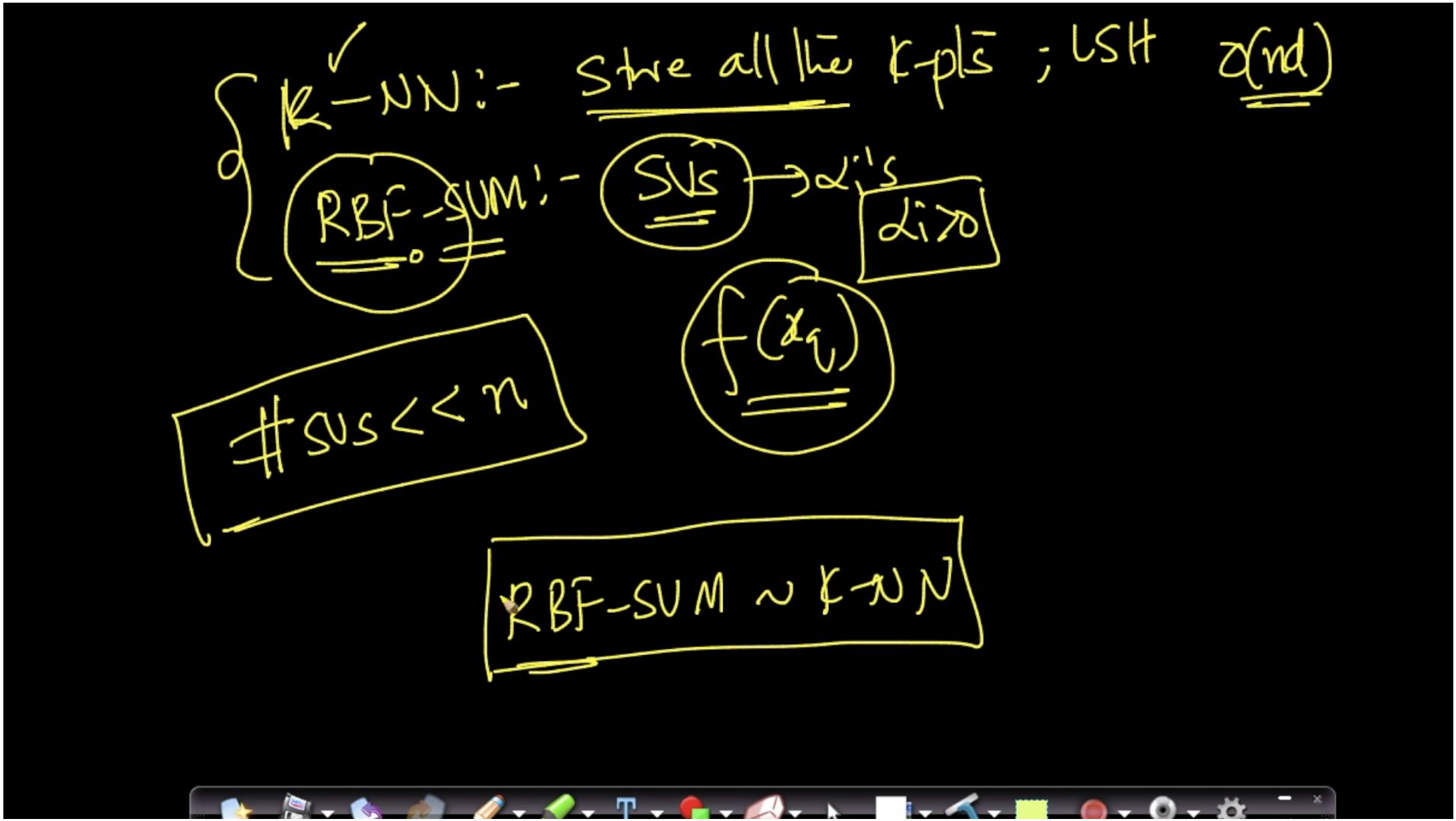
① $d_{12} \uparrow$; $K(x_1, x_2) \downarrow$

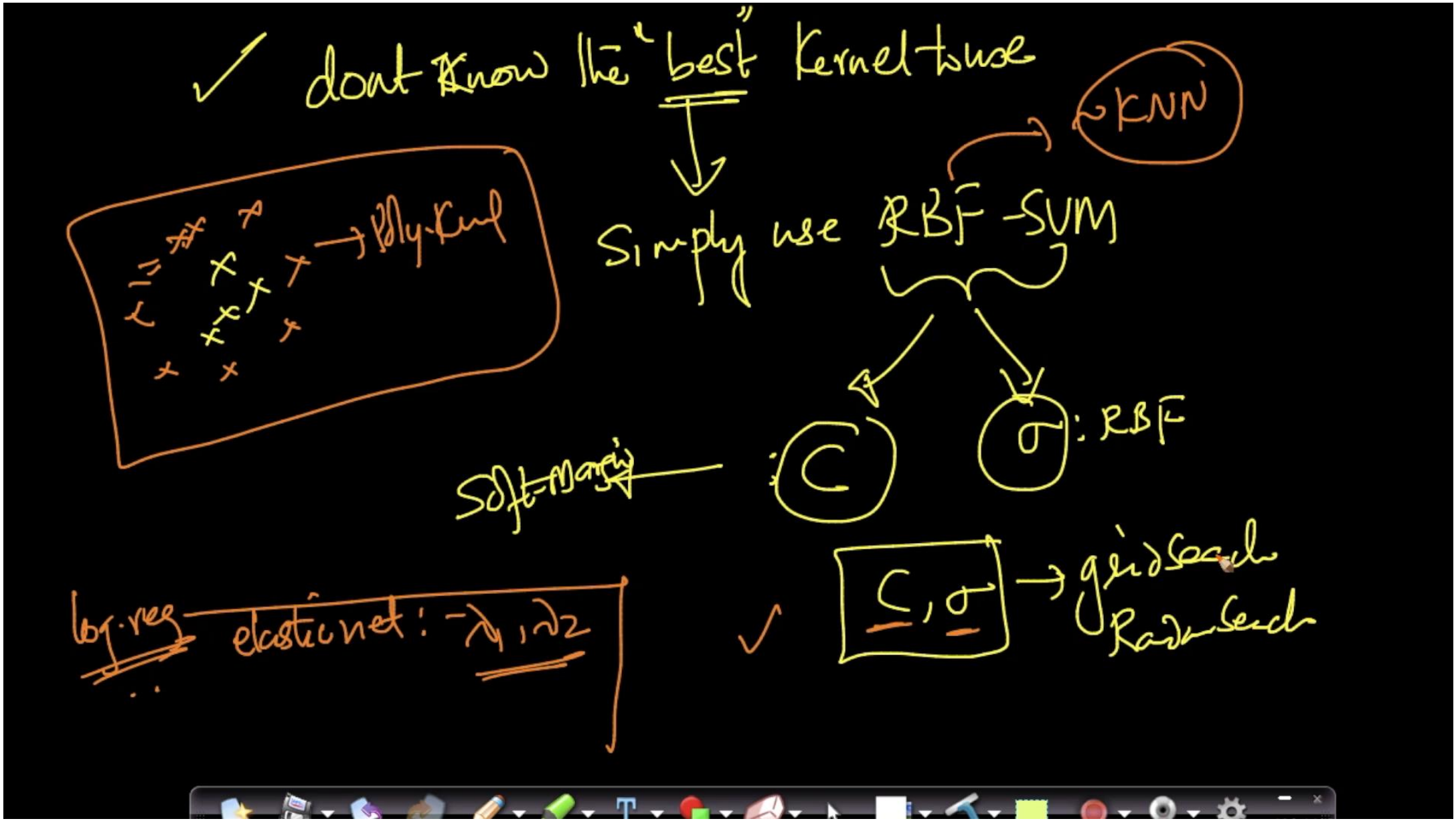
Similarity

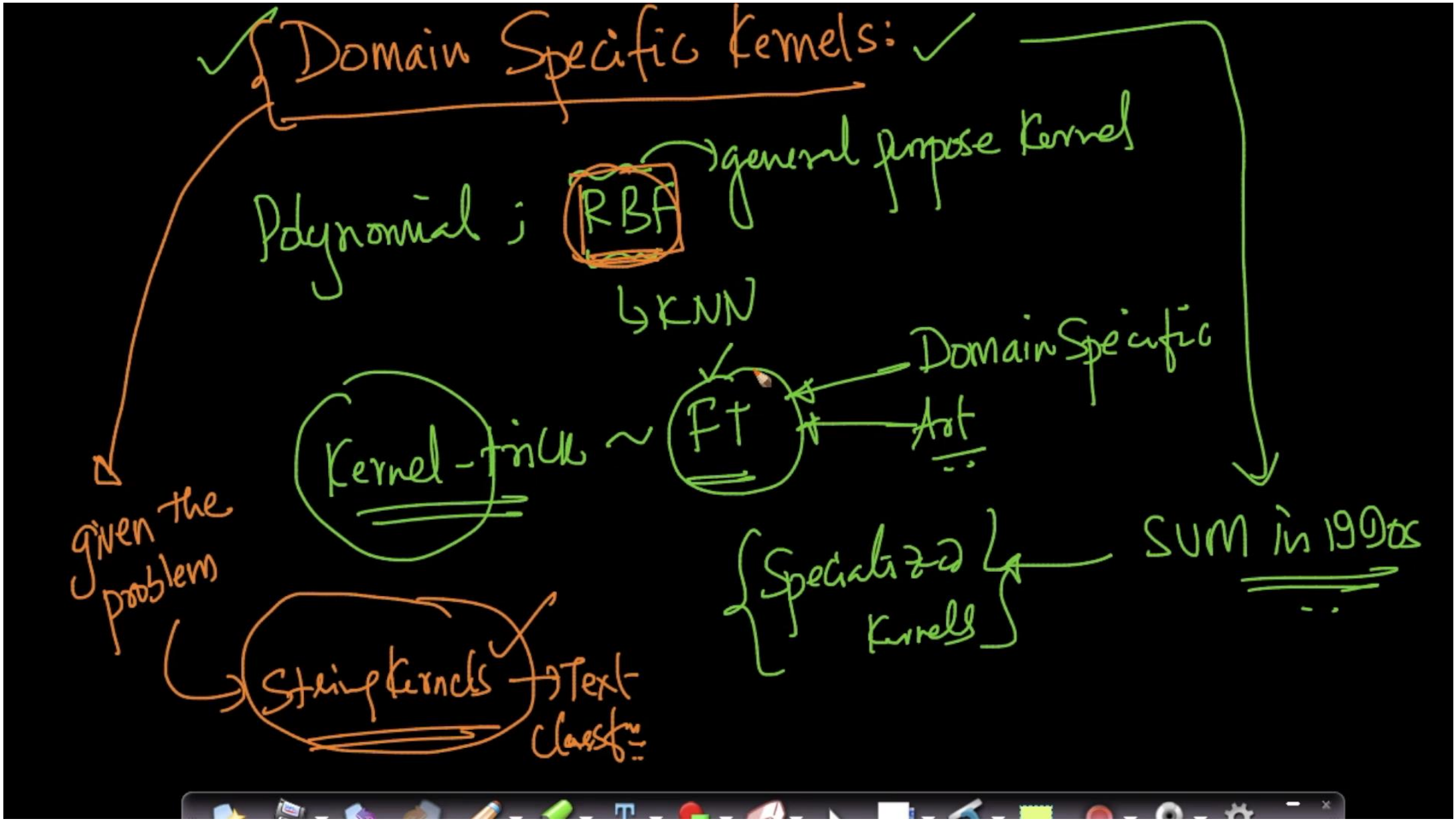


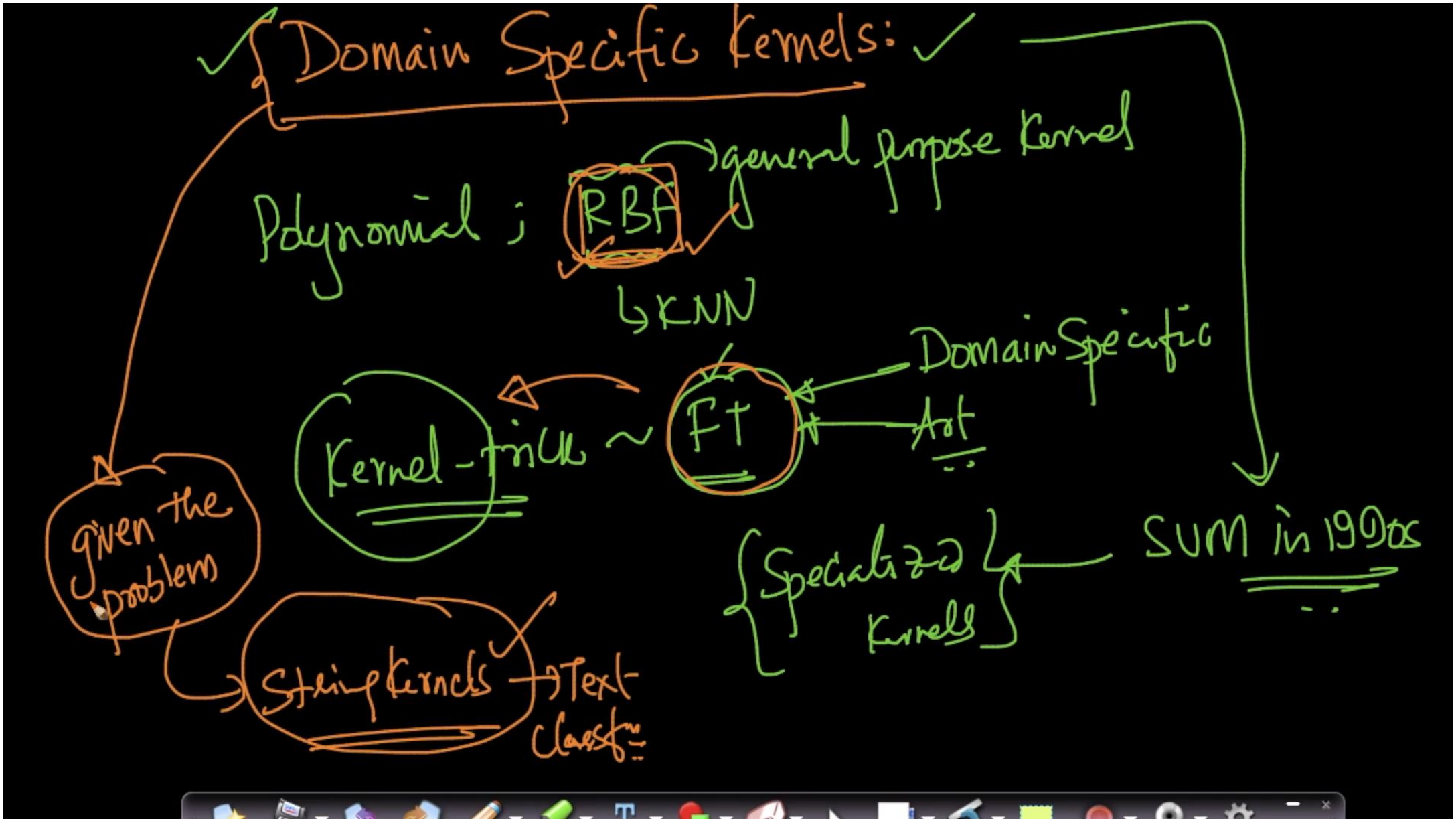
$K(x_1, x_2) > K(x_1, x_3)$


K-NN
K is large
K = Sim
σ = 10
σ = 1
σ = 0.1
dist
K(xi, xj) = 0
dissim ≈ dist
Sim ≈ K
σ ↑ ↓
σ = 1
σ = 0.1
Small K in K-NN
Sim > θ
xj
Sim > θ
xj
Sim > θ
xj
σ = 10
Sim > θ
xj
More K's
Large K in K-NN


K-NN :- Store all the k-pts ; LSH O(nd)
RBF-SUM :- SVs → αi's
αi > 0
f(xq)
# SVs << n
RBF-SVM ~ K-NN


dont Know the "best" Kernel to use
Poly-Kernel
Simply use RBF-SVM
KNN
Soft-Margin
C
σ : RBF
log-reg
elastic net : λ1, λ2
C, σ
grid Search
Rand Search


Domain Specific Kernels:
Polynomial ; RBF
general purpose Kernel
↳ KNN
Kernel-trick ~ Ft
Domain Specific
Art
given the problem
String Kernels → Text-Classif
{Specialized Kernels}
SVM in 1990s


Domain Specific Kernels:
Polynomial ; RBF
general purpose Kernel
↳ KNN
Domain Specific
Art
Ft
Kernel - trick ~
given the problem
String Kernels
Text-
Specialized Kernels
SVM in 1990s

SVM

Train & Run Time complexity of SVM

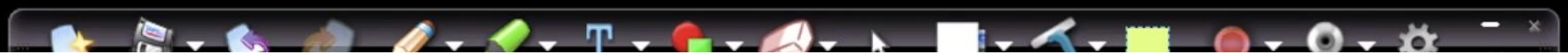
SMO, SVM

Train: → SGD

↳ Specialized algo (dual) → Sequential minimal optmzn (SMO)

✓ libSVM: best libraries for training SVMs

sklearn ✓

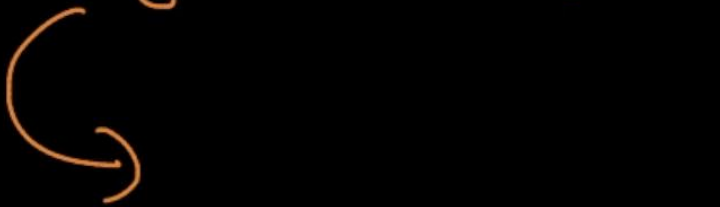


Traing Time :- $\sim O(n^2)$ for Kernel-SVMs

more opt algo (2007) :- $O(nd^2)$ if $d<n$

- if n is large → $O(n^2)$ M

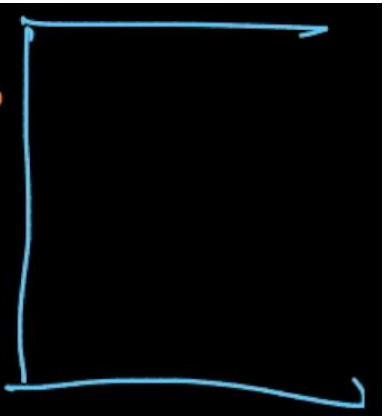
↑ "Typically" don't use "SVM" when n is large ...



Run-time: $f(x_q) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x_q) + b$

$x_q \rightarrow y_q$

$\alpha_i = 0$ for non SVs



SVs = K

if # SVs of small

$n = 100{,}000$

log-regrs: $O(d)$

$O(Kd)$

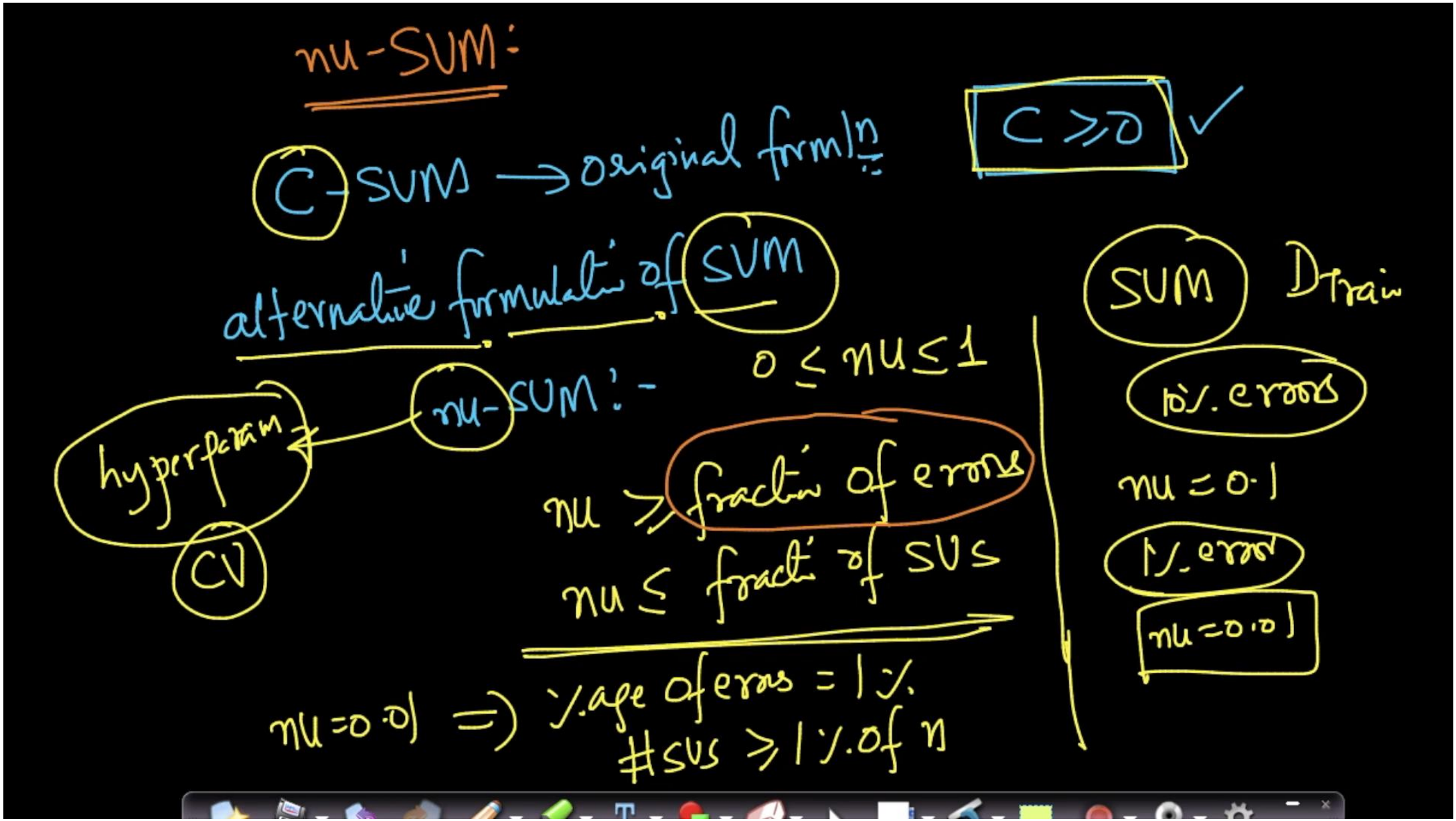
$1 \le K \le n$

$K = 100$

✓ RBF

Soft-SUM-formula: C, σ

nu-SUM


nu-SVM:
C-SVM → original form!!
C ≥ 0 ✓
alternative formulation of SVM
nu-SVM:-
0 ≤ nu ≤ 1
hyperparam
CV
nu ≥ fraction of errors
nu ≤ fracⁿ of SVs
nu = 0.01 ⇒ %age of errors = 1%
#SVs ≥ 1% of n
SVM
D_train
10% errors
nu = 0.1
1% error
nu = 0.01

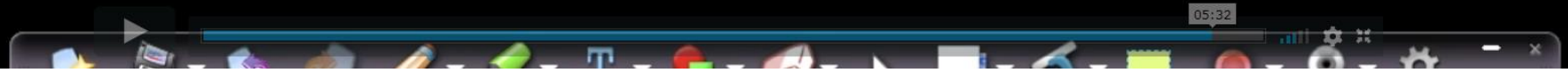
Run-time complx :- fewer SVs

$\eta \approx 0.01$ $\Rightarrow$ errors: $\leq 1\%$.

SVs $\geq 1\%$ of n

$n = 100,000$

$\#SVs = k \geq 1000$



✓ Support Vector regression (SVR)

$y_i \in \mathbb{R}$

SVM - Classfn :- SVC $\rightarrow y_i \in \{+1, -1\}$

Kernalized

Math:

$$\min_{w,b} \underbrace{\frac{1}{2}\|w\|^2}_{\text{reg}}$$

s.t $y_i - (w^T x_i + b) \leq \epsilon$ $\forall i$ (hyperparam)

$(w^T x_i + b) - y_i \leq \epsilon$

$\epsilon \geq 0$ ✓

$y_i = w^T x_i + b$

$f(x_i) = w^T x_i + b$

$= \hat{y}_i$

$$\begin{cases} y_i - \hat{y}_i \leq \epsilon \\ \hat{y}_i - y_i \leq \epsilon \end{cases}$$

Lr. form of SVR

→ Interpretability & Feature Importance → Kernel-SVM

↳ F.I for various features

↳ forward feature selcn

→ Outliers → v. little impact

↳ (SV)s that matter

↳ RBF with a small σ ⇒ K-NN with small K

→ Bias-Var :- $C \uparrow \Rightarrow$ overfit $\Rightarrow$ high-var

GridSearchCV $C \downarrow \Rightarrow$ underfit $\Rightarrow$ high-bias

RBF-SVM :- C, σ (loss) ↳ (RBF)

→ large (d) → V. good for SVM
→ Kernels (RBF)

(d) ⟷ (d')

→ Best cases: {right kernel} ✓

→ Worst cases :- (n) is large → Train time is high

→ internet based

logistic regn

(K) is large ⇒ low-latency is not possible

→ #SVs

f(x_q)

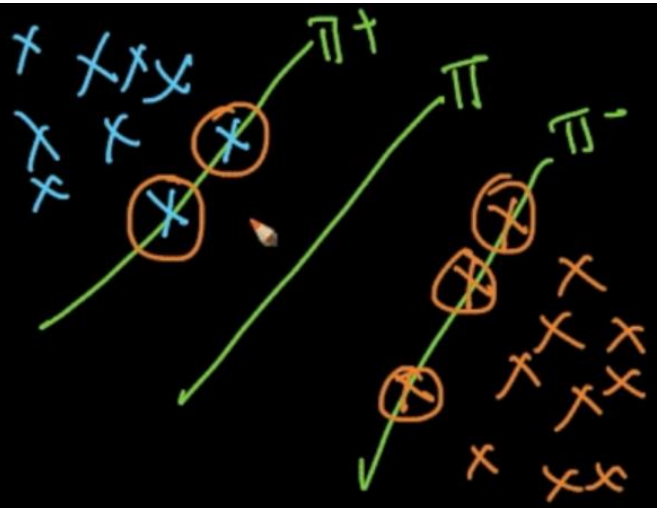
Loss-minimization : Hinge-loss

$= y_i (\vec{w}^T x_i + b) = z_i$

- $z_i > 0$:- x_i is correctly classified
- $z_i < 0$:- x_i is incorrectly classified

- log-reg :- logistic loss + reg
- lr-reg :- lr. loss + reg
- SUM :- hinge loss + reg

$$f(x_q) = \sum_{i=1}^{n} \alpha_i y_i x_i^T x_q + b$$



SVs :- $\alpha_i > 0$

non SVs :- $\alpha_i = 0$

$f(x_q)$:- only pts that matter are SVs

SVM

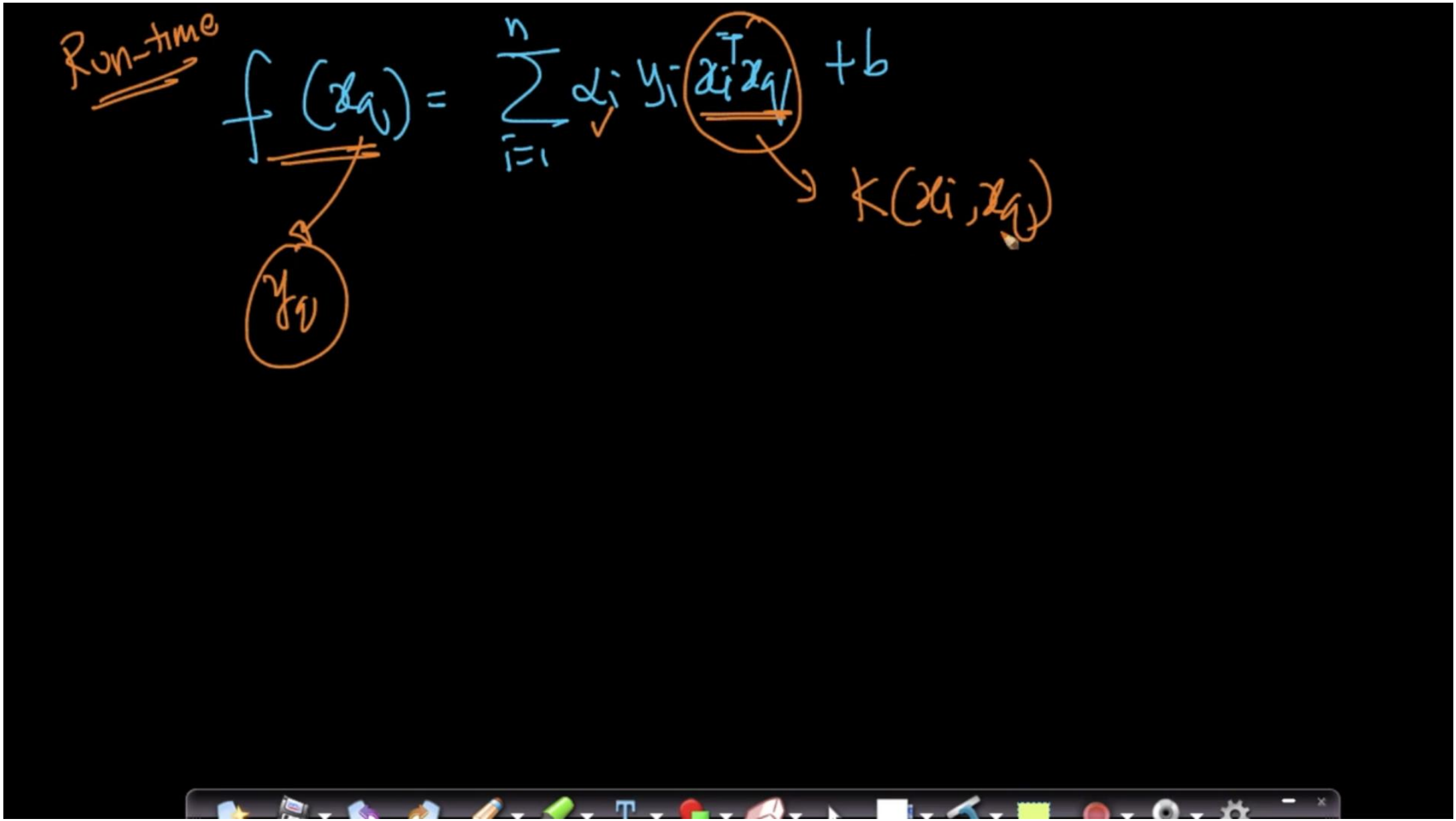
$$\max_{\alpha_i} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \underbrace{K(x_i, x_j)}_{\text{Kernel-fn}}$$

$$\text{s.t } \alpha_i \geq 0 \longrightarrow \begin{cases} \alpha_i = 0 \text{ for svs} \\ \alpha_i > 0 \text{ for non svs} \end{cases} \checkmark$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

Sim (x_i, x_j)

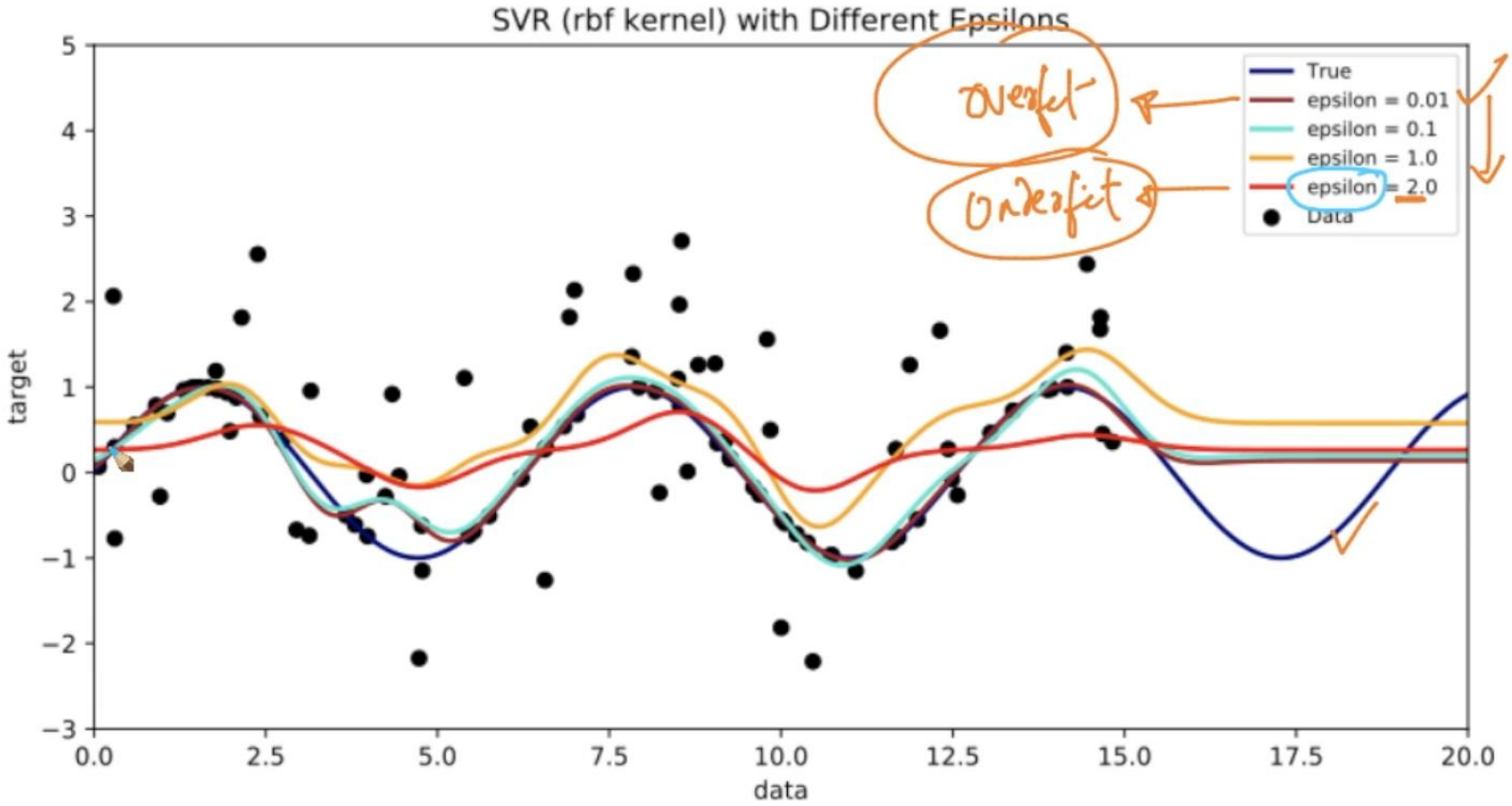
$$\begin{cases} x_i^T x_j = x_i \cdot x_j = \overbrace{\text{Cosine\_sim}(x_i, x_j)}^{\text{if } \|x_i\| = 1 \,;\, \|x_j\| = 1} \end{cases}$$


Run-time
$f(x_q) = \sum_{i=1}^{n} \alpha_i y_i x_i^T x_q + b$
y_q
$K(x_i, x_q)$

Contents - Google Docs | Svr epsilons demo - Support | Chekuri Srikan...

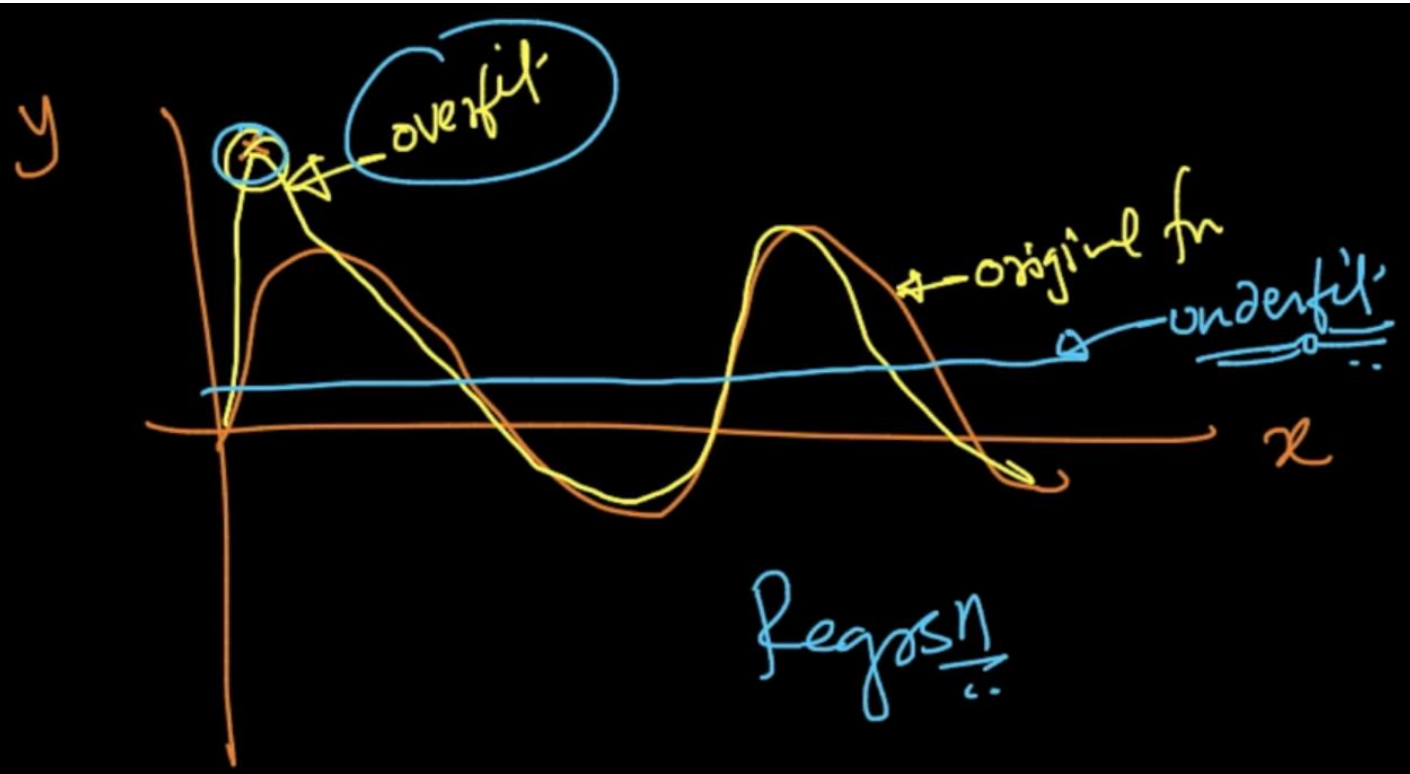
Secure | https://en.wikipedia.org/wiki/Support_vector_machine#/media/File:Svr_epsilons_demo.svg

Press Esc to exit full screen



Support Vector Regression (prediction) with different thresholds ε. As ε increases, the prediction becomes less sensitive to errors

More details


y
overfit
original fn
underfit
x
Regression