Machine Learning from Data

Lecture 8: Spring 2021

Today’s Lecture

- Linear Models
  - Classification and
  - Regression

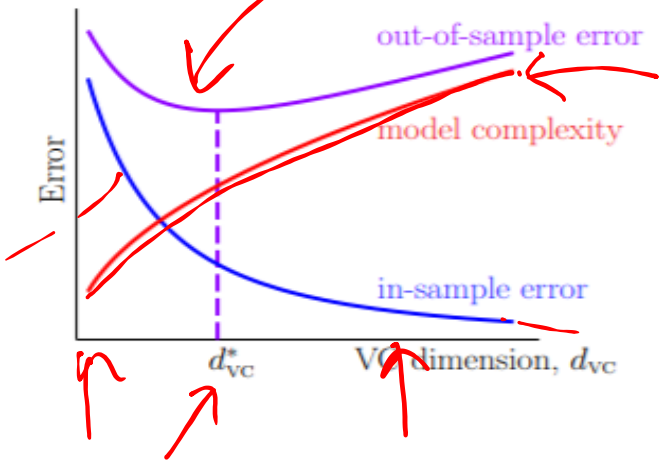
TRADE-OFF

VC Analysis

$$E_{\text{out}} \leq E_{\text{in}} + \Omega(d_{\text{VC}})$$

1. Did you fit your data well enough (E_{in})?
2. Are you confident your E_{in} will generalize to E_{out}

Approximation vs generalization



E_{in}

The VC Insuarance Co.

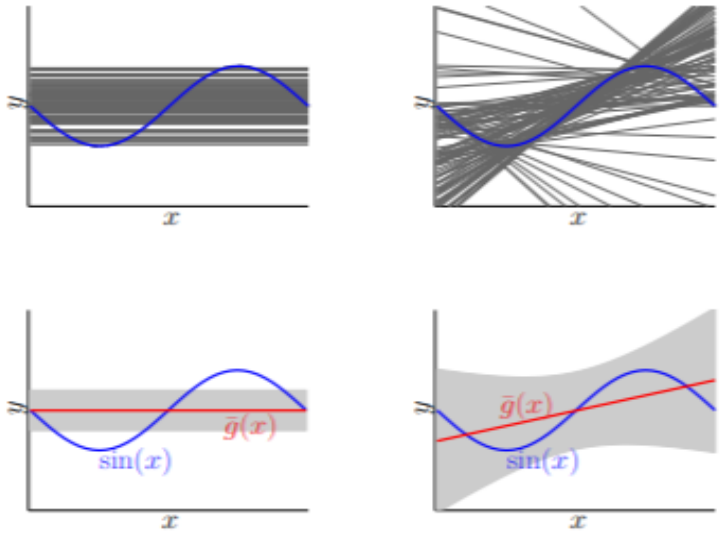
The VC warranty had conditions for becoming void:

You can't look at your data *before* choosing $\mathcal{H}$.
Data must be generated i.i.d from $P(\mathbf{x})$.
Data and test case from *same* $P(\mathbf{x})$ (same bin).

Bias-Variance Analysis

$$E_{\text{out}} = \textsf{bias} + \textsf{var}$$

1. How well can you fit your data (**bias**)?
2. How close to that best fit can you get (**var**)?

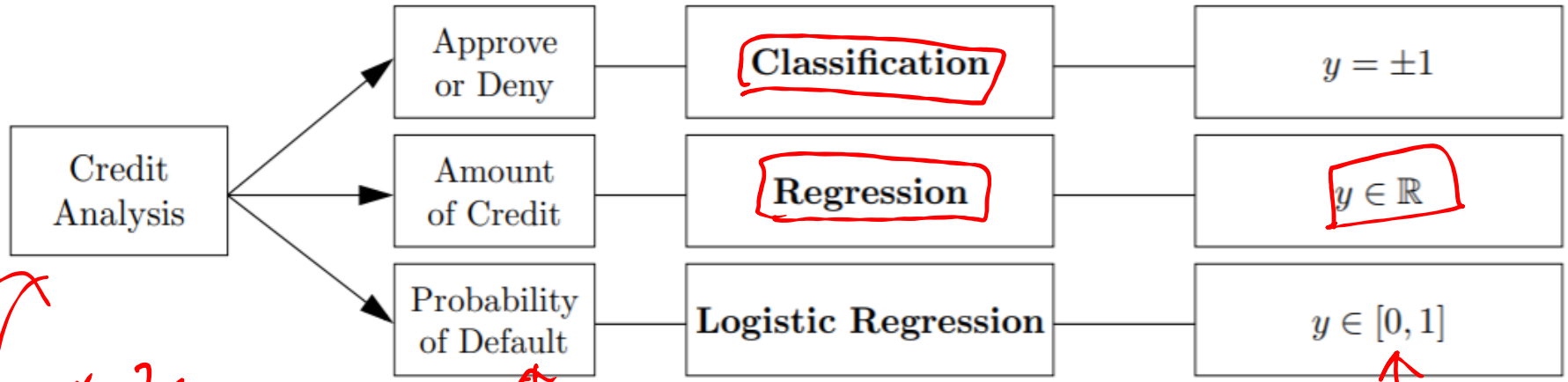


$\mathcal{H}_0$
bias $= 0.50$;
var $= 0.25$.
$E_{\text{out}} = 0.75$ ✓

$\mathcal{H}_1$
bias $= 0.21$;
var $= 1.69$.
$E_{\text{out}} = 1.90$

Linear Model

y → outcome variable



Chapter 2

A → 0.4 or 0.9

Linear Model : FUNDAMENTAL

i) On its own it's able to solve most ML problems

ii) It's a building block for other model. (NN, SVM (robust linear model))

$$S = W^T x$$

where $w \in \mathbb{R}^{d+1}$

$x \in 1 \times \mathbb{R}^{d+1}$

linear in weights.

linear in x

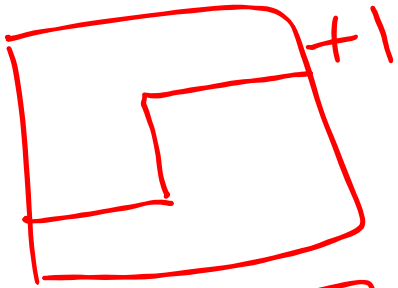
hyperplane $\xrightarrow{2d}$ line

Is it possible to solve any kind of problem using linear model?

TRANSFORM

$s = w^T x$ (linear signal)

→ sign function (classification)

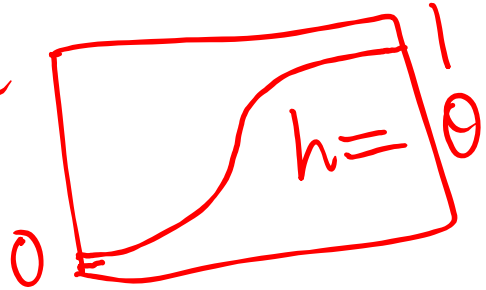


$h(x) = \text{sign}(w^T x)$

→ Identity transform (Regression)

$h(x) = w^T x$

→ Sigmoid function (Logistic Regression)



$h(x) = \theta(w^T x)$

Classification

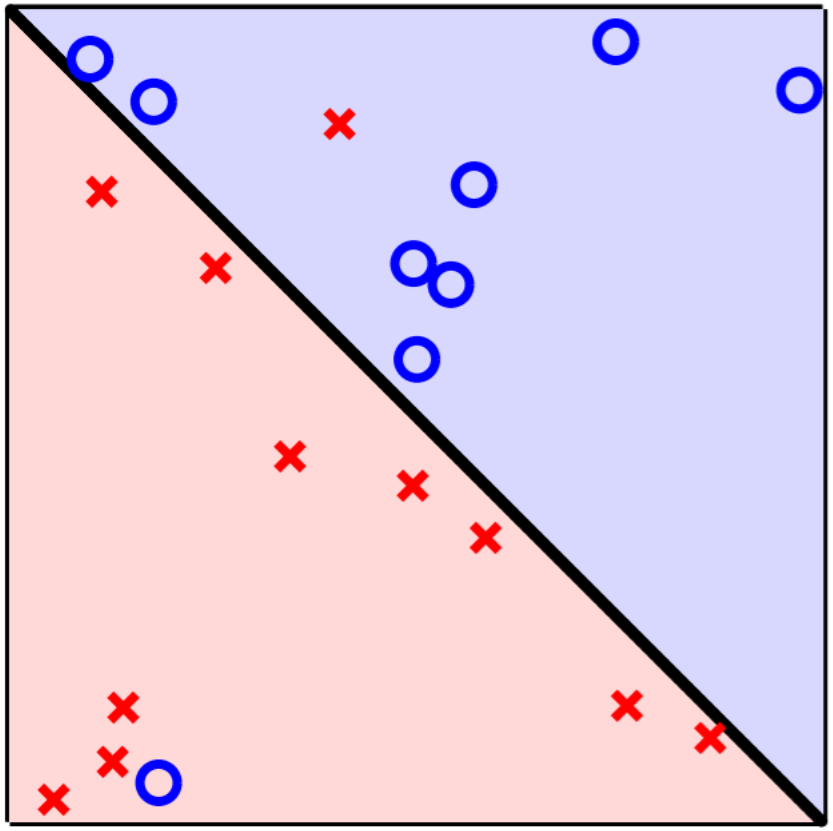
- $\mathcal{H}_{linear} = \{ h(x) = sign(w^T x) \mid w \in \mathbb{R}^{d+1} \}$ (Perceptron)

✓ i) $d_{VC} = d+1$, $\therefore E_{out} \leq E_{in} + O\left(\sqrt{d \frac{\ln N}{N}}\right)$

✓ ii) $E_{in} \simeq 0$, PLA $\longrightarrow$ accomplish that. $E_{in}(g) \simeq 0$

Q1. What happens if data is not linearly separable? $\rightarrow$ Tolerate error (Pocket Algorithm)

Q2. How can we ensure $E_{in}(g) \simeq 0$ $\rightarrow$ Feature Set



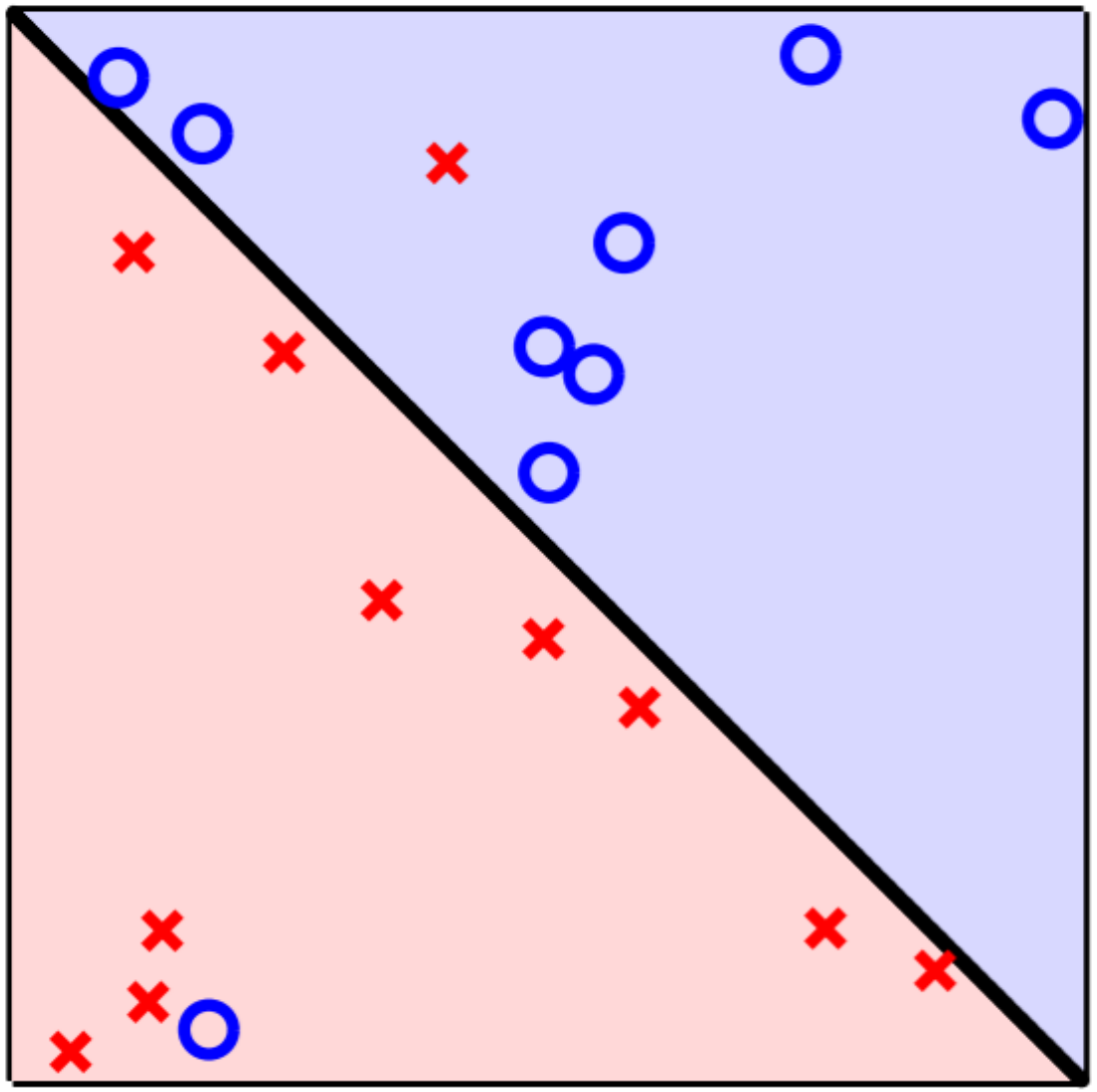
PLA : Pocket Algorithm

$w(t+1) \leftarrow w(t) + y_* x_*$

$w_t \longrightarrow E_{in}$ (best)

Pocket

Update



→ Get a better starting point.

→ Algorithms.

→ Minimizing E_{in} is a *hard* combinatorial problem.

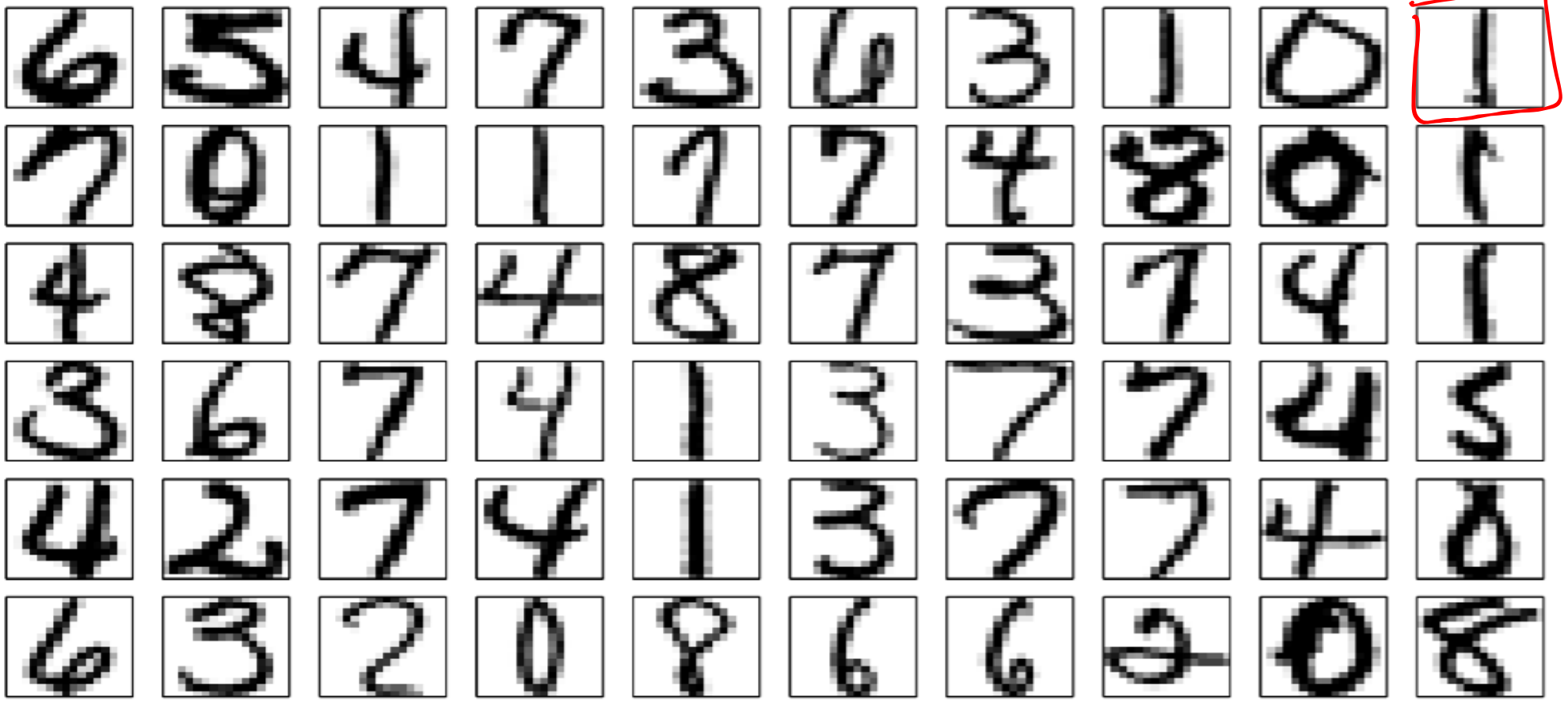
The Pocket Algorithm

- Run PLA ✓
- At each step keep the best E_{in} (and $\mathbf{w}$) so far.

(Its not rocket science, but it works.)

Stopping Criteria

1000 iterations

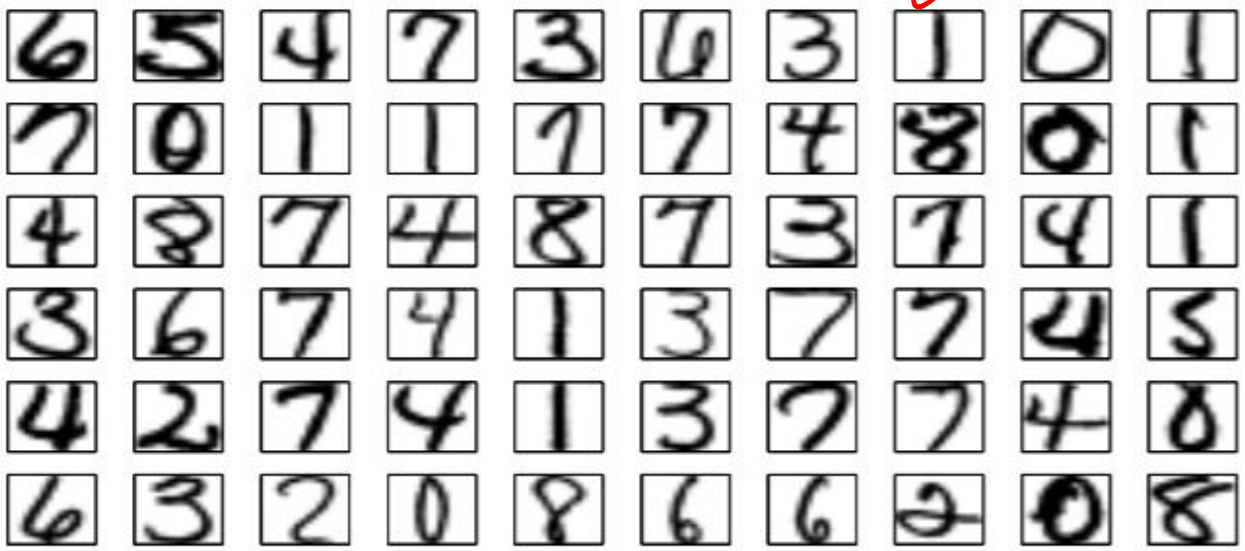


Each digit is a 16×16 image.

Pre-processing.

$E_{in} \simeq E_{out}$

$E_{in} \simeq 0$



500 points

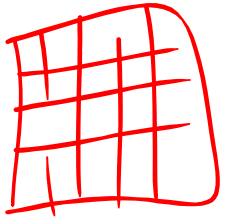
$d_{vc} = 5$

$d_{vc} \times 100$

25700

$d = 256$

$d_{vc} = 257$



Each digit is a 16×16 image. -1 to $+1$

[-1 -1 -1 -1 -1 -1 -1 -0.63 0.86 -0.17 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -0.99 0.3 1 0.31 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -0.41 1 0.99 -0.57 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -0.68 0.83 1 0.56 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -0.94 0.54
1 0.78 -0.72 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0.1 1 0.92 -0.44 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -0.26 0.95 1 -0.16 -1 -1
-1 -0.99 -0.71 -0.83 -1 -1 -1 -1 -1 -0.8 0.91 1 0.3 -0.96 -1 -1 -0.55 0.49 1 0.88 0.09 -1 -1 -1 -1 0.28 1 0.88 -0.8 -1
-0.9 0.14 0.97 1 1 1 0.99 -0.74 -1 -1 -0.95 0.84 1 0.32 -1 -1 0.35 1 0.65 -0.10 -0.18 1 0.98 -0.72 -1 -1 -0.63 1 1
0.07 -0.92 0.11 0.96 0.30 -0.88 -1 -0.07 1 0.64 -0.99 -1 -1 -0.67 1 1 0.75 0.34 1 0.70 -0.94 -1 -1 0.54 1 0.02 -1 -1
-1 -0.90 0.79 1 1 1 1 0.53 0.18 0.81 0.83 0.97 0.86 -0.63 -1 -1 -1 -1 -0.45 0.82 1 1 1 1 1 1 1 1 0.13 -1 -1 -1 -1 -1
-1 -0.48 0.81 1 1 1 1 1 1 0.21 -0.94 -1 -1 -1 -1 -1 -1 -1 -0.97 -0.42 0.30 0.82 1 0.48 -0.47 -0.99 -1 -1 -1 -1]

Feature Construction

- Summarize the input into features that are useful → d_{vc}

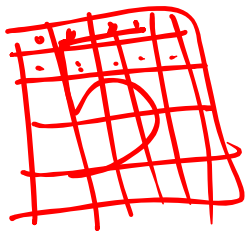
i) Reduce the dimension (d_{vc}) significantly

$E_{in} \simeq E_{out}$

ii) What is really important?

1 5 ✓ Intensity ✓ Symmetry

Mathematical form. → Algorithm



$d_{vc} = 3$!

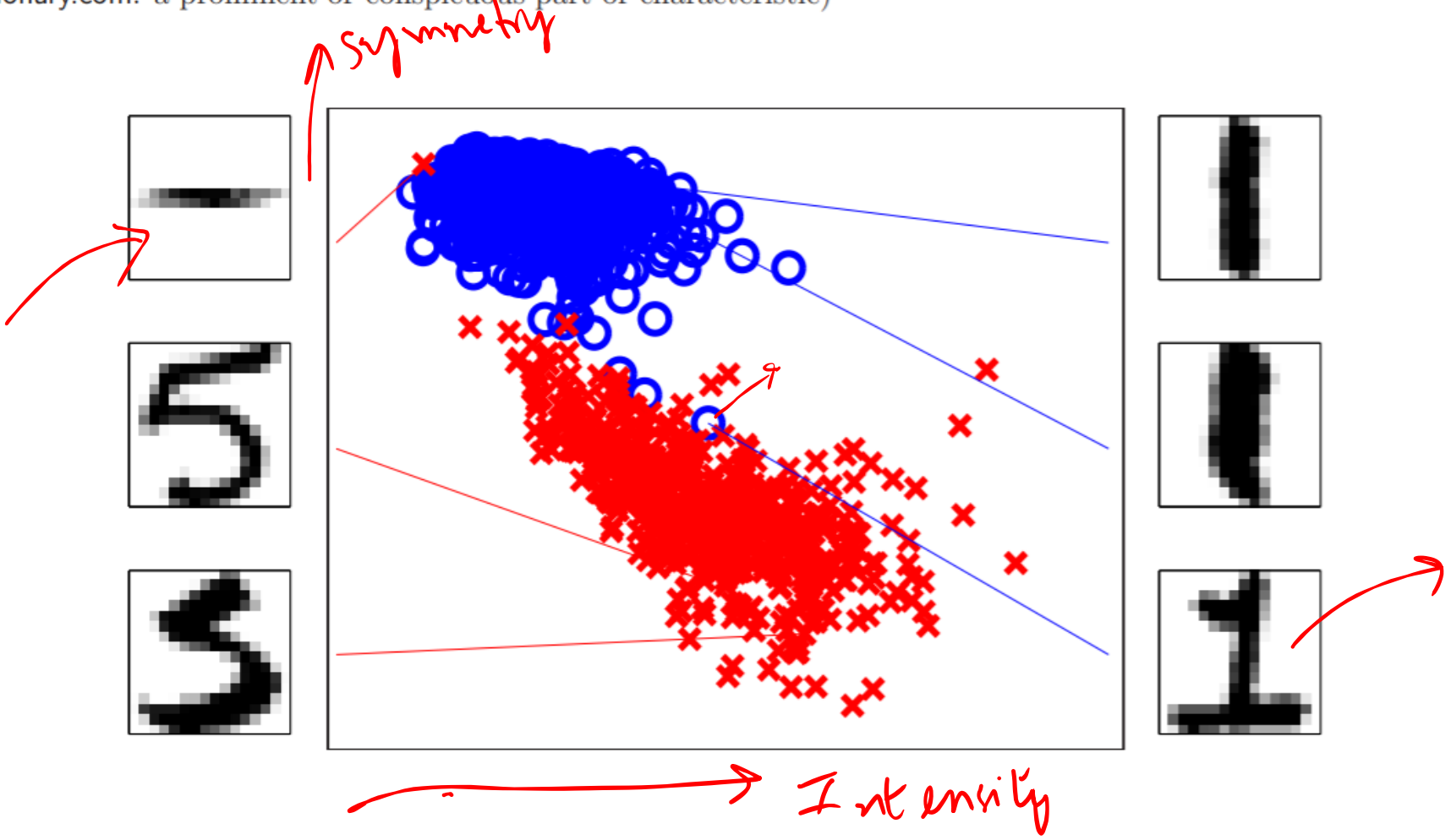
Linear Model

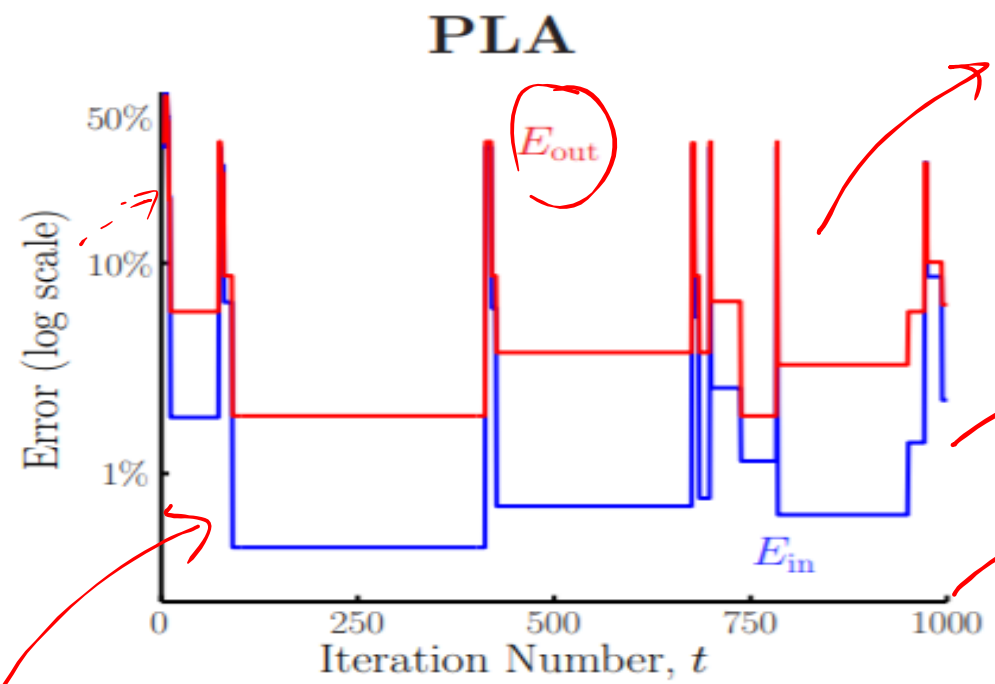
- Powerful . $E_{in} \simeq E_{out}$

  Feature Construction.

Intensity and Symmetry Features

feature: an important property of the input that you think is useful for classification.
(dictionary.com: a prominent or conspicuous part or characteristic)





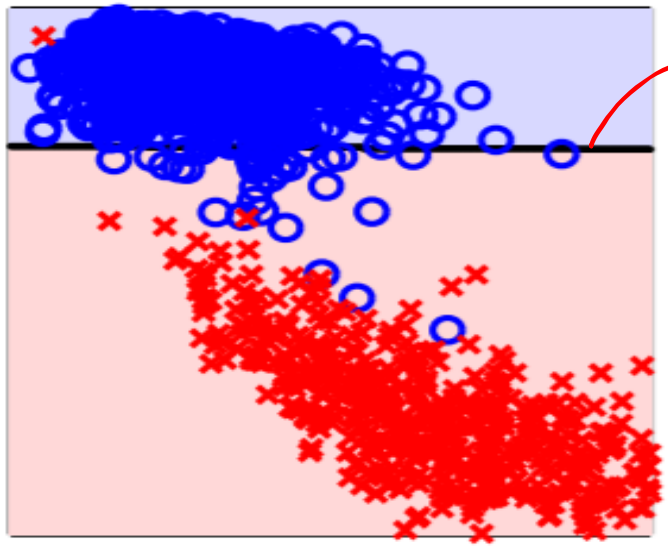
$d_{VC} = 3$

$E_{in} \simeq E_{out}$

tight

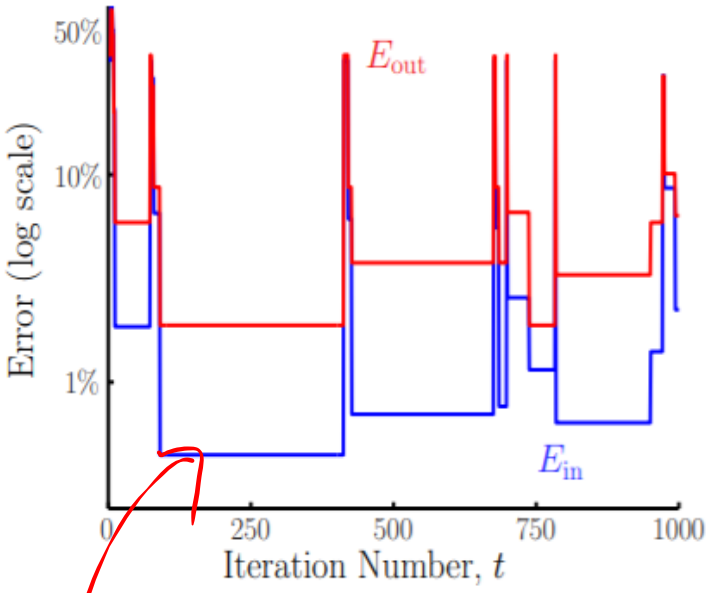
sync

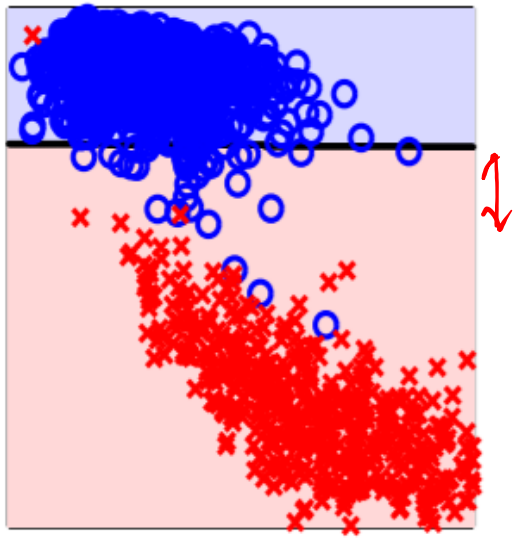
500 situations



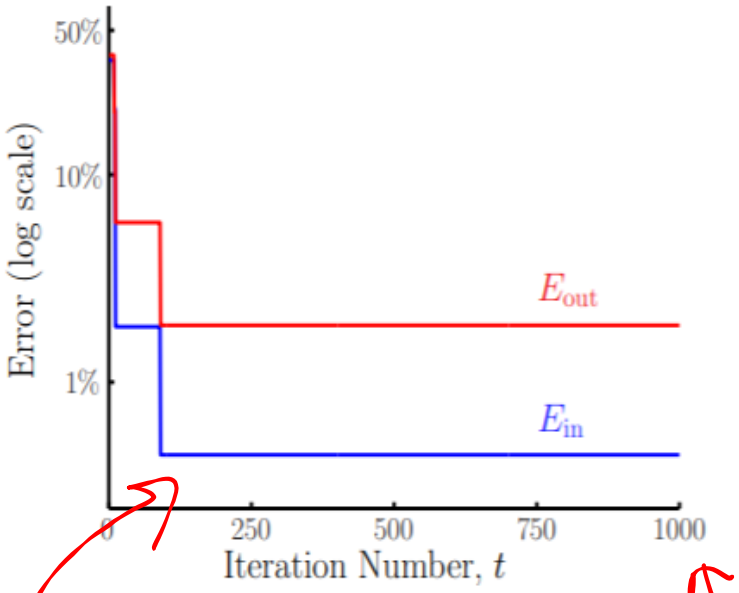
classifier

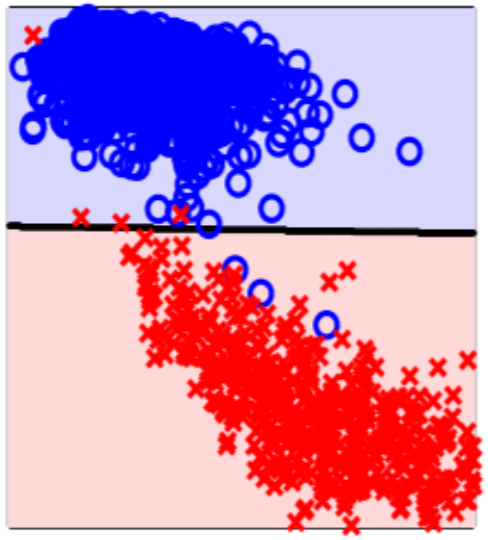
PLA


50%
10%
1%
Error (log scale)
E_{out}
E_{in}
0
250
500
750
1000
Iteration Number, t



Pocket


50%
10%
1%
Error (log scale)
E_{out}
E_{in}
250
500
750
1000
Iteration Number, t



Linear Regression

(credit card)

-

age	32 years
gender	male
salary	40,000
debt	26,000
years in job	1 year
years at home	3 years
...	...

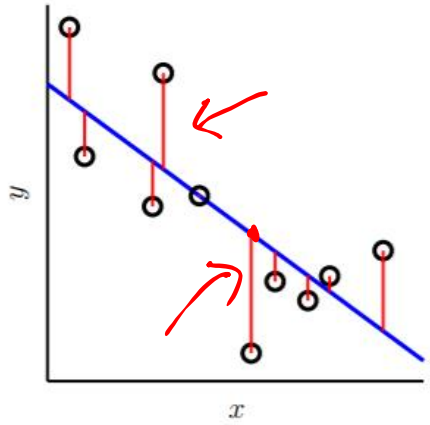
Classification: Approve/Deny

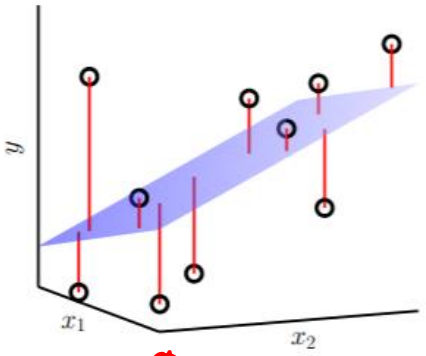
Regression: Credit Line (dollar amount)

regression $\equiv y \in \mathbb{R}$

$$h(\mathbf{x}) = \sum_{i=0}^{d} w_i x_i = \mathbf{w}^{\mathrm{T}}\mathbf{x}$$

Least Squares Linear Regression





$y = f(\mathbf{x}) + \epsilon$ ⟵ noisy target $P(y|\mathbf{x})$

in-sample error $E_{\text{in}}(h) = \frac{1}{N}\sum_{n=1}^{N}(h(\mathbf{x}_n) - y_n)^2$

out-of-sample error $E_{\text{out}}(h) = \mathbb{E}_{\mathbf{x}}[(h(\mathbf{x}) - y)^2]$

MSE

$h(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}$

$$h(x) = \boxed{w^T x}$$

$$E_{in}(w) = \frac{1}{N}\sum_{i=1}^{N}\left(h(x_i) - y_i\right)^2$$

(std. regression error)

$$= \frac{1}{N}\sum_{i=1}^{N}\left(w^T x_i - y_i\right)^2$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2 \quad \rightarrow (eq)$$

$$\left\{ \hat{y}_i = w^T x_i \right.$$

MATRIX NOTATION

$$\underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad \underline{\hat{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{bmatrix}$$

$$X = \begin{bmatrix} \text{---} & x_1^T & \text{---} \\ \text{---} & x_2^T & \text{---} \\ & \vdots & \\ \text{---} & x_N^T & \text{---} \end{bmatrix}$$

$$X \in \mathbb{R}^{N \times (d+1)}$$

$$(\hat{y}_i - y_i)^2$$

$$E_{in}(w) = \frac{1}{N} \| \hat{y} - y \|^2$$

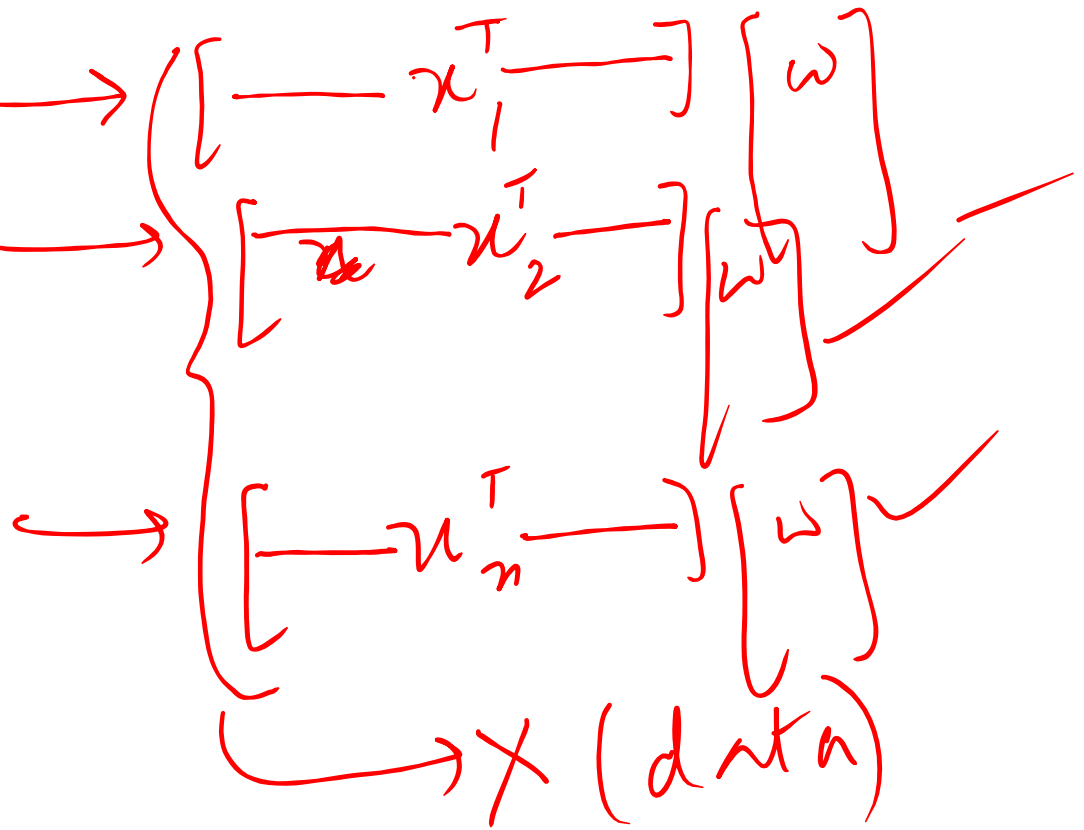
$\hat{y}_1 = w^T x_1 = x_1^T w$

$\hat{y}_2 = w^T x_2 = x_2^T w$

$\hat{y}_n =$

$\hat{y} = Xw$ (Verify)

$\therefore E_{in}(w) = \frac{1}{N} \| Xw - y \|^2$ → LA def.



$$\|z\|^2 = z^T z$$

$$\therefore E_{in}(w) = \frac{1}{N}(Xw - y)^T (Xw - y)$$

$$= \frac{1}{N}\left(w^T X^T X w - w^T X^T y - y^T X w + y^T y\right)$$

$$= \frac{1}{N}\left(w^T X^T X w - 2 w^T X^T y + y^T y\right)$$

$$\therefore \text{Set } \nabla E_{in}(w) \longrightarrow 0$$

$$E_{in}(w) = \frac{1}{N}\left(w^T X^T X w - 2 w^T X^T y + y^T y\right)$$

($y^T y$: constant)

$ax^2 + bx + c \longrightarrow 2ax + b$

$$\nabla E_{in}(w) = \frac{1}{N}\left(2X^T X w - 2X^T y\right) \longrightarrow 0$$

Normal eqns.

$$X^T X w = X^T y$$

$$w_{lin} = \left(X^T X\right)^{-1} X^T y$$

($X^T X$ is not invertible → pseudo inverse)

Pseudo Inverse Algo.

(g)

Summarize Psendo Inverse Algorithm

① Put your data in matrix form

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad X = \begin{bmatrix} — & x_1^T & — \\ — & x_2^T & — \\ & \vdots & \\ — & x_n^T & — \end{bmatrix}$$

② Appy psendo inverse Algorithm.

$$w_{lin} = (X^T X)^{-1} X^T y \xrightarrow{O(Nd^2 + d^3)} w_{lin} = X^\dagger y$$

$$X^\dagger = (X^T X)^{-1} X^T$$

least

③ $g(x) = w_{lin}^T x$

$g(x_i) \simeq y_i$

$$E_{out}(g) = E_{in}(g) + O\left(\frac{d}{N}\right)$$

Pocket Algorithm

$E_{in} \simeq 0$ (Hard)

Pseudo Inverse Algorithm $\longrightarrow w_{lin}$

y's $\longrightarrow +1, -1 \longrightarrow w_{lin}$

$$\text{sign}(w_{lin}^T x_i) \simeq y_i$$

$\longrightarrow$ Get w_{lin} for classification problem.

$\longrightarrow$ Make this your starting point

Regression for Classification.

Thanks!