Linear Algebra CS Lens

Kernel Method

10.11.2023

Afonso Bandeira

Let's revisit regression

Data points $x_1, \ldots, x_n$ (think images)
$(\in \mathbb{R}^m)$

and labels $l_1, \ldots, l_n$ for each point (think cat/dog)
+1/-1

Let's say we have features $\varphi^{(i)}$ (maybe color of the sky)
("amount of blue")

if we have p features then

$$x_k \longrightarrow \varphi(x_k) \in \mathbb{R}^p$$

let's call it φ_k (there are n of them)

Maybe we built good features and

$\forall_k$
$$y_k \sim \sum_{i=1}^{p} \alpha_i \varphi^{(i)}(x_k)$$

(or maybe it's $\operatorname{sign}(\cdot)$ or $\tanh(\cdot)$)

$$y_k \sim \varphi_k^T \alpha \quad \text{for some } \alpha \in \mathbb{R}^p$$

$$\Phi = \begin{bmatrix} \varphi_1^T \\ \vdots \\ \varphi_n^T \end{bmatrix}$$

Φ is $n \times p$.

we can do least squares

$$y \sim \Phi \alpha$$

So $\Phi^T \Phi \hat{\alpha} = \Phi^T y$

$$\Phi^T \Phi \hat{\alpha} = \Phi^T y \quad , \quad \Phi \text{ is } n \times p$$

let's suppose $\Phi^T \Phi$ is invertible
(implies $n \geq p$)

then
$$\hat{\alpha} = (\Phi^T \Phi)^{-1} \Phi^T y$$

let's write
$$\hat{\alpha} = \Phi^T \underbrace{\Phi (\Phi^T \Phi)^{-2} \Phi^T y}_{\hat{\beta} \in \mathbb{R}^n}$$

so $\hat{\alpha} = \Phi^T \beta$

let's substitute back
$$y \sim \Phi \hat{\alpha} = \Phi \Phi^T \hat{\beta}$$

so $y \sim \Phi \Phi^T \beta$ let's expand this ....

$$y_k \sim \varphi_k^T \left(\sum_{j=1}^{n} \varphi_j \beta_j \right) = \sum_{j=1}^{n} \varphi_k^T \varphi_j \beta_j$$

$$= \sum_{j=1}^{n} \beta_j \langle \varphi(x_k), \varphi(x_j) \rangle$$

Define $K(x,y) = \langle \varphi(x), \varphi(y) \rangle$ (called kernel measures "similarity")

then the regression is

$$y_k \approx \sum_{j=1}^{n} \beta_j K(x_k, x_j)$$

$$y_k \approx \sum_{j=1}^{n} \beta_j k(x_k, x_j)$$

→ when $\Phi^T \Phi$ isn't invertible, it's possible to use pseudoinverse (in ML/stats language, it's "ridge regression" "regularization")

→ Possible to do regression only with similarities, w/o features

→ Possible to do φ infinite dimensional

→ which kernels k correspond to features φ? Mercer Thm in functional Analysis

→ Key idea in Machine Learning/Statistics