Kernels (cont)

Fernando Lozano

Universidad de los Andes

October 19, 2022

Escogencia de un Kernel

Escogencia de un Kernel

- Kernel como medida de similaridad:

Escogencia de un Kernel

- Kernel como medida de similaridad:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$$

Escogencia de un Kernel

- Kernel como medida de similaridad:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$$

  - Producto punto en espacio de características.

Escogencia de un Kernel

- Kernel como medida de similaridad:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$$

  - Producto punto en espacio de características.
  - Coseno del ángulo entre $\phi(x_i)$ y $\phi(x_j)$.

Escogencia de un Kernel

- Kernel como medida de similaridad:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$$

  - Producto punto en espacio de características.
  - Coseno del ángulo entre $\phi(x_i)$ y $\phi(x_j)$.
  - Tamaño en espacio de características:

$$\|\phi(x)\|^2$$

Escogencia de un Kernel

- Kernel como medida de similaridad:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$$

  - Producto punto en espacio de características.
  - Coseno del ángulo entre $\phi(x_i)$ y $\phi(x_j)$.
  - Tamaño en espacio de características:

$$\|\phi(x)\|^2 = \langle \phi(x), \phi(x) \rangle_{\mathcal{H}}$$

Escogencia de un Kernel

- Kernel como medida de similaridad:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$$

  - Producto punto en espacio de características.
  - Coseno del ángulo entre $\phi(x_i)$ y $\phi(x_j)$.
  - Tamaño en espacio de características:

$$\|\phi(x)\|^2 = \langle \phi(x), \phi(x) \rangle_{\mathcal{H}} = k(x, x)$$

Escogencia de un Kernel

- Kernel como medida de similaridad:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$$

  - Producto punto en espacio de características.
  - Coseno del ángulo entre $\phi(x_i)$ y $\phi(x_j)$.
  - Tamaño en espacio de características:

$$\|\phi(x)\|^2 = \langle \phi(x), \phi(x) \rangle_{\mathcal{H}} = k(x, x)$$

  - Distancia en espacio de características:

$$\|\phi(x_i) - \phi(x_j)\|^2$$

Escogencia de un Kernel

- Kernel como medida de similaridad:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$$

  - Producto punto en espacio de características.
  - Coseno del ángulo entre $\phi(x_i)$ y $\phi(x_j)$.
  - Tamaño en espacio de características:

$$\|\phi(x)\|^2 = \langle \phi(x), \phi(x) \rangle_{\mathcal{H}} = k(x, x)$$

  - Distancia en espacio de características:

$$\|\phi(x_i) - \phi(x_j)\|^2 = \langle \phi(x_i) - \phi(x_j), \phi(x_i) - \phi(x_j) \rangle_{\mathcal{H}}$$

Escogencia de un Kernel

- Kernel como medida de similaridad:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$$

  - Producto punto en espacio de características.
  - Coseno del ángulo entre $\phi(x_i)$ y $\phi(x_j)$.
  - Tamaño en espacio de características:

$$\|\phi(x)\|^2 = \langle \phi(x), \phi(x) \rangle_{\mathcal{H}} = k(x, x)$$

  - Distancia en espacio de características:

$$\begin{aligned}\|\phi(x_i) - \phi(x_j)\|^2 &= \langle \phi(x_i) - \phi(x_j), \phi(x_i) - \phi(x_j) \rangle_{\mathcal{H}} \\ &= \langle \phi(x_i), \phi(x_i) \rangle_{\mathcal{H}} - 2 \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}} \\ &+ \langle \phi(x_j), \phi(x_j) \rangle_{\mathcal{H}}\end{aligned}$$

Escogencia de un Kernel

- Kernel como medida de similaridad:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$$

  - Producto punto en espacio de características.
  - Coseno del ángulo entre $\phi(x_i)$ y $\phi(x_j)$.
  - Tamaño en espacio de características:

$$\|\phi(x)\|^2 = \langle \phi(x), \phi(x) \rangle_{\mathcal{H}} = k(x, x)$$

  - Distancia en espacio de características:

$$\begin{aligned} \|\phi(x_i) - \phi(x_j)\|^2 &= \langle \phi(x_i) - \phi(x_j), \phi(x_i) - \phi(x_j) \rangle_{\mathcal{H}} \\ &= \langle \phi(x_i), \phi(x_i) \rangle_{\mathcal{H}} - 2 \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}} \\ &+ \langle \phi(x_j), \phi(x_j) \rangle_{\mathcal{H}} \\ &= k(x_i, x_i) - 2k(x_i, x_j) + k(x_j, x_j) \end{aligned}$$

Escogencia de un Kernel

- Kernel como medida de similaridad:

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$$

  - Producto punto en espacio de características.
  - Coseno del ángulo entre $\phi(x_i)$ y $\phi(x_j)$.
  - Tamaño en espacio de características:

$$\|\phi(x)\|^2 = \langle \phi(x), \phi(x) \rangle_{\mathcal{H}} = k(x, x)$$

  - Distancia en espacio de características:

$$\begin{aligned} \|\phi(x_i) - \phi(x_j)\|^2 &= \langle \phi(x_i) - \phi(x_j), \phi(x_i) - \phi(x_j) \rangle_{\mathcal{H}} \\ &= \langle \phi(x_i), \phi(x_i) \rangle_{\mathcal{H}} - 2 \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}} \\ &+ \langle \phi(x_j), \phi(x_j) \rangle_{\mathcal{H}} \\ &= k(x_i, x_i) - 2k(x_i, x_j) + k(x_j, x_j) \end{aligned}$$

- $\mathcal{X}$ no es necesariamente un espacio con producto punto.

- El kernel define una representación lineal de los datos, a través de un mapeo a un espacio de Hilbert.

- El kernel define una representación lineal de los datos, a través de un mapeo a un espacio de Hilbert.
- El kernel define el espacio de hipótesis con el cual se va a aprender.

- El kernel define una representación lineal de los datos, a través de un mapeo a un espacio de Hilbert.
- El kernel define el espacio de hipótesis con el cual se va a aprender.

$$f(x) = \sum_j \beta_j k(x_j, x)$$

- El kernel define una representación lineal de los datos, a través de un mapeo a un espacio de Hilbert.
- El kernel define el espacio de hipótesis con el cual se va a aprender.

$$f(x) = \sum_j \beta_j k(x_j, x)$$

  - Complejidad de la clase de hipótesis.

- El kernel define una representación lineal de los datos, a través de un mapeo a un espacio de Hilbert.
- El kernel define el espacio de hipótesis con el cual se va a aprender.

$$f(x) = \sum_j \beta_j k(x_j, x)$$

  - Complejidad de la clase de hipótesis.
- Otros puntos de vista: Regularización, Bayes...

Criterios

Criterios

- Conocimiento previo del problema y su solución.

Criterios

- Conocimiento previo del problema y su solución.
- Kernel es una manera directa de introducir conocimiento previo en el algoritmo de aprendizaje.

Criterios

- Conocimiento previo del problema y su solución.
- Kernel es una manera directa de introducir conocimiento previo en el algoritmo de aprendizaje.
- Medida de similaridad entre el tipo de datos específicos (en espacio de características).

Criterios

- Conocimiento previo del problema y su solución.
- Kernel es una manera directa de introducir conocimiento previo en el algoritmo de aprendizaje.
- Medida de similaridad entre el tipo de datos específicos (en espacio de características).
- Kernel positivo definido.

Criterios

- Conocimiento previo del problema y su solución.
- Kernel es una manera directa de introducir conocimiento previo en el algoritmo de aprendizaje.
- Medida de similaridad entre el tipo de datos específicos (en espacio de características).
- Kernel positivo definido.
- Parámetros del kernel + Constante de regularización = Selección de modelo.

Criterios

- Conocimiento previo del problema y su solución.
- Kernel es una manera directa de introducir conocimiento previo en el algoritmo de aprendizaje.
- Medida de similaridad entre el tipo de datos específicos (en espacio de características).
- Kernel positivo definido.
- Parámetros del kernel + Constante de regularización = Selección de modelo.
- Combinación de kernels.

Criterios

- Conocimiento previo del problema y su solución.
- Kernel es una manera directa de introducir conocimiento previo en el algoritmo de aprendizaje.
- Medida de similaridad entre el tipo de datos específicos (en espacio de características).
- Kernel positivo definido.
- Parámetros del kernel + Constante de regularización = Selección de modelo.
- Combinación de kernels.
- Aprendizaje del kernel (o combinación de kernels).

Kernels de rango uno

- Sea f una función real en $\mathcal{X}$, y $\mathrm{f} = \begin{bmatrix} f(x_1) & f(x_2) & \dots & f(x_n) \end{bmatrix}$

Kernels de rango uno

- Sea f una función real en $\mathcal{X}$, y $\mathrm{f} = \begin{bmatrix} f(x_1) & f(x_2) & \dots & f(x_n) \end{bmatrix}$
- La matrixz de Gramm:

$$\mathrm{K} = \mathrm{f}\,\mathrm{f}^T$$

Kernels de rango uno

- Sea f una función real en $\mathcal{X}$, y $\mathbf{f} = \begin{bmatrix} f(x_1) & f(x_2) & \dots & f(x_n) \end{bmatrix}$
- La matrixz de Gramm:

$$\mathsf{K} = \mathbf{f}\,\mathbf{f}^T$$

y

$$\boldsymbol{\alpha}^T \mathsf{K} \boldsymbol{\alpha}$$

Kernels de rango uno

- Sea f una función real en $\mathcal{X}$, y $\mathsf{f} = \begin{bmatrix} f(x_1) & f(x_2) & \dots & f(x_n) \end{bmatrix}$
- La matrixz de Gramm:

$$\mathsf{K} = \mathsf{f}\,\mathsf{f}^T$$

y

$$\boldsymbol{\alpha}^T \mathsf{K} \boldsymbol{\alpha} = \boldsymbol{\alpha}^T \mathsf{f}\,\mathsf{f}^T \boldsymbol{\alpha}$$

Kernels de rango uno

- Sea f una función real en $\mathcal{X}$, y $\mathsf{f} = \begin{bmatrix} f(x_1) & f(x_2) & \dots & f(x_n) \end{bmatrix}$
- La matrixz de Gramm:

$$\mathsf{K} = \mathsf{f}\,\mathsf{f}^T$$

y

$$\boldsymbol{\alpha}^T \mathsf{K} \boldsymbol{\alpha} = \boldsymbol{\alpha}^T \mathsf{f}\,\mathsf{f}^T \boldsymbol{\alpha} = (\mathsf{f}^T \boldsymbol{\alpha})^2$$

Kernels de rango uno

- Sea f una función real en $\mathcal{X}$, y $\mathrm{f} = \begin{bmatrix} f(x_1) & f(x_2) & \dots & f(x_n) \end{bmatrix}$
- La matrixz de Gramm:

$$\mathrm{K} = \mathrm{f}\,\mathrm{f}^T$$

y

$$\boldsymbol{\alpha}^T \mathrm{K} \boldsymbol{\alpha} = \boldsymbol{\alpha}^T \mathrm{f}\,\mathrm{f}^T \boldsymbol{\alpha} = (\mathrm{f}^T \boldsymbol{\alpha})^2 \geq 0$$

Kernels de rango uno

- Sea f una función real en $\mathcal{X}$, y $\mathsf{f} = \begin{bmatrix} f(x_1) & f(x_2) & \dots & f(x_n) \end{bmatrix}$
- La matrixz de Gramm:

$$\mathsf{K} = \mathsf{f}\,\mathsf{f}^T$$

y

$$\boldsymbol{\alpha}^T\mathsf{K}\boldsymbol{\alpha} = \boldsymbol{\alpha}^T\mathsf{f}\,\mathsf{f}^T\boldsymbol{\alpha} = (\mathsf{f}^T\boldsymbol{\alpha})^2 \geq 0$$

- Luego $k(x_i, x_j) = f(x_i)f(x_j)$ es un kernel.

Construcción de kernels

Construcción de kernels

Si $k_1, k_2, \ldots$ son kernels

Construcción de kernels

Si $k_1, k_2, \ldots$ son kernels

- $\alpha_1 k_1 + \alpha_2 k_2$ con $\alpha_1, \alpha_2 \geq 0$ es un kernel.

Construcción de kernels

Si $k_1, k_2, \ldots$ son kernels

- $\alpha_1 k_1 + \alpha_2 k_2$ con $\alpha_1, \alpha_2 \geq 0$ es un kernel.
- Si $k(x, x') = \lim_{n\to\infty} k_n(x, x')$ existe $\forall x, x'$, k es un kernel.

Construcción de kernels

Si $k_1, k_2, \ldots$ son kernels

- $\alpha_1 k_1 + \alpha_2 k_2$ con $\alpha_1, \alpha_2 \geq 0$ es un kernel.
- Si $k(x, x') = \lim_{n\to\infty} k_n(x, x')$ existe $\forall x, x'$, k es un kernel.
- $(k_1 k_2)(x, x') = k_1(x, x')k_2(x, x')$ es un kernel.

Construcción de kernels

Si $k_1, k_2, \ldots$ son kernels

- $\alpha_1 k_1 + \alpha_2 k_2$ con $\alpha_1, \alpha_2 \geq 0$ es un kernel.
- Si $k(x, x') = \lim_{n\to\infty} k_n(x, x')$ existe $\forall x, x'$, k es un kernel.
- $(k_1 k_2)(x, x') = k_1(x, x')k_2(x, x')$ es un kernel.
  - Por ejemplo, para una función real f podemos construir el kernel $k_f(x, x') = f(x)k(x, x')f(x')$

Construcción de kernels

Si $k_1, k_2, \ldots$ son kernels

- $\alpha_1 k_1 + \alpha_2 k_2$ con $\alpha_1, \alpha_2 \geq 0$ es un kernel.
- Si $k(x,x') = \lim_{n\to\infty} k_n(x,x')$ existe $\forall x, x'$, k es un kernel.
- $(k_1 k_2)(x,x') = k_1(x,x') k_2(x,x')$ es un kernel.
  - Por ejemplo, para una función real f podemos construir el kernel $k_f(x,x') = f(x)k(x,x')f(x')$
  - Esta transformación no afecta ángulos:

Construcción de kernels

Si $k_1, k_2, \ldots$ son kernels

- $\alpha_1 k_1 + \alpha_2 k_2$ con $\alpha_1, \alpha_2 \geq 0$ es un kernel.
- Si $k(x, x') = \lim_{n\to\infty} k_n(x, x')$ existe $\forall x, x'$, k es un kernel.
- $(k_1 k_2)(x, x') = k_1(x, x')k_2(x, x')$ es un kernel.
  - Por ejemplo, para una función real f podemos construir el kernel $k_f(x, x') = f(x)k(x, x')f(x')$
  - Esta transformación no afecta ángulos:

$$\cos(\angle(\phi_f(x), \phi_f(x'))) = \frac{f(x)k(x, x')f(x')}{\sqrt{f(x)k(x, x)f(x)}\sqrt{f(x')k(x', x')f(x')}}$$

Construcción de kernels

Si $k_1, k_2, \ldots$ son kernels

- $\alpha_1 k_1 + \alpha_2 k_2$ con $\alpha_1, \alpha_2 \geq 0$ es un kernel.
- Si $k(x,x') = \lim_{n\to\infty} k_n(x,x')$ existe $\forall x, x'$, k es un kernel.
- $(k_1 k_2)(x,x') = k_1(x,x')k_2(x,x')$ es un kernel.
  - Por ejemplo, para una función real f podemos construir el kernel $k_f(x,x') = f(x)k(x,x')f(x')$
  - Esta transformación no afecta ángulos:

$$\begin{aligned}\cos(\angle(\phi_f(x), \phi_f(x'))) &= \frac{f(x)k(x,x')f(x')}{\sqrt{f(x)k(x,x)f(x)}\sqrt{f(x')k(x',x')f(x')}} \\ &= \frac{k(x,x')}{\sqrt{k(x,x)}\sqrt{k(x',x')}}\end{aligned}$$

Construcción de kernels

Si $k_1, k_2, \ldots$ son kernels

- $\alpha_1 k_1 + \alpha_2 k_2$ con $\alpha_1, \alpha_2 \geq 0$ es un kernel.
- Si $k(x, x') = \lim_{n\to\infty} k_n(x, x')$ existe $\forall x, x'$, k es un kernel.
- $(k_1 k_2)(x, x') = k_1(x, x')k_2(x, x')$ es un kernel.
  - Por ejemplo, para una función real f podemos construir el kernel $k_f(x, x') = f(x)k(x, x')f(x')$
  - Esta transformación no afecta ángulos:

$$\begin{aligned}
\cos(\angle(\phi_f(x), \phi_f(x')) &= \frac{f(x)k(x, x')f(x')}{\sqrt{f(x)k(x, x)f(x)}\sqrt{f(x')k(x', x')f(x')}} \\
&= \frac{k(x, x')}{\sqrt{k(x, x)}\sqrt{k(x', x')}} \\
&= \cos(\angle(\phi(x), \phi(x'))
\end{aligned}$$

Construcción de kernels

Si $k_1, k_2, \ldots$ son kernels

- $\alpha_1 k_1 + \alpha_2 k_2$ con $\alpha_1, \alpha_2 \geq 0$ es un kernel.
- Si $k(x, x') = \lim_{n\to\infty} k_n(x, x')$ existe $\forall x, x'$, k es un kernel.
- $(k_1 k_2)(x, x') = k_1(x, x')k_2(x, x')$ es un kernel.
  - Por ejemplo, para una función real f podemos construir el kernel $k_f(x, x') = f(x)k(x, x')f(x')$
  - Esta transformación no afecta ángulos:

$$
\begin{aligned}
\cos(\angle(\phi_f(x), \phi_f(x'))) &= \frac{f(x)k(x, x')f(x')}{\sqrt{f(x)k(x, x)f(x)}\sqrt{f(x')k(x', x')f(x')}} \\
&= \frac{k(x, x')}{\sqrt{k(x, x)}\sqrt{k(x', x')}} \\
&= \cos(\angle(\phi(x), \phi(x')))
\end{aligned}
$$

- $(k_1 \otimes k_2(x_1, x_2, x_1', x_2') = k_1(x_1, x_2)k_2(x_1', x_2')$ es un kernel.

- $\psi(t) = \sum_{n=0}^{\infty} a_n t^n$, real, entera y $a_n \geq 0 \Rightarrow \psi\left(\langle x, x' \rangle_{\mathcal{H}}\right)$ es positivo definido para cualquier espacio de Hilbert $\mathcal{H}$.

- $\psi(t) = \sum_{n=0}^{\infty} a_n t^n$, real, entera y $a_n \geq 0 \Rightarrow \psi\left(\langle x, x' \rangle_{\mathcal{H}}\right)$ es positivo definido para cualquier espacio de Hilbert $\mathcal{H}$.
  - $k(x, x') = \langle x, x' \rangle^d$ es un kernel.

- $\psi(t) = \sum_{n=0}^{\infty} a_n t^n$, real, entera y $a_n \geq 0 \Rightarrow \psi\left(\langle x, x' \rangle_{\mathcal{H}}\right)$ es positivo definido para cualquier espacio de Hilbert $\mathcal{H}$.
  - $k(x, x') = \langle x, x' \rangle^d$ es un kernel.
  - $k(x, x') = e^{\langle x, x' \rangle / \sigma^2}$ es un kernel.

- $\psi(t) = \sum_{n=0}^{\infty} a_n t^n$, real, entera y $a_n \geq 0 \Rightarrow \psi\left(\langle x, x' \rangle_{\mathcal{H}}\right)$ es positivo definido para cualquier espacio de Hilbert $\mathcal{H}$.
  - $k(x, x') = \langle x, x' \rangle^d$ es un kernel.
  - $k(x, x') = e^{\langle x, x' \rangle / \sigma^2}$ es un kernel.
  - $$k(x, x') = e^{-\|x\|^2/2\sigma^2} e^{\langle x, x' \rangle / \sigma^2} e^{-\|x'\|^2/2\sigma^2}$$

- $\psi(t) = \sum_{n=0}^{\infty} a_n t^n$, real, entera y $a_n \geq 0 \Rightarrow \psi\left(\langle x, x' \rangle_{\mathcal{H}}\right)$ es positivo definido para cualquier espacio de Hilbert $\mathcal{H}$.
  - $k(x, x') = \langle x, x' \rangle^d$ es un kernel.
  - $k(x, x') = e^{\left\langle x, x' \right\rangle / \sigma^2}$ es un kernel.
  - $$k(x, x') = e^{-\|x\|^2/2\sigma^2} e^{\left\langle x, x' \right\rangle / \sigma^2} e^{-\|x'\|^2/2\sigma^2} = e^{-\|x-x'\|^2/(2\sigma^2)}$$

    es un kernel.

- $\psi(t) = \sum_{n=0}^{\infty} a_n t^n$, real, entera y $a_n \geq 0 \Rightarrow \psi\left(\langle x, x' \rangle_{\mathcal{H}}\right)$ es positivo definido para cualquier espacio de Hilbert $\mathcal{H}$.
    - $k(x, x') = \langle x, x' \rangle^d$ es un kernel.
    - $k(x, x') = e^{\langle x, x' \rangle / \sigma^2}$ es un kernel.
    - $$k(x, x') = e^{-\|x\|^2/2\sigma^2} e^{\langle x, x' \rangle / \sigma^2} e^{-\|x'\|^2/2\sigma^2} = e^{-\|x - x'\|^2/(2\sigma^2)}$$

      es un kernel.

Suma Directa

- k_1, k_2 kernels con

$$
\begin{aligned}
k_1 &: \mathcal{X}_1 \times \mathcal{X}_1 \to \mathbb{R} \\
k_2 &: \mathcal{X}_2 \times \mathcal{X}_2 \to \mathbb{R}
\end{aligned}
$$

Suma Directa

- k_1, k_2 kernels con

$$\begin{aligned} k_1 &: \mathcal{X}_1 \times \mathcal{X}_1 \to \mathbb{R} \\ k_2 &: \mathcal{X}_2 \times \mathcal{X}_2 \to \mathbb{R} \end{aligned}$$

- La suma directa $(x_1, x_1' \in \mathcal{X}_1,\ x_2, x_2' \in \mathcal{X}_2)$:

$$(k_1 \oplus k_2)(x_1, x_2, x_1', x_2') = k_1(x_1, x_1') + k_2(x_2, x_2')$$

Suma Directa

- k_1, k_2 kernels con

$$\begin{aligned} k_1 &: \mathcal{X}_1 \times \mathcal{X}_1 \to \mathbb{R} \\ k_2 &: \mathcal{X}_2 \times \mathcal{X}_2 \to \mathbb{R} \end{aligned}$$

- La suma directa ($x_1, x_1' \in \mathcal{X}_1$, $x_2, x_2' \in \mathcal{X}_2$):

$$(k_1 \oplus k_2)(x_1, x_2, x_1', x_2') = k_1(x_1, x_1') + k_2(x_2, x_2')$$

  es un kernel en $(\mathcal{X}_1 \times \mathcal{X}_2) \times (\mathcal{X}_1 \times \mathcal{X}_2)$.

Suma Directa

- k_1, k_2 kernels con

$$\begin{aligned} k_1 &: \mathcal{X}_1 \times \mathcal{X}_1 \to \mathbb{R} \\ k_2 &: \mathcal{X}_2 \times \mathcal{X}_2 \to \mathbb{R} \end{aligned}$$

- La suma directa ($x_1, x_1' \in \mathcal{X}_1$, $x_2, x_2' \in \mathcal{X}_2$):

$$(k_1 \oplus k_2)(x_1, x_2, x_1', x_2') = k_1(x_1, x_1') + k_2(x_2, x_2')$$

  es un kernel en $(\mathcal{X}_1 \times \mathcal{X}_2) \times (\mathcal{X}_1 \times \mathcal{X}_2)$.
- Util cuando la entrada consta de partes diferentes.

Kernels Convolucionales

- Suponga que $x \in \mathcal{X}$ está compuesto por partes $x_d \in \mathcal{X}_d$, $d = 1, \ldots, D$.

Kernels Convolucionales

- Suponga que $x \in \mathcal{X}$ está compuesto por partes $x_d \in \mathcal{X}_d$, $d = 1, \ldots, D$.
- Por ejemplo (D=2) :

$$ABC$$

Kernels Convolucionales

- Suponga que $x \in \mathcal{X}$ está compuesto por partes $x_d \in \mathcal{X}_d$, $d = 1, \ldots, D$.
- Por ejemplo (D=2) :

$$ABC = \color{red}{AB}\color{black}{C} = A\color{red}{BC}$$

Kernels Convolucionales

- Suponga que $x \in \mathcal{X}$ está compuesto por partes $x_d \in \mathcal{X}_d$, $d = 1, \ldots, D$.
- Por ejemplo (D=2) :

$$ABC = \color{red}{AB}\color{black}{C} = A\color{red}{BC}$$

- Conjunto de posibles descomposiciones: $R(x_1, \ldots, x_d, x)$.

Kernels Convolucionales

- Suponga que $x \in \mathcal{X}$ está compuesto por partes $x_d \in \mathcal{X}_d$, $d = 1, \ldots, D$.
- Por ejemplo (D=2) :

$$ABC = ABC = ABC$$

- Conjunto de posibles descomposiciones: $R(x_1, \ldots, x_d, x)$.
- Kernel para cada parte: $k_d \; : \; \mathcal{X}_d \times \mathcal{X}_d \to \mathbb{R}$.

Kernels Convolucionales

- Suponga que $x \in \mathcal{X}$ está compuesto por partes $x_d \in \mathcal{X}_d$, $d = 1, \ldots, D$.
- Por ejemplo (D=2) :

$$ABC = \textcolor{red}{AB}C = A\textcolor{red}{BC}$$

- Conjunto de posibles descomposiciones: $R(x_1, \ldots, x_d, x)$.
- Kernel para cada parte: $k_d \ : \ \mathcal{X}_d \times \mathcal{X}_d \to \mathbb{R}$.
- Kernel de convolución:

$$(k_1 \star \cdots \star k_D)(x, x') = \sum_R \prod_{d=1}^{D} k_d(x_d, x'_d)$$

- Estructuras discretas como strings, grafos, árboles...

String Kernels

- Strings: palabras, texto, secuencias de DNA.

String Kernels

- Strings: palabras, texto, secuencias de DNA.
- Por ejemplo en clasificación de texto se modela la entrada como una bolsa de palabras.

String Kernels

- Strings: palabras, texto, secuencias de DNA.
- Por ejemplo en clasificación de texto se modela la entrada como una bolsa de palabras.
- Bolsa de palabras: vector en el que cada posición cuenta la frecuencia de una palabra en el texto.

String Kernels

- Strings: palabras, texto, secuencias de DNA.
- Por ejemplo en clasificación de texto se modela la entrada como una bolsa de palabras.
- Bolsa de palabras: vector en el que cada posición cuenta la frecuencia de una palabra en el texto.
- La bolsa de palabras no tiene en cuenta la estructura del texto.

String Kernels

- Strings: palabras, texto, secuencias de DNA.
- Por ejemplo en clasificación de texto se modela la entrada como una bolsa de palabras.
- Bolsa de palabras: vector en el que cada posición cuenta la frecuencia de una palabra en el texto.
- La bolsa de palabras no tiene en cuenta la estructura del texto.
- String Kernel:
  - Medida de similaridad entre dos strings basada en el número de subcadenas comunes entre ellas.

String Kernels

- Strings: palabras, texto, secuencias de DNA.
- Por ejemplo en clasificación de texto se modela la entrada como una bolsa de palabras.
- Bolsa de palabras: vector en el que cada posición cuenta la frecuencia de una palabra en el texto.
- La bolsa de palabras no tiene en cuenta la estructura del texto.
- String Kernel:
  - Medida de similaridad entre dos strings basada en el número de subcadenas comunes entre ellas.
  - Menos peso a subcadenas que no son contiguas.

String Kernels

- Strings: palabras, texto, secuencias de DNA.
- Por ejemplo en clasificación de texto se modela la entrada como una bolsa de palabras.
- Bolsa de palabras: vector en el que cada posición cuenta la frecuencia de una palabra en el texto.
- La bolsa de palabras no tiene en cuenta la estructura del texto.
- String Kernel:
  - Medida de similaridad entre dos strings basada en el número de subcadenas comunes entre ellas.
  - Menos peso a subcadenas que no son contiguas.

Ejemplo

- Queremos comparar cat y cart.

Ejemplo

- Queremos comparar cat y cart.
- Subcadenas y longitud total:

	cat	cart
c	1	1
a	1	1
t	1	1
ca	2	2
at	2	3
ct	3	4
cat	3	4

Ejemplo

- Queremos comparar cat y cart.
- Subcadenas y longitud total:

	cat	cart
c	1	1
a	1	1
t	1	1
ca	2	2
at	2	3
ct	3	4
cat	3	4

- Con el factor de descuento λ se calcula el kernel:

$$k(\text{cat}, \text{cart}) = 2\lambda^7 + \lambda^5 + \lambda^4 + 3\lambda^2$$

En general

- Σ^nes el conjunto de strings de n símbolos sobre el alfabeto Σ, y Σ^* el conjunto de todos los strings en el alfabeto.

En general

- Σ^nes el conjunto de strings de n símbolos sobre el alfabeto Σ, y Σ^* el conjunto de todos los strings en el alfabeto.
- String s, subcadena s(i), con longitud total l(i).

En general

- Σ^nes el conjunto de strings de n símbolos sobre el alfabeto Σ, y Σ^* el conjunto de todos los strings en el alfabeto.
- String s, subcadena $s(\mathrm{i})$, con longitud total $l(\mathrm{i})$.
- Definimos mapeo ϕ con coordenadas ϕ_u ($u \in \Sigma^n$):

$$\phi_u(s) = \sum_{i\,:\,u=s(\mathrm{i})} \lambda^{l(\mathrm{i})}$$

En general

- Σ^nes el conjunto de strings de n símbolos sobre el alfabeto Σ, y Σ^* el conjunto de todos los strings en el alfabeto.
- String s, subcadena $s(\mathrm{i})$, con longitud total $l(\mathrm{i})$.
- Definimos mapeo ϕ con coordenadas ϕ_u ($u \in \Sigma^n$):

$$\phi_u(s) = \sum_{i\,:\,u=s(\mathrm{i})} \lambda^{l(\mathrm{i})}$$

- El kernel entre dos strings $s, t \in \Sigma^*$:

$$k_n(s,t) = \sum_{u \in \Sigma^n} \phi_u(s)\phi_u(t)$$

En general

- Σ^nes el conjunto de strings de n símbolos sobre el alfabeto Σ, y Σ^* el conjunto de todos los strings en el alfabeto.
- String s, subcadena $s(\mathrm{i})$, con longitud total $l(\mathrm{i})$.
- Definimos mapeo ϕ con coordenadas ϕ_u $(u \in \Sigma^n)$:

$$\phi_u(s) = \sum_{i\,:\,u=s(\mathrm{i})} \lambda^{l(\mathrm{i})}$$

- El kernel entre dos strings $s, t \in \Sigma^*$:

$$k_n(s,t) = \sum_{u\in\Sigma^n} \phi_u(s)\phi_u(t) = \sum_{u\in\Sigma^n} \sum_{i\,:\,u=s(\mathrm{i})} \sum_{j\,:\,u=t(\mathrm{j})} \lambda^{l(\mathrm{i})+l(\mathrm{j})}$$

En general

- Σ^nes el conjunto de strings de n símbolos sobre el alfabeto Σ, y Σ^* el conjunto de todos los strings en el alfabeto.
- String s, subcadena $s(\mathrm{i})$, con longitud total $l(\mathrm{i})$.
- Definimos mapeo ϕ con coordenadas ϕ_u $(u \in \Sigma^n)$:

$$\phi_u(s) = \sum_{i\,:\,u=s(\mathrm{i})} \lambda^{l(\mathrm{i})}$$

- El kernel entre dos strings $s, t \in \Sigma^*$:

$$k_n(s,t) = \sum_{u\in\Sigma^n} \phi_u(s)\phi_u(t) = \sum_{u\in\Sigma^n} \sum_{i\,:\,u=s(\mathrm{i})} \sum_{j\,:\,u=t(\mathrm{j})} \lambda^{l(\mathrm{i})+l(\mathrm{j})}$$

- Algoritmo recursivo calcula el kernel en $O\left(n|s||t|\right)$ (Lodhi et. al. 2001)

Kernels en Grafos

- Datos representados en un grafo $G = (V, E, w)$

Kernels en Grafos

- Datos representados en un grafo $G = (V, E, w)$
  - $V = \{v_1, \ldots, v_n\}$ vertices (objetos de interés).

Kernels en Grafos

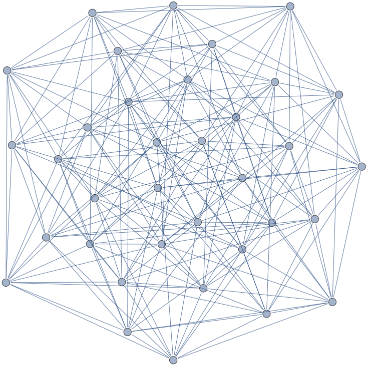
- Datos representados en un grafo $G = (V, E, w)$
  - $V = \{v_1, \ldots, v_n\}$ vertices (objetos de interés).
  - $E \subseteq V \times V$ arcos, relaciones entre objetos.

Kernels en Grafos

- Datos representados en un grafo $G = (V, E, w)$
  - $V = \{v_1, \ldots, v_n\}$ vertices (objetos de interés).
  - $E \subseteq V \times V$ arcos, relaciones entre objetos.
  - $w : E \to \mathbb{R}^+$ función de pesos.

Kernels en Grafos

- Datos representados en un grafo $G = (V, E, w)$
  - $V = \{v_1, \ldots, v_n\}$ vertices (objetos de interés).
  - $E \subseteq V \times V$ arcos, relaciones entre objetos.
  - $w : E \to \mathbb{R}^+$ función de pesos.



- Kernel: Similaridad entre vértices en el grafo.

- Kernel: Similaridad entre vértices en el grafo.
- Si $Q \in \mathbb{R}^{n \times n}$ es una matriz positiva (semi) definida, el producto punto con respecto a Q define un RKHS en el espacio de columnas de Q^{-1}.

- Kernel: Similaridad entre vértices en el grafo.
- Si $Q \in \mathbb{R}^{n \times n}$ es una matriz positiva (semi) definida, el producto punto con respecto a Q define un RKHS en el espacio de columnas de Q^{-1}.
  - El Laplaciano $L = D - W$

- Kernel: Similaridad entre vértices en el grafo.
- Si $Q \in \mathbb{R}^{n \times n}$ es una matriz positiva (semi) definida, el producto punto con respecto a Q define un RKHS en el espacio de columnas de Q^{-1}.
    - El Laplaciano $L = D - W$
    - El Laplaciano normalizado: $\tilde{L} = D^{-1/2} L D^{1/2} = I - D^{-1/2} W D^{1/2}$

- Kernel: Similaridad entre vértices en el grafo.
- Si $Q \in \mathbb{R}^{n \times n}$ es una matriz positiva (semi) definida, el producto punto con respecto a Q define un RKHS en el espacio de columnas de Q^{-1}.
  - El Laplaciano $L = D - W$
  - El Laplaciano normalizado: $\tilde{L} = D^{-1/2} L D^{1/2} = I - D^{-1/2} W D^{1/2}$
  - El kernel de difusión:

$$\begin{aligned} K &= e^{-\beta L} \\ &= \lim_{n \to \infty} \left(1 - \frac{\beta L}{n}\right)^n \end{aligned}$$

- Kernel: Similaridad entre vértices en el grafo.
- Si $Q \in \mathbb{R}^{n \times n}$ es una matriz positiva (semi) definida, el producto punto con respecto a Q define un RKHS en el espacio de columnas de Q^{-1}.
  - El Laplaciano $L = D - W$
  - El Laplaciano normalizado: $\tilde{L} = D^{-1/2} L D^{1/2} = I - D^{-1/2} W D^{1/2}$
  - El kernel de difusión:

$$
\begin{aligned}
K &= e^{-\beta L} \\
&= \lim_{n \to \infty} \left(1 - \frac{\beta L}{n}\right)^n \\
&= I - \beta L + \frac{\beta^2}{2!} L^2 - \frac{\beta^3}{3!} L^3 + \ldots
\end{aligned}
$$

Kernels en procesamiento de imágenes

Kernels en procesamiento de imágenes

- Kernel polinomial $k(x, x') = \langle x, x' \rangle^d$:

Kernels en procesamiento de imágenes

- Kernel polinomial $k(x, x') = \langle x, x' \rangle^d$: productos de d pixels.

Kernels en procesamiento de imágenes

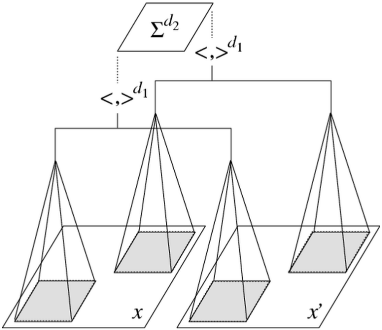
- Kernel polinomial $k(x, x') = \langle x, x' \rangle^d$: productos de d pixels.
- Usar todos los productos no es deseable:

Kernels en procesamiento de imágenes

- Kernel polinomial $k(x, x') = \langle x, x' \rangle^d$: productos de d pixels.
- Usar todos los productos no es deseable: correlaciones entre pixeles cercanos en la imagen son más importantes.

Kernels en procesamiento de imágenes

- Kernel polinomial $k(x, x') = \langle x, x' \rangle^d$: productos de d pixels.
- Usar todos los productos no es deseable: correlaciones entre pixeles cercanos en la imagen son más importantes.
- Campo receptivo piramidal:



1. Calcular nueva imagen $(x. * x')$.

1. Calcular nueva imagen $(x. * x')$.
2. $$z_{ij} = \sum_{i'j'} w(\max(|i-i'|, |j-j'|)(x. * x')_{i'j'}$$

1. Calcular nueva imagen $(x.*x')$.
2. $$z_{ij} = \sum_{i'j'} w(\max(|i-i'|,|j-j'|)(x.*x')_{i'j'}$$

   donde w es una función de peso.

1. Calcular nueva imagen $(x. * x')$.
2. $$z_{ij} = \sum_{i'j'} w(\max(|i - i'|, |j - j'|)(x. * x')_{i'j'}$$

   donde w es una función de peso. Por ejemplo $w(n) = \max(q - n, 0)$ da un campo receptivo de ancho $2q + 1$.

1. Calcular nueva imagen $(x. * x')$.
2. $$z_{ij} = \sum_{i'j'} w(\max(|i-i'|, |j-j'|)(x. * x')_{i'j'}$$

   donde w es una función de peso. Por ejemplo $w(n) = \max(q-n, 0)$ da un campo receptivo de ancho $2q+1$.
3. $z_{ij}^{d_1}$:

1. Calcular nueva imagen $(x. * x')$.
2. $$z_{ij} = \sum_{i'j'} w(\max(|i - i'|, |j - j'|)(x. * x')_{i'j'}$$
   donde w es una función de peso. Por ejemplo $w(n) = \max(q - n, 0)$ da un campo receptivo de ancho $2q + 1$.
3. $z_{ij}^{d_1}$:correlaciones locales de orden d_1.

1. Calcular nueva imagen $(x. * x')$.
2. $$z_{ij} = \sum_{i'j'} w(\max(|i - i'|, |j - j'|)(x. * x')_{i'j'}$$

   donde w es una función de peso. Por ejemplo $w(n) = \max(q - n, 0)$ da un campo receptivo de ancho $2q + 1$.
3. $z_{ij}^{d_1}$:correlaciones locales de orden d_1.
4. $\left(\sum_{ij} z_{ij}^{d_1}\right)^{d_2}$:

1. Calcular nueva imagen $(x.*x')$.
2. $$z_{ij} = \sum_{i'j'} w(\max(|i-i'|, |j-j'|)(x.*x')_{i'j'}$$

   donde w es una función de peso. Por ejemplo $w(n) = \max(q-n, 0)$ da un campo receptivo de ancho $2q+1$.
3. $z_{ij}^{d_1}$:correlaciones locales de orden d_1.
4. $\left(\sum_{ij} z_{ij}^{d_1}\right)^{d_2}$: Correlaciones de orden d_2 entre pixeles lejanos.