Adaptive Modelling of Complex Data: Kernels

Part 1: Kernels and feature space, ridge regression

Arthur Gretton, Dino Sejdinovic

Gatsby Unit, CSML, UCL

February 18, 2014

Course overview

Part 1:

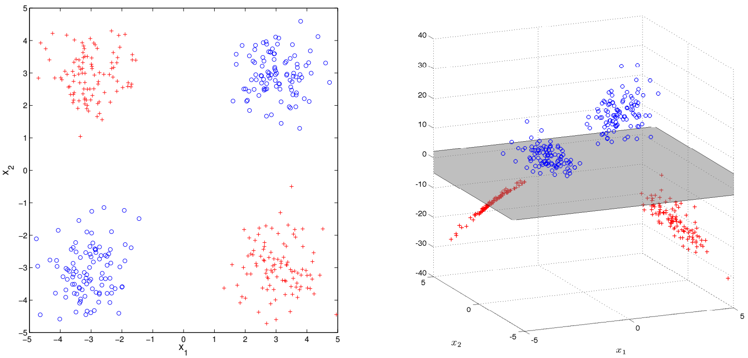
- What is a feature map, what is a kernel, and how do they relate?
- Applications: difference in means, kernel ridge regression

Part 2:

- Basics of convex optimization
- The support vector machine

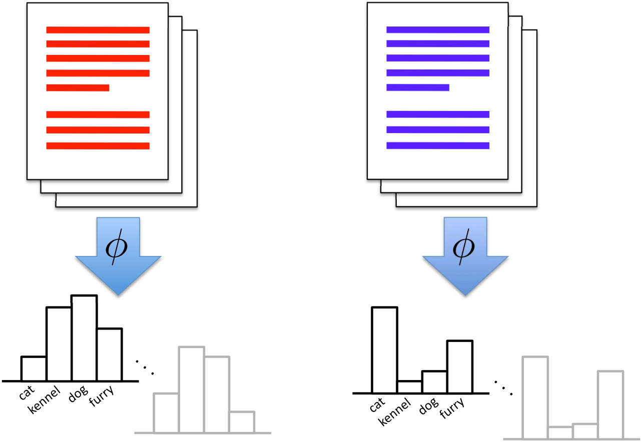
More detailed version of slides and lecture notes available at:
www.gatsby.ucl.ac.uk/~gretton/coursefiles/rkhscourse

Why kernel methods (1): XOR example



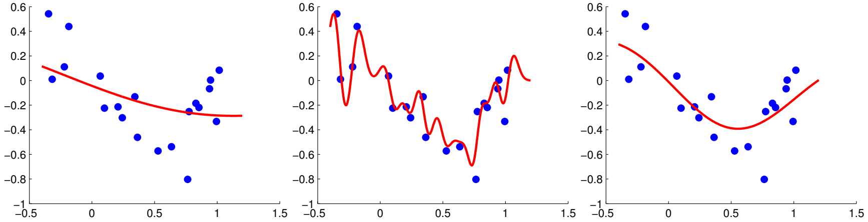
- No linear classifier separates red from blue
- Map points to **higher dimensional feature space**:
  $\phi(x) = \begin{bmatrix} x_1 & x_2 & x_1 x_2 \end{bmatrix} \in \mathbb{R}^3$

Why kernel methods (2): document classification



Kernels let us compare **objects** on the basis of **features**

Why kernel methods (3): smoothing



Kernel methods can control **smoothness** and **avoid overfitting/underfitting**.

Basics of reproducing kernel Hilbert spaces

Outline: reproducing kernel Hilbert space

We will describe in order:

1. Hilbert space (very simple)
2. Kernel (lots of examples: e.g. you can build kernels from simpler kernels)
3. Reproducing property

Hilbert space

Definition (Inner product)

Let $\mathcal{H}$ be a vector space over $\mathbb{R}$. A function $\langle \cdot, \cdot \rangle_{\mathcal{H}} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ is an inner product on $\mathcal{H}$ if

1. $\langle \alpha_1 f_1 + \alpha_2 f_2, g \rangle_{\mathcal{H}} = \alpha_1 \langle f_1, g \rangle_{\mathcal{H}} + \alpha_2 \langle f_2, g \rangle_{\mathcal{H}}$
2. $\langle f, g \rangle_{\mathcal{H}} = \langle g, f \rangle_{\mathcal{H}}$
3. $\langle f, f \rangle_{\mathcal{H}} \geq 0$ and $\langle f, f \rangle_{\mathcal{H}} = 0$ if and only if $f = 0$.

Norm induced by the inner product: $\|f\|_{\mathcal{H}} := \sqrt{\langle f, f \rangle_{\mathcal{H}}}$

Definition (Hilbert space)

"Well behaved" (complete) inner product space.

Hilbert space

Definition (Inner product)

Let $\mathcal{H}$ be a vector space over $\mathbb{R}$. A function $\langle \cdot, \cdot \rangle_{\mathcal{H}} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ is an inner product on $\mathcal{H}$ if

1. $\langle \alpha_1 f_1 + \alpha_2 f_2, g \rangle_{\mathcal{H}} = \alpha_1 \langle f_1, g \rangle_{\mathcal{H}} + \alpha_2 \langle f_2, g \rangle_{\mathcal{H}}$
2. $\langle f, g \rangle_{\mathcal{H}} = \langle g, f \rangle_{\mathcal{H}}$
3. $\langle f, f \rangle_{\mathcal{H}} \geq 0$ and $\langle f, f \rangle_{\mathcal{H}} = 0$ if and only if $f = 0$.

Norm induced by the inner product: $\|f\|_{\mathcal{H}} := \sqrt{\langle f, f \rangle_{\mathcal{H}}}$

Definition (Hilbert space)

"Well behaved" (complete) inner product space.

Hilbert space

Definition (Inner product)

Let $\mathcal{H}$ be a vector space over $\mathbb{R}$. A function $\langle\cdot,\cdot\rangle_{\mathcal{H}} : \mathcal{H}\times\mathcal{H}\to\mathbb{R}$ is an inner product on $\mathcal{H}$ if

1. $\langle\alpha_1 f_1 + \alpha_2 f_2, g\rangle_{\mathcal{H}} = \alpha_1 \langle f_1, g\rangle_{\mathcal{H}} + \alpha_2 \langle f_2, g\rangle_{\mathcal{H}}$
2. $\langle f, g\rangle_{\mathcal{H}} = \langle g, f\rangle_{\mathcal{H}}$
3. $\langle f, f\rangle_{\mathcal{H}} \geq 0$ and $\langle f, f\rangle_{\mathcal{H}} = 0$ if and only if $f = 0$.

Norm induced by the inner product: $\|f\|_{\mathcal{H}} := \sqrt{\langle f, f\rangle_{\mathcal{H}}}$

Definition (Hilbert space)

"Well behaved" (complete) inner product space.

Kernel: inner product between feature maps

Definition

Let $\mathcal{X}$ be a non-empty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a **kernel** if there exists a Hilbert space and a map $\phi : \mathcal{X} \to \mathcal{H}$ such that $\forall x, x' \in \mathcal{X}$,

$$k(x, x') := \langle \phi(x), \phi(x') \rangle_{\mathcal{H}} .$$

- Almost no conditions on $\mathcal{X}$ (eg, $\mathcal{X}$ itself doesn't need an inner product, eg. documents).
- Think of kernel as similarity measure between features

What are some simple kernels? E.g for books? For images?

- A single kernel can correspond to multiple sets of underlying features.

$$\phi_1(x) = x \qquad \text{and} \qquad \phi_2(x) = \left[\; x/\sqrt{2} \quad x/\sqrt{2} \;\right]$$

Kernel: inner product between feature maps

Definition

Let $\mathcal{X}$ be a non-empty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a **kernel** if there exists a Hilbert space and a map $\phi : \mathcal{X} \to \mathcal{H}$ such that $\forall x, x' \in \mathcal{X}$,

$$k(x, x') := \langle \phi(x), \phi(x') \rangle_{\mathcal{H}} .$$

- Almost no conditions on $\mathcal{X}$ (eg, $\mathcal{X}$ itself doesn't need an inner product, eg. documents).
- Think of kernel as similarity measure between features

What are some simple kernels? E.g for books? For images?

- A single kernel can correspond to multiple sets of underlying features.

$$\phi_1(x) = x \qquad \text{and} \qquad \phi_2(x) = \left[\ x/\sqrt{2} \quad x/\sqrt{2}\ \right]$$

New kernels from old: sums, transformations

The great majority of useful kernels are built from simpler kernels.

Theorem (Sums of kernels are kernels)

Given $\alpha > 0$ and k, k_1 and k_2 all kernels on $\mathcal{X}$, then αk and $k_1 + k_2$ are kernels on $\mathcal{X}$.

To prove this, just check inner product definition. A difference of kernels may not be a kernel (**why?**)

Theorem (Mappings between spaces)

Let $\mathcal{X}$ and $\widetilde{\mathcal{X}}$ be sets, and define a map $A : \mathcal{X} \to \widetilde{\mathcal{X}}$. Define the kernel k on $\widetilde{\mathcal{X}}$. Then the kernel $k(A(x), A(x'))$ is a kernel on $\mathcal{X}$.

Example: $k(x, x') = x^2 \, (x')^2$.

New kernels from old: sums, transformations

The great majority of useful kernels are built from simpler kernels.

Theorem (Sums of kernels are kernels)

Given $\alpha > 0$ and k, k_1 and k_2 all kernels on $\mathcal{X}$, then αk and $k_1 + k_2$ are kernels on $\mathcal{X}$.

To prove this, just check inner product definition. A difference of kernels may not be a kernel (**why?**)

Theorem (Mappings between spaces)

Let $\mathcal{X}$ and $\widetilde{\mathcal{X}}$ be sets, and define a map $A : \mathcal{X} \to \widetilde{\mathcal{X}}$. Define the kernel k on $\widetilde{\mathcal{X}}$. Then the kernel $k(A(x), A(x'))$ is a kernel on $\mathcal{X}$.

Example: $k(x, x') = x^2 (x')^2$.

New kernels from old: sums, transformations

The great majority of useful kernels are built from simpler kernels.

Theorem (Sums of kernels are kernels)

Given $\alpha > 0$ and k, k_1 and k_2 all kernels on $\mathcal{X}$, then αk and $k_1 + k_2$ are kernels on $\mathcal{X}$.

To prove this, just check inner product definition. A difference of kernels may not be a kernel (**why?**)

Theorem (Mappings between spaces)

Let $\mathcal{X}$ and $\widetilde{\mathcal{X}}$ be sets, and define a map $A : \mathcal{X} \to \widetilde{\mathcal{X}}$. Define the kernel k on $\widetilde{\mathcal{X}}$. Then the kernel $k(A(x), A(x'))$ is a kernel on $\mathcal{X}$.

Example: $k(x, x') = x^2 \, (x')^2$.

New kernels from old: products

Theorem (Products of kernels are kernels)

Given k_1 on $\mathcal{X}_1$ and k_2 on $\mathcal{X}_2$, then $k_1 \times k_2$ is a kernel on $\mathcal{X}_1 \times \mathcal{X}_2$. If $\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{X}$, then $k := k_1 \times k_2$ is a kernel on $\mathcal{X}$.

Proof.

Main idea only! $\mathcal{H}_1$ corresponding to k_1 is $\mathbb{R}^m$, and $\mathcal{H}_2$ corresponding to k_2 is $\mathbb{R}^n$. Define:

- $k_1 := u^\top v$ for $u, v \in \mathbb{R}^m$ (e.g.: kernel between two images)
- $k_2 := p^\top q$ for $p, q \in \mathbb{R}^n$ (e.g.: kernel between two captions)

Is the following a kernel?

$$K\left[(u,p);(v,q)\right] = k_1 \times k_2$$

(e.g. kernel between one image-caption pair and another)

New kernels from old: products

Proof.

(continued)

$$\begin{aligned} k_1 k_2 &= \left(u^\top v\right)\left(q^\top p\right) \\ &= \mathrm{trace}(u^\top v q^\top p) \\ &= \mathrm{trace}(p u^\top v q^\top) \\ &= \langle A, B \rangle, \end{aligned}$$

where $A := pu^\top$, $B := qv^\top$ (features of image-caption pairs)
Thus $k_1 k_2$ is a valid kernel, since inner product between $A, B \in \mathbb{R}^{m \times n}$ is

$$\langle A, B \rangle = \mathrm{trace}(AB^\top). \quad (1)$$

□

Sums and products $\implies$ polynomials

Theorem (Polynomial kernels)

Let $x, x' \in \mathbb{R}^d$ for $d \geq 1$, and let $m \geq 1$ be an integer and $c \geq 0$ be a positive real. Then

$$k(x, x') := \left(\langle x, x' \rangle + c\right)^m$$

is a valid kernel.

To prove: expand into a sum (with non-negative scalars) of kernels $\langle x, x' \rangle$ raised to integer powers. These individual terms are valid kernels by the product rule.

Infinite sequences

The kernels we've seen so far are dot products between finitely many features. E.g.

$$k(x, y) = \begin{bmatrix} \sin(x) & x^3 & \log x \end{bmatrix}^\top \begin{bmatrix} \sin(y) & y^3 & \log y \end{bmatrix}$$

where $\phi(x) = \begin{bmatrix} \sin(x) & x^3 & \log x \end{bmatrix}$

Can a kernel be a dot product between infinitely many features?

Infinite sequences

Definition

The space ℓ_p of p-summable sequences is defined as all sequences $(a_i)_{i\geq 1}$ for which

$$\sum_{i=1}^{\infty} a_i^p < \infty.$$

Kernels can be defined in terms of sequences in ℓ_2.

Theorem

Given sequence of functions $(f_i(x))_{i\geq 1}$ in ℓ_2 where $f_i : \mathcal{X} \to \mathbb{R}$. Then

$$k(x, x') := \sum_{i=1}^{\infty} f_i(x) f_i(x') \qquad (2)$$

is a kernel on $\mathcal{X}$.

Taylor series kernels (infinite polynomials)

Definition (Taylor series kernel)

For $r \in (0, \infty]$, with $a_n \geq 0$ for all $n \geq 0$

$$f(z) = \sum_{n=0}^{\infty} a_n z^n \qquad |z| < r, \; z \in \mathbb{R},$$

Define $\mathcal{X}$ to be the $\sqrt{r}$-ball in $\mathbb{R}^d$: $\|x\| < \sqrt{r}$,

$$k(x, x') = f\left(\langle x, x' \rangle\right) = \sum_{n=0}^{\infty} a_n \langle x, x' \rangle^n .$$

Example (Exponential kernel)

$$k(x, x') := \exp\left(\langle x, x' \rangle\right) .$$

Gaussian kernel

Example (Gaussian kernel)

The Gaussian kernel on $\mathbb{R}^d$ is defined as

$$k(x, x') := \exp\left(-\gamma^{-2}\left\|x - x'\right\|^2\right).$$

Proof: an exercise! Use product rule, exponential kernel.

Positive definite functions

If we are given a "measure of similarity" with two arguments, $k(x, x')$, how can we determine if it is a valid kernel?

1. Find a feature map?
   1. Sometimes this is not obvious (eg if the feature vector is infinite dimensional)
   2. In any case, the feature map is not unique.
2. A direct property of the function: positive definiteness.

Positive definite functions

Definition (Positive definite functions)

A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive definite if $\forall n \geq 1$, $\forall (a_1, \ldots a_n) \in \mathbb{R}^n$, $\forall (x_1, \ldots, x_n) \in \mathcal{X}^n$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_i, x_j) \geq 0.$$

The function $k(\cdot, \cdot)$ is strictly positive definite if for mutually distinct x_i, the equality holds only when all the a_i are zero.

Kernels are positive definite

Theorem

The kernel $k(x, y) := \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$ for Hilbert space $\mathcal{H}$ is positive definite.

Proof.

$$\begin{aligned} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_i, x_j) &= \sum_{i=1}^{n} \sum_{j=1}^{n} \langle a_i \phi(x_i), a_j \phi(x_j) \rangle_{\mathcal{H}} \\ &= \left\| \sum_{i=1}^{n} a_i \phi(x_i) \right\|_{\mathcal{H}}^2 \geq 0. \end{aligned}$$

□

Kernels are positive definite

Theorem

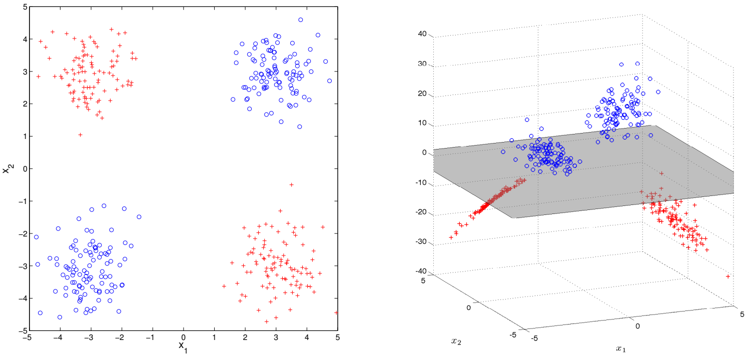
The kernel $k(x,y) := \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$ for Hilbert space $\mathcal{H}$ is positive definite.

- Reverse also holds: positive definite $k(x,x')$ is an inner product between $\phi(x)$ and $\phi(x')$ in some Hilbert space $\mathcal{H}$ (**Moore-Aronszajn** theorem)
- No need to explicitly specify features: This makes optimization *much* easier (e.g. when doing classification: Part II)

The reproducing kernel Hilbert space

First example: finite space, polynomial features

Reminder: XOR example:



First example: finite space, polynomial features

Reminder: Feature space from XOR motivating example:

$$\phi : \mathcal{X}(=\mathbb{R}^2) \rightarrow \mathcal{H}(=\mathbb{R}^3).$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \phi(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix},$$

with kernel

$$k(x,y) = \begin{bmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix}^\top \begin{bmatrix} y_1 \\ y_2 \\ y_1 y_2 \end{bmatrix}$$

(the standard inner product in $\mathbb{R}^3$ between features).

First example: finite space, polynomial features

Define a linear function f of the inputs x_1, x_2, and their product $x_1 x_2$ (linear on the feature space, **not** on the original space)

$$f(x) = f_1 x_1 + f_2 x_2 + f_3 x_1 x_2.$$

Then f is a function from $\mathcal{X} = \mathbb{R}^2$ to $\mathbb{R}$. Equivalent representation for f is:

$$f = [f_1 \; f_2 \; f_3]^\top.$$

(so we can also think of f as a vector in $\mathcal{H} = \mathbb{R}^3$– conversely, for every $h \in \mathcal{H}$, there is a corresponding linear function $\mathbb{R}^2 \to \mathbb{R}$).

$$f(x) = f^\top \phi(x) = \langle f, \phi(x) \rangle_{\mathcal{H}}$$

Evaluation of f at x is an **inner product in feature space** (here standard inner product in $\mathbb{R}^3$)

$\mathcal{H}$ can always be interpreted as a space of $\mathbb{R}$-valued functions.

First example: finite space, polynomial features

Define a linear function f of the inputs x_1, x_2, and their product $x_1 x_2$ (linear on the feature space, **not** on the original space)

$$f(x) = f_1 x_1 + f_2 x_2 + f_3 x_1 x_2.$$

Then f is a function from $\mathcal{X} = \mathbb{R}^2$ to $\mathbb{R}$. Equivalent representation for f is:

$$f = [f_1 \; f_2 \; f_3]^\top .$$

(so we can also think of f as a vector in $\mathcal{H} = \mathbb{R}^3$– conversely, for every $h \in \mathcal{H}$, there is a corresponding linear function $\mathbb{R}^2 \to \mathbb{R}$).

$$f(x) = f^\top \phi(x) = \langle f, \phi(x) \rangle_{\mathcal{H}}$$

Evaluation of f at x is an **inner product in feature space** (here standard inner product in $\mathbb{R}^3$)

$\mathcal{H}$ can always be interpreted as a space of $\mathbb{R}$-valued functions.

First example: finite space, polynomial features

$\phi(y)$ is also an element of $\mathcal{H} = \mathbb{R}^3 \ldots$
...which parametrizes a function (of x, indexed by y) mapping $\mathbb{R}^2$ to $\mathbb{R}$:

$$k(\cdot, y) := \begin{bmatrix} y_1 & y_2 & y_1 y_2 \end{bmatrix}^\top = \phi(y),$$

evaluated as:

$$k(x, y) = \langle k(\cdot, y), \phi(x) \rangle_{\mathcal{H}} = a x_1 + b x_2 + c x_1 x_2,$$

where $a = y_1$, $b = y_2$, and $c = y_1 y_2$
We can write $\phi(x) = k(\cdot, x)$ and $\phi(y) = k(\cdot, y)$ without ambiguity: canonical feature map– it suffices to specify a kernel function.

First example: finite space, polynomial features

$\phi(y)$ is also an element of $\mathcal{H} = \mathbb{R}^3$...
...which parametrizes a function (of x, indexed by y) mapping $\mathbb{R}^2$ to $\mathbb{R}$:

$$k(\cdot, y) := \begin{bmatrix} y_1 & y_2 & y_1 y_2 \end{bmatrix}^\top = \phi(y),$$

evaluated as:

$$k(x, y) = \langle k(\cdot, y), \phi(x) \rangle_{\mathcal{H}} = a x_1 + b x_2 + c x_1 x_2,$$

where $a = y_1$, $b = y_2$, and $c = y_1 y_2$
We can write $\phi(x) = k(\cdot, x)$ and $\phi(y) = k(\cdot, y)$ without ambiguity: canonical feature map– it suffices to specify a kernel function.

The reproducing property

This example illustrates the two defining features of an RKHS:

- **The reproducing property:**
  $\forall x \in \mathcal{X}, \forall f \in \mathcal{H}, \;\; \langle f, k(\cdot, x)\rangle_{\mathcal{H}} = f(x)$
- In particular, for any $x, y \in \mathcal{X}$,

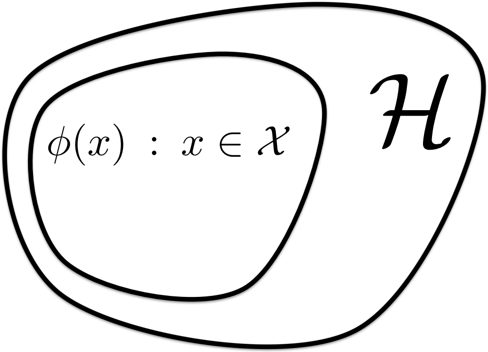
$$k(x, y) = \langle k(\cdot, x), k(\cdot, y)\rangle_{\mathcal{H}}.$$

Note: the feature map of every point is in the feature space:
$\forall x \in \mathcal{X}, \;\; k(\cdot, x) = \phi(x) \in \mathcal{H},$

RKHS is larger than $\{\phi(x) : x \in \mathcal{X}\}$

Another, more subtle point: $\mathcal{H}$ can be larger than all $\phi(x)$

Why?

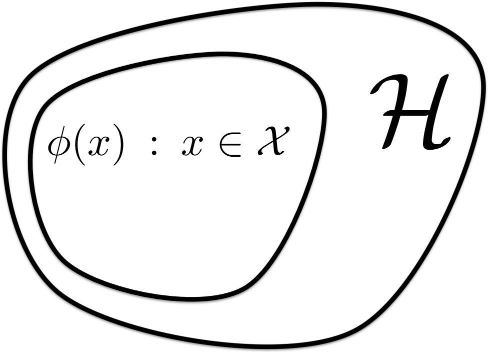


E.g. $f = [1\ 1\ -1] \in \mathcal{H}$ cannot be obtained by $\phi(x) = [x_1\ x_2\ (x_1 x_2)]$.

RKHS is larger than $\{\phi(x) : x \in \mathcal{X}\}$

Another, more subtle point: $\mathcal{H}$ can be larger than all $\phi(x)$

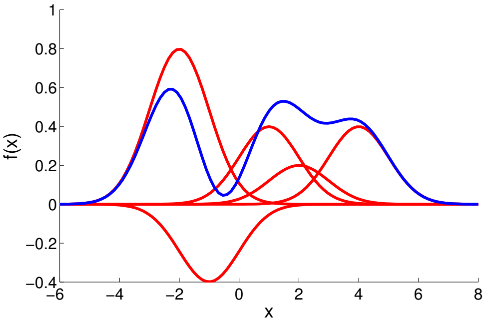
Why?



E.g. $f = [1\,1\,-1] \in \mathcal{H}$ cannot be obtained by $\phi(x) = [x_1\, x_2\, (x_1 x_2)]$.

Second example: infinite feature space

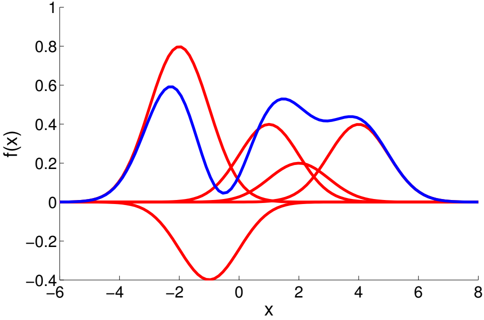
Reproducing property for function with Gaussian kernel:
$f(x) := \sum_{i=1}^{m} \alpha_i k(x_i, x) = \langle \sum_{i=1}^{m} \alpha_i \phi(x_i), \phi(x) \rangle_{\mathcal{H}}$.



- What do the features $\phi(x)$ look like (warning: there are infinitely many of them!)
- What do these features have to do with smoothness?

Second example: infinite feature space

Reproducing property for function with Gaussian kernel:
$f(x) := \sum_{i=1}^{m} \alpha_i k(x_i, x) = \left\langle \sum_{i=1}^{m} \alpha_i \phi(x_i), \phi(x) \right\rangle_{\mathcal{H}}$.



- What do the features $\phi(x)$ look like (warning: there are infinitely many of them!)
- What do these features have to do with smoothness?

Gaussian kernel example: infinite feature space

Under certain conditions (e.g Mercer's theorem), we can write

$$k(x,x') = \sum_{i=1}^{\infty} \lambda_i e_i(x) e_i(x'), \qquad \int_{\mathcal{X}} e_i(x) e_j(x) d\mu(x) = \begin{cases} 1, & i = j \\ 0, & i \neq j. \end{cases}$$

where this sum is guaranteed to converge whatever the x and x'.
Infinite-dimensional feature map can then be identified with a sequence:

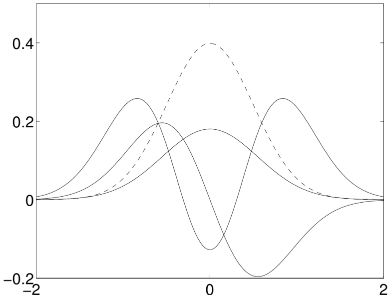
$$\phi(x) = \begin{bmatrix} \vdots \\ \sqrt{\lambda_i} e_i(x) \\ \vdots \end{bmatrix} \in \ell_2$$

Smoothness interpretation

Gaussian kernel, $k(x,y) = \exp\left(-\sigma \|x-y\|^2\right)$,

$$\begin{aligned} \lambda_j &\propto b^j \qquad b < 1 \\ e_j(x) &\propto \exp(-(c-a)x^2)H_j(x\sqrt{2c}), \end{aligned}$$

a, b, c are functions of σ, and H_j is jth order Hermite polynomial.



NOTE that $\|f\|_{\mathcal{H}}$ measures "smoothness": λ_j decay as e_j become "rougher" and for $f = \sum_j a_j e_j$:

$$\|f\|^2_{\mathcal{H}} = \sum_{j \in J} \frac{a_j^2}{\lambda_j}$$

(Figure from Rasmussen and Williams)

Reproducing kernel Hilbert space (1)

Definition

$\mathcal{H}$ a Hilbert space of $\mathbb{R}$-valued functions on non-empty set $\mathcal{X}$. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducing kernel of $\mathcal{H}$, and $\mathcal{H}$ is a reproducing kernel Hilbert space, if

- $\forall x \in \mathcal{X}, \quad k(\cdot, x) \in \mathcal{H}$,
- $\forall x \in \mathcal{X}, \forall f \in \mathcal{H}, \quad \langle f(\cdot), k(\cdot, x)\rangle_{\mathcal{H}} = f(x)$ (the reproducing property).

In particular, for any $x, y \in \mathcal{X}$,

$$k(x, y) = \langle k(\cdot, x), k(\cdot, y)\rangle_{\mathcal{H}}. \tag{3}$$

Original definition: kernel an inner product between feature maps. Then $\phi(x) = k(\cdot, x)$ a valid feature map.

Reproducing kernel Hilbert space (2)

Another RKHS definition:

Define δ_x to be the operator of evaluation at x, i.e.

$$\delta_x f = f(x) \quad \forall f \in \mathcal{H},\ x \in \mathcal{X}.$$

Definition (Reproducing kernel Hilbert space)

$\mathcal{H}$ is an RKHS if for all $f \in \mathcal{H}$, the evaluation operator δ_x is bounded: $\forall x \in \mathcal{X}$ there exists $\lambda_x \geq 0$ such that

$$|f(x)| = |\delta_x f| \leq \lambda_x \|f\|_{\mathcal{H}}$$

$\Longrightarrow$ two functions identical in RHKS norm agree at every point:

$$|f(x) - g(x)| = |\delta_x (f - g)| \leq \lambda_x \|f - g\|_{\mathcal{H}} \quad \forall f, g \in \mathcal{H}.$$

Simple Kernel Algorithms

Distance between means (1)

Sample $(x_i)_{i=1}^m$ from p and $(y_i)_{i=1}^m$ from q. What is the distance between their means *in feature space*?

$$\left\| \frac{1}{m}\sum_{i=1}^m \phi(x_i) - \frac{1}{n}\sum_{j=1}^n \phi(y_j) \right\|_{\mathcal{H}}^2$$

$$= \left\langle \frac{1}{m}\sum_{i=1}^m \phi(x_i) - \frac{1}{n}\sum_{j=1}^n \phi(y_j), \frac{1}{m}\sum_{i=1}^m \phi(x_i) - \frac{1}{n}\sum_{j=1}^n \phi(y_j) \right\rangle_{\mathcal{H}}$$

$$= \frac{1}{m^2}\left\langle \sum_{i=1}^m \phi(x_i), \sum_{i=1}^m \phi(x_i) \right\rangle + \ldots$$

$$= \frac{1}{m^2}\sum_{i=1}^m\sum_{j=1}^m k(x_i, x_j) + \frac{1}{n^2}\sum_{i=1}^n\sum_{j=1}^n k(y_i, y_j) - \frac{2}{mn}\sum_{i=1}^m\sum_{j=1}^m k(x_i, y_j).$$

Distance between means (1)

Sample $(x_i)_{i=1}^m$ from p and $(y_i)_{i=1}^m$ from q. What is the distance between their means *in feature space*?

$$\left\| \frac{1}{m}\sum_{i=1}^m \phi(x_i) - \frac{1}{n}\sum_{j=1}^n \phi(y_j) \right\|_{\mathcal{H}}^2$$

$$= \left\langle \frac{1}{m}\sum_{i=1}^m \phi(x_i) - \frac{1}{n}\sum_{j=1}^n \phi(y_j), \frac{1}{m}\sum_{i=1}^m \phi(x_i) - \frac{1}{n}\sum_{j=1}^n \phi(y_j) \right\rangle_{\mathcal{H}}$$

$$= \frac{1}{m^2}\left\langle \sum_{i=1}^m \phi(x_i), \sum_{i=1}^m \phi(x_i) \right\rangle + \ldots$$

$$= \frac{1}{m^2}\sum_{i=1}^m\sum_{j=1}^m k(x_i, x_j) + \frac{1}{n^2}\sum_{i=1}^n\sum_{j=1}^n k(y_i, y_j) - \frac{2}{mn}\sum_{i=1}^m\sum_{j=1}^m k(x_i, y_j).$$

Distance between means (2)

Sample $(x_i)_{i=1}^m$ from p and $(y_i)_{i=1}^m$ from q. What is the distance between their means *in feature space*?

$$\left\| \frac{1}{m}\sum_{i=1}^{m}\phi(x_i) - \frac{1}{n}\sum_{j=1}^{n}\phi(y_j) \right\|_{\mathcal{H}}^2$$

- When $\phi(x) = x$, distinguish means. When $\phi(x) = [x\ x^2]$, distinguish means and variances.

Nonparametric two-sample test.

There are kernels that can distinguish *any* two distributions (e.g. the Gaussian kernel, where the feature space is infinite).

Distance between means (2)

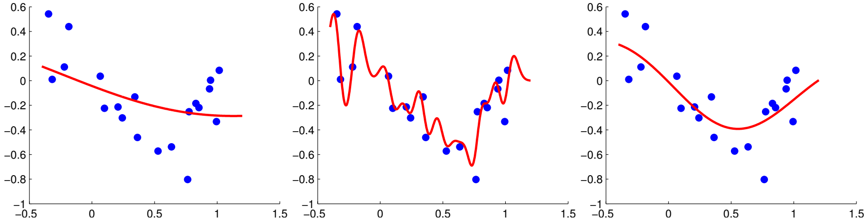
Sample $(x_i)_{i=1}^m$ from p and $(y_i)_{i=1}^m$ from q. What is the distance between their means *in feature space*?

$$\left\| \frac{1}{m}\sum_{i=1}^{m}\phi(x_i) - \frac{1}{n}\sum_{j=1}^{n}\phi(y_j) \right\|_{\mathcal{H}}^2$$

- When $\phi(x) = x$, distinguish means. When $\phi(x) = [x\ x^2]$, distinguish means and variances.

Nonparametric two-sample test.
There are kernels that can distinguish *any* two distributions (e.g. the Gaussian kernel, where the feature space is infinite).

Kernel ridge regression



Very simple to implement, works well when no outliers.

Ridge regression: case of $\mathbb{R}^D$

We are given n training points in $\mathbb{R}^D$:

$$X = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \in \mathbb{R}^{D\times n} \quad y := \begin{bmatrix} y_1 & \dots & y_n \end{bmatrix}^\top$$

Define some $\lambda > 0$. Our goal is:

$$f^* = \arg\min_{f\in\mathbb{R}^d} \left(\sum_{i=1}^n (y_i - f^\top x_i)^2 + \lambda \|f\|_2^2 \right)$$

The second term $\lambda\|f\|_2$ is chosen to avoid problems in high dimensional spaces.

Would like to replace with:

$$f^* = \arg\min_{f\in\mathcal{H}} \left(\sum_{i=1}^n \left(y_i - \langle f, \phi(x_i)\rangle_{\mathcal{H}}\right)^2 + \lambda \|f\|_{\mathcal{H}}^2 \right)$$

Ridge regression: case of $\mathbb{R}^D$

We are given n training points in $\mathbb{R}^D$:

$$X = \left[\begin{array}{ccc} x_1 & \ldots & x_n \end{array}\right] \in \mathbb{R}^{D\times n} \quad y := \left[\begin{array}{ccc} y_1 & \ldots & y_n \end{array}\right]^\top$$

Define some $\lambda > 0$. Our goal is:

$$f^* = \arg\min_{f\in\mathbb{R}^d} \left(\sum_{i=1}^n (y_i - f^\top x_i)^2 + \lambda \|f\|_2^2 \right)$$

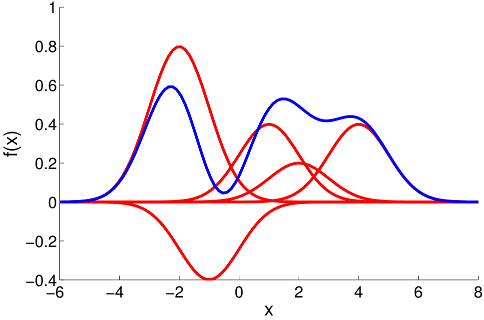
The second term $\lambda\|f\|_2$ is chosen to avoid problems in high dimensional spaces.
Would like to replace with:

$$f^* = \arg\min_{f\in\mathcal{H}} \left(\sum_{i=1}^n \left(y_i - \langle f, \phi(x_i)\rangle_{\mathcal{H}}\right)^2 + \lambda \|f\|_{\mathcal{H}}^2 \right)$$

Kernel ridge regression

We *begin* knowing f is a linear combination of feature space mappings of points (representer theorem: second set of notes)

$$f = \sum_{i=1}^{n} \alpha_i \phi(x_i) = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot).$$



Kernel ridge regression

We *begin* knowing f is a linear combination of feature space mappings of points (representer theorem: second set of notes)

$$f = \sum_{i=1}^{n} \alpha_i \phi(x_i) = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot).$$

Then

$$\begin{aligned}\sum_{i=1}^{n} \left(y_i - \langle f, \phi(x_i)\rangle_{\mathcal{H}}\right)^2 + \lambda \|f\|_{\mathcal{H}}^2 &= \|y - K\alpha\|_2^2 + \lambda \alpha^\top K \alpha \\ &= y^\top y - 2y^\top K\alpha + \alpha^\top \left(K^2 + \lambda K\right) \alpha\end{aligned}$$

Differentiating wrt α and setting this to zero, we get

$$\alpha^* = (K + \lambda I_n)^{-1} y.$$

Recall: $\frac{\partial \alpha^\top U \alpha}{\partial \alpha} = (U + U^\top)\alpha, \qquad \frac{\partial v^\top \alpha}{\partial \alpha} = \frac{\partial \alpha^\top v}{\partial \alpha} = v$

Kernel ridge regression

We *begin* knowing f is a linear combination of feature space mappings of points (representer theorem: second set of notes)

$$f = \sum_{i=1}^{n} \alpha_i \phi(x_i) = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot).$$

Then

$$\begin{aligned} \sum_{i=1}^{n} \left(y_i - \langle f, \phi(x_i)\rangle_{\mathcal{H}}\right)^2 + \lambda \|f\|_{\mathcal{H}}^2 &= \|y - K\alpha\|_2^2 + \lambda \alpha^\top K \alpha \\ &= y^\top y - 2y^\top K\alpha + \alpha^\top \left(K^2 + \lambda K\right) \alpha \end{aligned}$$

Differentiating wrt α and setting this to zero, we get

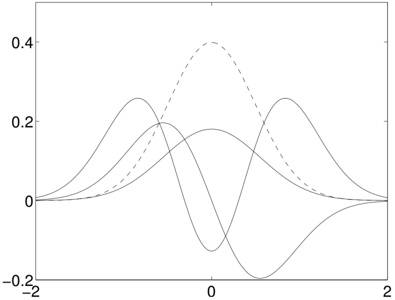
$$\alpha^* = (K + \lambda I_n)^{-1} y.$$

Recall: $\frac{\partial \alpha^\top U \alpha}{\partial \alpha} = (U + U^\top)\alpha, \qquad \frac{\partial v^\top \alpha}{\partial \alpha} = \frac{\partial \alpha^\top v}{\partial \alpha} = v$

Smoothness

What does a small $\|f\|_{\mathcal{H}}$ achieve? Smoothness!

Recall that for $f = \sum_j a_j e_j$: $\|f\|^2_{\mathcal{H}} = \sum_{j \in J} \frac{a_j^2}{\lambda_j}$, (where $\lambda_j \to 0$)



- the smaller the norm, the faster the a_j have to decay, hence the smaller the weight on the high frequency features.

Parameter selection for KRR

Given the objective

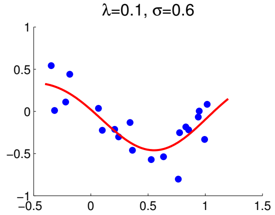
$$f^* = \arg\min_{f \in \mathcal{H}} \left(\sum_{i=1}^{n} (y_i - \langle f, \phi(x_i) \rangle_{\mathcal{H}})^2 + \lambda \|f\|_{\mathcal{H}}^2 \right).$$

How do we choose

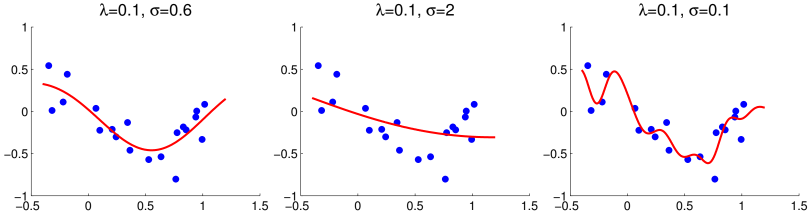
- The regularization parameter λ?
- The kernel parameter: for Gaussian kernel, σ in

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

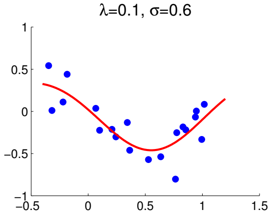
Choice of σ



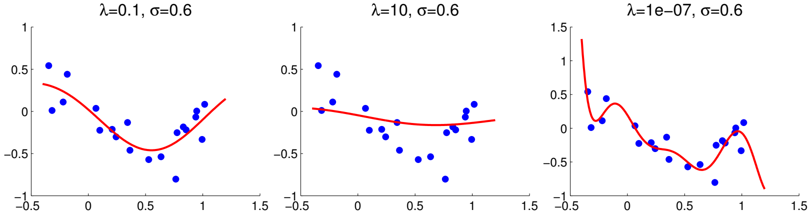
Choice of σ



Choice of λ



Choice of λ

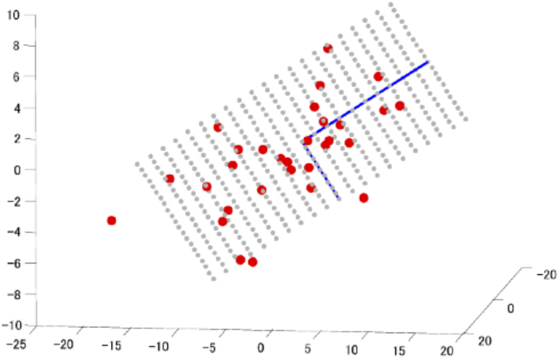


Cross-validation

- Split data into training set size n_{tr} and **test set** size $n_{\mathrm{te}} = 1 - n_{\mathrm{tr}}$.
- Split trainining set into m equal chunks of size $n_{\mathrm{val}} = n_{\mathrm{tr}}/m$. Call these $X_{\mathrm{val},i}, Y_{\mathrm{val},i}$ for $i \in \{1, \ldots, m\}$
- For each λ, σ pair
  - For each $X_{\mathrm{val},i}, Y_{\mathrm{val},i}$
    - Train ridge regression on remaining trainining set data $X_{\mathrm{tr}} \setminus X_{\mathrm{val},i}$ and $Y_{\mathrm{tr}} \setminus Y_{\mathrm{val},i}$,
    - Evaluate its error on the validation data $X_{\mathrm{val},i}, Y_{\mathrm{val},i}$
  - Average the errors on the validation sets to get the average validation error for λ, σ.
- Choose λ^*, σ^* with the lowest average validation error
- Finally, measure the performance on the test set $X_{\mathrm{te}}, Y_{\mathrm{te}}$.

PCA (1)

Goal of classical PCA: to find a d-dimensional subspace of a higher dimensional space (D-dimensional, $\mathbb{R}^D$) containing the directions of maximum variance.



(Figure from Kenji Fukumizu)

Applicationof kPCA: image denoising

What is the purpose of kernel PCA?

We consider the problem of **denoising** hand-written digits.

We are given a noisy digit x^*.

$$P_d\phi(x^*) = P_{f_1}\phi(x^*) + \ldots + P_{f_d}\phi(x^*)$$

is the projection of $\phi(x^*)$ onto one of the first d eigenvectors from kernel PCA (these are orthogonal).

Define the nearest point $y^* \in \mathcal{X}$ to this feature space projection as

$$y^* = \arg\min_{y \in \mathcal{X}} \|\phi(y) - P_d\phi(x^*)\|^2_{\mathcal{H}}.$$

In many cases,not possible to reduce the squared error to zero, as no single y^* corresponds to exact solution.

Applicationof kPCA: image denoising

What is the purpose of kernel PCA?

We consider the problem of **denoising** hand-written digits.

We are given a noisy digit x^*.

$$P_d\phi(x^*) = P_{f_1}\phi(x^*) + \ldots + P_{f_d}\phi(x^*)$$

is the projection of $\phi(x^*)$ onto one of the first d eigenvectors from kernel PCA (these are orthogonal).

Define the nearest point $y^* \in \mathcal{X}$ to this feature space projection as

$$y^* = \arg\min_{y \in \mathcal{X}} \|\phi(y) - P_d\phi(x^*)\|^2_{\mathcal{H}}.$$

In many cases,not possible to reduce the squared error to zero, as no single y^* corresponds to exact solution.

Applicationof kPCA: image denoising

What is the purpose of kernel PCA?

We consider the problem of **denoising** hand-written digits.
We are given a noisy digit x^*.

$$P_d\phi(x^*) = P_{f_1}\phi(x^*) + \ldots + P_{f_d}\phi(x^*)$$

is the projection of $\phi(x^*)$ onto one of the first d eigenvectors from kernel PCA (these are orthogonal).
Define the nearest point $y^* \in \mathcal{X}$ to this feature space projection as

$$y^* = \arg\min_{y\in\mathcal{X}} \|\phi(y) - P_d\phi(x^*)\|^2_{\mathcal{H}}.$$

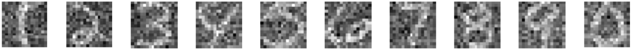
In many cases,not possible to reduce the squared error to zero, as no single y^* corresponds to exact solution.

Applicationof kPCA: image denoising

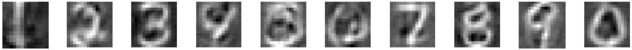
Projection onto PCA subspace for denoising. kPCA: data may not be Gaussian distributed, but can lie in a submanifold in input space.

USPS hand-written digits data:
7191 images of hand-written digits of 16 × 16 pixels.

Sample of original images (not used for experiments)



Sample of noisy images



Sample of denoised images (linear PCA)

Sample of denoised images (kernel PCA, Gaussian kernel)

What is PCA?

First principal component (max. variance)

$$\begin{aligned} u_1 &= \arg\max_{\|u\|\leq 1} \frac{1}{n}\sum_{i=1}^n \left(u^\top \left(x_i - \frac{1}{n}\sum_{i=1}^n x_i\right)\right)^2 \\ &= \arg\max_{\|u\|\leq 1} u^\top C u \end{aligned}$$

where

$$C = \frac{1}{n}\sum_{i=1}^n \left(x_i - \frac{1}{n}\sum_{i=1}^n x_i\right)\left(x_i - \frac{1}{n}\sum_{i=1}^n x_i\right)^\top = \frac{1}{n} X H X^\top,$$

$X = \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix}$, $H = I - n^{-1}\mathbf{1}_{n\times n}$, $\mathbf{1}_{n\times n}$ a matrix of ones.

Definition (Principal components)

These are eigenvalues of $n\lambda_i u_i = C u_i$.

PCA in feature space

Kernel version, first principal component:

$$f_1 = \arg\max_{\|f\|_{\mathcal{H}}\leq 1} \frac{1}{n}\sum_{i=1}^{n}\left(\left\langle f, \phi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\right\rangle_{\mathcal{H}}\right)^2$$
$$= \arg\max_{\|f\|_{\mathcal{H}}\leq 1} \mathrm{var}(f).$$

We can write

$$f = \sum_{i=1}^{n}\alpha_i\left(\phi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\right),$$
$$= \sum_{i=1}^{n}\alpha_i\tilde{\phi}(x_i),$$

since any component orthogonal to the span of $\tilde{\phi}(x_i) := \phi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)$ vanishes.

PCA in feature space

Kernel version, first principal component:

$$f_1 = \arg\max_{\|f\|_{\mathcal{H}}\leq 1} \frac{1}{n}\sum_{i=1}^n \left(\left\langle f, \phi(x_i) - \frac{1}{n}\sum_{i=1}^n \phi(x_i)\right\rangle_{\mathcal{H}}\right)^2$$
$$= \arg\max_{\|f\|_{\mathcal{H}}\leq 1} \mathrm{var}(f).$$

We can write

$$f = \sum_{i=1}^n \alpha_i \left(\phi(x_i) - \frac{1}{n}\sum_{i=1}^n \phi(x_i)\right),$$
$$= \sum_{i=1}^n \alpha_i \tilde{\phi}(x_i),$$

since any component orthogonal to the span of $\tilde{\phi}(x_i) := \phi(x_i) - \frac{1}{n}\sum_{i=1}^n \phi(x_i)$ vanishes.

How to solve kernel PCA

We can also define an infinite dimensional analog of the covariance:

$$
\begin{aligned}
C &= \frac{1}{n}\sum_{i=1}^{n}\left(\phi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\right) \otimes \left(\phi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\right), \\
&= \frac{1}{n}\sum_{i=1}^{n}\tilde{\phi}(x_i) \otimes \tilde{\phi}(x_i)
\end{aligned}
$$

where we use the definition

$$
(a \otimes b)c := \langle b, c\rangle_{\mathcal{H}}\, a \tag{4}
$$

this is analogous to the case of finite dimensional vectors, $(ab^\top)c = a(b^\top c)$.

How to solve kernel PCA (1)

Eigenfunctions of kernel covariance:

$$
\begin{aligned}
f_\ell \lambda_\ell &= C f_\ell \\
&= \left(\frac{1}{n} \sum_{i=1}^{n} \tilde{\phi}(x_i) \otimes \tilde{\phi}(x_i) \right) f_\ell \\
&= \frac{1}{n} \sum_{i=1}^{n} \tilde{\phi}(x_i) \left\langle \tilde{\phi}(x_i), \sum_{j=1}^{n} \alpha_{\ell j} \tilde{\phi}(x_j) \right\rangle_{\mathcal{H}} \\
&= \frac{1}{n} \sum_{i=1}^{n} \tilde{\phi}(x_i) \left(\sum_{j=1}^{n} \alpha_{\ell j} \tilde{k}(x_i, x_j) \right)
\end{aligned}
$$

$\tilde{k}(x_i, x_j)$ is the (i, j)th entry of the matrix $\tilde{K} := HKH$ (exercise!).

How to solve kernel PCA (1)

Eigenfunctions of kernel covariance:

$$\begin{aligned} f_\ell \lambda_\ell &= C f_\ell \\ &= \left(\frac{1}{n} \sum_{i=1}^{n} \tilde{\phi}(x_i) \otimes \tilde{\phi}(x_i) \right) f_\ell \\ &= \frac{1}{n} \sum_{i=1}^{n} \tilde{\phi}(x_i) \left\langle \tilde{\phi}(x_i), \sum_{j=1}^{n} \alpha_{\ell j} \tilde{\phi}(x_j) \right\rangle_{\mathcal{H}} \\ &= \frac{1}{n} \sum_{i=1}^{n} \tilde{\phi}(x_i) \left(\sum_{j=1}^{n} \alpha_{\ell j} \tilde{k}(x_i, x_j) \right) \end{aligned}$$

$\tilde{k}(x_i, x_j)$ is the (i, j)th entry of the matrix $\tilde{K} := HKH$ (exercise!).

How to solve kernel PCA (2)

We can now project both sides of

$$f_\ell \lambda_\ell = C f_\ell$$

onto all of the $\tilde{\phi}(x_q)$:

$$\left\langle \tilde{\phi}(x_q), \mathrm{LHS} \right\rangle_{\mathcal{H}} = \lambda_\ell \left\langle \tilde{\phi}(x_q), f_\ell \right\rangle = \lambda_\ell \sum_{i=1}^{n} \alpha_{\ell i} \tilde{k}(x_q, x_i) \qquad \forall q \in \{1 \dots n\}$$

$$\left\langle \tilde{\phi}(x_q), \mathrm{RHS} \right\rangle_{\mathcal{H}} = \left\langle \tilde{\phi}(x_q), C f_\ell \right\rangle_{\mathcal{H}} = \frac{1}{n} \sum_{i=1}^{n} \tilde{k}(x_q, x_i) \left(\sum_{j=1}^{n} \alpha_{\ell j} \tilde{k}(x_i, x_j) \right)$$

Writing this as a matrix equation,

$$n \lambda_\ell \widetilde{K} \alpha_\ell = \widetilde{K}^2 \alpha_\ell \qquad n \lambda_\ell \alpha_\ell = \widetilde{K} \alpha_\ell.$$

How to solve kernel PCA (2)

We can now project both sides of

$$f_\ell \lambda_\ell = C f_\ell$$

onto all of the $\tilde{\phi}(x_q)$:

$$\left\langle \tilde{\phi}(x_q), \mathrm{LHS} \right\rangle_{\mathcal{H}} = \lambda_\ell \left\langle \tilde{\phi}(x_q), f_\ell \right\rangle = \lambda_\ell \sum_{i=1}^{n} \alpha_{\ell i} \tilde{k}(x_q, x_i) \qquad \forall q \in \{1 \ldots n\}$$

$$\left\langle \tilde{\phi}(x_q), \mathrm{RHS} \right\rangle_{\mathcal{H}} = \left\langle \tilde{\phi}(x_q), C f_\ell \right\rangle_{\mathcal{H}} = \frac{1}{n} \sum_{i=1}^{n} \tilde{k}(x_q, x_i) \left(\sum_{j=1}^{n} \alpha_{\ell j} \tilde{k}(x_i, x_j) \right)$$

Writing this as a matrix equation,

$$n \lambda_\ell \widetilde{K} \alpha_\ell = \widetilde{K}^2 \alpha_\ell \qquad n \lambda_\ell \alpha_\ell = \widetilde{K} \alpha_\ell.$$

Projection onto kernel PC

How do you project a new point x^* onto the principal component f? Assuming f is properly normalised, the projection is

$$\begin{aligned} P_f \phi(x^*) &= \langle \phi(x^*), f \rangle_{\mathcal{H}} f \\ &= \sum_{i=1}^{n} \alpha_i \left(\sum_{j=1}^{n} \alpha_j k(x_j, x^*) \right) \tilde{\phi}(x_i). \end{aligned}$$