- Kernel Methods
- SVMs

$\theta^T x$

↑ input

$\phi : \mathbb{R}^d \to \mathbb{R}^p$ feature map

attributes features

$\phi(x)$: "features"

p : very high $p > n$ $p \to \infty$

LMS using gradient descent

Loop {

$$\theta := \theta + \alpha \sum_{i=1}^{n} (y^{(i)} - \underbrace{\theta^T}_{\mathbb{R}^p} \underbrace{\phi(x^{(i)})}_{\mathbb{R}^p}) \phi(x^{(i)})$$

}

each iteration: $O(np)$ time

Key observation:

$$\theta = \sum_{i=1}^{n} \beta_i \phi(x^{(i)}) \quad \text{for some } \underbrace{\beta_1 \cdots \beta_n}_{\beta \in \mathbb{R}^n} \in \mathbb{R}$$

$\theta \in \mathbb{R}^p$

New algo : update β

$$\theta = \sum_{i=1}^{n} \underbrace{(\beta_i + \alpha (y^{(i)} - \theta^T \phi(x^{(i)})))}_{\text{new } \beta_i} \phi(x^{(i)})$$

p parameters ⟶ n parameters

$$\beta_i := \beta_i + \alpha\left(y^{(i)} - \theta^T \phi(x^{(i)})\right)$$

$$= \beta_i + \alpha\left(y^{(i)} - \left(\sum_{j=1}^{n} \beta_j \phi(x^{(j)})\right)^T \phi(x^{(i)})\right)$$

$$= \beta_i + \alpha\left(y^{(i)} - \sum_{j=1}^{n} \beta_j \underbrace{\langle \phi(x^{(j)}), \phi(x^{(i)}) \rangle}_{p}\right)$$

(n under the sum)

Precompute

① $\langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle$

$\langle a, b \rangle = \sum_{i=1}^{p} a_i b_i$

$a = (a_1 \dots a_p)$

$b = (b_1 \dots b_p)$

② $\langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle$ can often be computed much faster without explicitly computing $\phi(\,)$

eg. Cubic polynomials

$$\phi(x) = \begin{bmatrix} 1 \\ x_i \\ x_i x_j \\ x_i x_j x_k \end{bmatrix} \begin{matrix} 1 \\ \} d \\ \} d^2 \\ \} d^3 \end{matrix}$$

$$\langle \phi(x), \phi(z) \rangle = [1 \dots x_i \dots x_i x_j \dots x_i x_j x_k \dots] \begin{bmatrix} 1 \\ \vdots \\ z_i \\ \vdots \\ z_i z_j \\ \vdots \\ z_i z_j z_k \\ \vdots \end{bmatrix}$$

$$= 1 + \sum_{i=1}^{d} x_i z_i + \sum_{i,j=1}^{d} x_i x_j z_i z_j + \sum_{i,j,k=1}^{d} x_i x_j x_k z_i z_j z_k$$

$$\sum_{i,j=1}^{d} u_i w_j = \left(\sum_{i=1}^{d} u_i\right)\left(\sum_{j=1}^{d} w_j\right)$$

$u_i \to x_i z_i, \quad w_j \to x_j z_j$

$$= \left(\sum_{i=1}^{d} x_i z_i\right)\left(\sum_{j=1}^{d} x_j z_j\right) = \langle x, z \rangle^2$$

$O(d)$ time

$$= \left(\sum_{i=1}^{d} x_i z_i\right)\left(\sum_{j=1}^{d} x_j z_j\right)\left(\sum_{k=1}^{d} x_k z_k\right)$$

$$= \langle x, z\rangle^3 \qquad O(d) \text{ time}$$

$$\langle \phi(x), \phi(z)\rangle = 1 + \langle x, z\rangle + \langle x, z\rangle^2 + \langle x, z\rangle^3$$

$O(d)$ time $\quad (p = 1 + d + d^2 + d^3 = \Theta(d^3))$

$K(x, z) = \langle \phi(x), \phi(z)\rangle$ — Mercer Kernels

$K(\cdot,\cdot)$ is Kernel function

$K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$

Compute $K(x^{(i)}, x^{(j)})$ $\forall\, i, j$

n^2 entries $\quad O(n^2 p)$ $O(n^2 d)$ time

$\beta = 0$

Loop $\{ \beta_i = \beta_i + \alpha\left(y^{(i)} - \sum_{j=1}^{n} \beta_j \langle \phi(x^{(i)}), \phi(x^{(j)})\rangle\right)$

$= \beta_i + \alpha\left(y^{(i)} - \sum_{j=1}^{n} \beta_j \cdot K(x^{(i)}, x^{(j)})\right)\}$

$K \in \mathbb{R}^{n \times n}$ Kernel matrix

$K_{ij} = K(x^{(i)}, x^{(j)})$

$\beta := \beta + \alpha(\vec{y} - K\beta)$ $\qquad O(n^2)$ time

Test time: given x, how to predict $\theta^T \phi(x)$

$$\theta^T \phi(x) = \left(\sum_{i=1}^{n} \beta_i \phi(x^{(i)})\right)^T \phi(x)$$

$$= \sum_{i=1}^{n} \beta_i \langle \phi(x^{(i)}), \phi(x)\rangle = \sum_{i=1}^{n} \beta_i K(x^{(i)}, x)$$

linear in # examples, independent of p

training: Preprocessing : $O(n^2 d)$
training: $O(n^2)$ × # iterations

Test time: $O(nd)$ assuming $K(\cdot,\cdot)$ can be computed in $O(d)$ time

Deeper Observation

- the only thing you need is $K(\cdot,\cdot)$
  function $K(\cdot,\cdot)$ is valid (Kernel fn)
  if $\exists\ \phi$ s.t. $K(x,z) = \langle \phi(x), \phi(z) \rangle$

{ Design some $K(\cdot,\cdot)$
Verify validity (by math)
Run algo

Other algos can also be "kernelized"
perceptron, logistic regression
- algo for linear $\theta^T x$
- replace x by $\phi(x)$
- rewrite algo s.t. it only depends on $\langle \phi(x), \phi(z) \rangle$

Kernel fns :

$K(x,z) = 1 + x^T z + (x^T z)^2 + (x^T z)^3$

$K(x,z) = (x^T z)^2$

$K(x,z) = (x^T z + c)^2$ $\quad \phi(x) = \begin{bmatrix} c \\ \sqrt{2c}\, x_i \\ x_i x_j \end{bmatrix}$

polynomial kernel $K(x,z) = (x^T z + c)^K \sim \binom{d+K}{K}$ monomials

$K(x,z) = \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right) = \langle \phi(x), \phi(z) \rangle$

ϕ ∞ dimensional

Valid Kernel?

Necessary cond^n

n examples $x^{(1)} \ldots x^{(n)}$

Kernel matrix $K_{ij} = K(x^{(i)}, x^{(j)})$

Claim: Kernel matrix is positive semidefinite

$K \succeq 0$

$z^T K z \geq 0 \quad \forall z \in \mathbb{R}^n$

Also sufficient

Theorem (Mercer, 1909)

$K: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a valid kernel fn

if for any $n < \infty$ and any $x^{(1)} \ldots x^{(n)} \in \mathbb{R}^d$

the kernel matrix $K_{ij} = K(x^{(i)}, x^{(j)})$ is

positive semidefinite

eg. Protein sequence classification

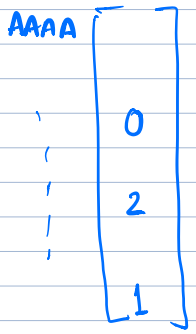
20 amino acids

A, B, C ...

$\phi(x)$ AAAA

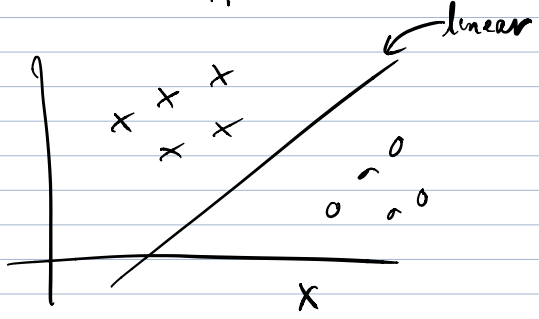
AAAB

$20^4 = 160{,}000$

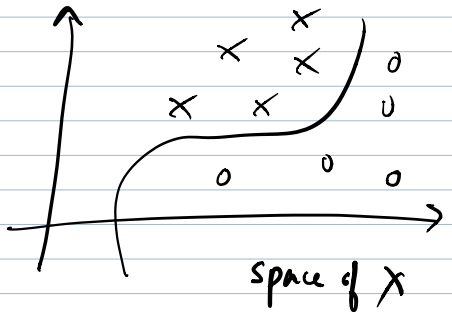


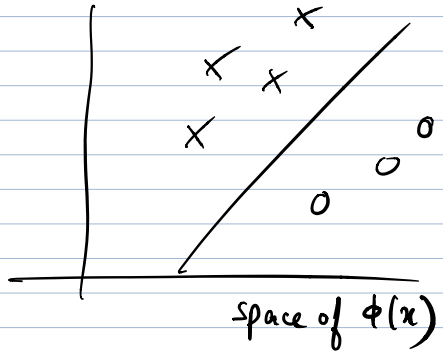
$\langle \phi(x), \phi(z) \rangle$ can be computed via dynamic programming

SVM : support vector machines



$\{x : w^T x + b = 0\}$





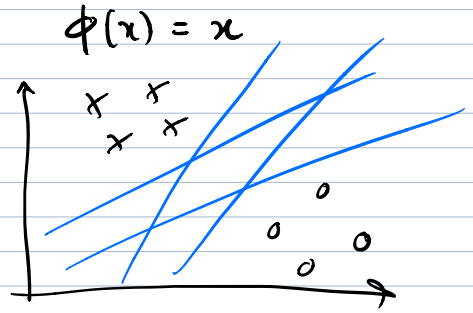
$y^{(i)} \in \{-1, +1\}$

$\{x : w^T x + b = 0\}$

$\{x : w^T \phi(x) + b = 0\}$

linear in Kernel space

Find w, b

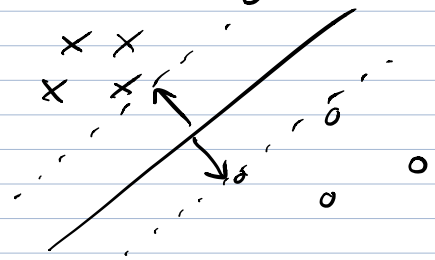


Find w, b s.t.

if $y^{(i)} = 1$, $w^T x^{(i)} + b > 0$ ①

$y^{(i)} = -1$, $w^T x^{(i)} + b < 0$ ②

Choose the (w, b) that gives the most separation



Among all (w, b) satisfy ①, ②

$$\max_{w,b} \left[\min_i \text{dist}\left(x^{(i)}, \text{decision boundary}\right) \right]$$

①, ② $\Rightarrow$ $y^{(i)}\left(w^T x^{(i)} + b\right) > 0 \quad \forall\, i$

Fact: $\text{dist}\left(x^{(i)}, \text{boundary}\right) = \dfrac{\left|w^T x^{(i)} + b\right|}{\|w\|_2}$

$$= \frac{y^{(i)}\left(w^T x^{(i)} + b\right)}{\|w\|_2}$$

$$\max_{w,b} \min_{i \in \{1, \ldots n\}} \frac{y^{(i)}\left(w^T x^{(i)} + b\right)}{\|w\|_2}$$

scaling invariant $(w,b) \rightarrow (10w, 10b)$

$\Longleftrightarrow$ $\min \frac{1}{2}\|w\|_2^2$

s.t. $y^{(i)}(w^T x^{(i)} + b) \geq 1 \quad \forall i$

Facts (non trivial) (need KKT condn)

① Optimal soln w^*, b^* satisfies

$$w^* = \sum_{i=1}^{n} \alpha_i x^{(i)} y^{(i)} \qquad \alpha_i \geq 0 \qquad \alpha_i \in \mathbb{R}$$

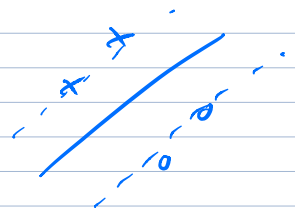
② α is optimal soln of program

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$K(x^{(i)}, x^{(j)})$

s.t. $\alpha_i \geq 0$

$$\sum_{i=1}^{n} \alpha_i y^{(i)} = 0$$



$$w^* = \sum_{i=1}^{n} \alpha_i \phi(x^{(i)}) y^{(i)}$$

test time:

$$w^{*T} \phi(x) = \sum_{i=1}^{n} \alpha_i \langle \phi(x^{(i)}) \phi(x) \rangle y^{(i)}$$

$$= \sum_{i=1}^{n} \alpha_i K(x^{(i)}, x) y^{(i)}$$