Métodos Kernel: Construcción del kernel y del aproximador óptimo

Antonio Sala Piqueras

Identificación de sistemas complejos

Dept. Ing. Sistemas y Automatica (DISA)

Universitat Politècnica de València (UPV)

Video-presentación disponible en:

http://personales.upv.es/asala/YT/V/**ktrick2**.html

Presentación

Motivación:

Existen problemas técnicos con "pocas" muestras N de "muchos" elementos q (aprender de 25 imágenes de 1Mpx, de 10 lotes de producto con 90 medidas/lote, etc.), que se abordan con métodos Kernel .

Objetivos:

Revisar "truco del Kernel" (cambio de vble), proponer cómo construir dicho Kernel y comprender cómo deshacer el cambio.

Contenidos:

Planteamiento del problema. Truco Kernel. Función generadora $\kappa(x, y)$. Reconstrucción del aproximador original. Conclusiones.

Revisión: "truco" del Kernel

$$\min_{\Theta} J(\Theta_{m\times q}) = \min_{\Theta}[c \cdot traza(\Theta\Theta^T) + L(Y - \Theta_{m\times q}X)] \underbrace{=}_{[\Theta = wX^T]}$$

$$= \min_{w}[c \cdot traza(w_{m\times N}X^T X w^T) + L(Y - w_{m\times N} \cdot (X^T)_{N\times q}X_{q\times N})]$$

$$= \min_{w}[c \cdot traza(wKw^T) + L(Y - wK)] = \min_{w} J(w_{m\times N})$$

siendo K una matriz de productos escalares de las "features":

$$K_{N\times N} = X^T X = \begin{pmatrix} \phi^T(x_1)\phi(x_1) & \phi^T(x_1)\phi(x_2) & \cdots & \phi^T(x_1)\phi(x_N) \\ \vdots & \ddots & & \vdots \\ \phi^T(x_N)\phi(x_1) & \phi^T(x_N)\phi(x_2) & \cdots & \phi^T(x_N)\phi(x_N) \end{pmatrix}$$

A la matriz K se le denomina **Kernel matrix**.

Construcción del Kernel

- Si se conocen **explícitamente** las funciones del vector ϕ, se evalúa "directamente".
Por ejemplo: identidad (regresión lineal), monomios hasta un cierto grado, valores de intensidad de pixels, entalpías, energías, etc...

- En la literatura, existen propuestas de Kernel que se construyen a partir de funciones elementales $\kappa(x, y)$ sobre pares de datos, $\kappa : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$, para las que existe cierto ϕ **implícito** tal que $\kappa(x, y) = \phi^T(x)\phi(y)$.
  - En ese caso, la matriz se construye a partir de los datos de entrenamiento con $K_{ij} := \kappa(x_i, x_j)$.

Ejemplo: Kernel lineal

Regresión Lineal: Consideremos $\phi(x) = \begin{pmatrix} 1 \\ x \end{pmatrix}$,

$K_{ij} := \phi^T(x_i)\phi(x_j) = 1 + x_i^T x_j$ motiva proponer

$$\kappa(x, y) := 1 + x^T y$$

Operar con este Kernel es equivalente a hacer regresión **lineal**:
$\hat{y} = \theta\phi = \theta_0 + \theta_1 x$.

*Obviamente, la formulación "clásica" (sin "kernel trick") es aconsejable si $q < N$ (p.ej. $x \in \mathbb{R}^2$, $N = 500$) , y la "Kernel" si $q > N$ (p.ej. $x \in \mathbb{R}^{50}$, $N = 22$).

[No olvidar regularización, claro]

Ejemplo: Kernel cuadrático

Cuadrático: Si $x_i = \begin{pmatrix} a_i \\ b_i \end{pmatrix} \in \mathbb{R}^2$, y $\phi(x) = \begin{pmatrix} 1; \\ \sqrt{2}a \\ \sqrt{2}b \\ \sqrt{2}ab \\ a^2 \\ b^2 \end{pmatrix}$ ... entonces

$$K_{ij} = \phi^T(x_i)\phi(x_j) = 1 + 2a_1 \cdot a_2 + 2b_1 \cdot b_2 + 2a_1 b_1 \cdot a_2 b_2 + a_1^2 \cdot a_2^2 + b_1^2 \cdot b_2^2 =$$
$$(1 + a_1 a_2 + b_1 b_2)^2 = (1 + (a_1 \ b_1) \begin{pmatrix} a_2 \\ b_2 \end{pmatrix})^2 = (1 + x_i^T x_j)^2.$$

Esto motiva la propuesta en literatura de $\kappa(x, y) := (1 + x^T y)^2$.

Otros kernels

Kernel polinomial:

Si vector de regresores $\phi(x)$ está formado por todos los monomios de variables en x hasta grado d, que son $q = \frac{(n+d)!}{n!d!}$ (muchos si n y d grandes), se puede formular un problema equivalente de regresión con la matriz K siendo $K_{ij} = (1 + x_i^T x_j)^d$, esto es:

$$\kappa(x, y) := (1 + x^T y)^d$$

Kernel "Radial Basis" (gaussiano):

$$\kappa(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$$

Más opciones: https://data-flair.training/blogs/svm-kernel-functions/ y libros de machine learning.

Reconstrucción del modelo de regresión

Tras calcular óptimo (w^*), el modelo original planteaba $\hat{y}(x) = \Theta \cdot \phi(x)$.

- Como $\Theta^* = w^* X^T$, si se conoce **explícitame** el vector ϕ, evaluado sobre los datos de entrenamiento produce $X = [\phi(x_1) \ \ldots \ \phi(x_N)]$, y θ puede ser calculado y usado.
- Si se ha usado un K de literatura sin formar ϕ, esto es, con un $\kappa(x, y)$, entonces **puede operarse con w^* directamente**. En efecto, $\hat{y}(x) = \Theta^* \cdot \phi(x) = w^* X^T \phi(x) = w \cdot K(X, x)$ siendo

$$K(X, x) := \begin{pmatrix} \phi^T(x_1)\phi(x) \\ \phi^T(x_2)\phi(x) \\ \vdots \\ \phi^T(x_N)\phi(x) \end{pmatrix} = \begin{pmatrix} \kappa(x_1, x) \\ \kappa(x_2, x) \\ \vdots \\ \kappa(x_N, x) \end{pmatrix}$$

Conclusiones

- En problemas con más regresores q que muestras de entrenamiento N, conviene usar métodos **Kernel**.
- Pueden construirse matrices Kernel a partir de características explícitamente diseñadas o a partir de propuestas $\kappa(x, y)$ en literatura, evaluadas sobre las N^2 parejas (x, y) de datos de entrenamiento.
- Una vez obtenido w^*, de dimensión N, no es necesario calcular el parámetro original θ de dimensión q: las mismas $\kappa(x, y)$ pueden ser usadas para construir el modelo de aproximador.

