Ciencia de Datos

Victor Muñiz
victor_m@cimat.mx

Asistente:
Víctor Gómez
victor.gomez@cimat.mx

Maestría en Cómputo Estadístico.
Centro de Investigación en Matemáticas.
Unidad Monterrey.

Enero-Junio 2021

Aprendizaje de variedades (Manifold Learning)

Manifold learning

Manifold (mathworld.wolfram.com)

A manifold is a topological space that is locally Euclidean (i.e., around every point, there is a neighborhood that is topologically the same as the open unit ball in $\mathbb{R}^d$)

Nosotros, consideraremos una variedad como una representación (embedding) particular de datos en alta dimensión.

Veremos algunos métodos que nos permiten recuperar ésta representación completa, de baja dimensión, de una variedad no-lineal desconocida $\mathcal{M}$, pero que se puede aprender mediante nuestros datos. Esta variedad se considera embebida en un espacio de mayor dimensión dado por nuestros datos en el espacio de entrada $\mathcal{X}$.

Manifold learning

Manifold (mathworld.wolfram.com)

A manifold is a topological space that is locally Euclidean (i.e., around every point, there is a neighborhood that is topologically the same as the open unit ball in $\mathbb{R}^d$)

Nosotros, consideraremos una variedad como una representación (embedding) particular de datos en alta dimensión.

Veremos algunos métodos que nos permiten recuperar ésta representación completa, de baja dimensión, de una variedad no-lineal desconocida $\mathcal{M}$, pero que se puede aprender mediante nuestros datos. Esta variedad se considera embebida en un espacio de mayor dimensión dado por nuestros datos en el espacio de entrada $\mathcal{X}$.

Manifold learning

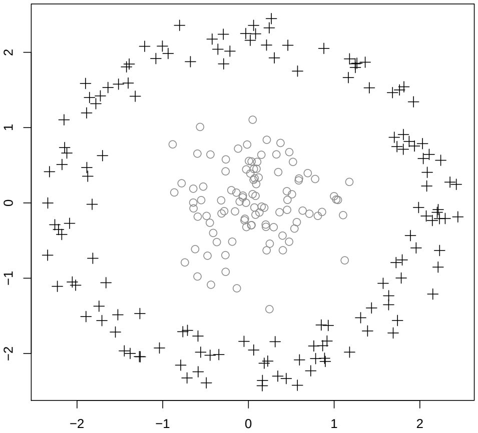
Manifold (mathworld.wolfram.com)

A manifold is a topological space that is locally Euclidean (i.e., around every point, there is a neighborhood that is topologically the same as the open unit ball in $\mathbb{R}^d$)

Nosotros, consideraremos una variedad como una representación (embedding) particular de datos en alta dimensión.
Veremos algunos métodos que nos permiten recuperar ésta representación completa, de baja dimensión, de una variedad no-lineal desconocida $\mathcal{M}$, pero que se puede aprender mediante nuestros datos. Esta variedad se considera embebida en un espacio de mayor dimensión dado por nuestros datos en el espacio de entrada $\mathcal{X}$.

Métodos de kernel

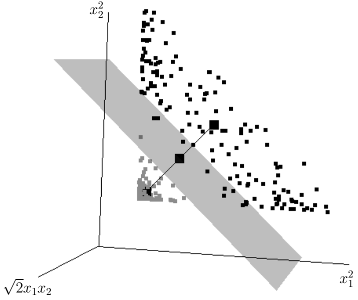
Patrones no-lineales y transformaciones implícitas.



Métodos de kernel

Considera el mapeo ϕ de $\mathbb{R}^2$ a $\mathbb{R}^3$:

$$\phi : \mathbf{x} = (x_1, x_2) \mapsto \phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2);$$



Métodos de kernel

Sean $\mathbf{x}$ y $\mathbf{z}$ dos puntos en $\mathbb{R}^2$, su producto punto en el espacio transformado es:

$$\begin{aligned}\langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{z}) \rangle &= \langle (x_1^2, x_2^2, \sqrt{2}x_1x_2), (z_1^2, z_2^2, \sqrt{2}z_1z_2) \rangle \\ &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1x_2z_1z_2 \\ &= (x_1z_1 + x_2z_2)^2 \\ &= \langle \mathbf{x}, \mathbf{z} \rangle^2 = k(\mathbf{x}, \mathbf{z})\end{aligned}$$

es decir, la función $k(\mathbf{x}, \mathbf{z})$ nos permite calcular el producto punto de dos puntos transformados **sin tener explícitamente sus coordenadas en tal espacio**.

Métodos de kernel

Truco del kernel

Reemplazar los productos punto por un kernel $k(\cdot, \cdot)$ adecuado.
Requisito: el método en cuestión debe poder expresarse en términos de productos punto.

Métodos de kernel

Definición (kernel)

Sean $\mathbf{x}, \mathbf{z} \in \mathcal{X}$*, donde* $\mathcal{X}$ *denota el espacio de entrada. Un kernel* k *es una función que calcula el producto punto de dos puntos transformados mediante cierta función* ϕ*:*

$$k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle \tag{1}$$

donde ϕ *es un mapeo de* $\mathcal{X}$ *a un espacio de productos punto* $\mathcal{H}$*, al que llamaremos espacio de características:*

$$\phi : \mathbf{x} \in \mathcal{X} \mapsto \phi(\mathbf{x}) \in \mathcal{H}.$$

Métodos de kernel

El hecho de considerar el mapeo del espacio de entrada $\mathcal{X}$ al espacio de productos punto $\mathcal{H}$ tiene una utilidad específica, y es que, como se vio en el ejemplo anterior, el producto punto nos permite definir medidas de similaridad entre objetos provenientes de $\mathcal{X}$ aún cuando no sea sencillo obtenerlos en este espacio original.

¿Cómo definir un kernel válido, es decir, que cumpla (1)?

Métodos de kernel

La clase de kernels que satisfacen la definición 1 corresponden a los kernels semi definidos positivos.

Definición (kernel semi definido positivo)

Sea k un kernel según la definición 1. Se dice que k es un kernel semi definido positivo si, para todo $n \in \mathbb{N}$, $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ y $c_1, \ldots, c_n \in \mathbb{R}$, se cumple que

$$\sum_{i,j \in \{1,\ldots,n\}} c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \tag{2}$$

Métodos de kernel

La definición anterior puede expresarse en términos de una matriz formada por evaluaciones de la función kernel, y tiene una función primordial en el análisis y desarrollo de todos los métodos basados en kernels.

Definición (Matriz de Gram)

Dado un kernel k y un conjunto de datos de entrada $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, la matriz $\mathbf{K}$ de $n \times n$ con entradas

$$K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) \tag{3}$$

es llamada matriz de Gram (o matriz de kernel) de k con respecto a $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

Métodos de kernel

Generalmente se consideran kernels simétricos, es decir $k(\mathbf{x}, \mathbf{z}) = k(\mathbf{z}, \mathbf{x})$, por lo tanto $\mathbf{K}$ será simétrica. Una matriz real y simétrica $\mathbf{K}$ es semi definida positiva si, para todo $\mathbf{c} \in \mathbb{R}^n$,

$$\mathbf{c}^T \mathbf{K} \mathbf{c} \geq 0, \tag{4}$$

que es equivalente a (2), por lo tanto, un kernel será semi definido positivo si la matriz de Gram formada con este kernel es semi definida positiva.

Métodos de kernel

Se puede demostrar que una función simétrica k es un kernel si y solo si k es semi definido positivo. Una parte de la demostración viene dada por el hecho de que las matrices de Gram son semi definidas positivas, como se muestra a continuación.

Resultado

Las matrices de Gram son semi definidas positivas. Para ver esto, notemos que, para cualquier vector $\mathbf{v}$

$$\begin{aligned}\mathbf{v}^T\mathbf{K}\mathbf{v} &= \sum_{i,j=1} v_i v_j k(\mathbf{x}_i,\mathbf{x}_j)\\ &= \sum_{i,j=1} v_i v_j \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\rangle\\ &= \langle \sum_i v_i \boldsymbol{\phi}(\mathbf{x}_i), \langle \sum_j v_j \boldsymbol{\phi}(\mathbf{x}_j)\rangle\\ &= \| \sum_i v_i \boldsymbol{\phi}(\mathbf{x}_i)\|^2 \geq 0,\end{aligned}$$

donde la desigualdad se debe a la no negatividad de la norma.

Métodos de kernel

Para completar la demostración tiene que probarse el hecho de que, partiendo de un kernel semi definido positivo, puede construirse un mapeo ϕ dentro de un espacio de características asociado a k de tal forma que se cumpla el producto punto (1). Puede consultarse en (Shawe-Cristianini).

Métodos de kernel

Algunos ejemplos de kernels:

- **Kernel polinomial de grado p:**

$$k(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + c)^p,$$

  con $p \in \mathbb{N}$ y $c \geq 0$.

- **Kernel Gaussiano:** Este kernel, con parámetro $\sigma > 0$, se define como:

$$k(\mathbf{x}, \mathbf{z}) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right). \tag{5}$$

Métodos de kernel

Caracterización con el kernel Gaussiano.

- Cuando tenemos n observaciones diferentes $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ y $\sigma \neq 0$, puede mostrarse que K de $n \times n$ es de rango completo, es decir,$\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_n)$ son linealmente independientes y generan un subespacio $n-$dimensional en $\mathcal{H}$.
- cada dato mapeado está normalizado, ya que, como $k(\mathbf{x}, \mathbf{x}) = 1$ implica que $\|\boldsymbol{\phi}(\mathbf{x})\| = 1$ y además, como el producto punto entre cualquier par de puntos mapeados es positivo, todos los puntos en el espacio de características se encuentran en el mismo octante de una hiperesfera en $\mathbb{R}^{\infty}$.

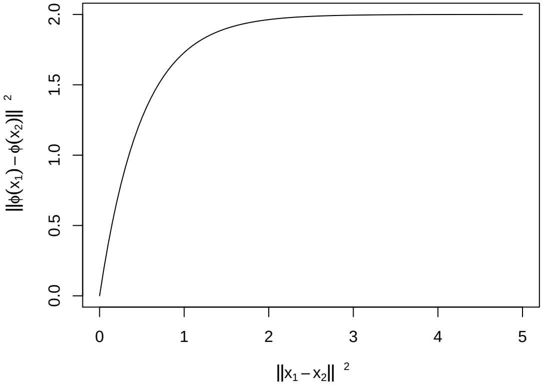
Métodos de kernel

Caracterización con el kernel Gaussiano.

- La expresión $\|\frac{1}{n}\sum_j \phi(\mathbf{x}_j)\|$ está relacionada con la variabilidad de los datos y $\|\phi(\mathbf{x}_i) - \frac{1}{n}\sum_j \phi(\mathbf{x}_j)\|$ con la densidad de las observaciones alrededor de $\mathbf{x}_i$.
- La distancia entre dos puntos transformados está dada por

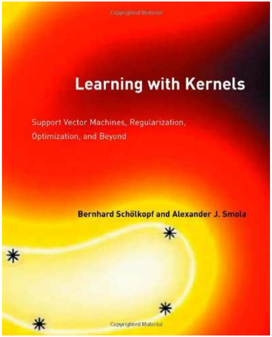
$$\begin{aligned} \|\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)\|^2 &= k(\mathbf{x}_1, \mathbf{x}_1) - 2k(\mathbf{x}_1, \mathbf{x}_2) + k(\mathbf{x}_2, \mathbf{x}_2) \\ &= 2(1 - e^{-\frac{||\mathbf{x}_1 - \mathbf{x}_2||^2}{2\sigma^2}}), \end{aligned}$$

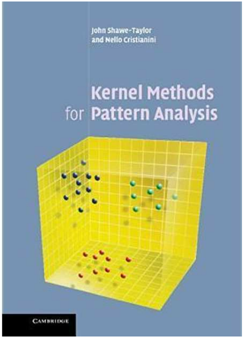
Métodos de kernel



No crece de forma arbitraria!!!

Kernels como base para definir espacios de funciones





Kernels y espacios de funciones

Espacios de Hilbert

Ya vimos que un kernel es la representación del producto punto de sus argumentos en algún espacio, por lo tanto, el producto punto tiene una importancia fundamental en el análisis y caracterización de kernels.

Consideremos un espacio lineal (o vectorial) $\mathcal{H}$. Un producto punto es válido si, para todo $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{H}$ cumple con las siguientes características:

- $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$ (simetría),
- $\langle \alpha\mathbf{x}, \mathbf{y} \rangle = \alpha\langle \mathbf{x}, \mathbf{y} \rangle$ y $\langle \mathbf{x} + \mathbf{z}, \mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{z}, \mathbf{y} \rangle$, donde $\alpha \in \mathbb{R}$ (linearidad) y
- $\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$ (positividad).

A través del producto punto podemos definir una norma mediante $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$.

Kernels y espacios de funciones

Espacios de Hilbert

Ya vimos que un kernel es la representación del producto punto de sus argumentos en algún espacio, por lo tanto, el producto punto tiene una importancia fundamental en el análisis y caracterización de kernels.

Consideremos un espacio lineal (o vectorial) $\mathcal{H}$. Un producto punto es válido si, para todo $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{H}$ cumple con las siguientes características:

- $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$ (simetría),
- $\langle \alpha\mathbf{x}, \mathbf{y} \rangle = \alpha\langle \mathbf{x}, \mathbf{y} \rangle$ y $\langle \mathbf{x} + \mathbf{z}, \mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{z}, \mathbf{y} \rangle$, donde $\alpha \in \mathbb{R}$ (linearidad) y
- $\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$ (positividad).

A través del producto punto podemos definir una norma mediante $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$.

Kernels y espacios de funciones

El espacio lineal $\mathcal{H}$ que tiene un producto punto válido es llamado también **espacio de productos punto**, y uno de especial interés es el llamado espacio de Hilbert.

Definición (Espacio de Hilbert)

Un espacio de Hilbert $\mathcal{H}$ es un espacio lineal, de dimensión (posiblemente) infinita dotado de un producto punto, y que además es completo.

La característica de completitud asegura que cualquier secuencia $\{\mathbf{z}_n\}_{n\geq 1}$ de elementos de $\mathcal{H}$ converge a un elemento $\mathbf{z}$ de $\mathcal{H}$ con respecto a la distancia definida por el producto punto, es decir: $\|\mathbf{z}_n - \mathbf{z}\| \to 0$ cuando $n \to \infty$.

Kernels y espacios de funciones

El espacio lineal $\mathcal{H}$ que tiene un producto punto válido es llamado también **espacio de productos punto**, y uno de especial interés es el llamado espacio de Hilbert.

Definición (Espacio de Hilbert)

Un espacio de Hilbert $\mathcal{H}$ es un espacio lineal, de dimensión (posiblemente) infinita dotado de un producto punto, y que además es completo.

La característica de completitud asegura que cualquier secuencia $\{\mathbf{z}_n\}_{n\geq 1}$ de elementos de $\mathcal{H}$ converge a un elemento $\mathbf{z}$ de $\mathcal{H}$ con respecto a la distancia definida por el producto punto, es decir: $\|\mathbf{z}_n - \mathbf{z}\| \to 0$ cuando $n \to \infty$.

Kernels y espacios de funciones

Ejemplos:

- $\mathbb{R}^d$, el espacio euclidiano de dimensión d
- Funciones lineales $\mathcal{H}_L$ reales:

$$\mathcal{H}_L = \{f : f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle, \ \mathbf{w}, \mathbf{x} \in \mathcal{X}\},$$

con norma $\|f\|_{\mathcal{H}_L} = \|\mathbf{w}\|$.

Kernels y espacios de funciones

Ejemplos:

- Funciones cuadradas integrables y real valuadas en el intervalo $[a, b]$:

$$L_2[a, b] = \left\{ f : \int_a^b f(x)^2 dx < \infty \right\},$$

con producto punto

$$\langle f, g \rangle = \int_a^b f(x) g(x) dx.$$

Aplicaciones de esto puedes encontrarlo en Análisis Funcional, donde nuestros datos son funciones que dependen, por ejemplo, del tiempo:

$$x = x(t), \quad t = 0, 1, \ldots$$

ver por ejemplo, Ramsay & Silverman, *Functional Data Analysis*, 2005.

Kernels y espacios de funciones

Reproducing Kernel Hilbert Spaces (RKHS)

Lo definimos como el espacio de funciones:

$$\mathcal{H} = \left\{ f : f(\cdot) = \sum_{i=1}^{n} \alpha_i k(\cdot, \mathbf{x}_i) : n \in \mathbb{N}, \alpha_i \in \mathbb{R}, \mathbf{x}_i \in \mathcal{X} \right\},$$

donde k es un kernel simétrico, positivo definido y real-valuado.

Consideremos $f, g \in \mathcal{H}$ dadas por

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i k(\cdot, \mathbf{x}_i), \quad \text{y} \quad g(\cdot) = \sum_{j=1}^{m} \beta_j k(\cdot, \mathbf{z}_j), \qquad (6)$$

con $m \in \mathbb{N}, \beta_j \in \mathbb{R}$ y $\mathbf{z}_j \in \mathcal{X}$.

Kernels y espacios de funciones

El producto punto está definido como

$$\langle f, g \rangle = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j k(\mathbf{x}_i, \mathbf{z}_j) = \sum_{i=1}^{n} \alpha_i g(\mathbf{x}_i) = \sum_{j=1}^{m} \beta_j f(\mathbf{z}_j), \tag{7}$$

puede demostrarse que el espacio de funciones $\mathcal{H}$ es un espacio de Hilbert (según la definición que dimos), y partiendo de la definición de f y su producto punto, obtenemos la siguiente propiedad fundamental.

Kernels y espacios de funciones

Propiedad de reproducción (reproducing property) del kernel

$$\langle k(\cdot, \mathbf{x}), f \rangle = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i) = f(\mathbf{x}). \qquad (8)$$

En particular

$$\langle k(\cdot, \mathbf{x}), k(\cdot, \mathbf{z}) \rangle = k(\mathbf{x}, \mathbf{z}).$$

Ahora, recuerda la definición que dimos antes (1): $k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$. Con un kernel definido positivo y usando la propiedad de reproducción del kernel, podemos ver que

$$\phi(\mathbf{x}) = k(\mathbf{x}, \cdot),$$

es decir, tenemos un kernel válido.

Por la propiedad de reproducción del kernel, el espacio de funciones $\mathcal{H}$ definido es llamado **Reproducing Kernel Hilbert Space (RKHS)**.

Kernels y espacios de funciones

Propiedad de reproducción (reproducing property) del kernel

$$\langle k(\cdot, \mathbf{x}), f \rangle = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i) = f(\mathbf{x}). \tag{8}$$

En particular

$$\langle k(\cdot, \mathbf{x}), k(\cdot, \mathbf{z}) \rangle = k(\mathbf{x}, \mathbf{z}).$$

Ahora, recuerda la definición que dimos antes (1): $k(\mathbf{x}, \mathbf{z}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{z}) \rangle$. Con un kernel definido positivo y usando la propiedad de reproducción del kernel, podemos ver que

$$\boldsymbol{\phi}(\mathbf{x}) = k(\mathbf{x}, \cdot),$$

es decir, tenemos un kernel válido.

Por la propiedad de reproducción del kernel, el espacio de funciones $\mathcal{H}$ definido es llamado **Reproducing Kernel Hilbert Space (RKHS)**.

Kernels y espacios de funciones

Propiedad de reproducción (reproducing property) del kernel

$$\langle k(\cdot, \mathbf{x}), f \rangle = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i) = f(\mathbf{x}). \tag{8}$$

En particular

$$\langle k(\cdot, \mathbf{x}), k(\cdot, \mathbf{z}) \rangle = k(\mathbf{x}, \mathbf{z}).$$

Ahora, recuerda la definición que dimos antes (1): $k(\mathbf{x}, \mathbf{z}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{z}) \rangle$. Con un kernel definido positivo y usando la propiedad de reproducción del kernel, podemos ver que

$$\boldsymbol{\phi}(\mathbf{x}) = k(\mathbf{x}, \cdot),$$

es decir, tenemos un kernel válido.

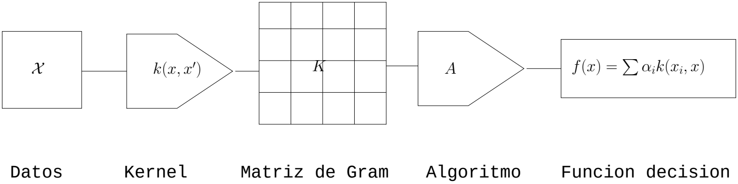
Por la propiedad de reproducción del kernel, el espacio de funciones $\mathcal{H}$ definido es llamado **Reproducing Kernel Hilbert Space (RKHS)**.

Kernels y espacios de funciones

¿Para qué nos sirve lo anterior?

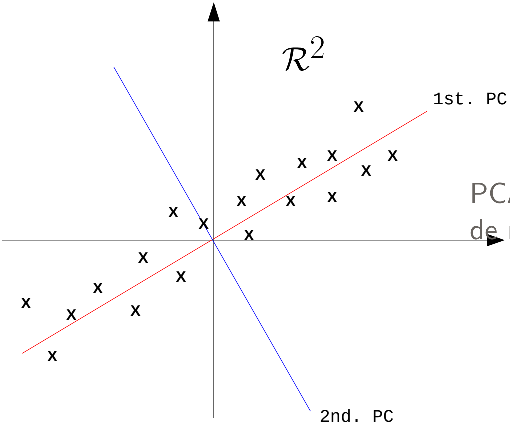
Entre otras cosas, para definir una familia de métodos de aprendizaje muy poderosos:

Métodos de Kernel



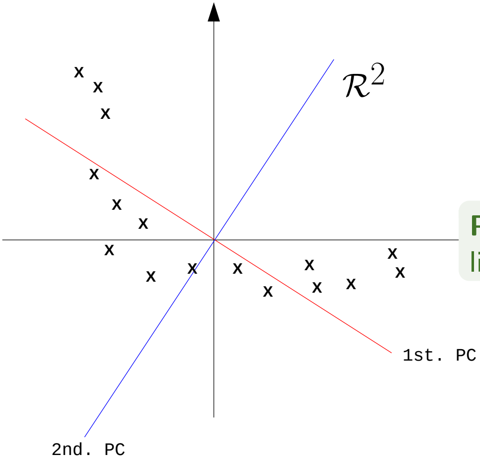
Kernel PCA. Una versión no lineal de PCA.

Kernel PCA



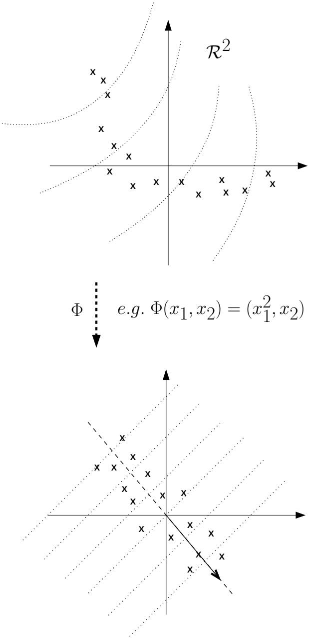
PCA busca direcciones de máxima varianza.

Kernel PCA



Problema: patrones no lineales en los datos.

Kernel PCA



Solución usando transformaciones.

Kernel PCA, una versión no lineal de PCA

- Consideraremos nuestro espacio de entrada $\mathcal{X}$ como el espacio euclideano de dimensión d, entonces $\mathbf{x} \in \mathbb{R}^d$ una observación con d características. $\mathbf{X}$ de $n \times d$ será nuestra matriz de datos (centrada por columnas) con n observaciones.
- La matriz de covarianzas estimada está dada por (simplificando la notación):

$$\mathbf{C} = \mathbf{X}^T \mathbf{X}.$$

- Considerando el producto punto en $\mathbb{R}^d$, la matriz de Gram puede escribirse como

$$\mathbf{K} = \mathbf{X}\mathbf{X}^T,$$

donde $K_{i,j} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$.

Kernel PCA, una versión no lineal de PCA

Puede mostrarse que la solución de PCA implica encontrar eigenvalores $\lambda \geq 0$ y eigenvectores $\mathbf{u} \neq 0$ tales que

$$\begin{aligned}\mathbf{C}\mathbf{u} &= \lambda\mathbf{u}\\ (\mathbf{X}^T\mathbf{X})\mathbf{u} &= \lambda\mathbf{u},\end{aligned}$$

multiplicando por $\mathbf{X}$ por la izquierda

$$\begin{aligned}\mathbf{X}\mathbf{X}^T(\mathbf{X}\mathbf{u}) &= \lambda(\mathbf{X}\mathbf{u})\\ \mathbf{K}(\mathbf{X}\mathbf{u}) &= \lambda(\mathbf{X}\mathbf{u}),\end{aligned}$$

Entonces $(\lambda^{-1/2}\mathbf{X}\mathbf{u}, \lambda)$ es un par eigenvector-eigenvalor normalizado de $\mathbf{K}$.

Podemos definir una relación similar partiendo de la matriz de Gram:

$$\begin{aligned}\mathbf{K}\mathbf{v} &= \lambda\mathbf{v}\\ (\mathbf{X}\mathbf{X}^T)\mathbf{v} &= \lambda\mathbf{v},\end{aligned}$$

multiplicando por $\mathbf{X}^T$ por la izquierda

$$\begin{aligned}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{v}) &= \lambda\mathbf{X}^T\mathbf{v}\\ \mathbf{C}(\mathbf{X}^T\mathbf{v}) &= \lambda(\mathbf{X}^T\mathbf{v}),\end{aligned}$$

por lo que $(\lambda^{-1/2}\mathbf{X}^T\mathbf{v}, \lambda)$ es un par eigenvector-eigenvalor normalizado de $\mathbf{C}$.

Las proyecciones en los componentes principales $P_{(\lambda^{-1/2}\mathbf{X}^T\mathbf{v})}(\mathbf{x})$ se obtienen mediante $\langle \mathbf{x}, \lambda^{-1/2}\mathbf{X}^T\mathbf{v}\rangle$, entonces, solamente necesitamos productos punto.

Lo anterior muestra la relación entre los eigenvalores y eigenvectores de $\mathbf{C}$ y $\mathbf{K}$. La conclusión es que podemos realizar PCA usando la matriz de covarianzas $\mathbf{C}$ o la matriz de productos punto $\mathbf{K}$.

Kernel PCA, una versión no lineal de PCA

Puede mostrarse que la solución de PCA implica encontrar eigenvalores $\lambda \geq 0$ y eigenvectores $\mathbf{u} \neq 0$ tales que

$$\begin{aligned} \mathbf{C}\mathbf{u} &= \lambda\mathbf{u} \\ (\mathbf{X}^T\mathbf{X})\mathbf{u} &= \lambda\mathbf{u}, \end{aligned}$$

multiplicando por $\mathbf{X}$ por la izquierda

$$\begin{aligned} \mathbf{X}\mathbf{X}^T(\mathbf{X}\mathbf{u}) &= \lambda(\mathbf{X}\mathbf{u}) \\ \mathbf{K}(\mathbf{X}\mathbf{u}) &= \lambda(\mathbf{X}\mathbf{u}), \end{aligned}$$

Entonces $(\lambda^{-1/2}\mathbf{X}\mathbf{u}, \lambda)$ es un par eigenvector-eigenvalor normalizado de $\mathbf{K}$.

Podemos definir una relación similar partiendo de la matriz de Gram:

$$\begin{aligned} \mathbf{K}\mathbf{v} &= \lambda\mathbf{v} \\ (\mathbf{X}\mathbf{X}^T)\mathbf{v} &= \lambda\mathbf{v}, \end{aligned}$$

multiplicando por $\mathbf{X}^T$ por la izquierda

$$\begin{aligned} \mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{v}) &= \lambda\mathbf{X}^T\mathbf{v} \\ \mathbf{C}(\mathbf{X}^T\mathbf{v}) &= \lambda(\mathbf{X}^T\mathbf{v}), \end{aligned}$$

por lo que $(\lambda^{-1/2}\mathbf{X}^T\mathbf{v}, \lambda)$ es un par eigenvector-eigenvalor normalizado de $\mathbf{C}$.

Las proyecciones en los componentes principales $P_{(\lambda^{-1/2}\mathbf{X}^T\mathbf{v})}(\mathbf{x})$ se obtienen mediante $\langle \mathbf{x}, \lambda^{-1/2}\mathbf{X}^T\mathbf{v} \rangle$, entonces, solamente necesitamos productos punto.

Lo anterior muestra la relación entre los eigenvalores y eigenvectores de $\mathbf{C}$ y $\mathbf{K}$. La conclusión es que podemos realizar PCA usando la matriz de covarianzas $\mathbf{C}$ o la matriz de productos punto $\mathbf{K}$.

Kernel PCA, una versión no lineal de PCA

Puede mostrarse que la solución de PCA implica encontrar eigenvalores $\lambda \geq 0$ y eigenvectores $\mathbf{u} \neq 0$ tales que

$$\begin{aligned} \mathbf{C}\mathbf{u} &= \lambda \mathbf{u} \\ (\mathbf{X}^T\mathbf{X})\mathbf{u} &= \lambda \mathbf{u}, \end{aligned}$$

multiplicando por $\mathbf{X}$ por la izquierda

$$\begin{aligned} \mathbf{X}\mathbf{X}^T(\mathbf{X}\mathbf{u}) &= \lambda(\mathbf{X}\mathbf{u}) \\ \mathbf{K}(\mathbf{X}\mathbf{u}) &= \lambda(\mathbf{X}\mathbf{u}), \end{aligned}$$

Entonces $(\lambda^{-1/2}\mathbf{X}\mathbf{u}, \lambda)$ es un par eigenvector-eigenvalor normalizado de $\mathbf{K}$.

Podemos definir una relación similar partiendo de la matriz de Gram:

$$\begin{aligned} \mathbf{K}\mathbf{v} &= \lambda \mathbf{v} \\ (\mathbf{X}\mathbf{X}^T)\mathbf{v} &= \lambda \mathbf{v}, \end{aligned}$$

multiplicando por $\mathbf{X}^T$ por la izquierda

$$\begin{aligned} \mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{v}) &= \lambda \mathbf{X}^T\mathbf{v} \\ \mathbf{C}(\mathbf{X}^T\mathbf{v}) &= \lambda(\mathbf{X}^T\mathbf{v}), \end{aligned}$$

por lo que $(\lambda^{-1/2}\mathbf{X}^T\mathbf{v}, \lambda)$ es un par eigenvector-eigenvalor normalizado de $\mathbf{C}$.

Las proyecciones en los componentes principales $P_{(\lambda^{-1/2}\mathbf{X}^T\mathbf{v})}(\mathbf{x})$ se obtienen mediante $\langle \mathbf{x}, \lambda^{-1/2}\mathbf{X}^T\mathbf{v}\rangle$, entonces, solamente necesitamos productos punto.

Lo anterior muestra la relación entre los eigenvalores y eigenvectores de $\mathbf{C}$ y $\mathbf{K}$. La conclusión es que podemos realizar PCA usando la matriz de covarianzas $\mathbf{C}$ o la matriz de productos punto $\mathbf{K}$.

Kernel PCA, una versión no lineal de PCA

Puede mostrarse que la solución de PCA implica encontrar eigenvalores $\lambda \geq 0$ y eigenvectores $\mathbf{u} \neq 0$ tales que

$$\begin{aligned} \mathbf{Cu} &= \lambda \mathbf{u} \\ (\mathbf{X}^T\mathbf{X})\mathbf{u} &= \lambda \mathbf{u}, \end{aligned}$$

multiplicando por $\mathbf{X}$ por la izquierda

$$\begin{aligned} \mathbf{X}\mathbf{X}^T(\mathbf{Xu}) &= \lambda(\mathbf{Xu}) \\ \mathbf{K}(\mathbf{Xu}) &= \lambda(\mathbf{Xu}), \end{aligned}$$

Entonces $(\lambda^{-1/2}\mathbf{Xu}, \lambda)$ es un par eigenvector-eigenvalor normalizado de $\mathbf{K}$.

Podemos definir una relación similar partiendo de la matriz de Gram:

$$\begin{aligned} \mathbf{Kv} &= \lambda \mathbf{v} \\ (\mathbf{X}\mathbf{X}^T)\mathbf{v} &= \lambda \mathbf{v}, \end{aligned}$$

multiplicando por $\mathbf{X}^T$ por la izquierda

$$\begin{aligned} \mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{v}) &= \lambda \mathbf{X}^T\mathbf{v} \\ \mathbf{C}(\mathbf{X}^T\mathbf{v}) &= \lambda(\mathbf{X}^T\mathbf{v}), \end{aligned}$$

por lo que $(\lambda^{-1/2}\mathbf{X}^T\mathbf{v}, \lambda)$ es un par eigenvector-eigenvalor normalizado de $\mathbf{C}$.

Las proyecciones en los componentes principales $P_{(\lambda^{-1/2}\mathbf{X}^T\mathbf{v})}(\mathbf{x})$ se obtienen mediante $\langle \mathbf{x}, \lambda^{-1/2}\mathbf{X}^T\mathbf{v} \rangle$, entonces, solamente necesitamos productos punto.

Lo anterior muestra la relación entre los eigenvalores y eigenvectores de $\mathbf{C}$ y $\mathbf{K}$. La conclusión es que podemos realizar PCA usando la matriz de covarianzas $\mathbf{C}$ o la matriz de productos punto $\mathbf{K}$.

Kernel PCA, una versión no lineal de PCA

Podemos reescribir las expresiones para los vectores propios de $\mathbf{C}$ mediante los vectores propios de $\mathbf{K}$:

$$\mathbf{u}_j = \lambda_j^{-1/2} \sum_{i=1}^{n} (\mathbf{v}_j)_i \mathbf{x}_i = \sum_{i=1}^{n} \alpha_i^j \mathbf{x}_i, \quad j = 1, \ldots, t = \text{rango}(\mathbf{K}) = \text{rango}(\mathbf{C})$$

con el vector

$$\boldsymbol{\alpha}^j = \lambda_j^{-1/2} \mathbf{v}_j,$$

y la proyección

$$P_{\mathbf{u}_j}(\mathbf{x}) = \langle \mathbf{u}_j, \mathbf{x} \rangle = \left\langle \sum_{i=1}^{n} \alpha_i^j \mathbf{x}_i, \mathbf{x} \right\rangle = \sum_{i=1}^{n} \alpha_i^j \langle \mathbf{x}_i, \mathbf{x} \rangle.$$

Kernel PCA, una versión no lineal de PCA

Podemos reescribir las expresiones para los vectores propios de $\mathbf{C}$ mediante los vectores propios de $\mathbf{K}$:

$$\mathbf{u}_j = \lambda_j^{-1/2} \sum_{i=1}^{n} (\mathbf{v}_j)_i \mathbf{x}_i = \sum_{i=1}^{n} \alpha_i^j \mathbf{x}_i, \quad j = 1, \ldots, t = \text{rango}(\mathbf{K}) = \text{rango}(\mathbf{C})$$

con el vector

$$\boldsymbol{\alpha}^j = \lambda_j^{-1/2} \mathbf{v}_j,$$

y la proyección

$$P_{\mathbf{u}_j}(\mathbf{x}) = \langle \mathbf{u}_j, \mathbf{x} \rangle = \left\langle \sum_{i=1}^{n} \alpha_i^j \mathbf{x}_i, \mathbf{x} \right\rangle = \sum_{i=1}^{n} \alpha_i^j \langle \mathbf{x}_i, \mathbf{x} \rangle.$$

Kernel PCA, una versión no lineal de PCA

La expresión anterior también es válida para la transformación $\mathbf{x}$ en $\phi(\mathbf{x}) \in \mathcal{H}$, entonces

$$P_{\mathbf{u}_j}(\boldsymbol{\phi}(\mathbf{x})) = \sum_{i=1}^{n} \alpha_i^j \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}) \rangle = \sum_{i=1}^{n} \alpha_i^j k(\mathbf{x}_i, \mathbf{x}),$$

$$j = 1, 2, \ldots \text{rango}(\mathbf{K}) \leq \text{mín}(d, n)$$

Esto es lo que se conoce **Kernel PCA.**

En caso de que los datos transformados no están centrados, puede usarse la versión centrada de la matriz de Gram:

$$\tilde{\mathbf{K}} = \mathbf{K} - \frac{1}{n}\mathbf{JK} - \mathbf{K}\frac{1}{n}\mathbf{J} + \frac{1}{n^2}\mathbf{JKJ} = (\mathbf{I} - \frac{1}{n}\mathbf{J})\mathbf{K}(\mathbf{I} - \frac{1}{n}\mathbf{J}),$$

donde $\mathbf{J}$ es una matriz de $n \times n$ con todas las entradas igual a 1.

Kernel PCA, una versión no lineal de PCA

La expresión anterior también es válida para la transformación $\mathbf{x}$ en $\phi(\mathbf{x}) \in \mathcal{H}$, entonces

$$P_{\mathbf{u}_j}(\boldsymbol{\phi}(\mathbf{x})) = \sum_{i=1}^{n} \alpha_i^j \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}) \rangle = \sum_{i=1}^{n} \alpha_i^j k(\mathbf{x}_i, \mathbf{x}),$$

$$j = 1, 2, \ldots \text{rango}(\mathbf{K}) \leq \text{mín}(d, n)$$

Esto es lo que se conoce **Kernel PCA.**

En caso de que los datos transformados no están centrados, puede usarse la versión centrada de la matriz de Gram:

$$\tilde{\mathbf{K}} = \mathbf{K} - \frac{1}{n}\mathbf{J}\mathbf{K} - \mathbf{K}\frac{1}{n}\mathbf{J} + \frac{1}{n^2}\mathbf{J}\mathbf{K}\mathbf{J} = (\mathbf{I} - \frac{1}{n}\mathbf{J})\mathbf{K}(\mathbf{I} - \frac{1}{n}\mathbf{J}),$$

donde $\mathbf{J}$ es una matriz de $n \times n$ con todas las entradas igual a 1.

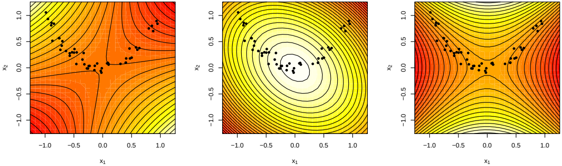
Kernel PCA, una versión no lineal de PCA

Algorithm 1 Kernel PCA

1: Obtener la matriz de Gram $\mathbf{K}$ según la definición 3.
2: Calcular la versión centrada $\tilde{\mathbf{K}}$ si es necesario.
3: Obtener la descomposición espectral $[\mathbf{\Lambda}, \mathbf{V}]$ de $\mathbf{K}$ (o $\tilde{\mathbf{K}}$).
4: Calcular los vectores propios normalizados $\boldsymbol{\alpha}^j$.
5: Calcular las proyecciones en los componentes principales $P_{\mathbf{u}_j}(\boldsymbol{\phi}(\mathbf{x}))$.

Kernel PCA, una versión no lineal de PCA

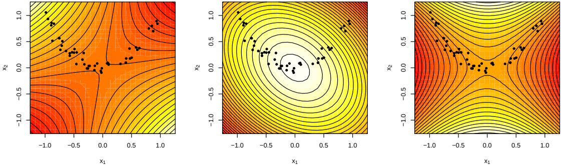
Ejemplo usando un kernel polinomial de grado 2:



Observemos que la solución está dada en términos del número de datos n, sin importar la dimensión del espacio de características (la transformación es implícita mediante algún kernel).

Kernel PCA, una versión no lineal de PCA

Ejemplo usando un kernel polinomial de grado 2:



Observemos que la solución está dada en términos del número de datos n, sin importar la dimensión del espacio de características (la transformación es implícita mediante algún kernel).

Algunas caracterizaciones de Kernel PCA

Kernel PCA como un problema de regularización

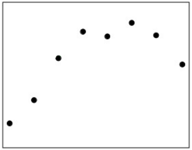
Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. *A generalized representer theorem*. COLT 01: Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory. 2001. Springer-Verlag.

Ejemplo:

Kernel PCA como un problema de regularización

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. *A generalized representer theorem*. COLT 01: Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory. 2001. Springer-Verlag.

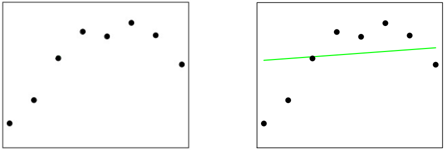
Ejemplo:



Kernel PCA como un problema de regularización

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. *A generalized representer theorem*. COLT 01: Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory. 2001. Springer-Verlag.

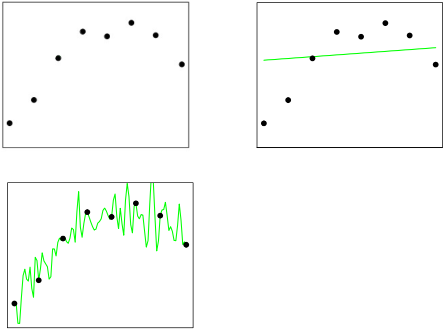
Ejemplo:



Kernel PCA como un problema de regularización

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. *A generalized representer theorem*. COLT 01: Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory. 2001. Springer-Verlag.

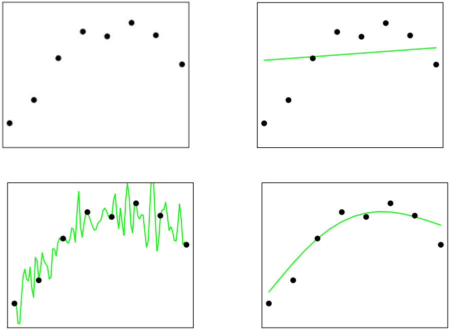
Ejemplo:



Kernel PCA como un problema de regularización

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. *A generalized representer theorem*. COLT 01: Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory. 2001. Springer-Verlag.

Ejemplo:



Kernel PCA como un problema de regularización

Consideremos funciones $f(\cdot)$ en un RKHS $\mathcal{H}$, con norma asociada $\|f\|_{\mathcal{H}}$.
El problema de componentes principales puede expresarse como:

$$\max_{\mathbf{u}} \frac{1}{n} \sum_{i} \left(f(\mathbf{x}_i) - \overline{f(\mathbf{x})} \right)^2, \quad \text{sujeto a} \quad \|f\|_{\mathcal{H}}^2 = 1.$$

Por ejemplo, si consideramos funciones lineales $f(\mathbf{x}) = \langle \mathbf{u}, \mathbf{x} \rangle$, con norma $\|f\|_{\mathcal{H}_L} = \|\mathbf{u}\|$ tenemos PCA ordinario.

Kernel PCA como un problema de regularización

El problema anterior puede escribirse como (Schölkopf, Herbrich, Smola. 2001):

$$\min \|f\|^2_{\mathcal{H}} \quad \text{sujeto a} \quad \frac{1}{n}\sum_i \left(f(\mathbf{x}_i) - \overline{f(\mathbf{x})}\right)^2 = 1,$$

y podemos definir entonces una función de costo:

$$\mathcal{C}\left(\{\mathbf{x}_i\}, \{y_i\}, \{f(\mathbf{x}_i)\}\right) = \begin{cases} 0 & \text{si } \frac{1}{n}\sum_i \left(f(\mathbf{x}_i) - \overline{f(\mathbf{x})}\right)^2 = 1 \\ \infty & \text{en caso contrario} \end{cases}$$

La función de costo **regularizada** a minimizar será entonces:

$$\mathcal{C}\left(\{\mathbf{x}_i\}, \{y_i\}, \{f(\mathbf{x}_i)\}\right) + \|f\|^2_{\mathcal{H}}.$$

Puede demostrarse (Teorema de Representación, Schölkopf y Smola) que la solución podrá representarse en la forma

$$f(\mathbf{x}) = \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}).$$

Kernel PCA como un problema de regularización

Ahora, para Kernel PCA, considera un kernel Gaussiano con $\mathcal{H}_G$ su RKHS correspondiente y norma $\|f\|_{\mathcal{H}_G}$.

- La norma $\|f\|_{\mathcal{H}_G}$ es menor mientras más suave es f (la demostración es muy técnica, la norma de éste kernel está relacionada con su transformada de Fourier).
- La suavidad está controlada por el parámetro σ del kernel.
- De aquí concluimos que

Kernel PCA busca funciones de proyección suaves y de máxima varianza

Kernel PCA es sensible a contrastes en densidades

¿Cómo interpretar el comportamiento de Kernel PCA con un kernel Gaussiano en el espacio original de los datos?

Considera la función de proyección del primer PC de Kernel PCA:

$$P_{\mathbf{u}_1}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i^1 k(\mathbf{x}_i, \mathbf{x}).$$

Definamos funciones de la forma

$$f_{\boldsymbol{\alpha}}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}),$$

para cualquier $\boldsymbol{\alpha}$.

Kernel PCA es sensible a contrastes en densidades

Consideremos el siguiente problema de optimización:

$$\max_{\boldsymbol{\alpha}} \sum_j \left(f_{\boldsymbol{\alpha}}(\mathbf{x}_j) - \frac{1}{n} \sum_t k(\mathbf{x}_j, \mathbf{x}_t) \right)^2$$

sujeto a $\|\boldsymbol{\alpha}\|^2 = 1, \ \sum_i \alpha_i = 0$.

Observa que el término en rojo es la estimación de densidad con un kernel Gaussiano (estimador de Parzen).

Puede mostrarse que la solución al problema anterior es $f^*_{\boldsymbol{\alpha}} = P_{\mathbf{u}_1}$, es decir, **el primer PC de Kernel PCA**. Entonces

Kernel PCA con un kernel Gaussiano busca contrastes en las densidades de los datos

Kernel PCA es sensible a contrastes en densidades

Consideremos el siguiente problema de optimización:

$$\max_{\boldsymbol{\alpha}} \sum_j \left(f_{\boldsymbol{\alpha}}(\mathbf{x}_j) - \frac{1}{n} \sum_t k(\mathbf{x}_j, \mathbf{x}_t) \right)^2$$

sujeto a $\|\boldsymbol{\alpha}\|^2 = 1, \ \sum_i \alpha_i = 0.$

Observa que el término en rojo es la estimación de densidad con un kernel Gaussiano (estimador de Parzen).
Puede mostrarse que la solución al problema anterior es $f^*_{\boldsymbol{\alpha}} = P_{\mathbf{u}_1}$, es decir, **el primer PC de Kernel PCA**.
Entonces

Kernel PCA con un kernel Gaussiano busca contrastes en las densidades de los datos

Kernel PCA es sensible a contrastes en densidades

Código

`notebooks/7-manifolds.ipynb`