Generative model

(1) learn CCD from data $p(x|y)$

(2) BDR to get classifier $p(y|x) = \frac{p(x|y) \cdot p(y)}{p(x)}$

- Note: data is used only in step 1 to get CCD. Classifier is secondary.
- Density estimation is an ill-posed problem. (Gaussian, GMM, gamma, ...?)

Vapnik advice: "When solving a given problem, avoid solving a more general problem as an intermediate step."

Discriminative solution: Solve the decision boundary or $p(y|x)$ directly.

"use the data to learn to discriminate classes, rather than generating data"

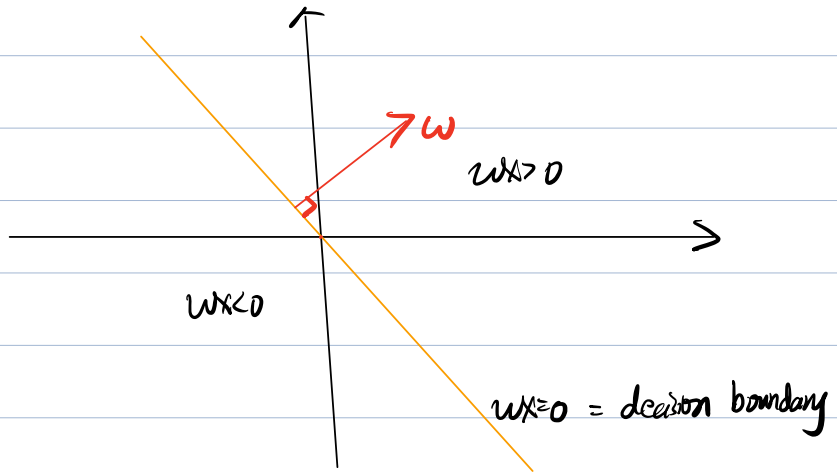
Linear classifier

output: $y \in \{+1, -1\}$ binary class

input: $x \in \mathbb{R}^d$

Linear function $f(x) = w^T x$

w separates the space into 2 half-spaces



Decision Rule

$$y^* = \text{sign}(w^T x) = \begin{cases} +1, & w^T x \geq 0 \\ -1, & w^T x < 0 \end{cases}$$

Note bias can be included as another dimension

$$\tilde{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}$$
$$\tilde{w} = \begin{bmatrix} w \\ b \end{bmatrix}$$

Training Set

$$D = \{(x_i, y_i)\}_{i=1}^{n}$$
$$D = \{X, y\}, \quad X = [x_1, \ldots, x_n]$$
$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Note: Given w, $y_i w^T x_i > 0$ $\Rightarrow$ in general, idea is that we successfully predict.

a) $+1$ >0

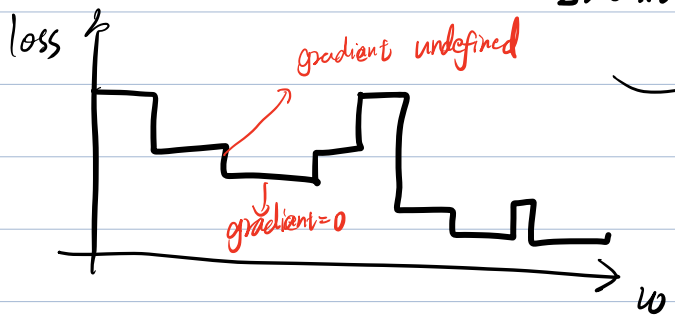
b) -1 <0

$y_i w^T x_i < 0$ $\Rightarrow$ misclassified x_i

Ideal case: 0-1 loss

opt # of misclassified samples.

$$w^* = \arg\min_w \sum_i^n \begin{cases} 0, & y_i w^T x_i > 0 \\ 1, & y_i w^T x_i < 0 \end{cases}$$



(not continuous)

no opt algorithm can optimize it.

Hence, alternatively

Least Square classification (Label regression)

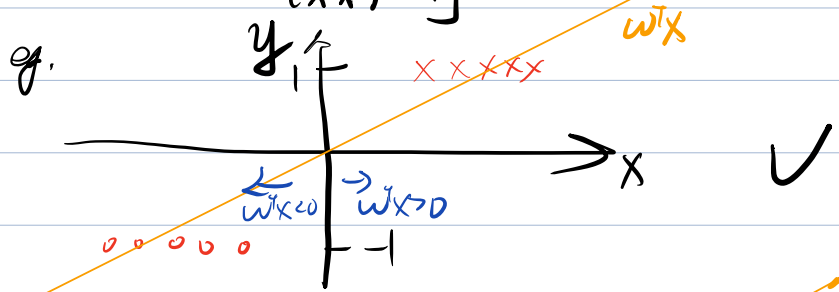
- Ignore the fact y is discrete.

$$\hat{w} = \arg\min_w \|y - X^T w\|^2$$

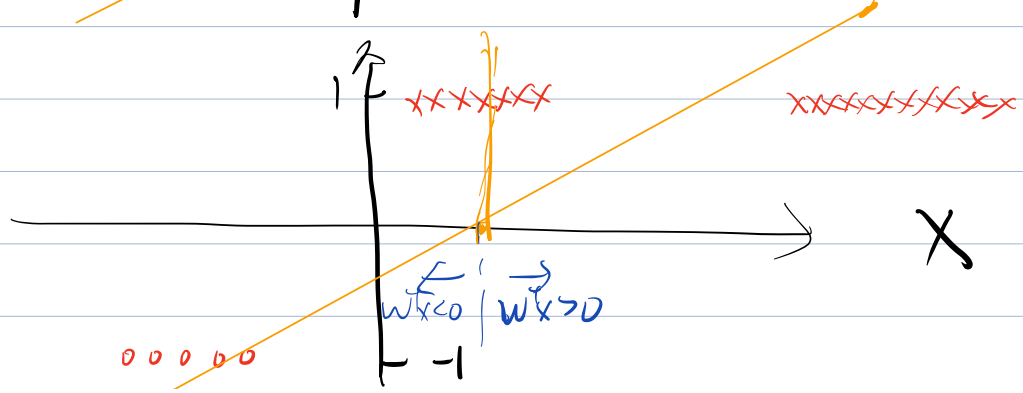
$$\downarrow$$

$$\hat{w} = (XX^T)^{-1} Xy$$

eg.



✓



- not robust to outliers
- squared error penalizes predictions that are "too correct"

Perceptron (Rosenblatt, 1962)

Perceptron criteria - only look at misclassified points

$$E(w) = \sum_{i \in M} -y_i w^T x_i$$

M: misclassified points

higher loss for x_i that are badly misclassified: $y_i w^T x_i << 0$

Perceptron Algorithm

$$w^* = \text{argmin}_w E(w) = \text{argmin}_w \sum_{i \in M} -y_i w^T x_i$$

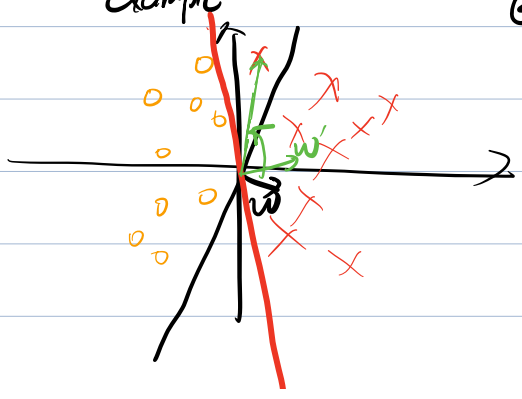
- Computers were slow in 60's
- Apply "stochastic gradient descent" (SGD)
  - use one data point at a time:

$$w^{(t+1)} = w^{(t)} + \eta y_i x_i$$

η: learning rate

① rotate w towards the positive class

Example



② The length of w increases

which means each point has diminishing effect.

- Rosenblatt proved that it converges

Logistic Regression (probabilistic approach)

- Consider 2-class problems $y \in \{0, 1\}$
- PRML 4.2.2 for cases of Gaussians
  $\Rightarrow$ posterior $p(y|x)$