Machine Learning

Kernel Methods

Mayson Ma

Courant Institute of Mathematical Sciences - New York University

June 2021

Table of Contents

Kernels

Motivation

- Non-linear decision boundary
- Efficient computation of inner products in high dimension

Non-Linear Separation

- Linear separation impossible in most problems
- Non-linear mapping from input space to high-dimensional feature space: $\Phi : X \mapsto F$.

Kernel Methods

Idea

Define $K : X \times X \mapsto \mathbb{R}$, called **kernel**, such that

$$\Phi(x) \cdot \Phi(y) = K(x, y)$$

Benefits

- Efficiency: K is often more efficient to compute than Φ and the dot product.
- Flexibility: K can be chosen arbitrarily so long as the existence of Φ is guaranteed.

Polynomial Kernels

Definition

$\forall x, y \in \mathbb{R}^d, K(x, y) = (x^\intercal y + 1)^p$

Theorem

$(x^\intercal y + 1)^p = \Phi(x)^\intercal \Phi(y)$, where $\Phi(x)$ contains every monomial in x of degree $0, 1, \dots, p$.

Example

for $d = 2$ and $p = 2$,

$$\begin{aligned} K(x, y) &= (x_1 y_1 + x_2 y_2 + 1)^2 \\ &= x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 y_1 x_2 y_2 + 2x_1 y_1 + 2x_2 y_2 + 1 \\ &= \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2} x_1 x_2 \\ \sqrt{2} x_1 \\ \sqrt{2} x_2 \\ 1 \end{bmatrix} \begin{bmatrix} y_1^2 \\ y_2^2 \\ \sqrt{2} y_1 y_2 \\ \sqrt{2} y_1 \\ \sqrt{2} y_2 \\ 1 \end{bmatrix} = \Phi(x)^\intercal \Phi(y) \end{aligned}$$

The Gaussian Kernel

Definition

$\forall x, y \in \mathbb{R}^d, K(x, y) = \exp\left(-\frac{(x-z)^2}{2\sigma^2}\right)$

Example

for $d = 1$, $\Phi(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)\left[1 \quad \frac{x}{\sigma\sqrt{1!}} \quad \frac{x^2}{\sigma^2\sqrt{2!}} \quad \cdots\right]^\mathsf{T}$

Key observation

hypothesis $h(z) = \sum_{j=1}^{n} a_j K(x_j, z)$ is a linear combination of Gaussian centered at sample points.

The Gaussian Kernel

Very popular in practice, Why?

- Gives very smooth h
- Behaves like K-nearest neighbors
- Oscillates less than polynomials (depend on σ)
- $K(x, y)$ interpreted as a similarity measure. Maximum when $z = x$; goes to 0 as distance increases
- Sample points vote for value at z, but closer get weightier vote.

σ trade off: bias vs. variance

larger $\sigma \rightarrow$ wider Gaussian, smoother $h \rightarrow$ more bias, less variance

PDS Condition

Definition

A kernel $K : X \times X \mapsto \mathbb{R}$ is positive definite symmetric (PDS) if for any $\{x_1, ..., x_m\} \subseteq X$, the matrix $\mathrm{K} = [K(x_i, x_j)]_{ij} \in \mathbb{R}^{m \times m}$ is symmetric positive semi-definite (SPSD).

SPSD

K SPSD if symmetric and one of the 2 equivalent conditions holds:

- its eigenvalues are non-negative
- for any $c \in \mathbb{R}^m$, $c^\intercal \mathrm{K} c = \sum_{i,j=1}^{m} c_i c_j K(x_i, x_j) \geq 0$.

Kernel-based algorithms

Observation

In many learning algorithms, the weights can be written as a linear combination of sample points. We can use inner products of $\Phi(x)$'s only and do not need to compute $\Phi(x)$.

Idea

Suppose $w = X^\top a = \sum_{i=1}^n a_i X_i$ for some $a \in \mathbb{R}^n$.
Substitute this identity into algorithm and optimize n dual weights a, compared to d primal weights.

Kernel Ridge Regression

Observation

Recall the normal equation $(X^\intercal X + \lambda I)w = X^\intercal y$, and $w = \frac{1}{\lambda}(X^\intercal y - X^\intercal X w) = X^\intercal a$, where $a = \frac{1}{\lambda}(y - Xw)$. This shows that w is a linear combination of sample points.

Optimization Problem

$\min_a F(a) = ||XX^\intercal a - y||^2 + \lambda ||X^\intercal a||^2$

Solution

$a = \frac{1}{\lambda}(y - Xw) \Rightarrow \lambda a = y - XX^\intercal a \Rightarrow a = (XX^\intercal + \lambda I)^{-1} y$

Regression Function

$h(z) = w^\intercal z = a^\intercal X z = \sum_{i=1}^{n} a_i (x_i^\intercal z)$, linear combination of inner products.

Kernel Ridge Regression

Kernel matrix

Let $K(x, y) = x^\intercal y$ be kernel function.
Let $\mathrm{K} = XX^\intercal$ be $n \times n$ **kernel matrix**. Note $\mathrm{K}_{ij} = K(x_i, x_j)$
K is singular if $n > d$. In that case, no solution if $\lambda = 0$.

Dual ridge regression algorithm

Train:
$\forall i, j\ \mathrm{K}_{ij} \leftarrow K(x_i, x_j) \Leftarrow O(n^2 d)$
Solve $(\mathrm{K} + \lambda I)a = y$ for $a \Leftarrow O(n^2)$ for $n \times n$ system
Test:
for each test point z **do**
$h(z) \leftarrow \sum_{i=1}^{n} a_i K(x_i, z) \Leftarrow O(nd)$
end for

Does not use x_i directly! Only K.
Dual: solve $n \times n$ system $O(n^3 + n^2 d)$ time
Primal: solve $(d + 1) \times (d + 1)$ system $O(d^3 + d^2 n)$ time
We prefer dual when $d > n$.