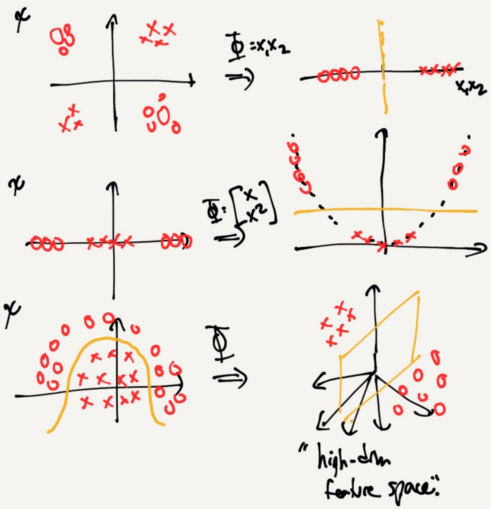
Lecture 10 Nonlinear Classifiers & Kernels

Linear classifiers – SVM, LR, perceptron, etc.

What if the data is nonlinearly separable?



IDEA – reuse linear classifiers

- mapping $\Phi(\cdot): \mathcal{X} \to \mathcal{Z}$
- learn linear classifier in new space.
- if new dimension is large enough, the data becomes linearly separable. $\dim(\mathcal{Z}) > \dim(\mathcal{X})$

In the limit, $\dim(\mathcal{Z}) \to \infty$

- we are mapping each point x_i into a function

$$\Phi(x) \to \begin{bmatrix} x_1 \\ \vdots \\ x_D \end{bmatrix} \to \phi(x; t)$$

CS5487 Lecture Notes (2022B)
Prof. Antoni B. Chan
Dept of Computer Science
City University of Hong Kong

Kernel SVM

Consider the SVM dual problem, but replace $x_i \to \Phi(x_i)$

Training
$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \underbrace{\Phi(x_i)^T \Phi(x_j)}_{k(x_i, x_j)}$$
$$\text{s.t.} \sum_i \alpha_i y_i = 0, \quad \alpha_i \geq 0$$

Decision Boundary: $w^* = \sum_i \alpha_i y_i \Phi(x_i)$ (w is in same space as $\Phi(x)$)

Bias term
$$b = \frac{1}{|SV|} \sum_{i \in SV} (y_i - w^T \Phi(x_i)) = \frac{1}{|SV|} \sum_{i \in SV} (y_i - \sum_j \alpha_j y_j \underbrace{\Phi(x_j)^T \Phi(x_i)}_{k(x_i, x_j)})$$
SV: set of support vectors.

Decision function
$$y^* = \text{sign}(f(x^*))$$
$$f(x^*) = w^T \Phi(x^*) + b = \sum_i \alpha_i y_i \underbrace{\Phi(x_i)^T \Phi(x^*)}_{k(x_i, x^*)} + b$$

Note: entire algorithm depends only on $\Phi(x_i)^T \Phi(x_j)$.

Define function: $k(x_i, x_j) = \Phi(x_i)^T \Phi(x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$
dot product kernel

- The dual SVM can be rewritten w/ this kernel function → nonlinear classifier
- Just define $k(x_i, x_j)$ directly w/o explicitly calculating $\Phi(x)$ ("kernel trick")
  - Good: saves time/computation of calculating $\Phi(x)$
  - Bad: need $O(n^2)$ terms of $k(x_i, x_j)$, store this kernel matrix.

$$K = [k(x_i, x_j)]_{ij}$$

Example

polynomial kernel: $x = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$

$$K(x,x') = (x^T x')^2 = \left(\sum_{i=1}^{d} x_i x_i'\right)^2$$

$$= \sum_{i=1}^{d}\sum_{j=1}^{d} x_i x_i' x_j x_j' = \sum_{i=1}^{d}\sum_{j=1}^{d} (x_i x_j)(x_i' x_j')$$

$$= \underbrace{\begin{bmatrix} x_1x_1 & x_1x_2 & \cdots & x_dx_d \end{bmatrix}}_{\Phi(x)^T} \overbrace{\begin{bmatrix} x_1'x_1' \\ x_1'x_2' \\ \vdots \\ x_d'x_d' \end{bmatrix}}^{\Phi(x')}$$

$$= \Phi(x)^T \Phi(x')$$

Hence, $\Phi(x): \mathbb{R}^d \to \mathbb{R}^{d^2}$

$$K(x,x') = \underbrace{\Phi(x)^T \Phi(x')}_{\text{calculation: } O(d^2)} = \underbrace{(x^T x')^2}_{O(d)}$$

★ more efficient

Look SVM decision function:

$$f(x) = \sum_i \alpha_i y_i K(x, x_i) + b$$

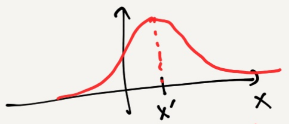
$$= \sum_i \alpha_i y_i (x^T x_i)^2 + b = \sum_i \alpha_i y_i (x^T x_i x_i^T x) + b$$

$$= \underbrace{x^T \underbrace{\left(\sum_i \alpha_i y_i x_i x_i^T\right)}_{A} x + b}_{\text{quadratic function}}$$

★ the kernel specifies the class of functions that are used.

Gaussian Kernel / Radial Basis function (RBF) kernel

$$K(x,x') = e^{-\frac{1}{\sigma^2}\|x-x'\|^2}$$



- Is it a dot product kernel?
- What is the $\Phi(x)$?

Kernel Functions

Kernel trick depends on $K(x_i, x_j)$ is a dot-product kernel

Defn a mapping $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a dot-product kernel iff

$$k(x,x') = \langle \Phi(x), \Phi(x') \rangle$$

where $\Phi: \mathcal{X} \to \mathcal{H}$, $\mathcal{H}$ is a vector space

$\langle \cdot, \cdot \rangle$ is dot product in $\mathcal{H}$

How to check w/o knowing $\Phi(x)$ & $\langle \cdot, \cdot \rangle$?

Defn $k(x,x')$ is a positive definite kernel

if $\forall n$ & $\forall \{x_1, \dots, x_n\}$, $x_i \in \mathcal{X}$ (all datasets of all sizes)

the Kernel matrix $K = [k(x_i, x_j)]_{i,j}$ is a positive defn matrix.

{ K is posdef iff
1) $y^T K y > 0$, $\forall y$
2) eigenvalues of $K > 0$
3) $K = XX^T$, X has independent columns.

★ $\boxed{k(x,x') \text{ is a dot-product kernel iff it is posdef kernel.}}$

Given a posdef kernel, what is the high-dim xform $\Phi(x)$?

Let $\mathcal{H}$ = space of all linear combos (combination) of function $k(\cdot, x_i)$

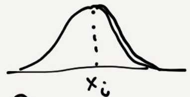
function of "·"

fixed

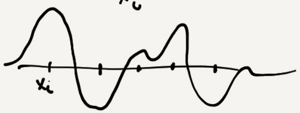
$$\mathcal{H} = \left\{ f(\cdot) \;\middle|\; f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i), \forall m, \forall x_i \in \mathcal{X} \right\}$$

e.g. use Gaussian kernel

$$k(\cdot, x_i) = e^{-\frac{1}{\sigma^2}\|\cdot - x_i\|^2}$$



$$f(\cdot) = \sum_i \alpha_i k(\cdot, x_i)$$



---

let $f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i)$, $g(\cdot) = \sum_{i=1}^{m} \beta_j k(\cdot, x_j)$

Can show the dot-product btwn them in $\mathcal{H}$ is:

$$\langle f, g \rangle = \sum_i \sum_j \alpha_i \beta_j k(x_i, x_j)$$

e.g. Gaussian kernel $\langle f, g \rangle = \sum_i \sum_j \alpha_i \beta_j e^{-\frac{1}{\sigma^2}\|x_i - x_j\|^2}$

(like a non-linear similarity function in $\mathcal{X}$)

Special case:

$\alpha_i = 1$, $\alpha_{\neg i} = 0$, $\beta_j = 1$, $\beta_{\neg j} = 0$

$$\Rightarrow \langle k(\cdot, x_i), k(\cdot, x_j) \rangle = k(x_i, x_j)$$

Hence, $k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$

where $\Phi : \mathcal{X} \to \mathcal{H}$

$$x_i \to \Phi(x_i) = k(\cdot, x_i)$$

xformation is from X to a very-high-dim space (infinite-dim space).

e.g. Gaussian kernel

$$x \to e^{-\frac{1}{\sigma^2}\|\cdot - x\|^2}$$

polynomial kernel

$$x \to \cdot^T (x x^T) \cdot$$

Final Note

$$f(\cdot) = \sum_i \alpha_i k(\cdot, x_i)$$

$$\langle k(\cdot, x), f(\cdot) \rangle = \sum_i \alpha_i k(x, x_i) = f(x)$$

dotproduct btwn function & kernel is the function.

"reproducing property" - like convolving a signal w/ delta function, which gives back the signal.

$\mathcal{H}$ is called a "Reproducing Kernel Hilbert Space" (RKHS)

vector space + dot product

RKHS uniquely specifies the kernel function & vice versa.

(sometimes called a Mercer Kernel)

Representer Thm

Empirical Risk: $R_{emp} = \sum_i L(y_i, f(x_i))$

Regularizer: $\Omega(\|f\|_p)$, $\Omega \geq 0$ & strictly monotonically increasing

, $k(x,x')$ & RKHS

Then for:

$$f^* = \arg\min_f R_{emp}(f) + \lambda \Omega(\|f\|_p)$$

$\Rightarrow$ f^* has the form $f^*(x) = \sum_{i=1}^n \alpha_i k(x, x_i)$, $f^* \in$ RKHS

- optim over inf. dim space of functions $\rightarrow$ finite dim space of α_i's (only n)
- many ML algorithms fit this framework $\rightarrow$ they can be kernelized

Kernel functions

Kernels on $\mathbb{R}^d$:

linear kernel: $k(x,x') = x^T x'$

$k(x,x') = x^T A x'$, A is pos def

poly kernel: $k(x,x') = (x^T x')^d \Rightarrow$ purely d order ter

$k(x,x') = (x^T x' + 1)^d \Rightarrow$ get all terms with order $\leq d$.

Gaussian/RBF: $k(x,x') = e^{-\frac{1}{\sigma^2}\|x-x'\|^2}$

exponential: $k(x,x') = e^{-\frac{1}{\sigma^2}\|x-x'\|}$

Combine kernels: $k(x,x') = \underbrace{x^T x'}_{\text{linear}} + \underbrace{e^{-\frac{1}{\sigma^2}\|x-x'\|^2}}_{\text{RBF}}$

Kernels on histogram

correlation kernel: $k(x,x') = x^T x' = \sum_{i=1}^d x_i x_i'$

Bhattacharyya kernel: $k(x,x') = \sum_{i=1}^d \sqrt{x_i}\sqrt{x_i'}$

χ^2-RBF kernel: $k(x,x') = e^{-\frac{1}{\sigma^2}\chi^2(x,x')}$

$$\chi^2(x,x') = \sum_i \frac{(x_i - x_i')^2}{\frac{1}{2}(x_i + x_i')}$$

Histogram intersection: $k(x,x') = \sum_i \min(x_i, x_i')$

Kernels on sets:

$\mathcal{X} = \{x_1, \dots, x_n\}$

$\mathcal{X}' = \{x_1', \dots, x_m'\}$ } $n \neq m$

Intersection kernel: $k(\mathcal{X}, \mathcal{X}') = 2^{|\mathcal{X} \cap \mathcal{X}'|}$ ← # of common elements

pairwise distance: $k(\mathcal{X}, \mathcal{X}') = e^{-\frac{1}{\sigma^2}\sum_i \sum_j d(x_i, x_j)}$

pyramid match kernel: approx to sum of min. distances btwn points.

Kernels on strings/trees/graphs

$k(x,x') = \sum_s w_s \phi_s(x) \phi_s(x')$, $\phi_s(x) =$ # of times substring s appears in x.

$w_s > 0$, coefficient (weight)

Kernels on probability densities: $p(x), q(x)$

corr. kernel: $k(p,q) = \int p(x) q(x) dx$

prob. product kernel: $k(p,q) = \int p(x)^s q(x)^s dx$

Fisher kernel, ...