INTRODUCCIÓN A LOS KERNELS, SUS PROPIEDADES Y APLICACIONES EN MACHINE LEARNING

Abraham Jain Jiménez

Contents

1 Generalidades de los kernels y su aplicación en ML

- El concepto de kernel, hablando con respecto al área del Machine Learning, fue expuesto por primera vez en el artículo:

  Aizerman, M.A., Braverman, E.M. and Rozonoer, L.I. (1964) Theoretical Foundations of Potential Function Method in Pattern Recognition. Automation and Remote Control, 25, 917-936.

- Los kernels en Machine Learning son relevantes en los llamados métodos de kernels, los cuales se utilizan cuando se requiere abordar un problema no lineal, transformándolo y resolviéndolo como un sistema lineal.

- Los kernels son funciones que se basan en una comparación de dos puntos en el espacio de características que representan a nuestros datos:

$$\vec{x}, \vec{x}' \in \mathbb{R}^d$$

2 Introducción a los kernels

Dado un conjunto de datos asociado a un problema de aprendizaje supervisado:

$$D = \{\vec{x}_k \in \mathcal{X} \subseteq \mathbb{R}^d, t_k \in \mathbb{R}\}_{k=1}^M$$

O de aprendizaje no supervisado:

$$D = \{\vec{x}_k \in \mathcal{X} \subseteq \mathbb{R}^d\}_{k=1}^M$$

Definimos una función de comparación k que recibe dos puntos $\vec{x} \in \mathcal{X}$ y regresa un escalar referente a una medida sobre qué tan parecidos son.

$$k = \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

Con k construimos una representación matricial para la comparación del conjunto de pares $(\vec{x}, \vec{x}') \in \mathcal{X} \times \mathcal{X}$ tal que

$$[K]_{ij} = k(\vec{x}_i, \vec{x}_j)$$

Esta matriz, llamada matriz de Gram, contiene toda la info del sistema, pues compara todos los puntos entre sí. Por ejemplo, en el caso en que $i, j = 1, 2, 3$, la matriz de Gram es:

$$K = \begin{pmatrix} k(\vec{x}_1, \vec{x}_1) & k(\vec{x}_1, \vec{x}_2) & k(\vec{x}_1, \vec{x}_3) \\ k(\vec{x}_2, \vec{x}_1) & k(\vec{x}_2, \vec{x}_2) & k(\vec{x}_2, \vec{x}_3) \\ k(\vec{x}_3, \vec{x}_1) & k(\vec{x}_3, \vec{x}_2) & k(\vec{x}_3, \vec{x}_3) \end{pmatrix}$$

Propiedades de K:

- K dice cómo se correlacionan dos puntos en el espacio.
- Es matriz simétrica.
- En general se exige que la diagonal sea $k(\vec{x}_i, \vec{x}_i) = 1$. Es decir, puntos máximamente correlacionados, que la distancia $d(\vec{x}_i, \vec{x}_i)$ sea nula.
- K caracteriza por completo a los datos.

3 Restricciones sobre los kernels

Restricciones en $k(\vec{x}, \vec{x}')$:

1. Debe ser una función simétrica (misma correlación entre $\vec{x}_i$ y $\vec{x}_j$). Es decir:
$$k(\vec{x}_i, \vec{x}_j) = k(\vec{x}_j, \vec{x}_i)$$
2. Dados $\{a_i\} \in \mathbb{R}$, se debe cumplir que:
$$\sum_{i=1}^{M} \sum_{j=1}^{M} a_i a_j k(\vec{x}_i, \vec{x}_j) \geq 0$$

El segundo punto impone características para los valores que puede tomar el kernel. Por otro lado, los dos puntos anteriores convierten a K en una matriz definida positiva (positive definite).

En Machine Learning, imponer estos dos puntos a la representación matricial de los datos y llevarla a algún modelo hace al problema soluble, es decir, el modelo puede aprender.

4 El kernel lineal y otros tipos de kernels

Un caso sencillo de un kernel que sea positive definite es el kernel lineal, que definimos a continuación.

Sea $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ mediante

$$k(\vec{x}, \vec{x}') = \langle \vec{x}, \vec{x}' \rangle$$

Comprobando que sea positive definite:

1. El producto escalar es conmutativo: $\langle \vec{x}, \vec{x}' \rangle = \langle \vec{x}', \vec{x} \rangle$

2.

$$\sum_i \sum_j a_i a_j k(\vec{x}_i, \vec{x}_j) = \sum_i \sum_j a_i a_j \langle \vec{x}_i, \vec{x}_j \rangle = \sum_i \sum_j \langle a_i \vec{x}_i, a_j \vec{x}_j \rangle$$

$$= \langle \sum_i a_i \vec{x}_i, \sum_j a_j \vec{x}_j \rangle = || \sum_{j=1}^{M} a_j \vec{x}_j ||^2 \geq 0 \text{ (por definición)}$$

Así como el kernel lineal, hay otros tipos de funciones que cumplen con estas condiciones. Algunas de las mas usadas en Machine Learning son:

- **Kernel Lineal:**

$$k(\vec{x}_i, \vec{x}_j) = \vec{x}_i^T \vec{x}_j$$

- **Kernel Polinomial:**

$$k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i^T \vec{x}_j + c)^d$$

- **Kernel Gaussiano (o Radial Basis Function):**

$$k(\vec{x}_i, \vec{x}_j) = \exp(-\frac{||\vec{x}_i - \vec{x}_j||^2)}{2\sigma^2}$$

  A veces escrito más general mediante un hiperparámetro γ como

$$k(\vec{x}_i, \vec{x}_j) = \exp(-\gamma ||\vec{x}_i - \vec{x}_j||^2)$$

- **Kernel Sigmoidal:**

$$k(\vec{x}_i, \vec{x}_j) = \tanh(\alpha \vec{x}_i^T \vec{x}_j + c)$$

5 Transformando un problema no lineal en uno lineal

La idea de transformar un problema no lineal en Machine Learning en uno lineal es transformar el espacio de los datos, al cual se asocia el problema no lineal, en un espacio donde el problema pueda abordarse linealmente, ya sea regresión o clasificación.

Esto se hace, comunmente, aumentando el número características en el conjunto de datos, es decir, aumentando la dimensión del espacio de características en donde los datos estén representados.

Por ejemplo, hablando de un problema de clasificación no lineal, queremos transformarlo a un problema lineal, esto lo hacemos encontrando una transformación del espacio donde no se puede clasificar usando una frontera lineal (un

hiperplano) a un espacio donde sí se puede.

Supongamos que tenemos un conjunto de datos de entrenamiento $D = \{\vec{x}_k \in \mathbb{R}^2, t_k \in \{-1, +1\}\}_{k=1}^{M}$ tal que tenemos un clasificador binario no lineal que operó de la siguiente forma:

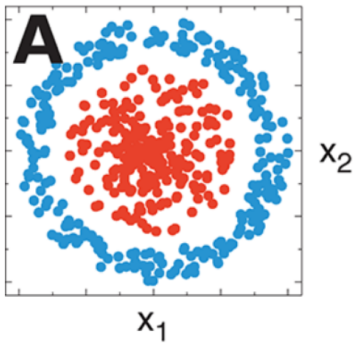

Figure 1: Unke etal. Chem. Rev. 2021, 121, 16, 10142-10186

Definimos una transformación que convierta el espacio asociado al problema no lineal en un espacio al que se le asocie el problema de manera lineal:

$$\vec{\phi} : \mathbb{R}^2 \to \mathbb{R}^3$$

Particularmente

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

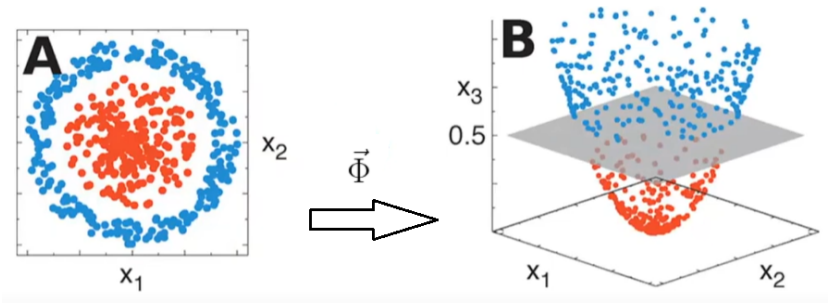

Figure 2: Unke etal. Chem. Rev. 2021, 121, 16, 10142-10186

Entonces, aumentar la dimensión en la descripción de los datos, permite separarlos mediante un hiperplano.

En los métodos de kernels se diseña la estructura de la transformación $\vec{\phi}$ tal que no se necesita saber la forma analítica de esta transformación. Podemos diseñar $\vec{\phi}$, pero en general solo nos interesa saber que existe y simplifica el problema.

6 Espacios de Hilbert y métodos de kernels

Supongamos que existe una transformación $\vec{\phi}$ de $\mathcal{X} \subseteq \mathbb{R}^d$ a un espacio de Hilbert $\mathcal{H}$ que es de mayor dimensión:

$$\vec{\phi} = \mathcal{X} \to \mathcal{H}$$

Por ejemplo, dado un problema de aprendizaje supervisado en donde $d = 2$ y $M = 3$:

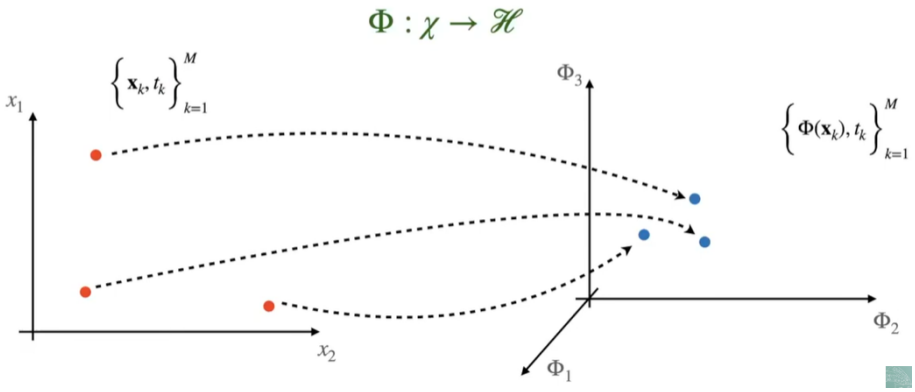

Figure 3: Créditos al Dr. Huziel Enoc Sauceda Felix

Notemos que la etiqueta t_k se conserva, lo que cambia es la representación numérica: $\vec{x}_k \to \vec{\phi}(\vec{x}_k)$.

Definimos el kernel usando $\vec{\phi}$:

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R},$$
$$k(\vec{x}, \vec{x}') := \langle \vec{\phi}(\vec{x}), \vec{\phi}(\vec{x}') \rangle = \vec{\phi}(\vec{x})^T \vec{\phi}(\vec{x}')$$

En una notación menos general se tiene que $\mathcal{H} = \mathbb{R}^q$, en donde se tiene que $q > d$.

El hecho de que la función k se pueda expresar como el producto escalar de dos vectores es consecuencia del teorema de Mercer:

- Una función kernel k puede ser expresada como $k(\vec{u},\vec{v}) = \langle\vec{\phi}(\vec{u}),\vec{\phi}(\vec{v})\rangle \iff \exists g \in L_2(\mathbb{R}^n) : \mathbb{R}^n \to \mathbb{R}$ tal que

$$\int [k(\vec{u},\vec{v})g(\vec{u})g(\vec{v})]d\vec{u}d\vec{v} \geq 0$$

(El espacio donde vive g es $L_2(\mathbb{R}^n) := \{f : \mathbb{R}^n \to \mathbb{R} \mid \int |f(\vec{w})|^2 d\vec{w} < \infty\}$)

Esto puede demostrarse para cada ejemplo de kernel expuesto.

Definir el kernel mediante un producto punto exige a $\vec{\phi}$ que mapee de un espacio arbitrario a uno de Hilbert, ya que este tiene un producto interior definido.

En Machine Learning hay algoritmos que solo trabajan correctamente sobre conjuntos de datos a los que se asocian relaciones lineales. Por ejemplo, exponemos tres casos:

- La regresión lineal regularizada de **Ridge** modela relaciones lineales entre las variables. Si los datos no siguen un comportamiento lineal, es decir, no son linealmente separables, el rendimiento de este modelo no sería el adecuado. Por ello, se usa un método de kernels para linearizar el problema, pero, ¿dónde entra el kernel?.

  En regresión Ridge kernelizada, el problema de optimización se define como:

$$\vec{w}^* = ArgMin_{\vec{w}}\{L(\vec{w})\} = ArgMin_{\vec{w}}\{\frac{1}{N}||\vec{f} - K\vec{w}||^2 + \lambda||\vec{w}||^2\}$$

  En donde K es la matriz de Gram del problema, cuyas entradas son

$$[K]_{ij} = k(\vec{x_i},\vec{x_j}) := \langle\vec{\phi}(\vec{x_i}),\vec{\phi}(\vec{x_j})\rangle = \vec{\phi}(\vec{x_i})^T\vec{\phi}(\vec{x_j})$$

  Teniendo que $X \in \mathbb{R}^{d\times M}$ es la matriz de diseño sociada al sistema, entonces la matriz que transforma a X es:

$$\Phi = \Phi(X) \in \mathbb{R}^{d\times q} \Rightarrow K = \Phi\Phi^T$$

- Otro ejemplo: **PCA** captura relaciones lineales entre las variables para la reducción de la dimensión, y se aplica un método de kernels para linearizar el problema cuando no se presenta esta estructura lineal en los datos.

- Las SVM se fundamentan en la clasificación lineal, es decir, mediante un hiperplano de separación. Cuando el problema es no lineal, se linealiza mediante un método de kernels.

—

7 Conslusión: Los kernels en ML

- Con $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, mediante $k(\vec{x_i}, \vec{x_j}) := \langle \vec{\phi}(\vec{x_i}), \vec{\phi}(\vec{x_j}) \rangle$, construimos la matriz K tal que $[K]_{ij} = k(\vec{x_i}, \vec{x_j})$.

A su vez, K permite:

- Aplicar operaciones lineales en espacios donde los datos no son linealmente separebles originalmente.
- Capturar relaciones de similitud para métodos basados en similitud (útil en clustering).
- Desarrollar algoritmos kernelizados.