# Machine Learning from Data

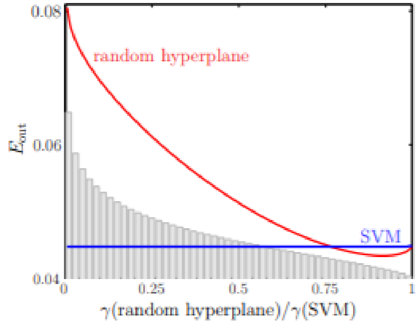
Lecture 25: Spring 2021

# Today’s Lecture

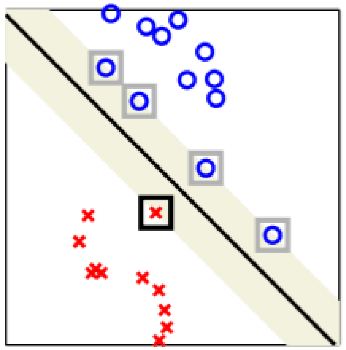
- The Kernel Trick ✓

RECAP: # Large Margin is Better

## Controling Overfitting

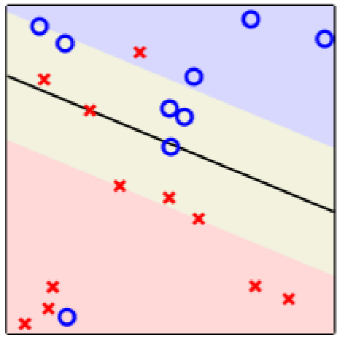



**Theorem.** $d_{\text{VC}}(\gamma) \leq \left\lceil \frac{R^2}{\gamma^2} \right\rceil + 1$



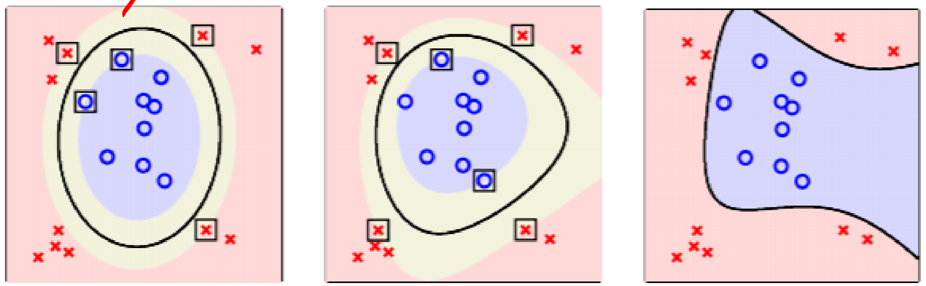
$$E_{\text{cv}} \leq \frac{\#\ \text{support vectors}}{N}$$

## Non-Separable Data



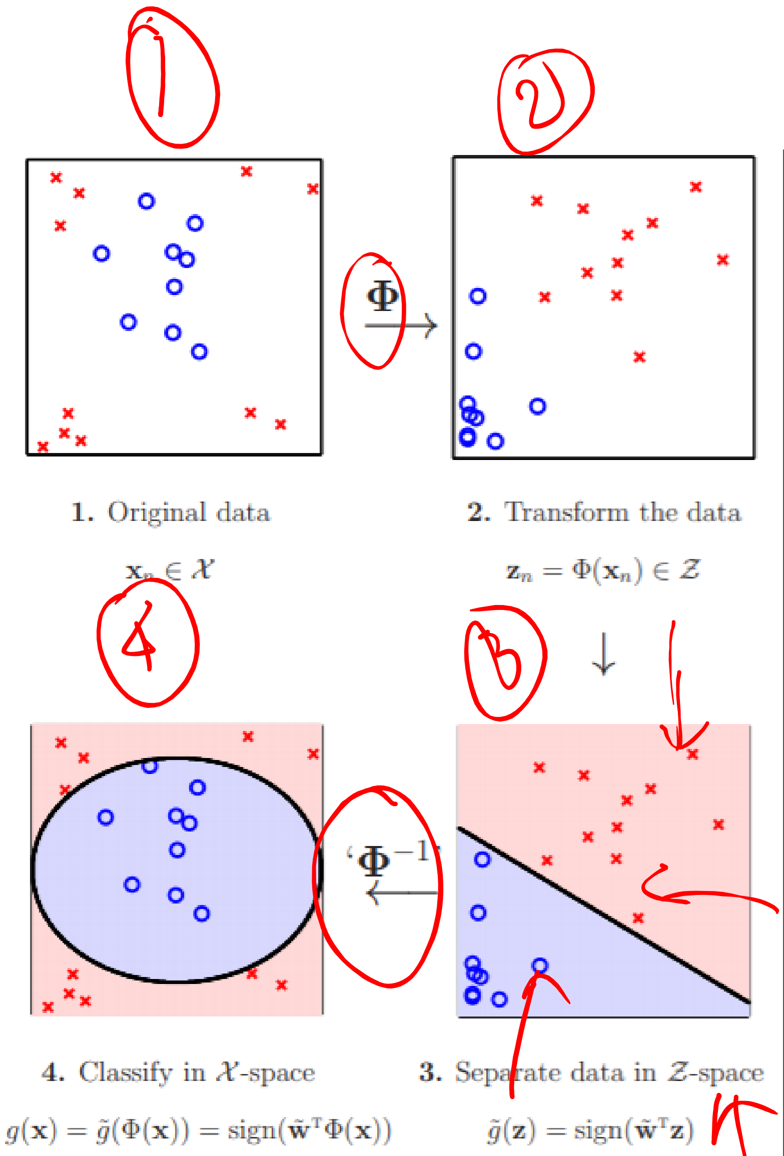
$$\begin{aligned} &\underset{b,\mathbf{w},\boldsymbol{\xi}}{\text{minimize}} && \tfrac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w} + C\sum_{n=1}^{N}\xi_n \\ &\text{subject to:} && y_n(\mathbf{w}^{\mathsf{T}}\mathbf{x}_n + b) \geq 1 - \xi_n \\ & && \xi_n \geq 0 \qquad\qquad \text{for } n = 1, \ldots, N \end{aligned}$$

5 +1 6



$\Phi_2$ + SVM $\Phi_3$ + SVM $\Phi_3$ + pseudoinverse algorithm

Complex hypothesis that does not overfit because it is 'simple', controlled by only a few support vectors.




| $\mathcal{X}$-space is $\mathbb{R}^d$ | $\mathcal{Z}$-space is $\mathbb{R}^{\tilde{d}}$ |
| --- | --- |
| $\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$ | $\mathbf{z} = \Phi(\mathbf{x}) = \begin{bmatrix} 1 \\ \Phi_1(\mathbf{x}) \\ \vdots \\ \Phi_{\tilde{d}}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 1 \\ z_1 \\ \vdots \\ z_{\tilde{d}} \end{bmatrix}$ |
| $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ | $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N$ |
| $y_1, y_2, \ldots, y_N$ | $y_1, y_2, \ldots, y_N$ |
| no weights | $\tilde{\mathbf{w}} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{\tilde{d}} \end{bmatrix}$ |
| $d_{\text{vc}} = d + 1$ | $\boldsymbol{d_{\text{vc}} = d + 1}$ |
| $g(\mathbf{x}) = \text{sign}(\tilde{\mathbf{w}}^{\mathsf{T}}\Phi(\mathbf{x}))$ | |

Have to **transform** the data to the $\mathcal{Z}$-space.

# Today’s Lecture

- How to use nonlinear transforms without physically transforming data to Z-space?

$$\left\{\begin{array}{ll} \min\limits_{\underline{w,b}} & \frac{1}{2} w^T w \\ \text{s.t.} & y_n(w^T x_n + b) \geq 1 \quad \forall n = 1, 2 \ldots \underline{N} \end{array}\right.$$

$\underline{QP}$

→ $d+1$ optimization variables, $N$ constraints.

→ Transformed Space → Large no. of features.

<u>PRIMAL</u> $\Longleftrightarrow$ <u>DUAL</u>

↓

Lagrange's multiplier. → $\alpha$

$$\min_{\alpha} \begin{cases} \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \underbrace{(x_n^T x_m)}_{\uparrow} - \sum_i \alpha_i \\ \text{s.t. } \sum_n \alpha_n y_n = 0 \\ \alpha_n \geq 0 \end{cases}$$

$\rightarrow$ QP

Output $\rightarrow \alpha_n \, (\alpha_n^*)$

$$\underbrace{w^* = \sum_{n=1}^{N} \alpha_n^* y_n x_n}$$

$$b^* = y_s - (w^*)^T x_s$$

classification

Pick some points $\alpha_s^* > 0 \; \{SV\}$

**Dual:** $N$ optimization variables, $N+1$ constraints (simple)

Does not depend on dimensions.

$$d \longleftrightarrow N$$

**Lagrangian**

$$\rightarrow L(w, b, \alpha) = \underbrace{\frac{1}{2} w^T w}_{} + \sum_{n=1}^{N} \underbrace{\alpha_n \left(1 - y_n (w^T x_n + b)\right)}_{\downarrow 0 \ \uparrow}$$

Lagrangian $\alpha_n \geq 0$

$\min L$ w.r.t. $w, b$ ✓

or $\max L$ w.r.t. $\alpha$'s $(\alpha_n \geq 0)$ ✓

"Proof":

$$1 - y_n (w^T x_n + b) \leq 0 \quad \text{— } 1$$

$$1 - y_n (w^T x_n + b) > 0 \quad \text{— } 2 \quad \text{for each } n$$

OR

$$\alpha_n \left(1 - y_n (w^T x_n + b)\right) > 0$$

$\alpha_n \longrightarrow$ large $\longrightarrow \infty \longrightarrow L \to \infty$ X

$$\underbrace{1-y_n(w^T x_n+b) \leq 0}_{} \Rightarrow \underbrace{y_n(w^T x_n+b) \geq 1}_{}$$

$(1-y_n(w^T x_n+b)) = 0$ ✓ or $\alpha_n = 0$

or $\alpha_n(1-y_n(w^T x_n+b)) \leq 0$ $\quad \alpha_n \geq 0$

$\Rightarrow$ -ve

## Summarize

$$\alpha_n \underbrace{\left(1-y_n(w^T x_n+b)\right)}=0 \checkmark$$

$$\downarrow$$

$$y_n(w^T x_n+b)=1 \checkmark \longrightarrow x_n \text{ is } \underline{\underline{SV}}$$

$$\alpha_n=0 \checkmark$$

$$L^* = \tfrac{1}{2} w^T w \leftarrow$$

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

What happens when we solve this problem?

$$\frac{\partial L}{\partial w} \Rightarrow w - \sum_{m=1}^{N} \alpha_n y_n x_n = 0 \Rightarrow \underline{w} = \underbrace{\sum_{n=1}^{N} \alpha_n y_n x_n}$$

$$\frac{\partial L}{\partial b} \Rightarrow -\sum_{n=1}^{N} \alpha_n y_n = 0 \Rightarrow \underbrace{\sum_{n=1}^{N} \alpha_n \underline{y_n} = 0}$$

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

$$L(\alpha) = \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m (x_n^T x_m) + \sum \alpha_n$$

$$- \sum_{n=1}^{N} \alpha_n y_n x_n^T \sum_{m=1}^{N} \alpha_m y_m x_m$$

$$L(\alpha) = \left( -\frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m \underbrace{y_n y_m \left(x_n^T x_m\right)}_{} + \sum \alpha_n \right)$$

$$\max \quad L(\alpha)$$

$$\text{s.t} \quad \alpha_n \geq 0$$

$$\sum_{n=1} \alpha_n y_n = 0$$

$$\min \quad L(\alpha)$$

$$\text{s.t.} \quad \sum_{n=1}^{N} \alpha_n y_n = 0$$

$$\alpha \geq 0$$

$$\Updownarrow$$

DUAL $\longrightarrow$ Primal

End Proof

Solving $\longrightarrow \alpha_n^*$

$$w_n^* = \sum_{n=1}^{N} \alpha_n^* y_n x_n$$

$$\underline{b = y_s - w^{*T} x_s}$$

$\alpha_n (1 - y_n(w^T x_n + b)) = 0$

If $\alpha_n \neq 0$, $\alpha_s > 0$

$y_s (w^T x_s + b) = 1$

<u>Summary</u>

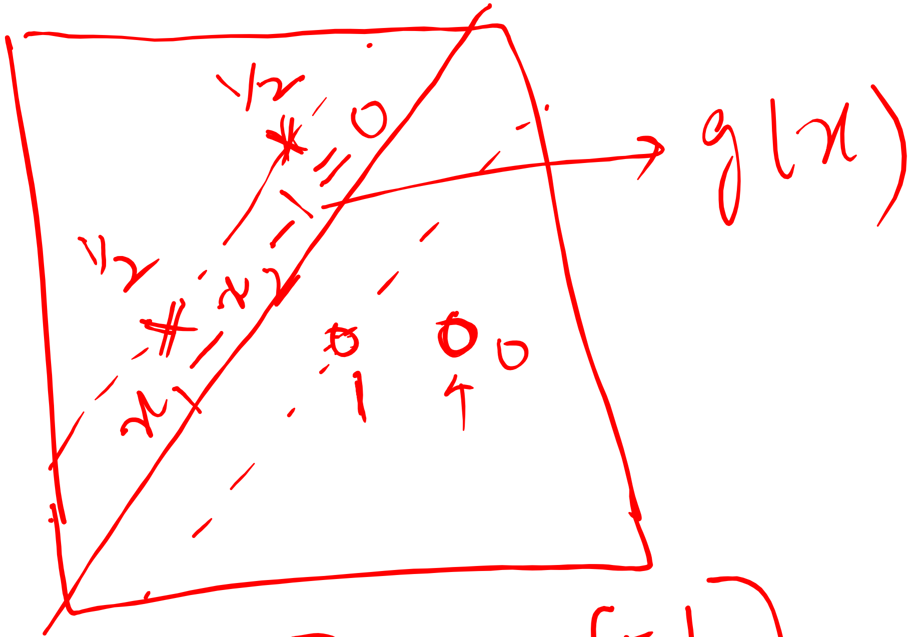
$$\min_{u} \; u^T Q u + p^T u$$

s.t. $\underline{A u \geq C}$

$$[X_s X_s^T]_{NM} = \begin{bmatrix} y_1 x_1^T \\ y_2 x_2^T \\ \vdots \\ y_N x_N^T \end{bmatrix} \begin{bmatrix} y_1 x_1^T & \cdots & y_N x_N^T \end{bmatrix}$$

$$= \underbrace{y_n x_n^T \; y_m x_m} \Big\} G_{nm}$$

$A$




$$X = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad y = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

$$X_s = \begin{bmatrix} 0 & 0 \\ -2 & -2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix}$$ signed.

$$G_m = X_s X_s^T \Rightarrow Q$$

$$P = -1^T$$

$A$ (constraints)

$$= \begin{bmatrix} y^T \\ -y^T \\ I \end{bmatrix}$$

$$C = \begin{bmatrix} 0 \\ 0 \\ 0_N \end{bmatrix}$$

$$\left\{ \begin{array}{l} y^T\alpha = 0 \rightarrow \\ \\ \alpha \geq 0 \rightarrow \end{array} \right. \quad \left. \begin{array}{r} y^T\alpha \geq 0 \\ -y^T\alpha \geq 0 \\ I\alpha \geq 0 \end{array} \right\}$$

Got QPAC $\longrightarrow$ QP Solver $\longrightarrow$ $u^*$

$$\downarrow$$

$w^*$ $b^*$

$$\alpha^* = \begin{bmatrix} 1/2 \\ 1/2 \\ 1 \\ 0 \end{bmatrix}$$

$$w^* = \sum_n \alpha_n^* y_n x_n$$

$$= \frac{1}{2} \cdot -1 \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \frac{1}{2} \cdot -1 \begin{bmatrix} 2 \\ 2 \end{bmatrix} + 1 \cdot 1 \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$b^* = -1 - 1 \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = -1$$

$$g(x) = \text{sign}(x_1 - x_2 - 1)$$

Primal Version $g(x) = \text{sign}(w^{*T} x + b^*)$

Dual Version: $g(x) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n^* y_n (x_n^T x) + b^*\right)$

where $\underline{b^*} = y_s - \sum_{n=1}^{N} \alpha_n^* y_n \underbrace{(x_n^T x_s)}_{\text{dot product}}$

$\alpha_n = 0$

→ Most of $\alpha$'s are zero, $\alpha_s$ (need) → huge saving in computation

→ Don't need all the data. $y, \alpha$

→ $[G]_{nm} \longrightarrow \underbrace{x_n \text{ dotted into } x_m}$ $\longrightarrow \alpha_n^*$ $x_n x_m$

$G_{nm} \rightarrow N \times N$ matrix of dot products $\Big] \alpha_n^*$

$y \longrightarrow N \times 1$

# INNER PRODUCT ALGORITHMS

1) $\underbrace{K(x, x')}_{\text{Kernel}} \longrightarrow x \cdot x'$

$$\min_{\alpha} \sum_{nm} \alpha_n \alpha_m y_n y_m K(x_n, x_m) - \sum \alpha_n$$

s.t $\sum_n \alpha_n y_n = 0$

$\alpha_n \geq 0$

$\downarrow$

$\alpha_n^*$

$$g(x) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n^* K(x_n, x) + b^*\right)$$

$$b^* = y_s - \sum_{n=1}^{N} \alpha_n y_n K(x_n, x_s)$$

$$\alpha_s > 0$$

→ Dot product regardless of the space.

$x_1 \quad x_2 \cdots x_N \longrightarrow z_1 \quad z_2 \cdots z_N$

Target $y_1 \quad y_2 \cdots y_N \longrightarrow y_1 \quad y_2 \cdots y_N$

$\underline{\underline{z_n^T z_m}}$ Test point $\longrightarrow \underline{\underline{z_n^T z}}$ ?

$$\left.\begin{array}{l} K(\underbrace{x_n \, x_m}) \xrightarrow{\checkmark} z_n^T z_n \\ K(x_n, x) \xrightarrow{\checkmark} z_n^T z \\ K(x_n x_s) \xrightarrow{\checkmark} z_n^T z_s \end{array}\right\}$$

## Example 1

(1) 2nd order polynomial transform.

$$Z \cdot Z' = \Phi(x)^T \, \Phi(x')$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \longrightarrow z = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \\ \sqrt{2}\, x_1 x_2 \\ x_2^2 \end{bmatrix}$$

$$z' = \begin{bmatrix} x_1' \\ x_2' \\ x_1'^2 \\ \sqrt{2}\, x_1' x_2' \\ x_2'^2 \end{bmatrix}$$

$$\underline{Z \cdot Z'} = x_1 x_1' + x_2 x_2' + x_1^2 (x_1')^2 + x_2^2 (x_2')^2 + 2\, x_1 x_2 x_1' x_2'$$

$$= \left( x_1 x_1' + x_2 x_2' + \tfrac{1}{2} \right)^2 - \tfrac{1}{4}$$

$$Z \cdot Z' = \Big( \underset{\downarrow \; x.\text{space}}{x \cdot x'} + \tfrac{1}{2} \Big)^2 - \tfrac{1}{4} \Rightarrow K(x, x')$$

$$K(x, x') = (x \cdot x' + 1/2)^{Q} - C$$

This kernel $\Rightarrow$ get dot product in Z-space without actually going there.

1) Kernel $\rightarrow$ dot products!

2) Huge computational saving.

**Separable data**

C (penalisation)

# Infinite dimensional Space

$$x \longrightarrow e^{-x^2}\begin{bmatrix} \sqrt{\frac{2}{0!}}\,x^0 \\ \sqrt{\frac{2^1}{1!}}\,x^1 \\ \vdots\ x^2 \\ \sqrt{\frac{2^3}{3!}}\,x^3 \\ x^4 \\ \vdots \\ \sqrt{\frac{2^k}{k!}}\,x^k \\ \vdots \end{bmatrix} = Z \qquad x' \longrightarrow e^{-x'^2}\begin{bmatrix} \sqrt{\frac{2}{0!}}\,x'^0 \\ \sqrt{\frac{2}{1!}}\,x'^1 \\ \vdots \\ \sqrt{\frac{2}{k!}}\,x'^k \end{bmatrix}$$

$$Z \cdot Z' = e^{-x^2}\, e^{-(x')^2} \Bigg[ \underbrace{\frac{2^0}{0!}(xx')^0 + \frac{2^1}{1!}(x^1 x'^1)^1 + \cdots}_{e^{2xx'}} \Bigg]$$

$$= e^{-(x-x')^2}$$

$$\underbrace{K(x, x')}_{\text{gaussian kernel}} = e^{-(x-x')^2} \Rightarrow e^{-\gamma(\|x-x'\|)^2}$$

Kernel → important.

Any kernel (symmetric, +ve definite)

Infinite dimensions.

1) Computationally feasible → Inf. dim

$K(x, x')$ non-falsifiable

$\underbrace{E_{in} \text{ VS } E_{out}}$

2) Regularize (A lot!)

↑

3) Maximally Regularize → Hyperplane.

4) Small no. of SVs.

# Thanks!