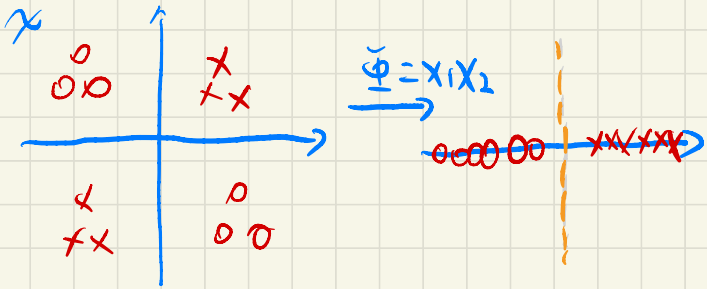
Lecture 10 NonLinear classifiers and kernals

Linear classifiers – SVM, LR, perceptron, etc.

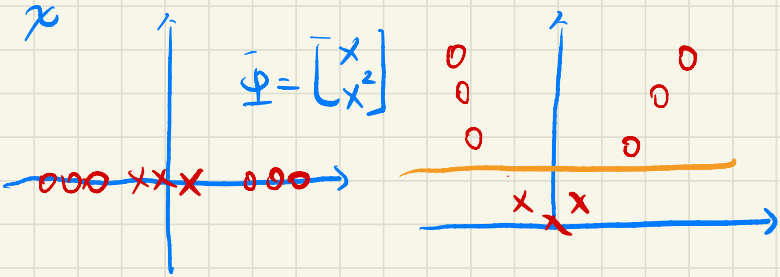
What if the data is nolinearly seperable?



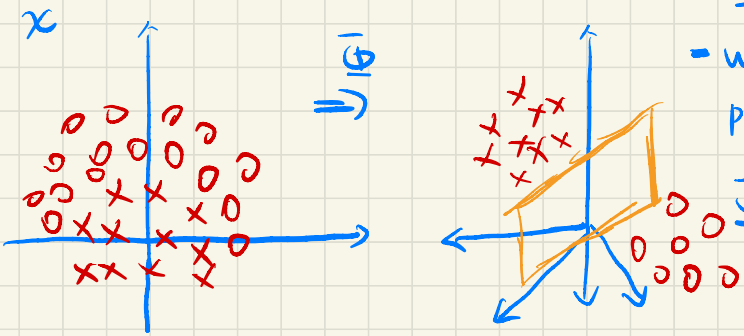
IDEA: reuse Linear classifiers.

- mapping: $\Phi(\cdot): x \to z$
- Learn Linear classifier in new space

if the new dimension is large enough, the data becomes, Linearly seperable. $\dim(z) > \dim(x)$



high-dim feature space



In the limit, $\dim(z) \to \infty$

- we are mapping each point x_i into a function

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

Kernal SVM:

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

Training $\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \underbrace{\Phi(x_i)^T \Phi(x_j)}_{K(x_i, x_j)}$

s.t. $\sum_i \alpha_i y_i = 0$, $\alpha_i \geq 0$

Decision Boundary: $w^* = \sum_i \alpha_i y_i \Phi(x_i)$

(w is in the same space as $\Phi(x)$)

Bias term: $b = \frac{1}{|SV|} \sum_{i \in SV} (y_i - w^T \Phi(x_i))$

$= \frac{1}{|SV|} \sum_{i \in SV} (y_i - \sum_j \alpha_j y_j \underbrace{\Phi(x_j)^T \Phi(x_i)}_{K(x_i, x_j)})$

Set of support vectors.

Decision function:

$y^* = \text{sign}(f(x^*))$

$f(x^*) = w^T \Phi(x^*) + b = \sum_i \alpha_i y_i \underbrace{\Phi(x_i)^T \Phi(x^*)}_{K(x_i, x_j)} + b$

Note: entire algorithm depends on

$\Phi(x_i)^T \Phi(x_j)$

dot product kernel

Define function: $K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$

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

- The dual SVM can be rewritten with this kernal function $\rightarrow$ nolinear classifier

- Just define $k(x_i, x_j)$ directly without explicitly calculating $\Phi(x)$ ("kernal trick")
  - Good: Saves time / computation of calculating $\Phi(x)$
  - Bad: need $O(n^2)$ terms of $k(x_i, x_j)$ to store this kernal matrix. $K = [k(x_i, x_j)]_{ij}$

Example.

Polynominal kernal: $x = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$

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

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

$$= \underbrace{[x_1 x_1, x_1 x_2, \cdots x_d d]}_{\Phi(x)^T} \underbrace{\begin{bmatrix} x_1' x_1' \\ x_1' x_2' \\ \vdots \\ x_d' x_d \end{bmatrix}}_{\Phi(x')}$$

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

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

$O(d)$: more efficient

$$K(x, x') = \underbrace{\Phi(x)^T}_{} \Phi(x') = (\overbrace{x^T x'})^2$$

Calculation: $O(d^2)$

Look SVM decision function:

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

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

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

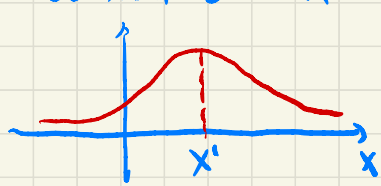
$$= \underbrace{x^T \underbrace{\left(\sum_i \alpha_i y_i x_i x_i^T\right)}_{A} x + b}_{\text{Quadratic function}}$$

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

Gaussian kernel / Radial Basis function (RBF) kernel

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

Gaussian centroid at x'



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

Kernal functions.

Kernal trick depends on $k(x_i, x_j)$ is a dot-product kernal.

Defin: A mapping $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a dot-product kernal iff:

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

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

$\langle \cdot, \cdot \rangle$ is a dot-product in H.

How to check without knowing $\Phi(x)$ and $\langle \cdot, \cdot \rangle$?

Defin: $K(x, x')$ is a positive definite kernal

if $\forall n$ and $\forall \{x_1, \cdots, x_n\}$, $x_i \in \mathcal{X}$

all datasets of all sizes

the kernal matrix $K = [k(x_i, x_j)]_{ij}$ is a positive defin 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 column

☆: $K(x, x')$ is a dot-product kernel iff it is a posdef kernel.

Given a pos def kernel, what is the high-dim x form $\Phi(x)$?

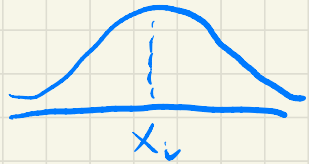
Let H = space of all linear combinations of functions $K(\cdot, x_i)$ (function of "$\cdot$")

fixed

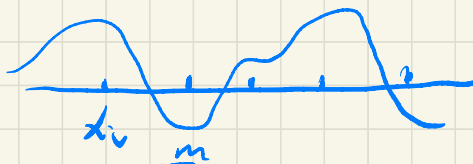
$$H = \left\{ f(\cdot) \mid f(\cdot) = \sum_{i=1}^{m} \alpha_i K(\cdot, x_i),\ \forall m, \forall x_i \in X \right\}$$

e.g. use Gaussian kernal

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



$$f(\cdot) = \sum_{i=1}^{n} \alpha_i K(\cdot, x_i)$$



Let $f(\cdot) = \sum_{i=1}^{m} \alpha_i K(\cdot, x_i)$ $\quad g(\cdot) = \sum_{j=1}^{m} \beta_j K(\cdot, x_j)$

Can show the dot-product between them in 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 x)

Special case:

$\alpha_i = 1$, $\alpha_{ri} = 0$, $\beta_j = 1$, $\beta_{rj} = 0$

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

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

where $\Phi : X \to \mathcal{H}$ $\quad x_i \to \Phi(x_i) = K(\cdot, x_i)$

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

e.g. Gaussian kernal

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

Polynomial kernal

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

Final Note: $f(\cdot) = \sum_i \alpha_i K(\cdot, x_i)$

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

dot-product between function and kernel is the function

"reproducing property" — like convolving a signal with delta function, which gives back the signal.

$\mathcal{H}$ is called a "Reproducing Kernal Hilbert space" (RKHS)

Hilbert space: vector space + dot product space

RKHS uniquely specifies the kernal function and vice versa. (Sometimes called a Mercer kernel)

Representer Theory:

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

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

$K(x, x')$ and RKHS

Therefore:

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

- Optim over infinite dimension of functions
  $\longrightarrow$ finite dimension space of α_i's
- many ML algorithms fit this framework $\rightarrow$ they can be kernerlized.

Kernel functions:

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

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

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

- poly kernel : $k(x, x') = (x^T x')^d$

  $k(x, x') = (x^T x' + 1)^d$

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

wiggly straight line

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 sizes: $\left.\begin{array}{l} x = \{x_1, \dots x_n\} \\ x' = \{x_1, \dots x_m\} \end{array}\right\} n \neq m$

Intersection kernel: $k(x, x') = 2^{|x \cap x'|}$

# of common elements

Pairwise kernel: $k(x, x') = e^{-\frac{1}{\sigma^2} \sum_i \sum_j d(x_i, x_j)}$

Pyramid match kernel: approx to sum of min distance between 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,

$$k(p, q) = \int p(x)^s q(x)^s dx.$$