Questions: We know (Representer Thm)

for hypothesis classes $\mathcal{H}$ that are RKHSes,

$$f^* = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \underline{\ell}(f(x_i), y_i) + \lambda \|f\|_{\mathcal{H}}^2$$

then

$$f^*(x) = \sum_{i=1}^{n} \underline{K}(x, x_i) \alpha_i$$

1) What are useful RKHSes / kernels / feature maps?

$\phi \Leftrightarrow K \Leftrightarrow$ RKHS $\mathcal{H}$

→ 2) How do we find $f^*$ efficiently? } }

How do we solve for $\alpha$?

# Examples of useful kernels

1) polynomial kernels of order $r$:

$$x \xrightarrow{\phi} \left[\begin{array}{l}\text{all monomial features}\\ \text{of order up to } r\\ \text{including } r\end{array}\right] \leftsquigarrow \sum_{i=1}^{r} \binom{d+r}{r} \text{ features}$$

ex: $r = 3$

$$x \xrightarrow{\phi} \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \\ x_1^2 \\ x_1 x_2 \\ \vdots \\ x_d^2 \\ x_1^3 \\ x_1^2 x_2 \\ x_1 x_2 x_3 \\ \vdots \\ x_d^3 \end{bmatrix}$$

corresponding kernel is

$$K(x, y) = \langle \phi(x), \phi(y) \rangle$$

$$= \left\langle \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d^3 \end{bmatrix}, \begin{bmatrix} 1 \\ y_1 \\ \vdots \\ y_d^3 \end{bmatrix} \right\rangle$$

very expensive to compute! $\Omega(d^3)$

In fact,

$$K(x, y) = (\langle x, y \rangle + 1)^3$$

takes $O(d)$

Ex: $r = 2 \quad d = 2$

Claim learning w/ $\phi = [1 \;\; x_1 \;\; x_2 \;\; x_1^2 \;\; x_1x_2 \;\; x_2^2]$

is the same as learning with $\tilde{\phi} = [1 \;\; x_1 \;\; x_2 \;\; x_1^2 \;\; \sqrt{2}x_1x_2 \;\; x_2^2]$

and note that

$$\underline{K(x,y)} = \langle \tilde{\phi}(x), \tilde{\phi}(y) \rangle = \left\langle \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}, \begin{bmatrix} 1 \\ y_1 \\ y_2 \\ y_1^2 \\ \sqrt{2}y_1y_2 \\ y_2^2 \end{bmatrix} \right\rangle$$

$$= 1 + x_1y_1 + x_2y_2 + x_1^2y_1^2 + 2x_1y_1x_2y_2 + x_2^2y_2^2$$

Claim is $K(x,y) = (\underline{\langle x, y \rangle} + 1)^2 = (x_1y_1 + x_2y_2 + 1)^2$

$= (x_1y_1 + x_2y_2 + 1)(x_1y_1 + x_2y_2 + 1)$

$$= (x_1y_1 + x_2y_2 + 1)(x_1y_1 + x_2y_2 + 1)$$

$$= x_1^2y_1^2 + x_2y_2x_1y_1 + x_1y_1$$

$$+ x_1y_1x_2y_2 + x_2^2y_2^2 + x_2y_2$$

$$+ x_1y_1 + x_2y_2 + 1$$

$$= 1 + x_1y_1 + x_2y_2 + x_1^2y_1^2 + x_2^2y_2^2$$

$$+ 2x_1y_1x_2y_2$$

$$= K(x, y)$$

Example of applications for polynomial kernels:

1) NLP applications w/ BOW features

2) Image processing: allow pixels in neighborhoods to interact

| | | |
|---|---|---|
| $x_7$ | $x_8$ | $x_9$ |
| $x_2$ | $x_1$ | $x_3$ |
| $x_4$ | $x_5$ | $x_6$ |

$$f(x) = \sum \cdots$$

$$+ \underbrace{x_1 x_2 x_3 x_4 \cdots x_9}$$

2) Gaussian kernel:

$$e^z = \sum_{n=0}^{\infty} \frac{1}{n!} z^n$$

$$x \xrightarrow[\phi]{} \mathbb{R}^{\infty}$$

Consider $d=1$ in this case $\phi_0(x) = 1$

$$\phi_n(x) = \frac{1}{\sqrt{n!}} e^{\frac{-x^2}{2}} x^n$$

then $K(x,y) = \langle \phi(x), \phi(y) \rangle$

$$= \sum_{n=0}^{\infty} \left[ \frac{1}{\sqrt{n!}} e^{-\frac{x^2}{2}} x^n \cdot \frac{1}{\sqrt{n!}} e^{-\frac{y^2}{2}} y^n \right]$$

$$= e^{-\frac{|x-y|^2}{2}}$$

$$= e^{\frac{-(x^2+y^2)}{2}} \sum_{n=0}^{\infty} \frac{1}{n!} (xy)^n$$

$$= e^{\left[-\frac{(x^2+y^2)}{2} + xy\right]}$$

when $d=1$, $K(x,y) = e^{-\frac{(x-y)^2}{2}}$

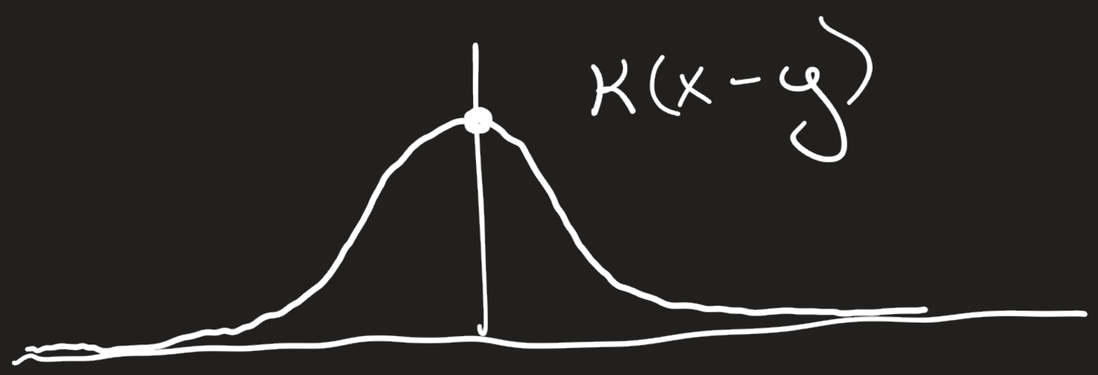
In general for $d>1$

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

where $\sigma$ is called the bandwidth of the kernel.

Note $K$ is <u>shift-invariant</u>:

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




$x-y$ $\left(\begin{matrix} d=1 \\ \sigma=1 \end{matrix}\right)$

In terms of the Gaussian kernel, reconsider regression model

$$f(x_{new}) = \sum_{i=1}^{n} K(x_{new}, x_i)\alpha_i$$

$$= \sum_{i=1}^{n} e^{-\frac{\|x_{new} - x_i\|_2^2}{2\sigma^2}} \alpha_i$$

## How to find $f^*$ ?

If $f^* = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2$

then

$$f^*(x) = \sum_i K(x, x_i) \alpha_i^*.$$

Furthermore

$$\alpha^* = \underset{\alpha \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell\left((K\alpha)_i, y_i\right) + \frac{\lambda}{2} \alpha^T K \alpha$$

($(K\alpha)_i = f^*(x_i)$; $\alpha^T K \alpha = \|f^*\|_{\mathcal{H}}^2$)

note this is a finite-dim problem

Intuition:

$$f^*(x_i) = \sum_{j=1}^{n} K(x_i, x_j)\alpha_j$$

$$= [K\alpha]_i$$

so we get $\alpha^*$ by plugging in $f^*$'s values on the $n$ training points, observing that

$$\|f^*\|_{\mathcal{H}}^2 = \alpha^{*T} K \alpha^*$$

and minimizing w.r.t. $\alpha$

## Examples

Kernel regression w/ $l_2$ loss

$$\alpha^* = \underset{\alpha}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} l((K\alpha)_i, y_i) + \frac{\lambda}{2} \alpha^T K \alpha$$

$$= \underset{\alpha}{\operatorname{argmin}} \frac{1}{n} \| K\alpha - y \|_2^2 + \frac{\lambda}{2} \alpha^T K \alpha$$

set $\nabla_\alpha (\text{objective}) = 0$ to see

$$\nabla_\alpha \left( \frac{1}{n} \alpha^{*T} K^T K \alpha^* + \frac{1}{n} y^T y - \frac{1}{n} 2 y^T K \alpha^* + \frac{\lambda}{2} \alpha^{*T} K \alpha^* \right) = 0$$

so at $\alpha^*$

$$\nabla_\alpha \left( \frac{1}{n} \alpha^{*T} K^T K \alpha^* + \frac{1}{n} y^T y - \frac{2}{n} y^T K \alpha^* + \frac{\lambda}{2} \alpha^{*T} K \alpha^* \right) = 0$$

$$\frac{2}{n} K^T K \alpha^* - \frac{2}{n} K^T y + \lambda K \alpha^* = 0$$

$$\Rightarrow \alpha^* = \left( K^T K + \frac{n\lambda}{2} K \right)^{-1} K^T y$$

Note $K = K^T$ because $K(x, y) = \langle \phi(x), \phi(y) \rangle$

$= \langle \phi(y), \phi(x) \rangle$

$= K(y, x)$

$$\Rightarrow \alpha^* = \left( K^2 + \frac{n\lambda K}{2} \right)^{-1} K y = \left( K + \frac{n\lambda I}{2} \right)^{-1} y$$

Recall from last class that solving RR w/ nonlinear features gives

$$f(x_{new}) = \sum_{i=1}^{n} K(x_{new}, x_i)\alpha_i$$

where $$\alpha = (K + n\lambda I)^{-1} y$$

Another example: kernelized logistic regression

Again: choose a kernel $K$ then solve

$$\alpha_* = \operatorname*{argmin}_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{y_i (K\alpha)_i}\right) + \frac{\lambda}{2} \alpha^T K \alpha$$