Machine Learning from Data

Lecture 24: Spring 2021

Today’s Lecture

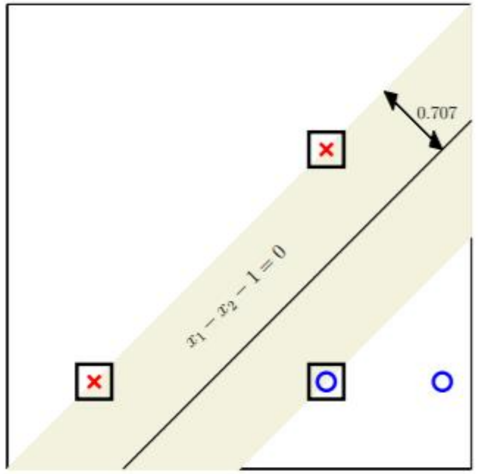
- Support Vector Machines (SVMs)
  - Why is fattest hyperplane better?
  - Non-separable Data

RECAP: **The Optimal Hyperplane**

The Optimal Hyperplane

The fattest hyperplane that separates the data

tolerates most measurement error



1. Can we efficiently find the fattest separator?
2. Is fatter better than thin?

The Algorithm

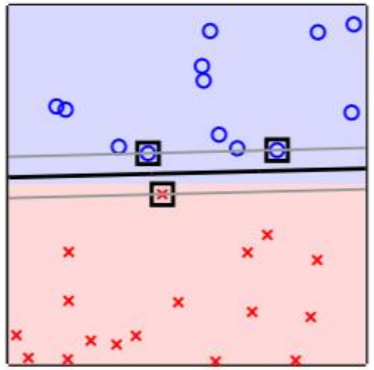
Quadratic Programming:

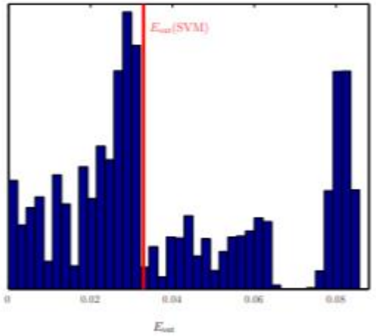
$$\underset{b,\mathbf{w}}{\text{minimize}} \quad \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\text{subject to: } y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1 \quad \text{for } n = 1, \ldots, N.$$

Support vectors: the data points that sit on the cushion.

Using only support vectors, the classifier does not change.





PLA depends on the (random) order of data

Link to Regularization

<u>optimal hyperplane</u>

$$\underset{b,\mathbf{w}}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w}$$

$$\text{subject to:} \quad y_n(\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b) \geq 1 \quad \text{for } n = 1, \ldots, N.$$

<u>regularization</u>

$$\underset{\mathbf{w}}{\text{minimize}} \quad E_{\text{in}}(\mathbf{w})$$

$$\text{subject to:} \quad \mathbf{w}^{\mathsf{T}}\mathbf{w} \leq C.$$

	optimal hyperplane	regularization
minimize	$\mathbf{w}^{\mathsf{T}}\mathbf{w}$	E_{in}
subject to	$E_{\text{in}} = 0$	$\mathbf{w}^{\mathsf{T}}\mathbf{w} \leq C$

The optimal hyperplane performs 'automatic' regularization.

Evidence that Larger Margin is Better

(1) Experimental: larger margin gives lower E_{out}; **bias** drops a little and **var** a lot.

(2) Bound for d_{VC} can be less than $d+1$ – fat hyperplanes generalize better.

(3) E_{cv} bound does not explicitly depend on d.

Larger Margin is better

1. Generate a separable dataset $(N=20)$
2. Randomly generate 50,000 separating hyperplanes
3. Repeat

$h_1 \longrightarrow r(h_1)/r_{opt} = \rho_1$

$h_2 \longrightarrow r(h_2)/r_{opt} = \rho_2$

$\vdots$

$h_{50K} \longrightarrow r(h_{50K})/r_{opt} = \rho_{50K}$

$E_{out}(h_1)$

$E_{out}(h_2)$

$\vdots$

$E_{out}(h_{50K})$

Optimal hyperplane

r_{opt}

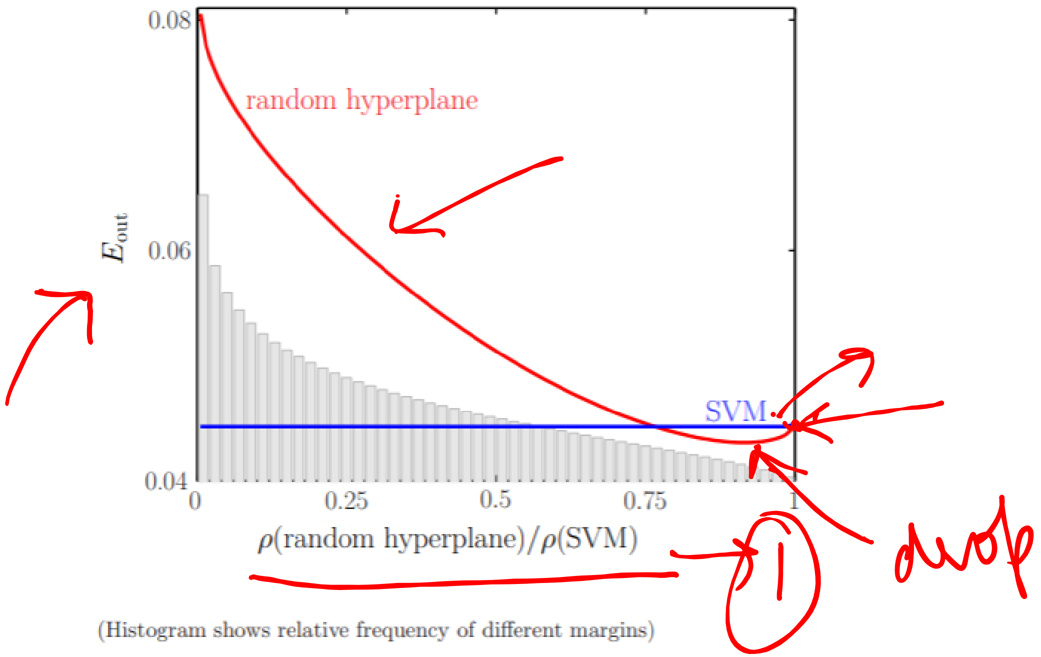
Larger Margin is Better

Generate a random separable data set ($N = 20$)

Select 50,000 random separating hyperplanes h

Compute E_{out} and $\rho(h)/\rho(SVM)$

Average over several thousands of random data sets



(Histogram shows relative frequency of different margins)

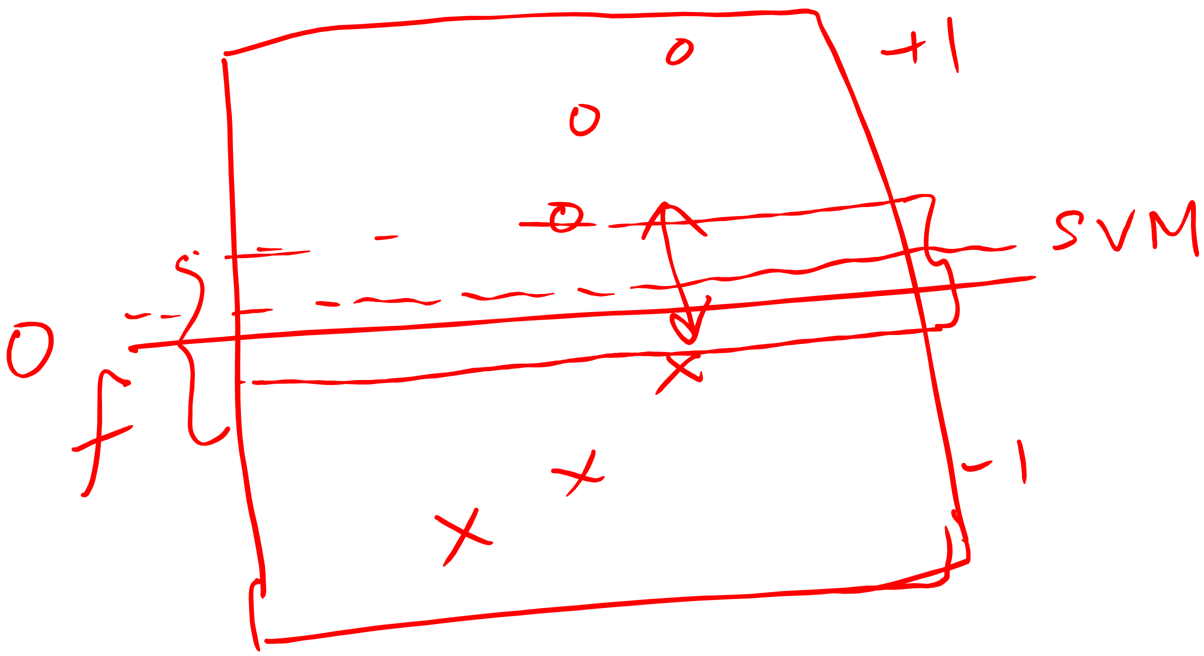
Bigger margin is generally better

Biggest is not best.

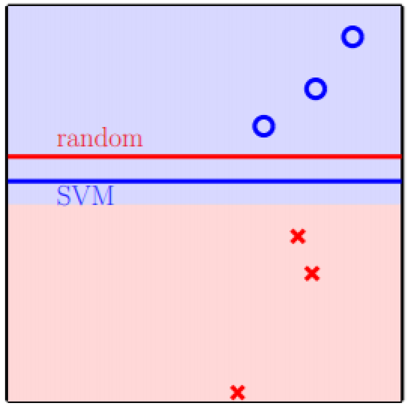
⟵ Data other than support vectors can have role in fine-tuning

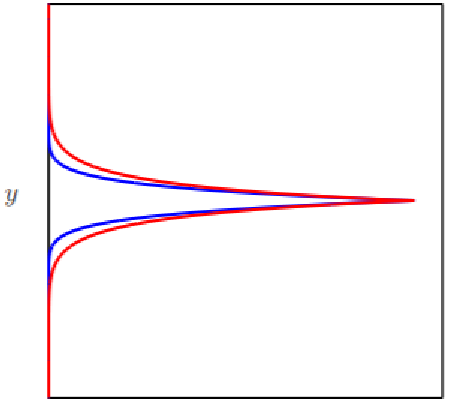
① Optimal

② Random

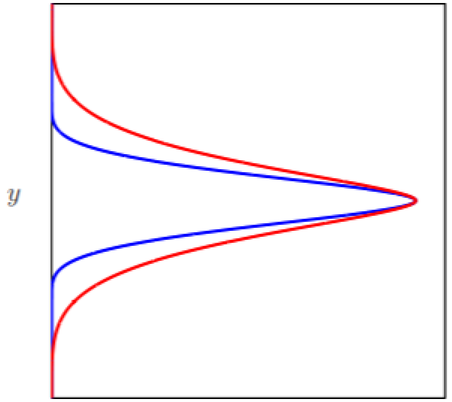


Bias and Variance



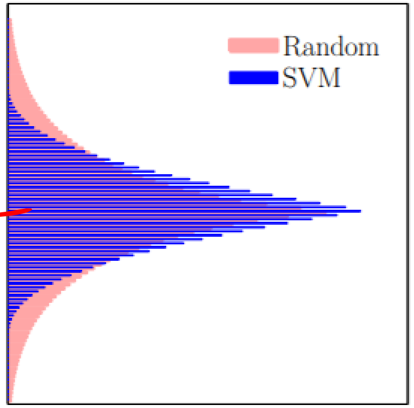


$\text{bias}_{\text{rand}}$ vs. bias_{SVM}



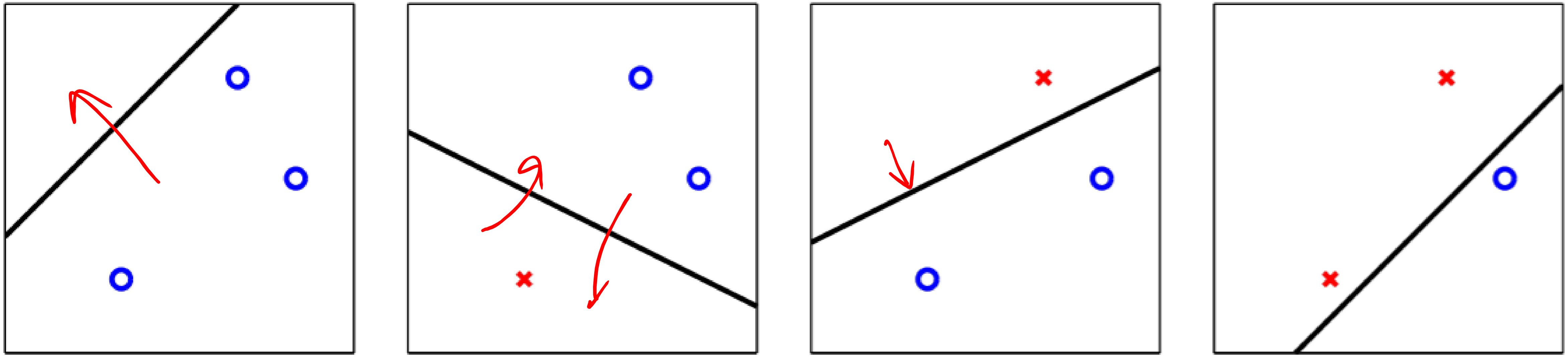
var_{rand} vs. var_{SVM}

0



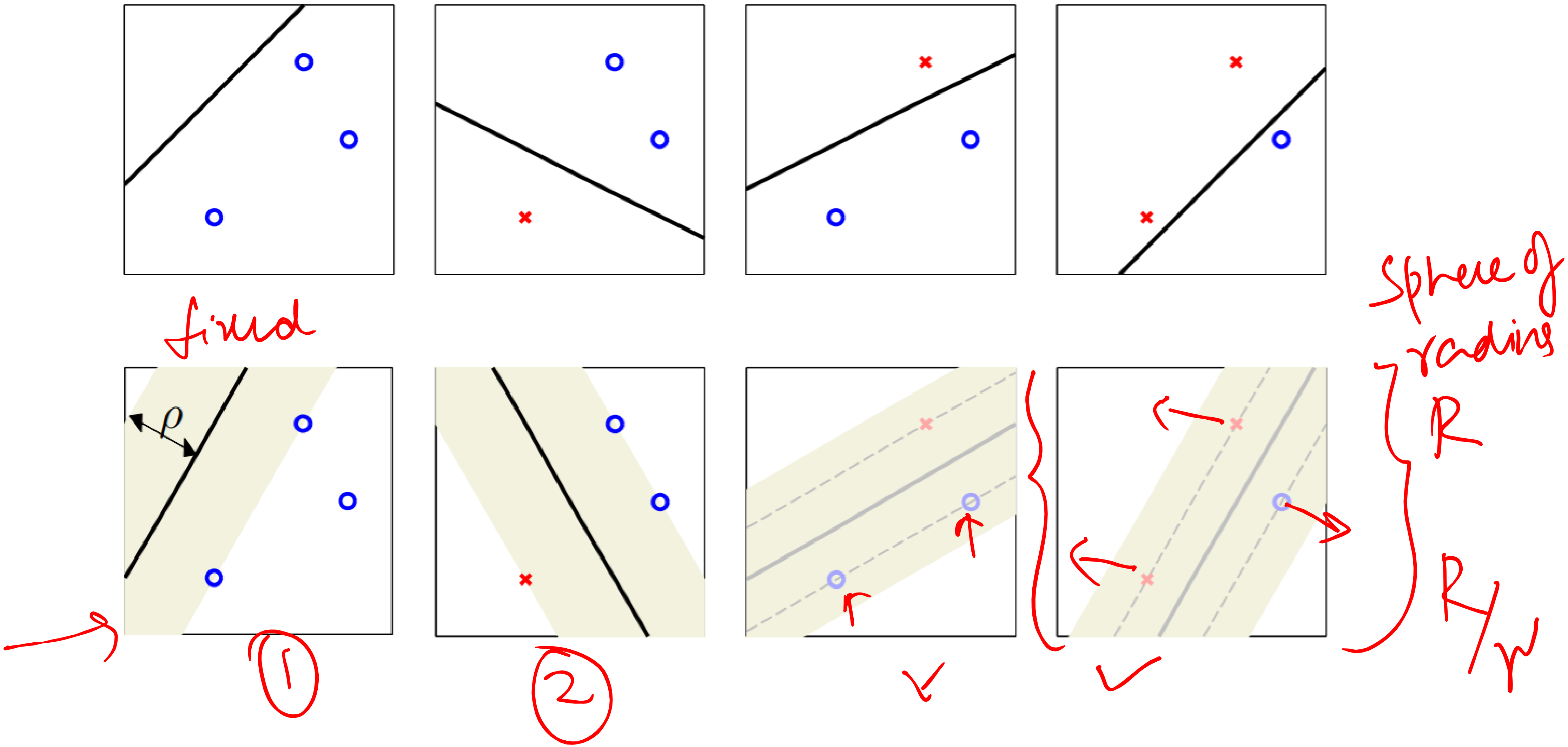
	Random	SVM	
bias	0.02	0.015	**-0.005**
var	0.059	0.038	**-0.021**
E_{out}	0.079	0.053	**-0.026**

Fat Hyperplanes Shatter Fewer Points



d_{VC}

Fat Hyperplanes Shatter Fewer Points



Theorem (intuition)

$$d_{vc}(\rho) \le \left\lceil \frac{R^2}{\rho^2} \right\rceil + 1$$

d_{vc} → lower E_{in} & E_{out}

ρ^2 → thickness of hyperplane

ρ → bigger

, $d_{vc} \le d+1$

$$d_{vc} \le \min\left(d+1, \left\lceil \frac{R^2}{\rho^2} \right\rceil + 1\right)$$

→ independent of dimensions.

→ ρ → big

R → small

E_{cv} (error) of max-margin (fattest) hyperplane

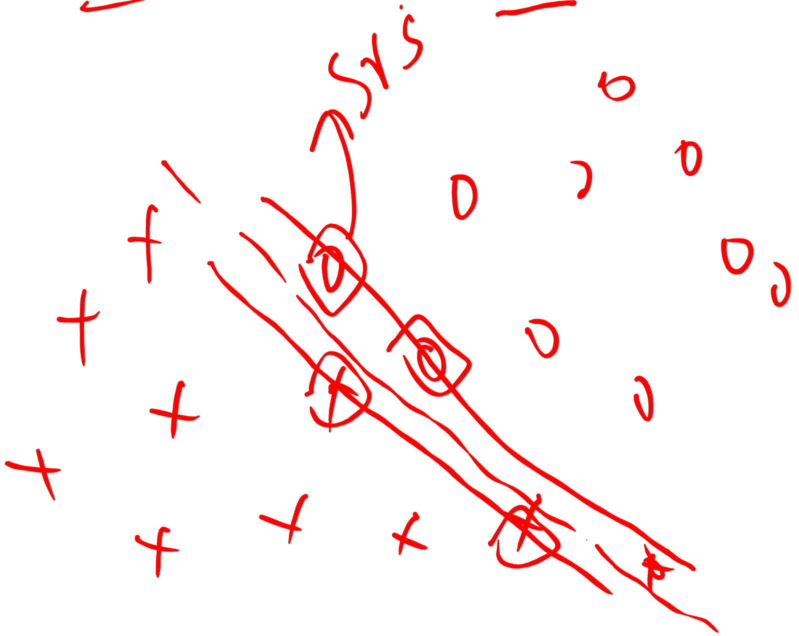
$E_{cv} \rightarrow$ unbiased estimate of

$\rightarrow E_{out}(N-1)$

Result to prove

E_{cv} (fattest hyperplane) $\leq$ # of support vectors / N

SV are imp.



$E_{cv} \rightarrow$ small $\rightarrow E_{out}$

$\downarrow$

control

Proof :

$$e_{LOO}(x_{\text{notsv}}) = 0$$

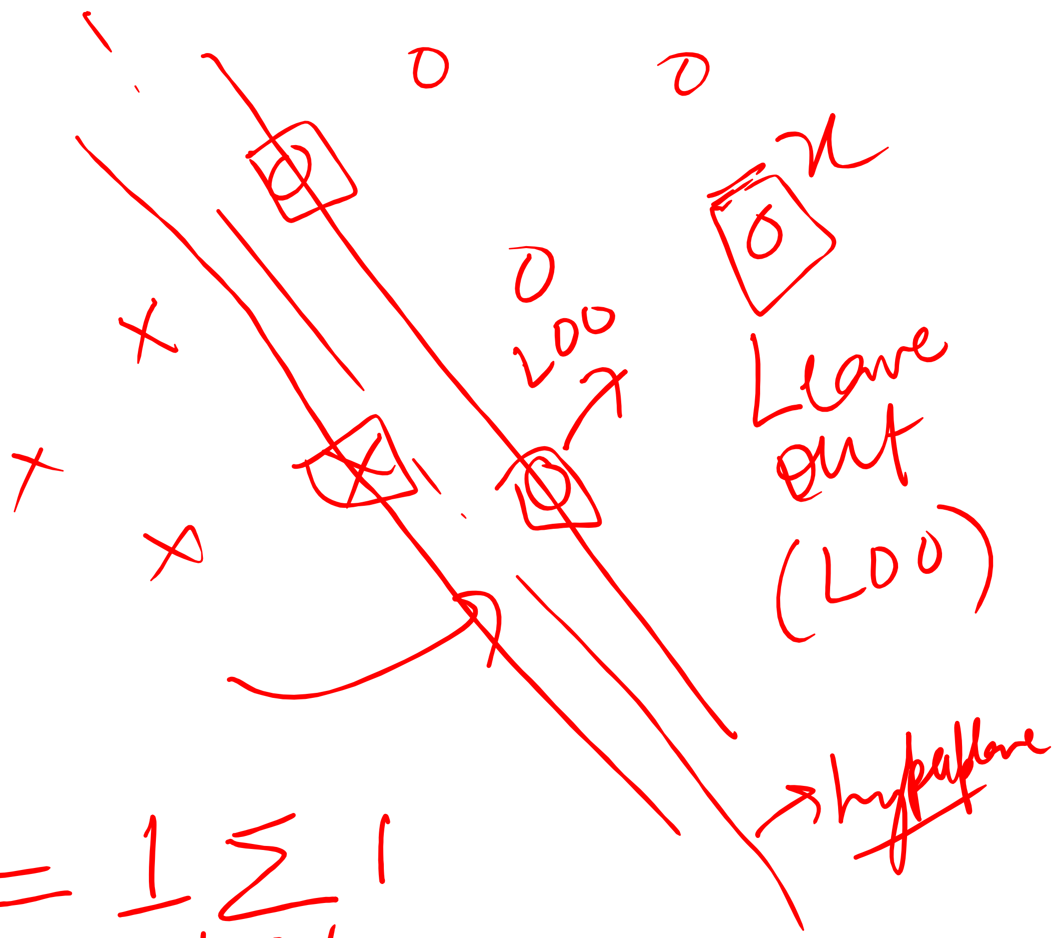
$$e_{LOO}(x_{sv}) \le 1$$

$$\therefore E_{cv} = \frac{1}{N}\sum_{n=1}^{N} e_{LOO}(x_n)$$

$$= \frac{1}{N}\sum_{SV} e_{LOO}(x_{sv}) = \frac{1}{N}\sum_{SV} 1$$

$$= \frac{\#\text{ of SVS}}{N}$$

$$E_{cv} \to E_{out}$$



Perceptron : $d_{vc} = d+1$

$\downarrow$

hyperplane

Optimal

1) Experimentally

Bias $\downarrow$ Variance $\downarrow\downarrow$ ($\because E_{out} \downarrow$)

2) $d_{vc} \leq \min\left(d, \left\lceil \frac{R^2}{\rho^2} \right\rceil\right) + 1$

$E_{in} \rightarrow E_{out}$

3) $E_{cv} \leq \frac{\# \text{ of SVs}}{N}$

(4) Algorithm : QP

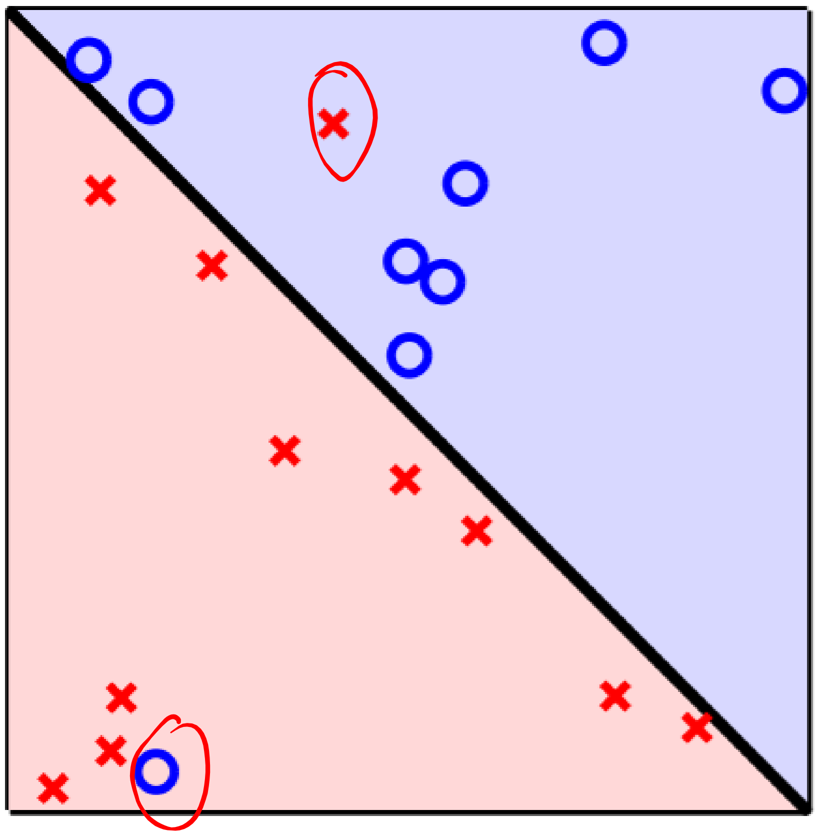
PLA

$$E_{cv} \le \frac{R^2}{N\rho^2}$$

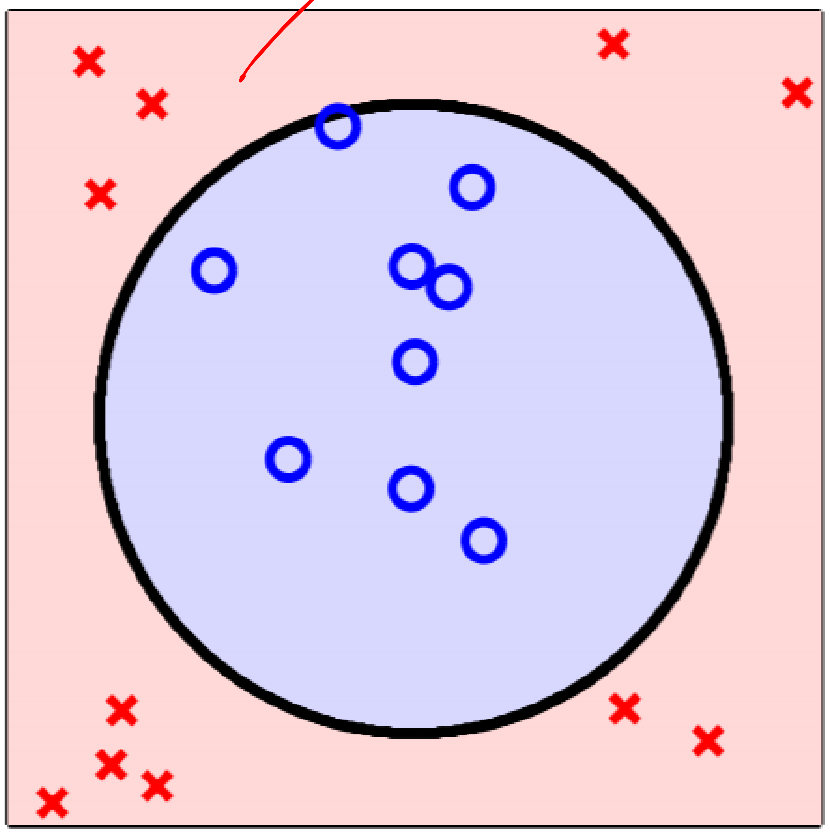
$\rho^2 \rightarrow$ optimal

Non-Separable Data

2-8pm

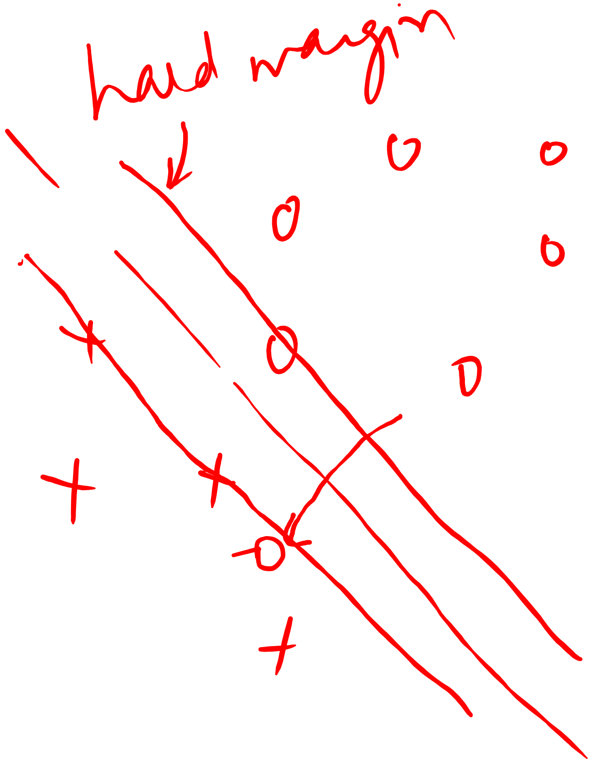


tolerate error



nonlinear transform

SOFT MARGIN OPTIMAL HYPERPLANE

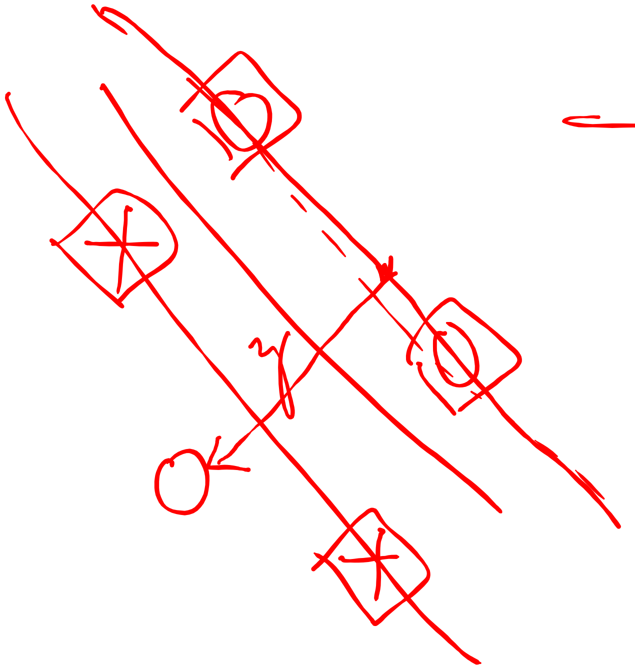


$$\min \frac{1}{2} w^T w$$

$$\text{s.t. } y_n(w^T x_n + b) \geq 1 \quad \Big\} \rightarrow \underline{\underline{\text{SVs}}}$$

Original condition: $\underline{y_n(w^T x_n + b) = 1}$

<u>saturating</u>



$$y_n(w^T x_n + b) \geq 1 - \xi_n$$

$$\xi_n \geq 0$$

$\xi_n \to \infty$

Penalty

amount

margin violation

Optimization:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + \underbrace{C \sum \xi_n}$$

s.t. $y_n(w^T x_n + b) \geq 1 - \xi_n$

$\xi_n \geq 0$

Regularization.

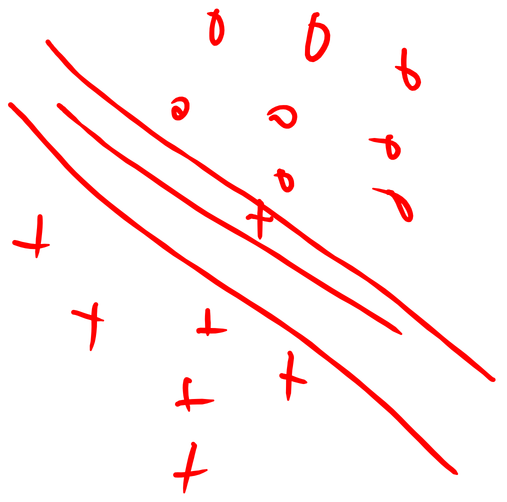
Soft Margin Optimal Hyperplane

Variables: w, b, ξ

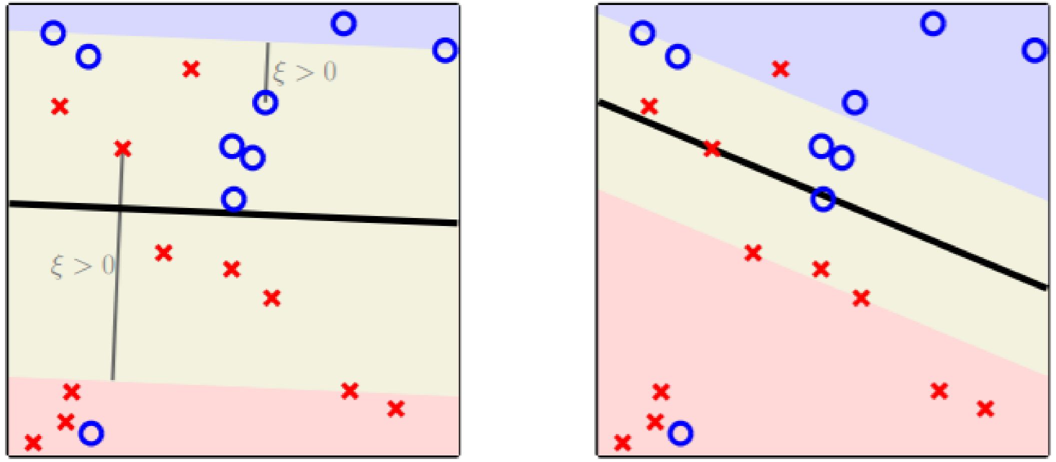
Regularization → penalty (complexity)
Soft Margin → Size of margin (robustness).

C → Regularization parameter.

Small C → more robustness → more regularized
Large C → less " , C → ∞ (Recovering the hard margin).



Non-Separable Data

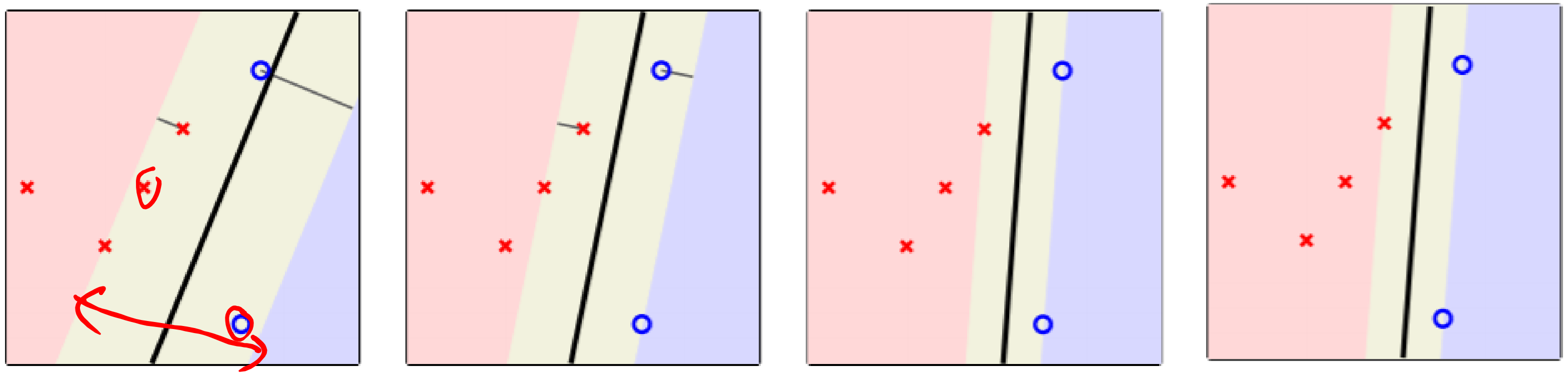


$C = 1$ $\qquad\qquad$ $C = 500$

Robustness VS In-sample error.

$$
\begin{aligned}
&\underset{b,\mathbf{w},\boldsymbol{\xi}}{\text{minimize}} && \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{n=1}^{N}\xi_n \\
&\text{subject to:} && y_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1 - \xi_n \\
& && \xi_n \geq 0 \qquad\qquad \text{for } n = 1, \ldots, N
\end{aligned}
$$

Soft Margin SVM With Separable Data



small C | medium C | large C | hard margin

$C \to \infty$ (big)

QPAC

$$\underset{b,\mathbf{w},\boldsymbol{\xi}}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w} + C\sum_{n=1}^{N}\xi_n$$

$$\text{subject to:} \quad y_n(\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b) \geq 1 - \xi_n$$

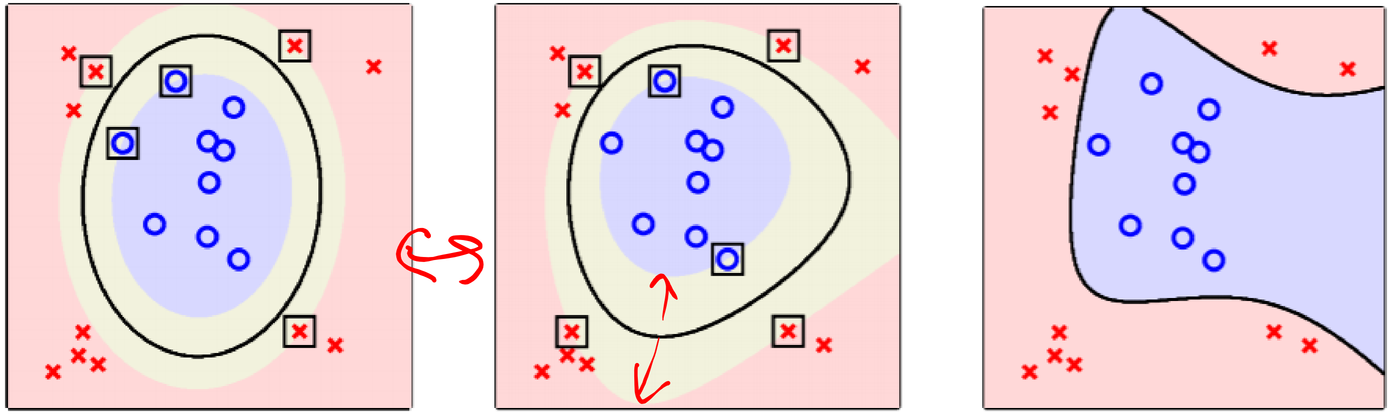
$$\xi_n \geq 0 \qquad \text{for } n = 1, \ldots, N$$

QP

Choice of C is **IMPORTANT**

Tolerance of in-sample error.

Nonlinear Transform and SVM

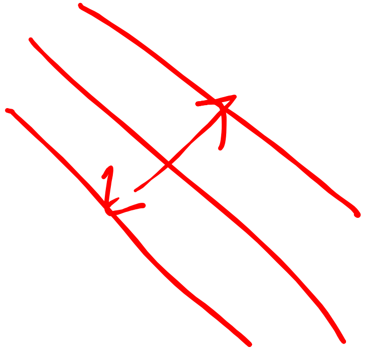


Φ_2+ SVM | Φ_3+ SVM | Φ_3+ pseudoinverse algorithm

overfit

Observations:

1. Φ_3 has almost $2\times$ the parameters of Φ_2
2. Φ_3-SVM does not display significant overfitting compared to Φ_3-regression
3. #support vectors did not double
4. Can go to higher dimensions if #support vectors stays small or margin stays large



	pseudoinverse regression		SVM	
	linear	nonlinear (ϕ)	linear	nonlinear (ϕ)
overfitting	little	lots	tiny	ok
boundary	linear	complex	linear	complex

Going to Even Higher Dimension

In higher dimension, can control overfitting with # support vectors or margin ρ

What about:

Efficiency?

Infinitely many dimensions?

E_{cv}

$\hookrightarrow E_{out}$

Thanks!