Kernel methods

Narayana Santhanam

EE 645
Jan 22, 2023

Brief recap

Linear to non-linear

Support Vector Classification

Ridge Regression

- ℓ_2 regularization makes it kernelizable

Gaussian process regression

- conditional means of Gaussians = ridge regression
- though ridge computes mean, this is Bayesian
- predictions gaussian (with known variance)

Complexity

If $\mathrm{x}_1, \ldots, \mathrm{x}_n$ are the training points, kernel $k(\cdot,\cdot)$, need the kernel Gram matrix:

$$\begin{bmatrix} k(\mathrm{x}_1,\mathrm{x}_1) & k(\mathrm{x}_1,\mathrm{x}_2) & \ldots & k(\mathrm{x}_1,\mathrm{x}_n) \\ \vdots & \vdots & \vdots & \vdots \\ k(\mathrm{x}_n,\mathrm{x}_1) & k(\mathrm{x}_n,\mathrm{x}_2) & \ldots & k(\mathrm{x}_n,\mathrm{x}_n) \end{bmatrix}$$

The above matrix has n^2 entries (and we often need to invert matrices of this size. Complexity is quadratic or worse.

Deeper look into kernels

In certain cases, we can get to linear complexity in n (training size)

Deeper look into kernels

In certain cases, we can get to linear complexity in n (training size)

approximate solutions, not exact

Deeper look into kernels

In certain cases, we can get to linear complexity in n (training size)

- approximate solutions, not exact
- NeurIPS Test of Time award for influential papers

Deeper look into kernels

In certain cases, we can get to linear complexity in n (training size)

- approximate solutions, not exact
- NeurIPS Test of Time award for influential papers

Makes large training sets feasible

Kernel methods have many of the "amazing" features neural nets have

Deeper look into kernels

In certain cases, we can get to linear complexity in n (training size)

approximate solutions, not exact

NeurIPS Test of Time award for influential papers

Makes large training sets feasible

Kernel methods have many of the "amazing" features neural nets have

Can often fit any random permutations of labels

Deeper look into kernels

In certain cases, we can get to linear complexity in n (training size)

- approximate solutions, not exact
- NeurIPS Test of Time award for influential papers

Makes large training sets feasible

Kernel methods have many of the "amazing" features neural nets have

- Can often fit any random permutations of labels
- ... yet do not misuse power and overfit!

Function positive definiteness

Function $k(\mathrm{x}, \mathrm{y})$ is said to be positive (semi-)definite if

Function positive definiteness

Function $k(\mathrm{x}, \mathrm{y})$ is said to be positive (semi-)definite if for all n and all $\mathrm{x}_1, \ldots ,\mathrm{x}_n$,

Function positive definiteness

Function $k(\mathrm{x}, \mathrm{y})$ is said to be positive (semi-)definite if for all n and all $\mathrm{x}_1, \ldots, \mathrm{x}_n$, the Gram matrix is positive semi-definite.

Function positive definiteness

Function $k(\mathrm{x},\mathrm{y})$ is said to be positive (semi-)definite if for all n and all $\mathrm{x}_1,\ldots,\mathrm{x}_n$, the Gram matrix is positive semi-definite.

This means that for all vectors $\mathrm{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}$

$$\begin{bmatrix} w_1 & \ldots & w_n \end{bmatrix} \begin{bmatrix} k(\mathrm{x}_1,\mathrm{x}_1) & k(\mathrm{x}_1,\mathrm{x}_2) & \ldots & k(\mathrm{x}_1,\mathrm{x}_n) \\ \vdots & \vdots & \vdots & \vdots \\ k(\mathrm{x}_n,\mathrm{x}_1) & k(\mathrm{x}_n,\mathrm{x}_2) & \ldots & k(\mathrm{x}_n,\mathrm{x}_n) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \geq 0$$

Not enough that all entries of Gram matrix ≥ 0

Any positive (semi-)definite k is allowed to be a kernel

Making new kernels from old

Linear combinations with non-negative coeffs

if k_1 and k_2 are two kernels, so is $\alpha k_1(\mathsf{x}, \mathsf{y}) + \beta k_2(\mathsf{x}, \mathsf{y})$

Making new kernels from old

Linear combinations with non-negative coeffs

if k_1 and k_2 are two kernels, so is $\alpha k_1(\mathrm{x}, \mathrm{y}) + \beta k_2(\mathrm{x}, \mathrm{y})$

Product of kernels

if k_1 and k_2 are two kernels, so is $k_1(\mathrm{x}, \mathrm{y}) k_2(\mathrm{x}, \mathrm{y})$

Making new kernels from old

Linear combinations with non-negative coeffs

if k_1 and k_2 are two kernels, so is $\alpha k_1(\mathrm{x},\mathrm{y}) + \beta k_2(\mathrm{x},\mathrm{y})$

Product of kernels

if k_1 and k_2 are two kernels, so is $k_1(\mathrm{x},\mathrm{y}) k_2(\mathrm{x},\mathrm{y})$

If $g(\mathrm{x})$ is any function $k(\mathrm{x},\mathrm{y}) = g(\mathrm{x}) g(\mathrm{y})$ is a kernel

Making new kernels from old

Linear combinations with non-negative coeffs

if k_1 and k_2 are two kernels, so is $\alpha k_1(\mathrm{x},\mathrm{y}) + \beta k_2(\mathrm{x},\mathrm{y})$

Product of kernels

if k_1 and k_2 are two kernels, so is $k_1(\mathrm{x},\mathrm{y}) k_2(\mathrm{x},\mathrm{y})$

If $g(\mathrm{x})$ is any function $k(\mathrm{x},\mathrm{y}) = g(\mathrm{x})g(\mathrm{y})$ is a kernel

If $k(\mathrm{x},\mathrm{y})$ is any kernel, so are $\exp(k(\mathrm{x},\mathrm{y}))$ and $k(f(\mathrm{x}), f(\mathrm{y}))$

Examples

Radial basis function (for scale parameter $s > 0$)

$$k(\mathrm{x}, \mathrm{y}) = \exp\left(-\frac{||\mathrm{x}-\mathrm{y}||^2}{2s}\right)$$

Examples

Radial basis function (for scale parameter $s > 0$)

$$k(\mathrm{x}, \mathrm{y}) = \exp\left(-\frac{||\mathrm{x}-\mathrm{y}||^2}{2s}\right)$$

This function is positive semi-definite because

$$\exp\left(-\frac{||\mathrm{x}-\mathrm{y}||^2}{2s}\right) = \exp\left(-\frac{||\mathrm{x}||^2}{2s}\right)\exp\left(-\frac{||\mathrm{y}||^2}{2s}\right)\exp\left(\frac{\mathrm{x}^T\mathrm{y}}{s}\right)$$

Examples

Exponential/Laplace kernel

$$k(\mathrm{x}, \mathrm{y}) = \exp(-||\mathrm{x} - \mathrm{y}||/\lambda)$$

Examples

Exponential/Laplace kernel

$$k(\mathrm{x}, \mathrm{y}) = \exp(-||\mathrm{x} - \mathrm{y}||/\lambda)$$

Positive semi-definiteness of this function not trivial
but follows easily from Bochner's theorem

Examples

Exponential/Laplace kernel

$$k(\mathrm{x}, \mathrm{y}) = \exp(-||\mathrm{x} - \mathrm{y}||/\lambda)$$

Positive semi-definiteness of this function not trivial
but follows easily from Bochner's theorem
... as for the whole class of Matern kernels and a host of others

Bochner's Theorem

Need this for two reasons

- finding kernels
- faster computation

Bochner's Theorem

Consider kernels $k(\mathrm{x}, \mathrm{y})$ where dependency only via $||\mathrm{x} - \mathrm{y}||$

rbf, Matern

not examples: polynomial

Bochner

$k(\mathrm{x} - \mathrm{y})$, $\mathrm{x}, \mathrm{y} \in \mathbb{R}^d$ is positive semi-definite iff it is the ($d-$dimensional) Fourier transform of a finite positive measure on $\mathbb{R}^d$ (think pdf).

Fourier transform

Let μ be absolutely continuous wrt to the Lebesgue measure (ignore if you haven't heard the terms). Let the pdf of μ be f_μ. Then

$$F(\mathsf{x}-\mathsf{y}) = \int_{\nu\in\mathbb{R}^d} e^{-j2\pi\nu^T(\mathsf{x}-\mathsf{y})} f_\mu(\nu) d\nu$$

is a valid kernel.

we interpret the kernel $k(\mathsf{x},\mathsf{y}) = F(\mathsf{x}-\mathsf{y})$.

we call f_μ the kernel spectral measure

Bochner's kernels

Familiar examples:

if measure is normal, radial basis kernel

similarly for Matern kernels

Bochner's kernels

Familiar examples:

if measure is normal, radial basis kernel

similarly for Matern kernels

Interestingly, these are also universal

any compactly supported function arbitrarily approximated

Computational speedups

Bochner's theorem can also speed up computations (stationary kernels)

From Bochner's theorem

$$k(\mathsf{x}, \mathsf{y}) = \mathbb{E} \exp\left(j 2\pi \nu^T (\mathsf{x} - \mathsf{y})\right)$$

ν random $d-$vector $\sim$ kernel spectral measure $f_\mu(\nu)$
$\mathbb{E}$ denotes expectation

Random Fourier Features

$\mathsf{z}_\nu(\mathsf{x}) = \cos(2\pi\nu^T\mathsf{x} + b)$, $\nu \sim f_\mu$ and b uniform

Random Fourier Features

$\mathsf{z}_\nu(\mathsf{x}) = \cos(2\pi\nu^T \mathsf{x} + b)$, $\nu \sim f_\mu$ and b uniform

Recall feature map $\mathsf{x} \to \phi(\mathsf{x})$, $k(\mathsf{x}, \mathsf{y}) = \phi(\mathsf{x})^T \phi(\mathsf{y})$

replace $\phi(\mathsf{x})$ with $\mathsf{z}(\mathsf{x})$ with same property

yet z is a vector with D coordinates (D small)

$\mathsf{z}^T(\mathsf{x}) = \left[\mathsf{z}_{\nu_1}(\mathsf{x}) \quad , \ldots , \mathsf{z}_{\nu_D}(\mathsf{x})\right]$

$k(\mathsf{x}, \mathsf{y}) = \mathbb{E}_\nu \mathsf{z}_\nu(\mathsf{x})^T \mathsf{z}_\nu(y) \approx \mathsf{z}(\mathsf{x})^T \mathsf{z}(\mathsf{y})$