Sheet 5 Kernel Machines (structured Kernels)

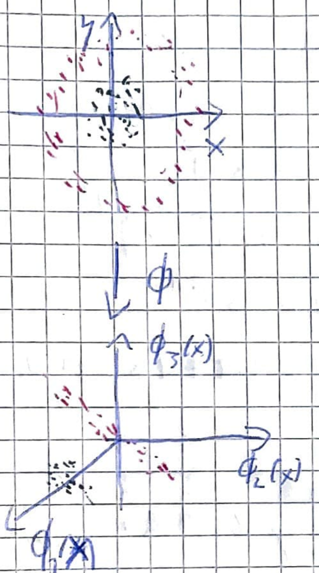
Linear Models

$X = x_1, \dots, x_N$ data

$f(x) = Y = y_1, \dots, y_N$ target

$f(x) = \beta^T x$

often useful and good but do not solve non-linear problems:



Solution: Apply non-linear feature map ϕ

$\phi(x_1, x_2) = (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2, x_1, x_2)$

Kernel Models

$$f(x) = \beta^T \phi(x)$$

Tip

Express feature map as dot-product of nonlinear function:

$$x_1^2 y_1^2 + 2 x_1 y_1 x_2 y_2 + x_2^2 y_2^2 = \left\langle \begin{pmatrix} x_1^2 \\ \sqrt{2}\, x_1 x_2 \\ x_2^2 \end{pmatrix}, \begin{pmatrix} y_1^2 \\ \sqrt{2}\, y_1 y_2 \\ y_2^2 \end{pmatrix} \right\rangle = \left\langle \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \right\rangle^2$$

$$\langle \phi(x), \phi(y) \rangle = k(x, y)$$

What is a kernel?

A (nonlinear) function $k: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$

Summary

- structured kernels can be used for data that is not in $\mathbb{R}^d$: strings, trees
- structured kernels can be designed to incorporate prior domain-specific models
- other algorithms (LR, SVMs, PCA, CCA, ...) can be used with structured kernels

Which Kernels induce feature maps?

Mercer Theorem: All symmetric $k(x,y) = k(y,x)$ and PSD kernels!

Positive semi-definite (PSD):

$$\sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j \, k(x_i, x_j) \geq 0 \qquad \forall x_1,\dots,x_N \in \mathbb{R}^d,\ c_1,\dots,c_N \in \mathbb{R}$$

$$\longrightarrow k(x,y) = \phi(x)^T \phi(y) \qquad \phi(x,y) \in \mathbb{R}^\infty$$

Weighted Degree Kernel

$$k(x,y) = \sum_{n=1}^{d} \beta_n \sum_{i=1}^{L+1-n} \delta(x_{i,\dots,i+n-1} = y_{\dots})$$

$L \in \{G, A, T, C\}$

d: neighbors to check

AAACAAATAAG
TACCTAATTAT

d=1, d=2, d=3

PSD Kernel Examples

Linear kernel	$k(x,y) = xy^T$
Polynomial	$k(x,y) = (xy^T + a)^b$
Gaussian	$k(x,y) = e^{-\beta \lVert x-y \rVert^2}$
t-Student	$k(x,y) = (a + \lVert x-y \rVert^2)^{-1}$

String Kernels

Alphabet A: finite set of discrete symbols $A = \{S_1, \dots, S_d\}$

Examples: DNA: $A \in \{G, A, T, C\}$, $A = \{a, b, c, \dots\}$

String: concatenation of symbols of A

Task: compare two sequences $x = (a, b, c, c, e, a, b, d) \in A^L$
$z = (a, b, a, c, c, c, b, e) \in A^L$

$$k(x,y) = \sum_{i=1}^{L} \mathbb{1}(x_i = z_i) \qquad \mathbb{1}(x=z)\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 0$$

$k: A^L \times A^L \to \mathbb{R}$ Kernel counts number of matching ~~[illegible]~~ symbols

Feature map: $\phi(x) = (\delta(x_i = a))_{a \in A,\, i \in \{1,\dots,L\}}$

$$= \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & \dots & L \end{array} \\ \begin{array}{c} a \\ b \\ c \\ \vdots \\ z \end{array} & \begin{pmatrix} 1 & & & \\ & 1 & & 1 \\ & & & \\ & & 1 & \end{pmatrix} \end{array} \in \mathbb{R}^{\mathrm{len}(A) \times \cancel{\mathrm{len}} L}$$

Bag of Words Implementation

sort words to save time when comparing!

Variation: ~~none~~ shift tolerance $\begin{smallmatrix} a & c & b \\ b & a & c \end{smallmatrix}$, full shift tolerance = Bag of Words, subsequences,