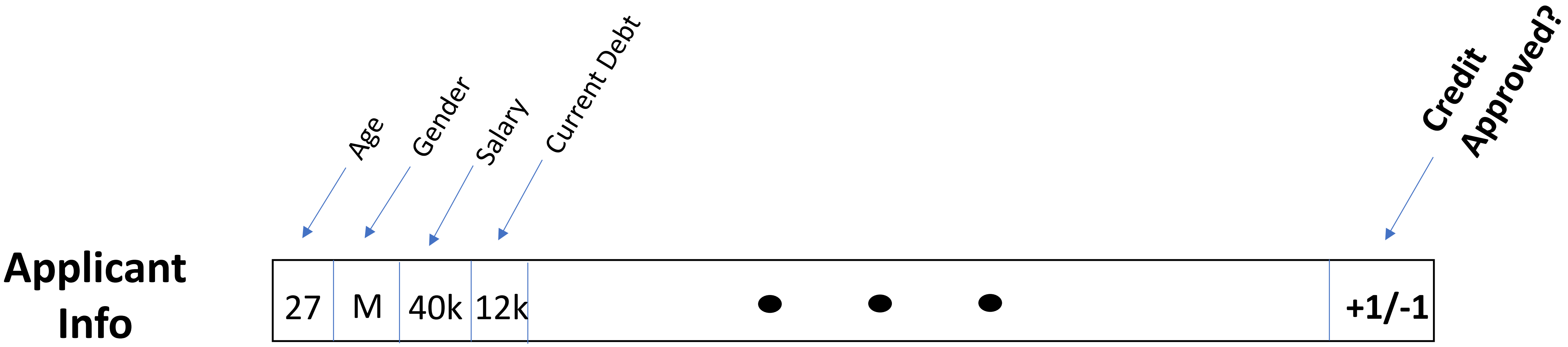
Lecture 3: Linear Model & Regression

EECS4404/5327
Introduction to Machine Learning
And Pattern Recognition

Amir Ashouri

Fall 2019

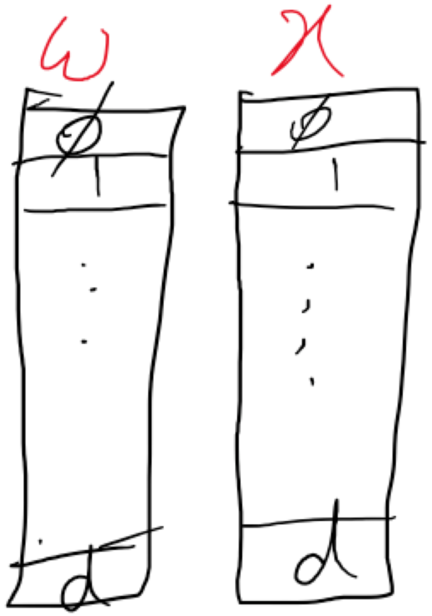
Recap (1/4)
Binary Linear Classification



Recap (2/4)
Perceptron Learning Algorithm (PLA)

$$h(x) = \mathbf{sign}(\sum_{\mathbf{i=0}}^{\mathbf{D}} \mathbf{w_i x_i})$$

$$h(x) = sign(w^T x)$$

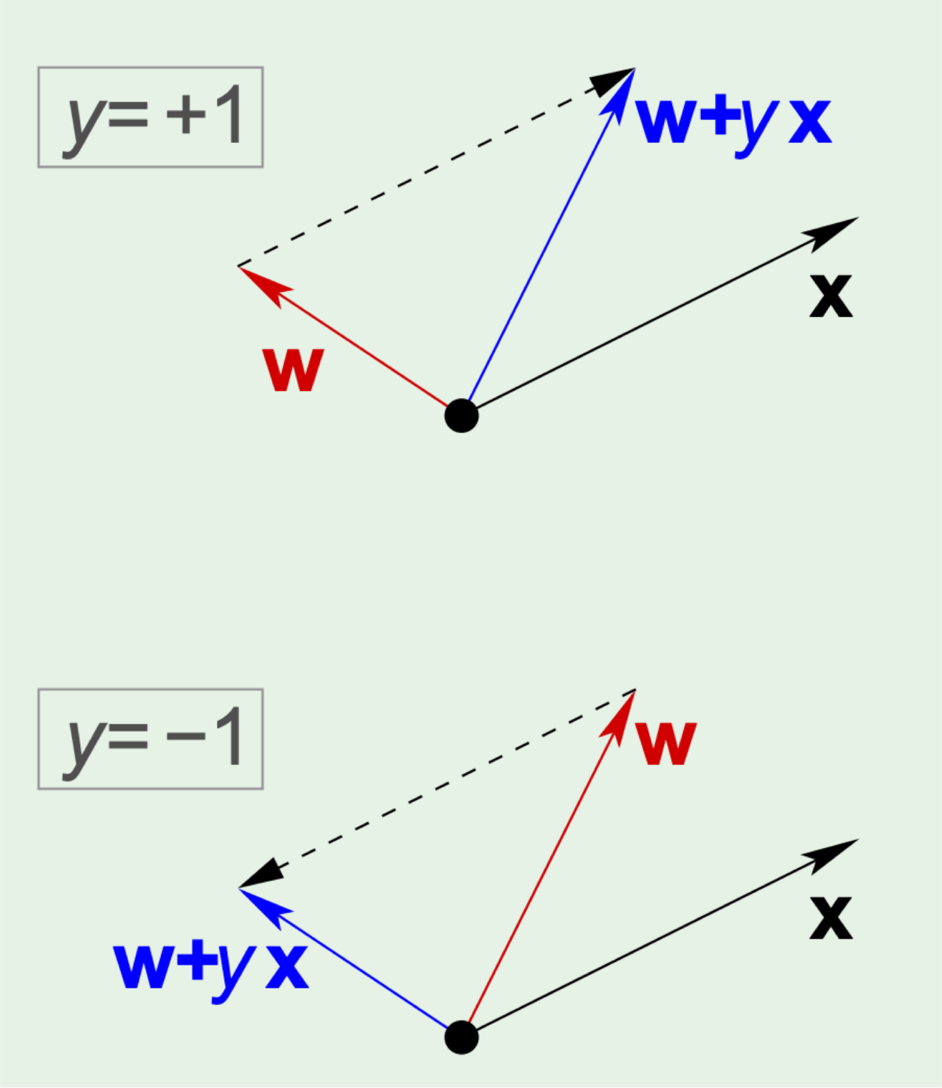


Recap (3/4)
Misclassifications and updates

In a binary linear classification, there are two possibilities:

$$sign(w^T x_i) \neq y_i$$

1. $y_i = +1$ for a $t_i = -1$
2. $y_i = -1$ for a $t_i = +1$



Recap (4/4)
PLA Algorithm

Input: $D = ((\mathbf{x}_1, t_1), \ldots (\mathbf{x}_N, t_N))$

Initialize: $\mathbf{w}^1 = \mathbf{0}$

For $t = 1, 2, \ldots$:

If there exits an i with $y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \leq 0$ (a misclassified point)
then update: $\mathbf{w}^{y+1} = \mathbf{w}^y + y_i \mathbf{x}_i$

Output: $\mathbf{w}^y$

Good tool to visualize PLA:
https://lecture-demo.ira.uka.de/neural-network-demo/?preset=Rosenblatt%20Perceptron

Outline
Lecture 3

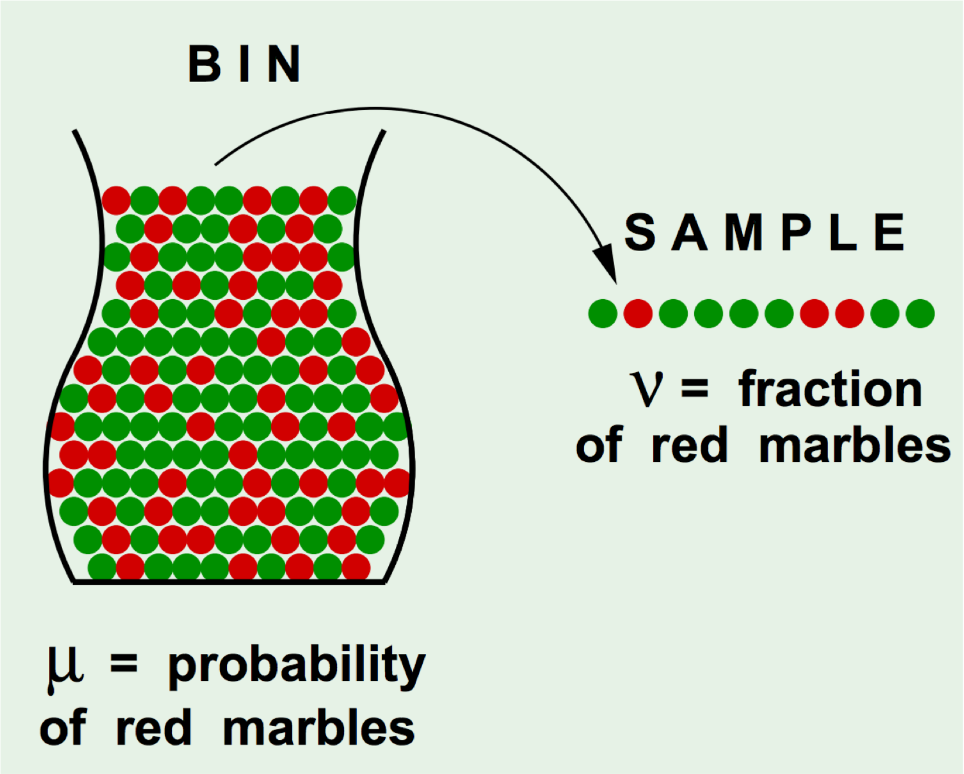
- Learning Notion
- Input Representation
- Pocket Algorithm
- Linear Regression (LR)
- Nonlinear Transformation

Feasibility of Learning
A Bin of Marbles

$\mathbb{P}$ [picking a **red** marble] = μ

$\mathbb{P}$ [picking a **green** marble] = $1 - \mu$

The value of μ is *unknown* .



Experiment:

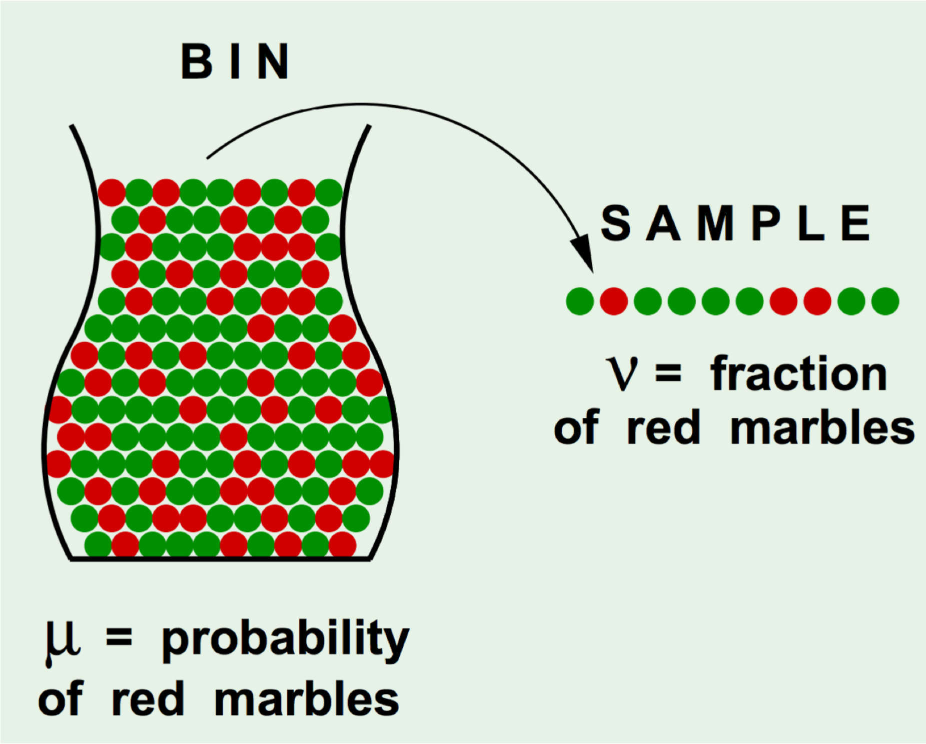
We pick ***N*** marbles independently.

The fraction of Red marbles in sample = ϑ

Relation Between μ and ϑ

Question 1

- Does ϑ say anything about μ ?
  - **NO**, Samples can be mostly red while the bin was mostly green
  - **However,** the sample frequency of these two are likely close to each other.



Question 2

- What does ϑ say about μ ?
  - In a big sample (large ***N***), ϑ is probably close to μ within a margin (ε)

Hoeffding's Inequality[1]

[1] Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, *58*(301), 13-30.

In a big sample (large N), ϑ is probably close to μ within ε

$$\mathbb{P}\left[\,|\vartheta - \mu| > \epsilon\,\right] \leq 2e^{-2\epsilon^2 N}$$

In other word, the statement "$\mu = \vartheta$" is **P.A.C**

Probably approximately correct

Notation for Learning

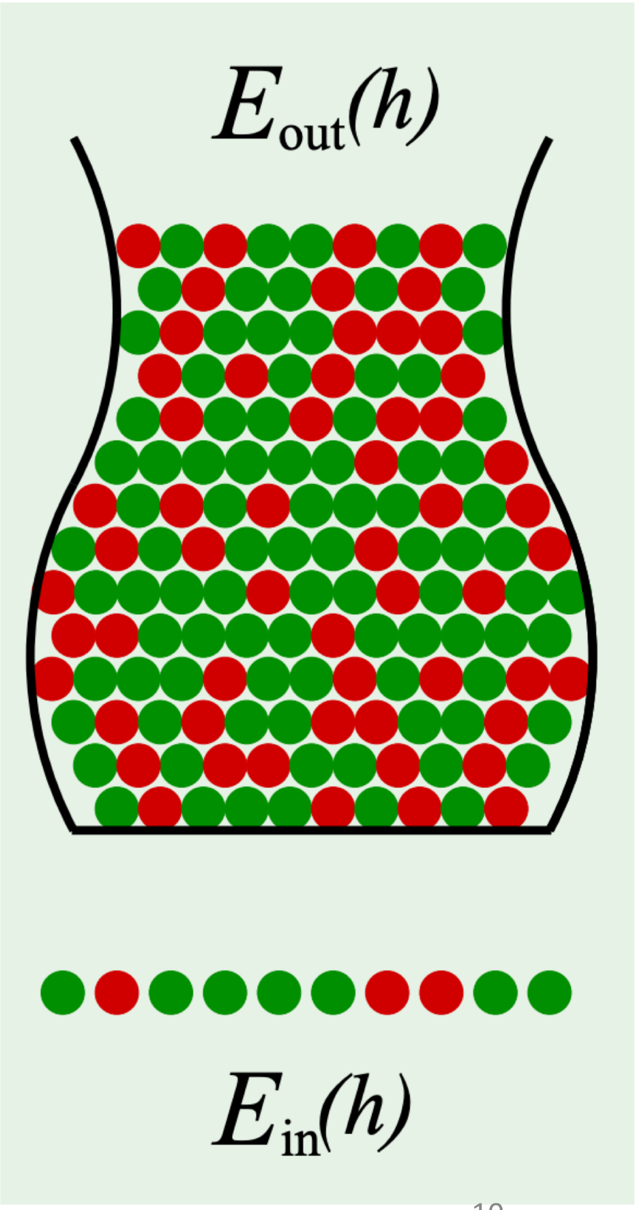
Both μ and ϑ depend on which hypothesis (h)

ϑ is in sample $\longrightarrow E_{in}(h)$

μ is "out of sample" $\longrightarrow E_{out}(h)$

Thus, Hoeffding inequality becomes:

$$\mathbb{P}\left[\,|E_{in}(h) - E_{out}(h)| > \epsilon\,\right] \le 2e^{-2\epsilon^2 N}$$



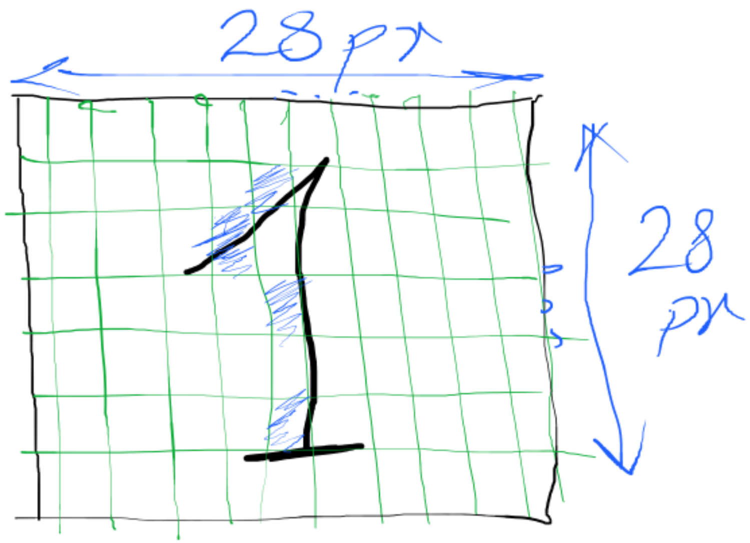
MNIST Dataset[1]

SVHN Dataset[2]

[1] LeCun, Yann. "The MNIST database of handwritten digits." http://yann. lecun. com/exdb/mnist/ (1998).
[2] http://ufldl.stanford.edu/housenumbers/

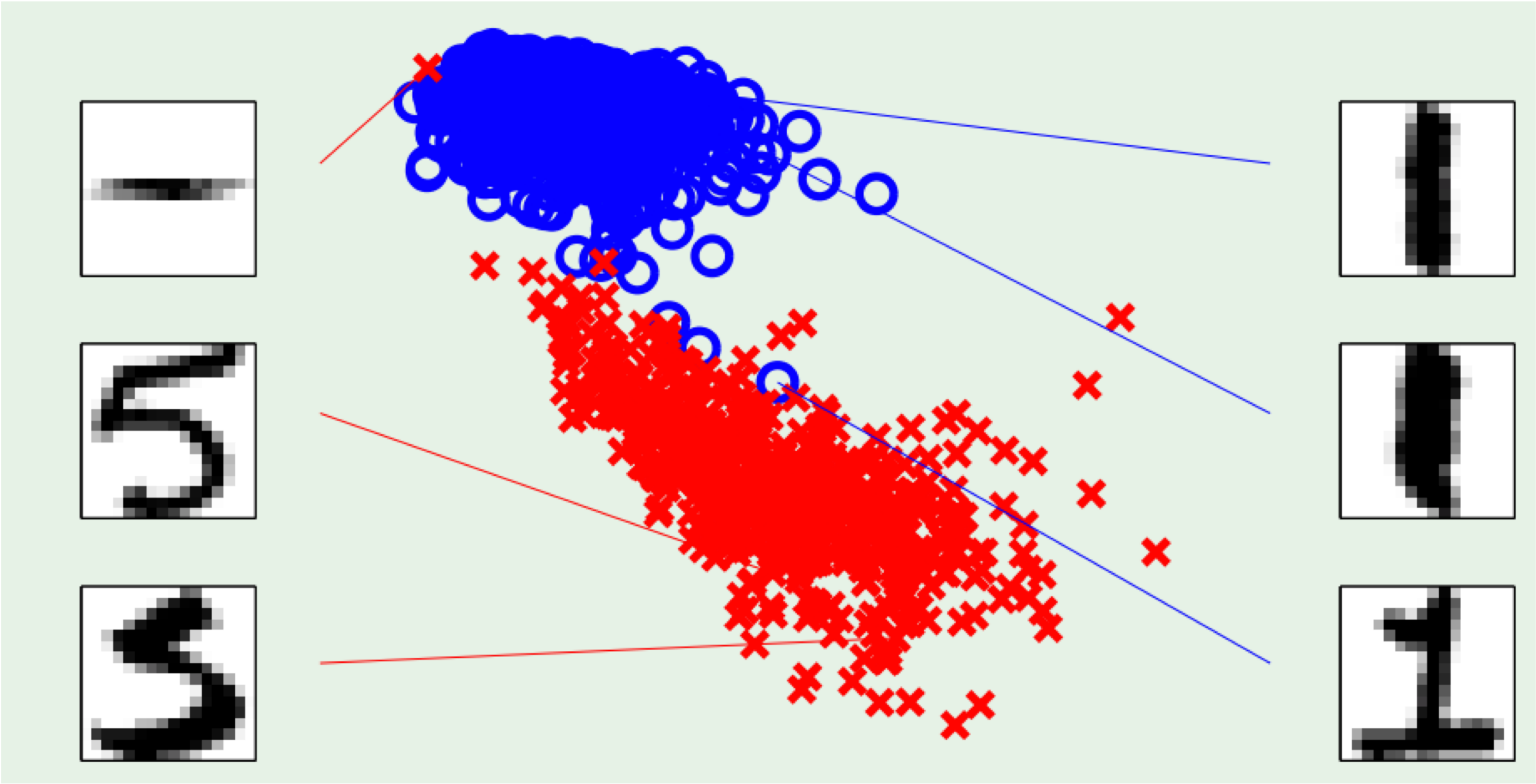
Representation

- Input
  - Each image is a 28*28 pixel
  - **X** =(x_0,x_1,x_2,…,x_{784})
- Model
  - Linear Model weights: (w_0,w_1,…,w_{784})
- Features
  - Downsizing the large vector of input:
    - Capturing only certain metrics instead of the raw data(e.g., *intensity*, *symmetry* (vertical, horizontal, diagonal), *sharpness*, etc.)

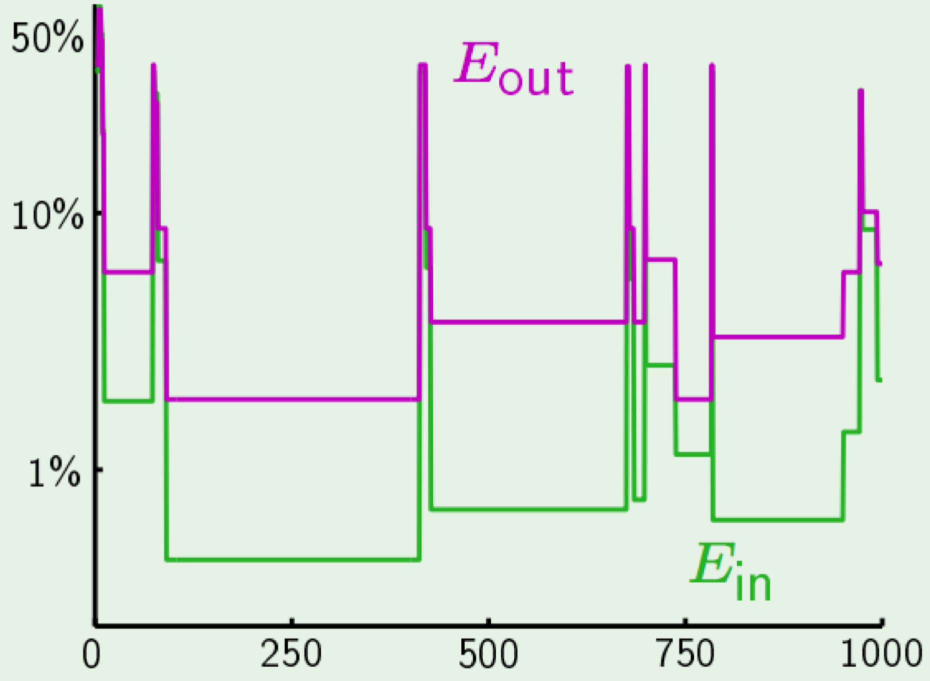


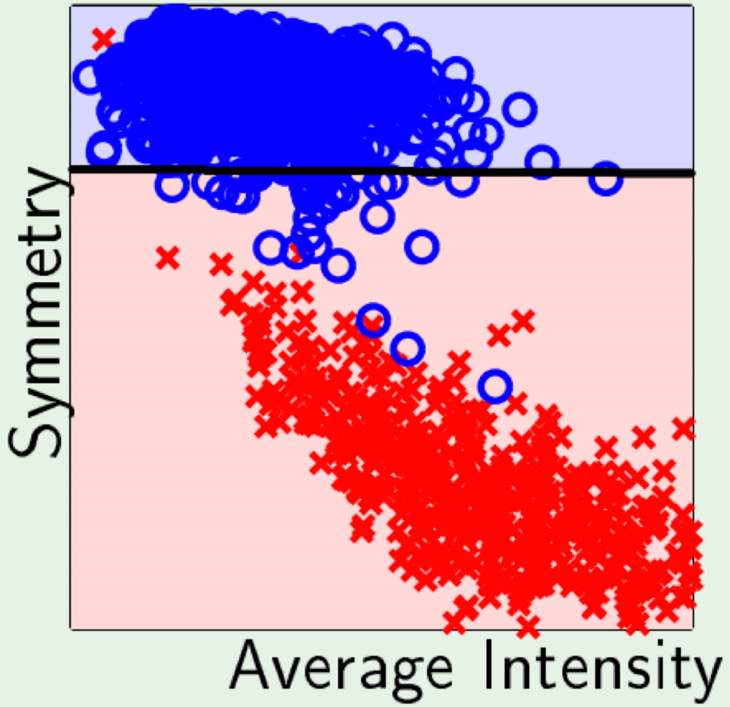
$$\text{linear model: } (x_0, x_1, x_2)$$

Representation (2)
Case of 1’s vs. 5’s



Applying PLA





Rosenblatt Theorem (1957)

Let w^* be the output of the PLA on a linearly separable dataset D. The PLA terminates in almost:

$$T \leq \frac{R}{P^2}$$

$$R = \max_{1 \leq n \leq N} ||X_n||^2$$

$$P = \min_{1 \leq n \leq N} \frac{|W^T X_n|}{||W*||}$$

R: Radius of dataset

P: Distance of D to the decision boundary

w* = margin

Pocket Algorithm

It is helpful when our $D = \{(x_1, y_1), \ldots, (x_n,y_n)\}$ is **not** linearly separable. Since PLA is not guaranteed to terminate.

Pocket algorithm, keep the "best weight vector" found up to iteration t in the pocket. It only replaces it if a better weight vector was found.

Pocket Algorithm Steps

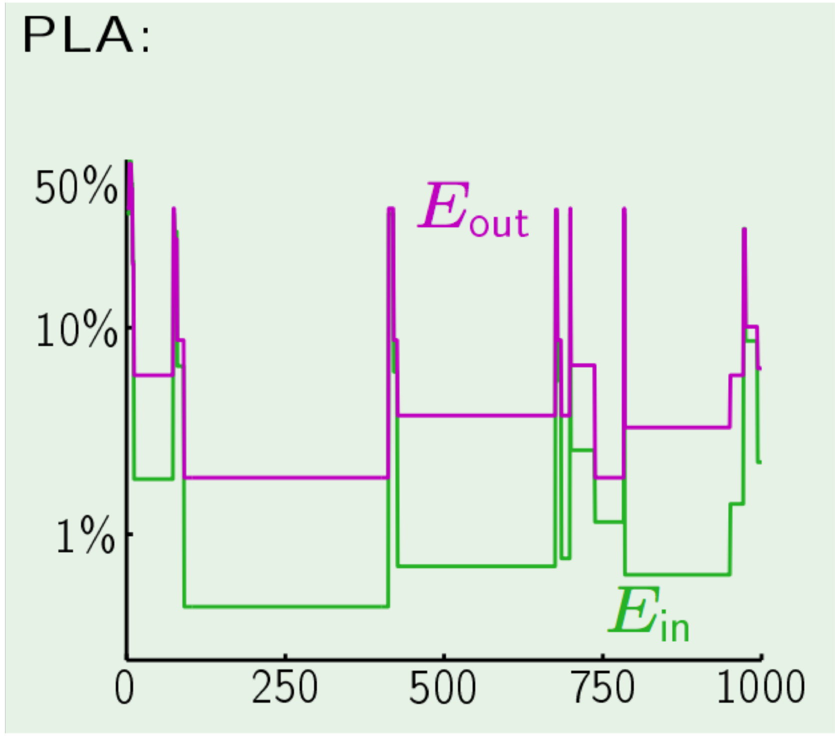
1) Set the pocket weight vector ($\hat{\underline{w}}$) to $\underline{w}(0)$ of PLA
2) For t = 0, 1, 2, . . . , t-1 do:

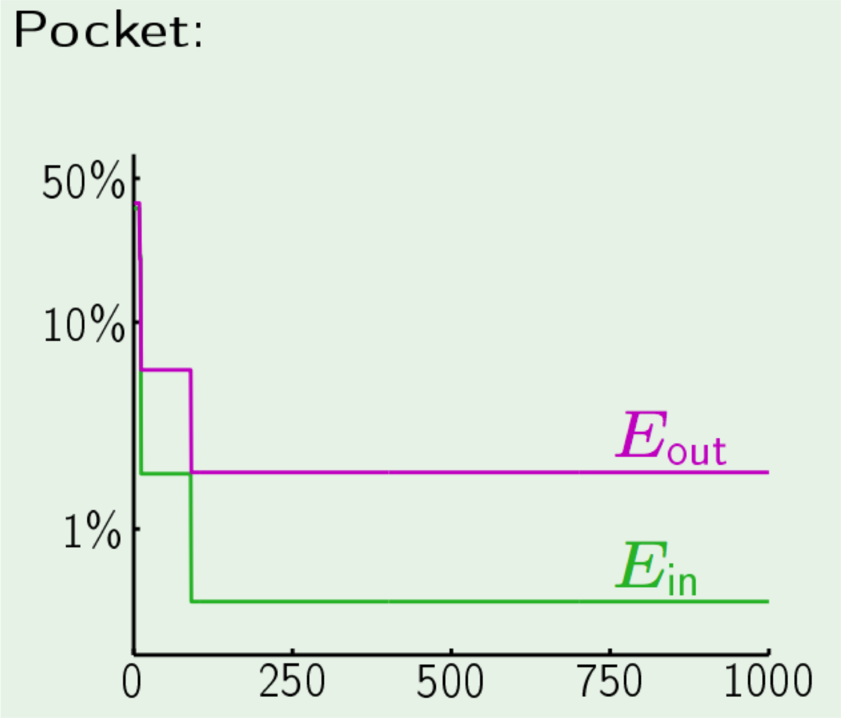
   - Run PLA for one update to get $\underline{w}(t+1)$
   - Evaluate $E_{in}(\underline{w}(t+1))$
   - If $E_{in}(\underline{w}(t+1)) \leq E_{in}(\underline{w}) \Rightarrow \hat{\underline{w}} = \underline{w}(t+1)$

3) Return $\hat{\underline{w}}$ at the end

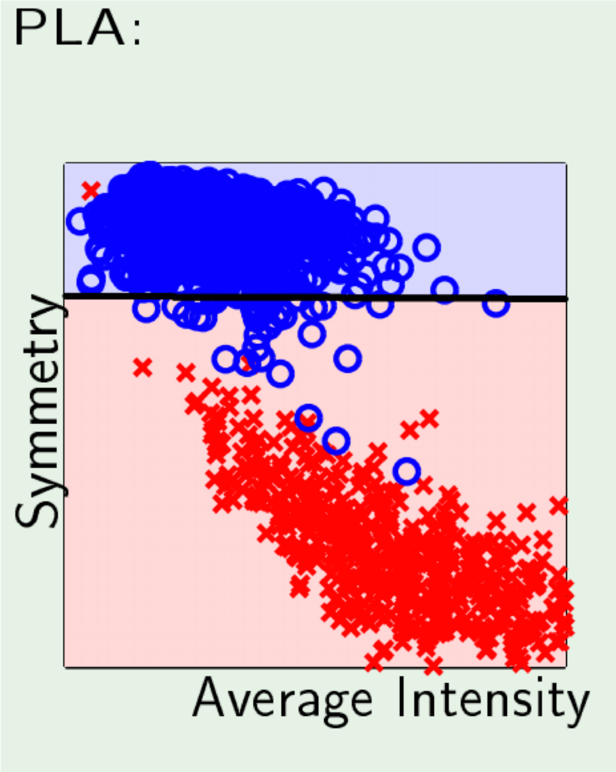
The Pocket Algorithm

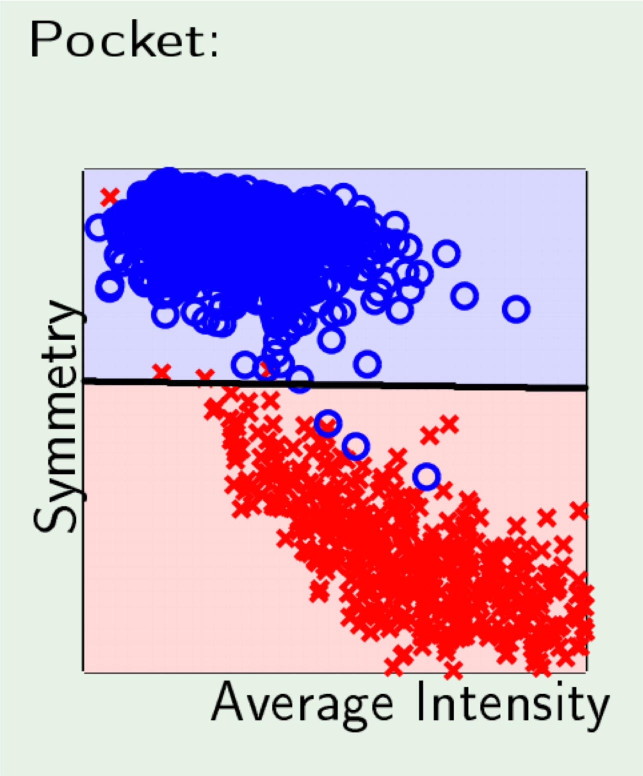
The algorithm saves the best found result until a better result is reached:





The Pocket Algorithm(2)
Classification Comparison





Credit Approval Revisited

Classification vs. Regression

Input

Each Customer Representative Features (Age, Salary, etc.)

$\mathbf{X} = (x_0, x_1, x_2, \ldots, x_d)$

Linear Regression Output:

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

Example #2
Exam Marks

Say we want to predict the mark on the exam of a student in this class. For a student, we collect the following “measurements”:

- x1 = number of hours they studied
- x2 = number of hours of sleep
- x3 =age
- x4 = height
- x5 = amount of alcohol consumed

- Our homegrown predictor:

$$\text{mark on exam} = \text{b} + 1 \cdot x_1 + .2x_2 + 0 \cdot x_3 + 0 \cdot x_4 + (-2) \cdot x_5$$

Will is work well on unseen data?

LR Formalization

Training Set $$D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$$

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

Prediction Function $$\hat{y} = h(\underline{\mathrm{x}})$$

Linear Model

$$(\underline{\mathrm{x}}) = w_0 + w_1 x_1 + \cdots + w_d x_d)$$

$$\underline{\mathrm{w}} = (w_0, w_1, \ldots, w_d) = \sum_{i=0}^{d} w_i x_i \quad (x_0 = 1)$$

$$\underline{\mathrm{x}} = (x_0, x_1, \ldots, x_d) = \underline{\mathrm{w}}^T \underline{\mathrm{x}}$$

Square Error vs. Absolute Error

- Square error provides better properties:

1. If X is a random variable (e.g., toss a coin), the estimator that minimizes the square error is **mean**, whereas **median** for absolute error.

   If **mean** → E(X+Y) = E(X) + E(Y)
   If **median** → E(X+Y) =! E(X) + E(Y)

2. If X is an independent variable (e.g., age, time, etc.):

   If **Sq.Err** → Var(X+Y) = Var(X) + Var(Y)
   If **Abs.Err** → Var(X+Y) =! Var(X) + Var(Y)

See more info about random variables property:
http://facweb.cs.depaul.edu/sjost/csc423/documents/rv-props.htm

LR Error Estimation

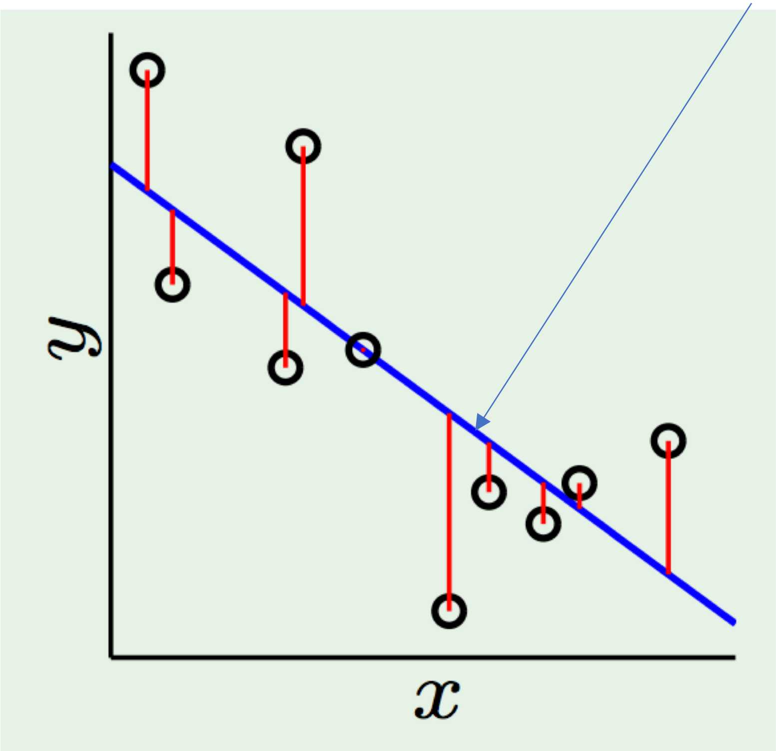
we need to compute average square error

$$E_m(\underline{\mathrm{w}}) = \frac{1}{N}\sum_{i=1}^{N}\underbrace{(y_i - \underline{\mathrm{w}}^T\underline{\mathrm{x}}_i)^2}_{e_i(w)}$$

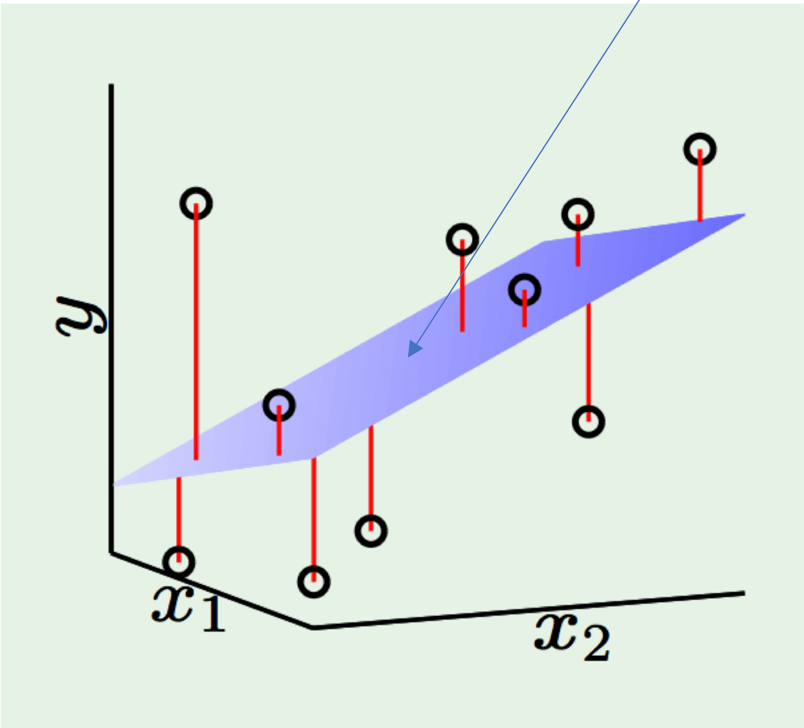
$e_i(w) =$ squared error on ith training example

Measuring Error
Linear Regression Illustration

2D Dimension



3D Dimension

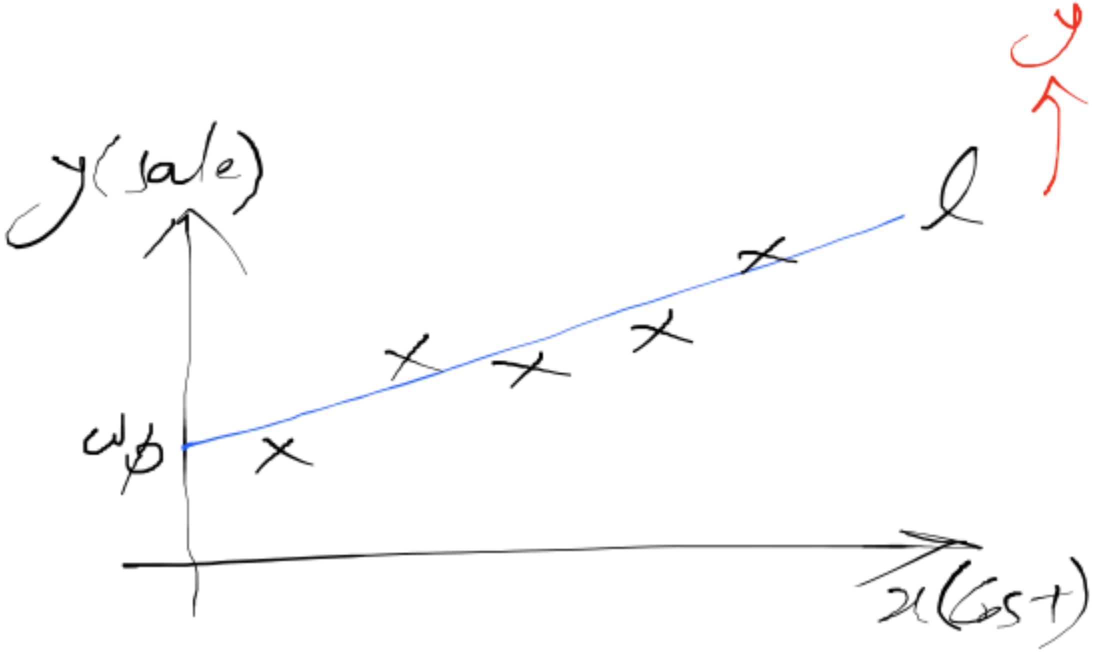


Example
Linear Regression

x = advertising cost in one week

y = sales in one week

historical data D; d = 1



We need to fit a linear model: $y = w_0 + w_1 x$

w_0 = sales when x = 0

w_1 = increase in sales, for unit increase in cost

Refined Model

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \text{TV ads (\$)} \\ \text{radio ads (\$)} \\ \text{newspaper ads (\$)} \end{bmatrix}$$

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

largest $w_i \Rightarrow$ most profitable x_i

Design Matrix

To obtain a concise notation, we write the collection of data points as rows of a matrix:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \dots \\ \dots \\ \mathbf{x}_N \end{pmatrix} = \begin{pmatrix} x_{10} & x_{11} & \dots & x_{1D} \\ x_{20} & x_{21} & \dots & x_{2D} \\ \dots & & & \\ \dots & & & \\ x_{N0} & x_{N1} & \dots & x_{ND} \end{pmatrix}$$

This is also called the **design matrix**.

Least Squares

- It is a standard approach in regressions to approximate the solution of problem.

- There are many least square methods:

1. MLE (Maximum likelihood Estimation)
2. MAP (Maximum A posteriori Probability)
3. Analytical Solution
4. Geometric Interpolation
5. ,…

Minimizing Error in LR

linear systems of equations: (i = 1, 2, 3, …, N)

$$y_i = w_0 + w_1 x_{i,1} + w_2 x_{i,2} + \cdots + w_d x_{i,d}$$

# of equations (N) { $y_1 = \ldots$; … ; $y_N = \ldots$ } — # of variables (d+1)

if $d+1 \geq N$: An exact solution exist (model is consistent)

otherwise: No exact solution $\xrightarrow{\text{approximate by}}$ $\min \sum_{i=1}^{N} (y_i - w^T x_i)^2$

Minimizing LR Error

Given D, find $\underline{\mathrm{w}} \in \mathbb{R}^{d+1}$ to minimize $E_{in}(\underline{\mathrm{w}})$

1. Analytic solution
2. Geometric solution

Reading: PRML 3.1.1, 3.1.2, 3.1.4