Modern Kernel Methods in Machine Learning: Part I

Danica J. Sutherland (she/her)
Computer Science, University of British Columbia
ETICS "summer" school, Oct 2022

Motivation

- Machine learning!

Motivation

- Machine learning! ...but how do we actually do it?

Motivation

- Machine learning! …but how do we actually do it?
- Linear models! $f(x) = w_0 + wx$, $\hat{y}(x) = \textbf{sign}(f(x))$

Motivation

- Machine learning! ...but how do we actually do it?
- Linear models! $f(x) = w_0 + wx$, $\hat{y}(x) = \textbf{sign}(f(x))$

Motivation

- Machine learning! ...but how do we actually do it?
- Linear models! $f(x) = w_0 + wx$, $\hat{y}(x) = \textbf{sign}(f(x))$

Motivation

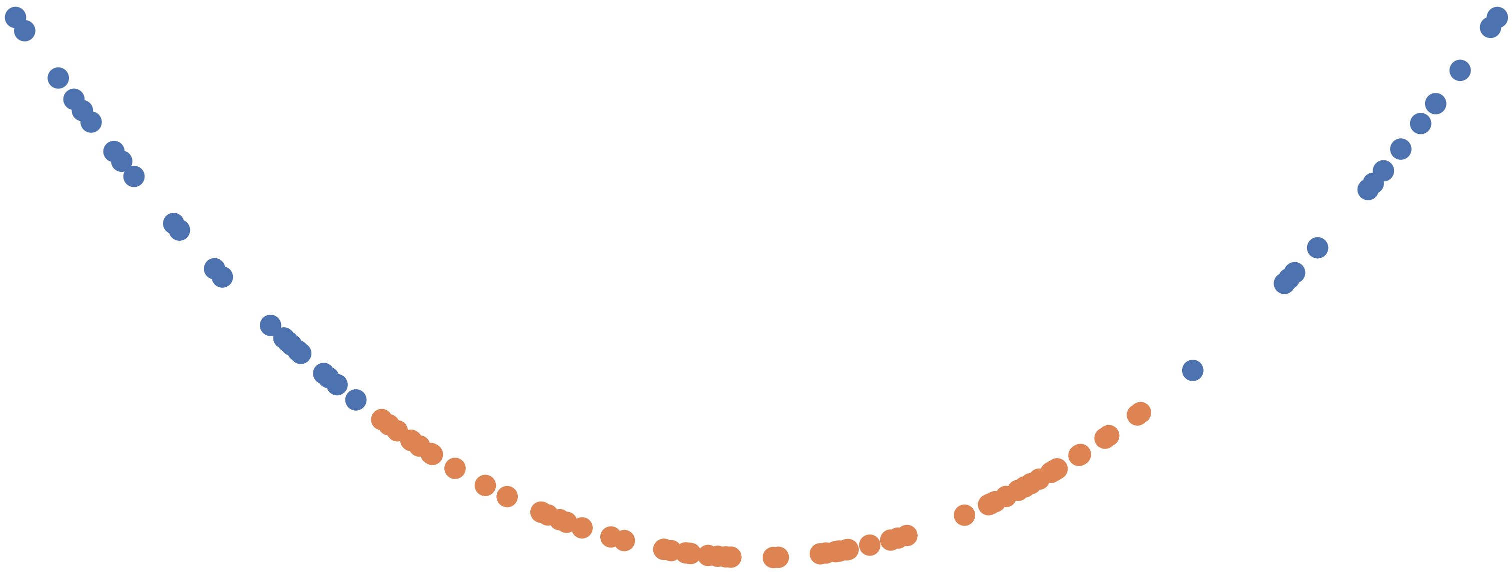
- Machine learning! ...but how do we actually do it?
- Linear models! $f(x) = w_0 + wx$, $\hat{y}(x) = \mathbf{sign}(f(x))$
- Extend x...

$$f(x) = w^\top(1, x, x^2) = w^\top \phi(x)$$

Motivation

- Machine learning! ...but how do we actually do it?
- Linear models! $f(x) = w_0 + wx$, $\hat{y}(x) = \mathbf{sign}(f(x))$
- Extend x...

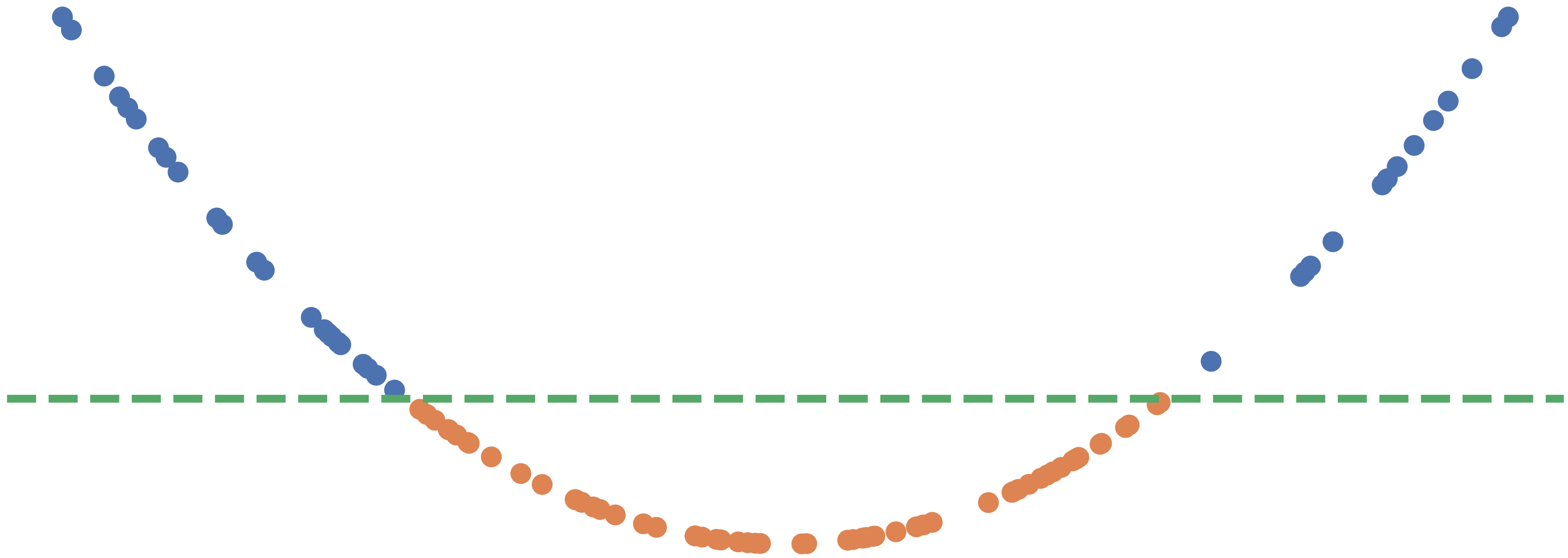
$$f(x) = w^{\mathsf{T}}(1, x, x^2) = w^{\mathsf{T}}\phi(x)$$



Motivation

- Machine learning! ...but how do we actually do it?
- Linear models! $f(x) = w_0 + wx$, $\hat{y}(x) = \textbf{sign}(f(x))$
- Extend x...

$$f(x) = w^{\mathsf{T}}(1, x, x^2) = w^{\mathsf{T}}\phi(x)$$



Motivation

- Machine learning! ...but how do we actually do it?
- Linear models! $f(x) = w_0 + wx$, $\hat{y}(x) = \textbf{sign}(f(x))$
- Extend x...

$$f(x) = w^{\mathsf{T}}(1, x, x^2) = w^{\mathsf{T}}\phi(x)$$

- Kernels are basically a way to study doing this with any, potentially very complicated, ϕ

Motivation

- Machine learning! ...but how do we actually do it?
- Linear models! $f(x) = w_0 + wx$, $\hat{y}(x) = \textbf{sign}(f(x))$
- Extend x...

$$f(x) = w^\mathsf{T}(1, x, x^2) = w^\mathsf{T}\phi(x)$$

- Kernels are basically a way to study doing this with any, potentially very complicated, ϕ
- Convenient way to make models on documents, graphs, videos, datasets, ...

Motivation

- Machine learning! ...but how do we actually do it?
- Linear models! $f(x) = w_0 + wx$, $\hat{y}(x) = \mathbf{sign}(f(x))$
- Extend x...

$$f(x) = w^\mathsf{T}(1, x, x^2) = w^\mathsf{T}\phi(x)$$

- Kernels are basically a way to study doing this with any, potentially very complicated, ϕ
- Convenient way to make models on documents, graphs, videos, datasets, ...
- ϕ will live in a *reproducing kernel Hilbert space*

Hilbert spaces

- A complete (real or complex) inner product space.

Hilbert spaces

- A complete (real ~~or complex~~) inner product space.

Hilbert spaces

- A complete (real ~~or complex~~) inner product space.
- Inner product space: a vector space with an **inner product**:
  - $\langle \alpha_1 f_1 + \alpha_2 f_2, g \rangle_{\mathcal{H}} = \alpha_1 \langle f_1, g \rangle_{\mathcal{H}} + \alpha_2 \langle f_2, g \rangle_{\mathcal{H}}$
  - $\langle f, g \rangle_{\mathcal{H}} = \langle g, f \rangle_{\mathcal{H}}$
  - $\langle f, f \rangle_{\mathcal{H}} > 0$ for $f \neq 0$, $\langle 0, 0 \rangle_{\mathcal{H}} = 0$

Hilbert spaces

- A complete (real ~~or complex~~) inner product space.
- Inner product space: a vector space with an **inner product**:
  - $\langle \alpha_1 f_1 + \alpha_2 f_2, g\rangle_{\mathcal{H}} = \alpha_1 \langle f_1, g\rangle_{\mathcal{H}} + \alpha_2 \langle f_2, g\rangle_{\mathcal{H}}$
  - $\langle f, g\rangle_{\mathcal{H}} = \langle g, f\rangle_{\mathcal{H}}$
  - $\langle f, f\rangle_{\mathcal{H}} > 0$ for $f \neq 0$, $\langle 0, 0\rangle_{\mathcal{H}} = 0$

  Induces a **norm**: $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f\rangle_{\mathcal{H}}}$

Hilbert spaces

- A complete (real ~~or complex~~) inner product space.
- Inner product space: a vector space with an **inner product**:
  - $\langle \alpha_1 f_1 + \alpha_2 f_2, g \rangle_{\mathcal{H}} = \alpha_1 \langle f_1, g \rangle_{\mathcal{H}} + \alpha_2 \langle f_2, g \rangle_{\mathcal{H}}$
  - $\langle f, g \rangle_{\mathcal{H}} = \langle g, f \rangle_{\mathcal{H}}$
  - $\langle f, f \rangle_{\mathcal{H}} > 0$ for $f \neq 0$, $\langle 0, 0 \rangle_{\mathcal{H}} = 0$

  Induces a **norm**: $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f \rangle_{\mathcal{H}}}$
- Complete: "well-behaved" (Cauchy sequences have limits in $\mathcal{H}$)

Kernel: an inner product between feature maps

- Call our domain $\mathcal{X}$, some set
  - $\mathbb{R}^d$, functions, distributions of graphs of images, …

Kernel: an inner product between feature maps

- Call our domain $\mathcal{X}$, some set
  - $\mathbb{R}^d$, functions, distributions of graphs of images, …
- $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel on $\mathcal{X}$ if there exists a Hilbert space $\mathcal{H}$ and a *feature map* $\phi : \mathcal{X} \to \mathcal{H}$ so that

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

Kernel: an inner product between feature maps

- Call our domain $\mathcal{X}$, some set
  - $\mathbb{R}^d$, functions, distributions of graphs of images, …
- $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel on $\mathcal{X}$ if there exists a Hilbert space $\mathcal{H}$ and a *feature map* $\phi : \mathcal{X} \to \mathcal{H}$ so that

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

- Roughly, k is a notion of “similarity” between inputs

Kernel: an inner product between feature maps

- Call our domain $\mathcal{X}$, some set
  - $\mathbb{R}^d$, functions, distributions of graphs of images, ...
- $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel on $\mathcal{X}$ if there exists a Hilbert space $\mathcal{H}$ and a *feature map* $\phi : \mathcal{X} \to \mathcal{H}$ so that

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

- Roughly, k is a notion of “similarity” between inputs
- *Linear kernel* on $\mathbb{R}^d$: $k(x, y) = \langle x, y \rangle_{\mathbb{R}^d}$

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Semi-related: kernel density estimation

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Semi-related: kernel density estimation
  - $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, usually symmetric, like RKHS kernel

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Semi-related: kernel density estimation
  - $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, usually symmetric, like RKHS kernel
  - Always requires $\int k(x, y) \mathrm{d}y = 1$, unlike RKHS kernel

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Semi-related: kernel density estimation
  - $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, usually symmetric, like RKHS kernel
  - Always requires $\int k(x, y) \mathrm{d}y = 1$, unlike RKHS kernel
  - Often requires $k(x, y) \geq 0$, unlike RKHS kernel

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Semi-related: kernel density estimation
  - $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, usually symmetric, like RKHS kernel
  - Always requires $\int k(x, y) \mathrm{d}y = 1$, unlike RKHS kernel
  - Often requires $k(x, y) \geq 0$, unlike RKHS kernel
  - Not required to be inner product, unlike RKHS kernel

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Semi-related: kernel density estimation
- Unrelated:

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Semi-related: kernel density estimation
- Unrelated:
  - The kernel (null space) of a linear map

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Semi-related: kernel density estimation
- Unrelated:
    - The kernel (null space) of a linear map
    - The kernel of a probability density

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Semi-related: kernel density estimation
- Unrelated:
  - The kernel (null space) of a linear map
  - The kernel of a probability density
  - The kernel of a convolution

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Semi-related: kernel density estimation
- Unrelated:
  - The kernel (null space) of a linear map
  - The kernel of a probability density
  - The kernel of a convolution
  - CUDA kernels

Aside: the name "kernel"

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Semi-related: kernel density estimation
- Unrelated:
  - The kernel (null space) of a linear map
  - The kernel of a probability density
  - The kernel of a convolution
  - CUDA kernels
  - The Linux kernel

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Semi-related: kernel density estimation
- Unrelated:
  - The kernel (null space) of a linear map
  - The kernel of a probability density
  - The kernel of a convolution
  - CUDA kernels
  - The Linux kernel
  - Popcorn kernels

Building kernels from other kernels

- Scaling: if $\gamma \geq 0$, $k_\gamma(x, y) = \gamma k(x, y)$ is a kernel

Building kernels from other kernels

- Scaling: if $\gamma \geq 0$, $k_\gamma(x, y) = \gamma k(x, y)$ is a kernel
  - $k_\gamma(x, y) = \gamma\langle\phi(x), \phi(y)\rangle_{\mathcal{H}} = \langle\sqrt{\gamma}\phi(x), \sqrt{\gamma}\phi(y)\rangle_{\mathcal{H}}$

Building kernels from other kernels

- Scaling: if $\gamma \geq 0$, $k_\gamma(x, y) = \gamma k(x, y)$ is a kernel
  - $k_\gamma(x, y) = \gamma \langle \phi(x), \phi(y) \rangle_{\mathcal{H}} = \langle \sqrt{\gamma}\phi(x), \sqrt{\gamma}\phi(y) \rangle_{\mathcal{H}}$
- Sum: $k_+(x, y) = k_1(x, y) + k_2(x, y)$ is a kernel

Building kernels from other kernels

- Scaling: if $\gamma \geq 0$, $k_\gamma(x, y) = \gamma k(x, y)$ is a kernel
  - $k_\gamma(x, y) = \gamma \langle \phi(x), \phi(y) \rangle_\mathcal{H} = \langle \sqrt{\gamma}\phi(x), \sqrt{\gamma}\phi(y) \rangle_\mathcal{H}$
- Sum: $k_+(x, y) = k_1(x, y) + k_2(x, y)$ is a kernel
  - $k_+(x, y) = \left\langle \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \end{bmatrix}, \begin{bmatrix} \phi_1(y) \\ \phi_2(y) \end{bmatrix} \right\rangle_{\mathcal{H}_1 \oplus \mathcal{H}_2}$

Building kernels from other kernels

- Scaling: if $\gamma \geq 0$, $k_\gamma(x, y) = \gamma k(x, y)$ is a kernel
  - $k_\gamma(x, y) = \gamma \langle \phi(x), \phi(y) \rangle_\mathcal{H} = \langle \sqrt{\gamma}\phi(x), \sqrt{\gamma}\phi(y) \rangle_\mathcal{H}$
- Sum: $k_+(x, y) = k_1(x, y) + k_2(x, y)$ is a kernel
  - $k_+(x, y) = \left\langle \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \end{bmatrix}, \begin{bmatrix} \phi_1(y) \\ \phi_2(y) \end{bmatrix} \right\rangle_{\mathcal{H}_1 \oplus \mathcal{H}_2}$
- Is $k_1(x, y) - k_2(x, y)$ necessarily a kernel?

Building kernels from other kernels

- Scaling: if $\gamma \geq 0$, $k_\gamma(x, y) = \gamma k(x, y)$ is a kernel
  - $k_\gamma(x, y) = \gamma \langle \phi(x), \phi(y) \rangle_{\mathcal{H}} = \langle \sqrt{\gamma}\phi(x), \sqrt{\gamma}\phi(y) \rangle_{\mathcal{H}}$
- Sum: $k_+(x, y) = k_1(x, y) + k_2(x, y)$ is a kernel
  - $k_+(x, y) = \left\langle \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \end{bmatrix}, \begin{bmatrix} \phi_1(y) \\ \phi_2(y) \end{bmatrix} \right\rangle_{\mathcal{H}_1 \oplus \mathcal{H}_2}$
- Is $k_1(x, y) - k_2(x, y)$ necessarily a kernel?
  - Take $k_1(x, y) = 0$, $k_2(x, y) = xy$, $x \neq 0$.
  - Then $k_1(x, x) - k_2(x, x) = -x^2 < 0$
  - But $k(x, x) = \|\phi(x)\|_{\mathcal{H}}^2 \geq 0$.

Positive definiteness

- A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (i.e. have $k(x, y) = k(y, x)$) is *positive semi-definite (psd)* if for all $n \geq 1$, $(a_1, \ldots, a_n) \in \mathbb{R}^n$, $(x_1, \ldots, x_n) \in \mathcal{X}^n$,

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

Positive definiteness

- A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (i.e. have $k(x, y) = k(y, x)$) is *positive semi-definite (psd)* if for all $n \geq 1$, $(a_1, \ldots, a_n) \in \mathbb{R}^n$, $(x_1, \ldots, x_n) \in \mathcal{X}^n$,

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

- Equivalently: *kernel matrix* K is PSD

$$K := \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \ldots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \ldots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \ldots & k(x_n, x_n) \end{bmatrix}$$

Positive definiteness

- A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (i.e. have $k(x, y) = k(y, x)$) is *positive semi-definite (psd)* if for all $n \geq 1$, $(a_1, \ldots, a_n) \in \mathbb{R}^n$, $(x_1, \ldots, x_n) \in \mathcal{X}^n$,

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

- Hilbert space kernels are psd

Positive definiteness

- A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (i.e. have $k(x, y) = k(y, x)$) is *positive semi-definite (psd)* if for all $n \geq 1$, $(a_1, \ldots, a_n) \in \mathbb{R}^n$, $(x_1, \ldots, x_n) \in \mathcal{X}^n$,

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

- Hilbert space kernels are psd

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \langle a_i \phi(x_i), a_j \phi(x_j) \rangle_{\mathcal{H}}$$

Positive definiteness

- A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (i.e. have $k(x, y) = k(y, x)$) is *positive semi-definite (psd)* if for all $n \geq 1$, $(a_1, \ldots, a_n) \in \mathbb{R}^n$, $(x_1, \ldots, x_n) \in \mathcal{X}^n$,

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

- Hilbert space kernels are psd

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \langle a_i \phi(x_i), a_j \phi(x_j) \rangle_{\mathcal{H}} = \left\langle \sum_{i=1}^{n} a_i \phi(x_i), \sum_{j=1}^{n} a_j \phi(x_j) \right\rangle_{\mathcal{H}}$$

Positive definiteness

- A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (i.e. have $k(x, y) = k(y, x)$) is *positive semi-definite (psd)* if for all $n \geq 1$, $(a_1, \ldots, a_n) \in \mathbb{R}^n$, $(x_1, \ldots, x_n) \in \mathcal{X}^n$,

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

- Hilbert space kernels are psd

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

Positive definiteness

- A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (i.e. have $k(x, y) = k(y, x)$) is *positive semi-definite (psd)* if for all $n \geq 1$, $(a_1, \ldots, a_n) \in \mathbb{R}^n$, $(x_1, \ldots, x_n) \in \mathcal{X}^n$,

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

- Hilbert space kernels are psd

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

Positive definiteness

- A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (i.e. have $k(x, y) = k(y, x)$) is *positive semi-definite (psd)* if for all $n \geq 1$, $(a_1, \ldots, a_n) \in \mathbb{R}^n$, $(x_1, \ldots, x_n) \in \mathcal{X}^n$,

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

- Hilbert space kernels are psd

Positive definiteness

- A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (i.e. have $k(x, y) = k(y, x)$) is *positive semi-definite (psd)* if for all $n \geq 1$, $(a_1, \ldots, a_n) \in \mathbb{R}^n$, $(x_1, \ldots, x_n) \in \mathcal{X}^n$,

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

- Hilbert space kernels are psd
- psd functions are Hilbert space kernels
  - Moore-Aronszajn Theorem; we'll come back to this

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m \to \infty} k_m(x, y)$ exists, k_∞ is psd
  - $\lim\limits_{m \to \infty} \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} a_i a_j k_m(x_i, x_j) \geq 0$

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m \to \infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y) k_2(x, y)$ is psd

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y)k_2(x, y)$ is psd
  - Let $V \sim \mathcal{N}(0, K_1)$, $W \sim \mathcal{N}(0, K_2)$ be independent
  - $\mathrm{Cov}(V_i W_i, V_j W_j) = \mathrm{Cov}(V_i, V_j)\,\mathrm{Cov}(W_i, W_j) = k_\times(x_i, x_j)$
  - Covariance matrices are psd, so $k_\times$ is too

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y)k_2(x, y)$ is psd

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y)k_2(x, y)$ is psd
- Powers: $k_n(x, y) = k(x, y)^n$ is pd for any integer $n \geq 0$

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y)k_2(x, y)$ is psd
- Powers: $k_n(x, y) = k(x, y)^n$ is pd for any integer $n \geq 0$

$$x^\mathsf{T} y$$

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y)k_2(x, y)$ is psd
- Powers: $k_n(x, y) = k(x, y)^n$ is pd for any integer $n \geq 0$

$$x^\mathsf{T} y + c$$

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m \to \infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y) k_2(x, y)$ is psd
- Powers: $k_n(x, y) = k(x, y)^n$ is pd for any integer $n \geq 0$

  $(x^\mathsf{T} y + c)^n$

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m \to \infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y) k_2(x, y)$ is psd
- Powers: $k_n(x, y) = k(x, y)^n$ is pd for any integer $n \geq 0$

  $\left(x^\mathsf{T} y + c\right)^n$, the **polynomial kernel**

Some more ways to build kernels

- Limits: if $k_\infty(x,y) = \lim_{m\to\infty} k_m(x,y)$ exists, k_∞ is psd
- Products: $k_\times(x,y) = k_1(x,y)k_2(x,y)$ is psd
- Powers: $k_n(x,y) = k(x,y)^n$ is pd for any integer $n \geq 0$
- Exponents: $k_{\exp}(x,y) = \exp(k(x,y))$ is pd

Some more ways to build kernels

- Limits: if $k_\infty(x,y) = \lim_{m\to\infty} k_m(x,y)$ exists, k_∞ is psd
- Products: $k_\times(x,y) = k_1(x,y)k_2(x,y)$ is psd
- Powers: $k_n(x,y) = k(x,y)^n$ is pd for any integer $n \geq 0$
- Exponents: $k_{\exp}(x,y) = \exp(k(x,y))$ is pd
    - $k_{\exp}(x,y) = \lim_{N\to\infty} \sum_{n=0}^{N} \frac{1}{n!} k(x,y)^n$

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y)k_2(x, y)$ is psd
- Powers: $k_n(x, y) = k(x, y)^n$ is pd for any integer $n \geq 0$
- Exponents: $k_{\exp}(x, y) = \exp(k(x, y))$ is pd

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y)k_2(x, y)$ is psd
- Powers: $k_n(x, y) = k(x, y)^n$ is pd for any integer $n \geq 0$
- Exponents: $k_{\exp}(x, y) = \exp(k(x, y))$ is pd
- If $f : \mathcal{X} \to \mathbb{R}$, $k_f(x, y) = f(x)k(x, y)f(y)$ is pd

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y)k_2(x, y)$ is psd
- Powers: $k_n(x, y) = k(x, y)^n$ is pd for any integer $n \geq 0$
- Exponents: $k_{\exp}(x, y) = \exp(k(x, y))$ is pd
- If $f : \mathcal{X} \to \mathbb{R}$, $k_f(x, y) = f(x)k(x, y)f(y)$ is pd
  - Use the feature map $x \mapsto f(x)\phi(x)$

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y)k_2(x, y)$ is psd
- Powers: $k_n(x, y) = k(x, y)^n$ is pd for any integer $n \geq 0$
- Exponents: $k_{\exp}(x, y) = \exp(k(x, y))$ is pd
- If $f : \mathcal{X} \to \mathbb{R}$, $k_f(x, y) = f(x)k(x, y)f(y)$ is pd

Some more ways to build kernels

- Limits: if $k_\infty(x,y) = \lim_{m\to\infty} k_m(x,y)$ exists, k_∞ is psd
- Products: $k_\times(x,y) = k_1(x,y)k_2(x,y)$ is psd
- Powers: $k_n(x,y) = k(x,y)^n$ is pd for any integer $n \geq 0$
- Exponents: $k_{\exp}(x,y) = \exp(k(x,y))$ is pd
- If $f : \mathcal{X} \to \mathbb{R}$, $k_f(x,y) = f(x)k(x,y)f(y)$ is pd

$$x^\mathsf{T} y$$

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y)k_2(x, y)$ is psd
- Powers: $k_n(x, y) = k(x, y)^n$ is pd for any integer $n \geq 0$
- Exponents: $k_{\exp}(x, y) = \exp(k(x, y))$ is pd
- If $f : \mathcal{X} \to \mathbb{R}$, $k_f(x, y) = f(x)k(x, y)f(y)$ is pd

$$\frac{1}{\sigma^2} x^\mathsf{T} y$$

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y)k_2(x, y)$ is psd
- Powers: $k_n(x, y) = k(x, y)^n$ is pd for any integer $n \geq 0$
- Exponents: $k_{\exp}(x, y) = \exp(k(x, y))$ is pd
- If $f : \mathcal{X} \to \mathbb{R}$, $k_f(x, y) = f(x)k(x, y)f(y)$ is pd

$$\exp\left(\frac{1}{\sigma^2} x^\mathsf{T} y\right)$$

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y)k_2(x, y)$ is psd
- Powers: $k_n(x, y) = k(x, y)^n$ is pd for any integer $n \geq 0$
- Exponents: $k_{\exp}(x, y) = \exp(k(x, y))$ is pd
- If $f : \mathcal{X} \to \mathbb{R}$, $k_f(x, y) = f(x)k(x, y)f(y)$ is pd

$$\exp\left(-\frac{1}{2\sigma^2}\|x\|^2\right)\exp\left(\frac{1}{\sigma^2}x^\mathsf{T} y\right)\exp\left(-\frac{1}{2\sigma^2}\|y\|^2\right)$$

Some more ways to build kernels

- Limits: if $k_\infty(x,y) = \lim_{m\to\infty} k_m(x,y)$ exists, k_∞ is psd
- Products: $k_\times(x,y) = k_1(x,y)k_2(x,y)$ is psd
- Powers: $k_n(x,y) = k(x,y)^n$ is pd for any integer $n \geq 0$
- Exponents: $k_{\exp}(x,y) = \exp(k(x,y))$ is pd
- If $f : \mathcal{X} \to \mathbb{R}$, $k_f(x,y) = f(x)k(x,y)f(y)$ is pd

$$\exp\left(-\frac{1}{2\sigma^2}\|x\|^2\right)\exp\left(\frac{1}{\sigma^2}x^\mathsf{T}y\right)\exp\left(-\frac{1}{2\sigma^2}\|y\|^2\right)$$

$$= \exp\left(-\tfrac{1}{2\sigma^2}\left[\|x\|^2 - 2x^\mathsf{T}y + \|y\|^2\right]\right)$$

Some more ways to build kernels

- Limits: if $k_\infty(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists, k_∞ is psd
- Products: $k_\times(x, y) = k_1(x, y)k_2(x, y)$ is psd
- Powers: $k_n(x, y) = k(x, y)^n$ is pd for any integer $n \geq 0$
- Exponents: $k_{\exp}(x, y) = \exp(k(x, y))$ is pd
- If $f : \mathcal{X} \to \mathbb{R}$, $k_f(x, y) = f(x)k(x, y)f(y)$ is pd

$$\exp\left(-\frac{1}{2\sigma^2}\|x\|^2\right)\exp\left(\frac{1}{\sigma^2}x^\mathsf{T}y\right)\exp\left(-\frac{1}{2\sigma^2}\|y\|^2\right)$$

$$= \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$$, the **Gaussian kernel**

Reproducing property

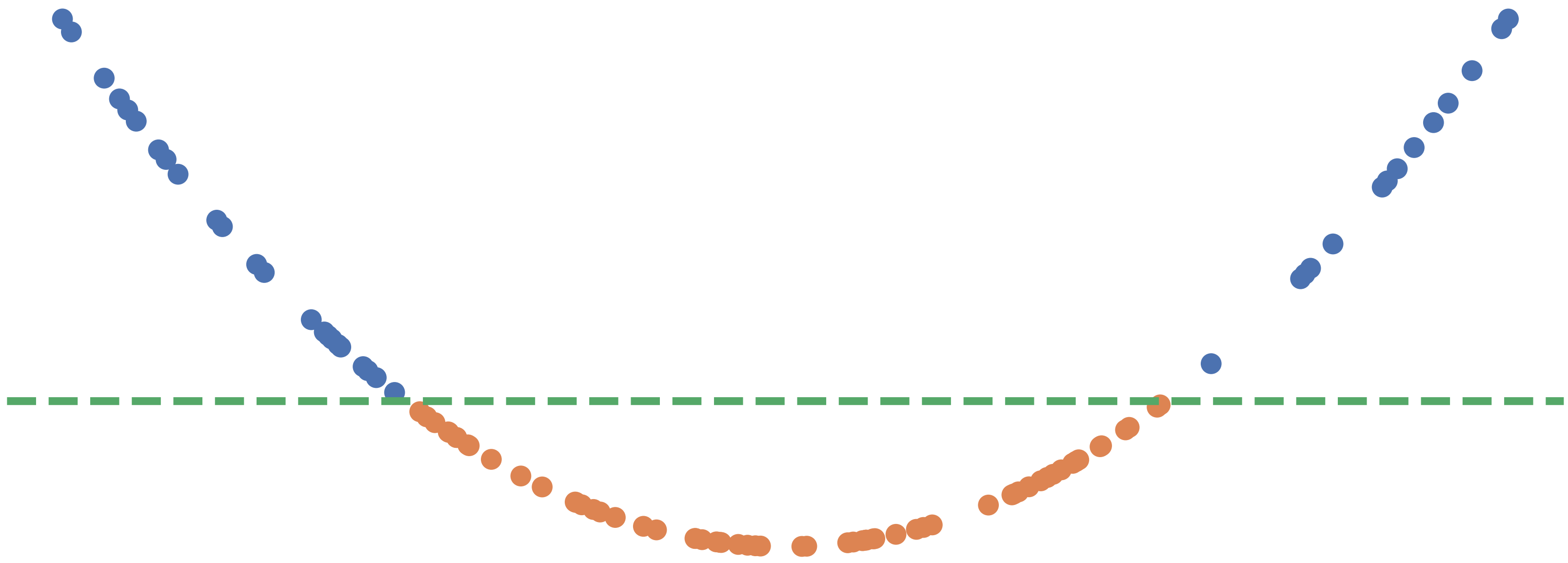
- Recall original motivating example with

$$\mathcal{X} = \mathbb{R} \qquad \phi(x) = (1, x, x^2) \in \mathbb{R}^3$$

Reproducing property

- Recall original motivating example with

$$\mathcal{X} = \mathbb{R} \qquad \phi(x) = (1, x, x^2) \in \mathbb{R}^3$$

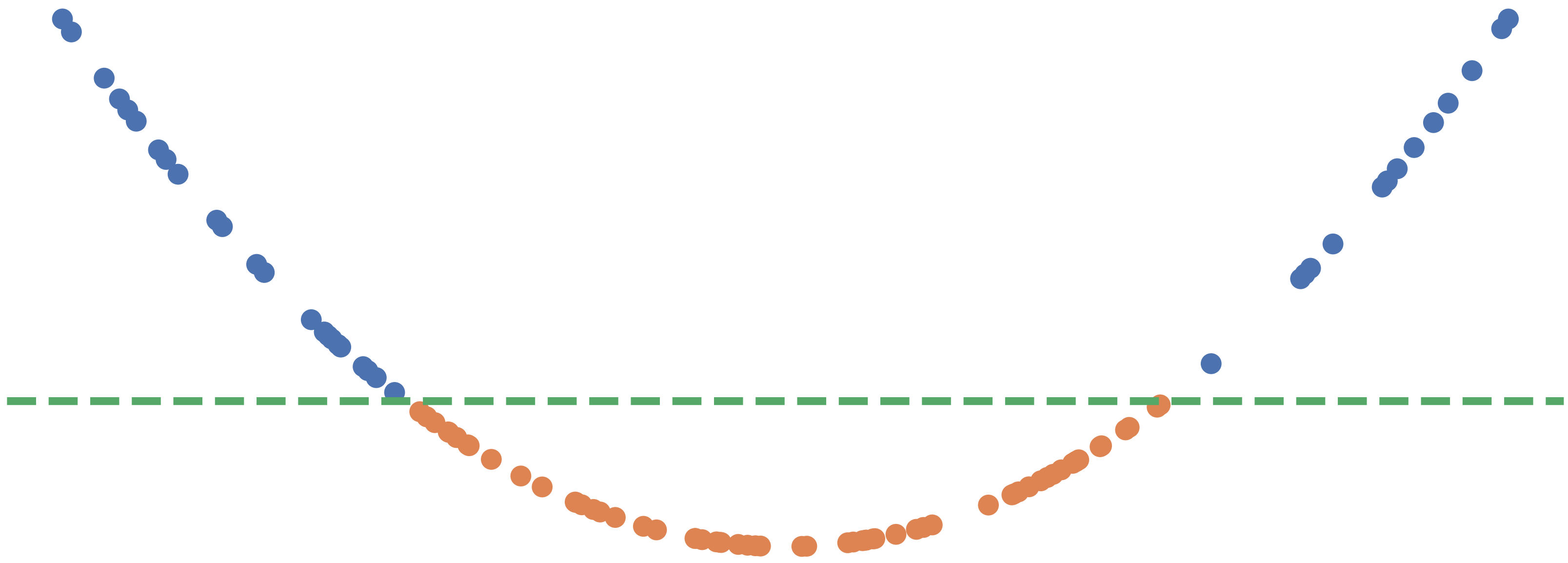


Reproducing property

- Recall original motivating example with

$$\mathcal{X} = \mathbb{R} \qquad \phi(x) = (1, x, x^2) \in \mathbb{R}^3$$

- Kernel is $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}} = 1 + xy + x^2 y^2$

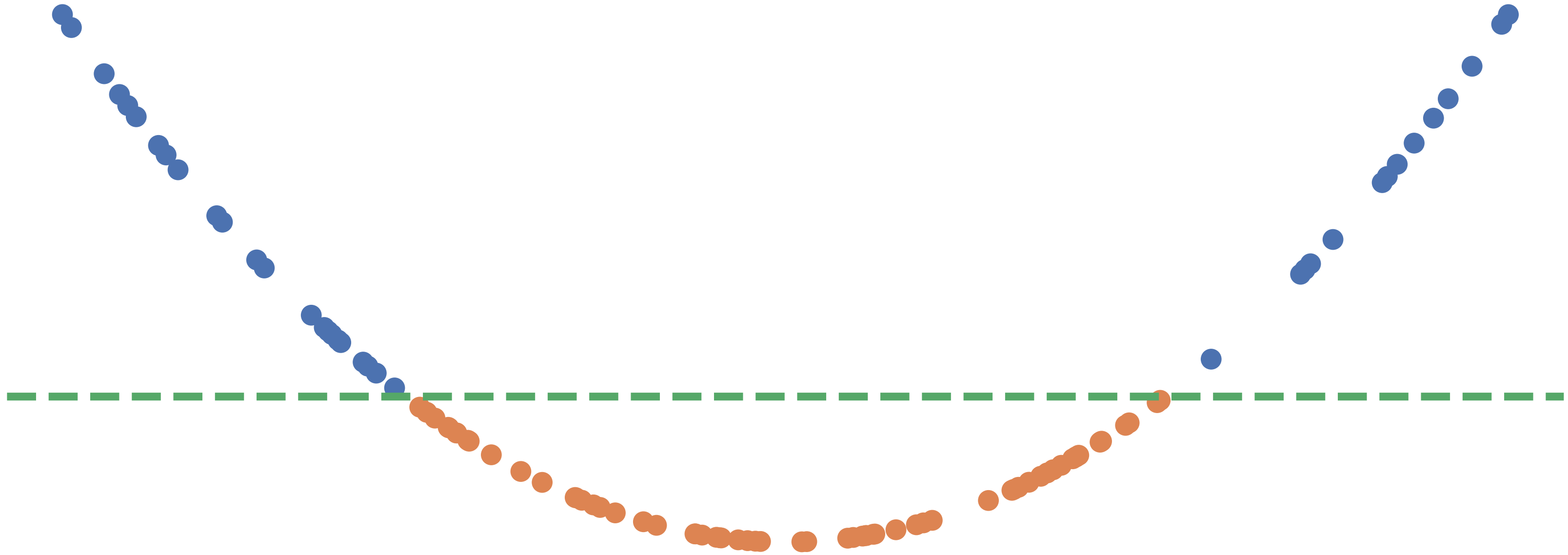


Reproducing property

- Recall original motivating example with

$$\mathcal{X} = \mathbb{R} \qquad \phi(x) = (1, x, x^2) \in \mathbb{R}^3$$

- Kernel is $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}} = 1 + xy + x^2 y^2$
- Classifier based on linear $f(x) = \langle w, \phi(x) \rangle_{\mathcal{H}}$



Reproducing property

- Recall original motivating example with

$$\mathcal{X} = \mathbb{R} \qquad \phi(x) = (1, x, x^2) \in \mathbb{R}^3$$

- Kernel is $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}} = 1 + xy + x^2 y^2$
- Classifier based on linear $f(x) = \langle w, \phi(x) \rangle_{\mathcal{H}}$
- $f(\cdot)$ is the function f itself; corresponds to vector w in $\mathbb{R}^3$
  $f(x) \in \mathbb{R}$ is the function evaluated at a point x

Reproducing property

- Recall original motivating example with

$$\mathcal{X} = \mathbb{R} \qquad \phi(x) = (1, x, x^2) \in \mathbb{R}^3$$

- Kernel is $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}} = 1 + xy + x^2y^2$
- Classifier based on linear $f(x) = \langle w, \phi(x) \rangle_{\mathcal{H}}$
- $f(\cdot)$ is the function f itself; corresponds to vector w in $\mathbb{R}^3$
  $f(x) \in \mathbb{R}$ is the function evaluated at a point x
- Elements of $\mathcal{H}$ are **functions**, $f : \mathcal{X} \to \mathbb{R}$

Reproducing property

- Recall original motivating example with

$$\mathcal{X} = \mathbb{R} \qquad \phi(x) = (1, x, x^2) \in \mathbb{R}^3$$

- Kernel is $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}} = 1 + xy + x^2y^2$
- Classifier based on linear $f(x) = \langle w, \phi(x) \rangle_{\mathcal{H}}$
- $f(\cdot)$ is the function f itself; corresponds to vector w in $\mathbb{R}^3$
  $f(x) \in \mathbb{R}$ is the function evaluated at a point x
- Elements of $\mathcal{H}$ are **functions**, $f : \mathcal{X} \to \mathbb{R}$
- **Reproducing prop.**: $f(x) = \langle f(\cdot), \phi(x) \rangle_{\mathcal{H}}$ for $f \in \mathcal{H}$

Reproducing kernel Hilbert space (RKHS)

- Every psd kernel k on $\mathcal{X}$ defines a (unique) Hilbert space, its RKHS $\mathcal{H}$, and a map $\phi : \mathcal{X} \to \mathcal{H}$ where
  - $$k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$$
  - Elements $f \in \mathcal{H}$ are functions on $\mathcal{X}$, with

    $$f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}}$$
- Combining the two, we sometimes write $k(x, \cdot) = \phi(x)$

Reproducing kernel Hilbert space (RKHS)

- Every psd kernel k on $\mathcal{X}$ defines a (unique) Hilbert space, its RKHS $\mathcal{H}$, and a map $\phi : \mathcal{X} \to \mathcal{H}$ where
  - $$k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$$
  - Elements $f \in \mathcal{H}$ are functions on $\mathcal{X}$, with

    $$f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}}$$
- Combining the two, we sometimes write $k(x, \cdot) = \phi(x)$
- $k(x, \cdot)$ is the **evaluation functional**

  An RKHS is defined by it being *continuous*, or

  $$|f(x)| \leq M_x \|f\|_{\mathcal{H}}$$

Moore-Aronszajn Theorem

- Building $\mathcal{H}$ for a given psd k:
  - Start with $\mathcal{H}_0 = \mathbf{span}(\{k(x, \cdot) : x \in \mathcal{X}\})$

Moore-Aronszajn Theorem

- Building $\mathcal{H}$ for a given psd k:
  - Start with $\mathcal{H}_0 = \mathbf{span}(\{k(x, \cdot) : x \in \mathcal{X}\})$
  - Define $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$ from $\langle k(x, \cdot), k(y, \cdot) \rangle_{\mathcal{H}_0} = k(x, y)$

Moore-Aronszajn Theorem

- Building $\mathcal{H}$ for a given psd k:
  - Start with $\mathcal{H}_0 = \mathbf{span}(\{k(x,\cdot) : x \in \mathcal{X}\})$
  - Define $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$ from $\langle k(x,\cdot), k(y,\cdot) \rangle_{\mathcal{H}_0} = k(x,y)$
  - Take $\mathcal{H}$ to be completion of $\mathcal{H}_0$ in the metric from $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$

Moore-Aronszajn Theorem

- Building $\mathcal{H}$ for a given psd k:
  - Start with $\mathcal{H}_0 = \mathbf{span}(\{k(x,\cdot) : x \in \mathcal{X}\})$
  - Define $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$ from $\langle k(x,\cdot), k(y,\cdot) \rangle_{\mathcal{H}_0} = k(x,y)$
  - Take $\mathcal{H}$ to be completion of $\mathcal{H}_0$ in the metric from $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$
  - Get that the reproducing property holds for $k(x,\cdot)$ in $\mathcal{H}$

Moore-Aronszajn Theorem

- Building $\mathcal{H}$ for a given psd k:
  - Start with $\mathcal{H}_0 = \textbf{span}(\{k(x,\cdot) : x \in \mathcal{X}\})$
  - Define $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$ from $\langle k(x,\cdot), k(y,\cdot) \rangle_{\mathcal{H}_0} = k(x,y)$
  - Take $\mathcal{H}$ to be completion of $\mathcal{H}_0$ in the metric from $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$
  - Get that the reproducing property holds for $k(x,\cdot)$ in $\mathcal{H}$
  - Can also show uniqueness

Moore-Aronszajn Theorem

- Building $\mathcal{H}$ for a given psd k:
  - Start with $\mathcal{H}_0 = \textbf{span}(\{k(x, \cdot) : x \in \mathcal{X}\})$
  - Define $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$ from $\langle k(x, \cdot), k(y, \cdot) \rangle_{\mathcal{H}_0} = k(x, y)$
  - Take $\mathcal{H}$ to be completion of $\mathcal{H}_0$ in the metric from $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$
  - Get that the reproducing property holds for $k(x, \cdot)$ in $\mathcal{H}$
  - Can also show uniqueness
- Theorem: k is psd iff it's the reproducing kernel of an RKHS

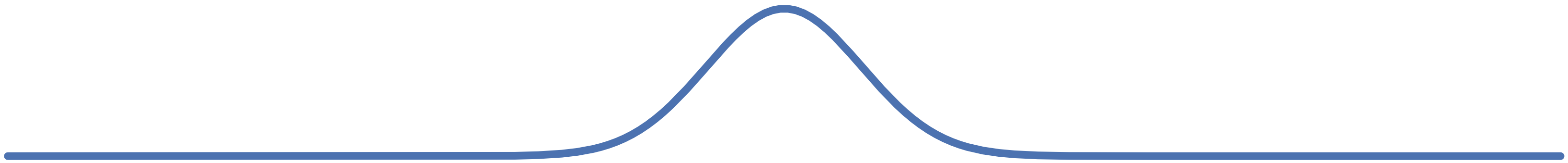
A quick check: linear kernels

- $k(x,y) = x^\mathsf{T} y$ on $\mathcal{X} = \mathbb{R}^d$
  - $k(x,\cdot) = [y \mapsto x^\mathsf{T} y]$ "corresponds to" x
- If $f(y) = \sum_{i=1}^{n} a_i k(x_i, y)$, then $f(y) = \left[\sum_{i=1}^n a_i x_i\right]^\mathsf{T} y$
- Closure doesn't add anything here, since $\mathbb{R}^d$ is closed
- So, linear kernel gives you RKHS of linear functions
- $\|f\|_\mathcal{H} = \sqrt{\sum_{i=1}^n \sum_{j=1}^n a_i a_j k(x_i, x_j)} = \|\sum_{i=1}^n a_i x_i\|$

More complicated: Gaussian kernels

$$k(x, y) = \exp(\frac{1}{2\sigma^2}\|x - y\|^2)$$

- $\mathcal{H}$ is *infinite-dimensional*



More complicated: Gaussian kernels

$$k(x, y) = \exp(\frac{1}{2\sigma^2} \|x - y\|^2)$$

- $\mathcal{H}$ is *infinite-dimensional*

More complicated: Gaussian kernels

$$k(x, y) = \exp(\frac{1}{2\sigma^2}\|x - y\|^2)$$

- $\mathcal{H}$ is *infinite-dimensional*

More complicated: Gaussian kernels

$$k(x, y) = \exp(\frac{1}{2\sigma^2}\|x - y\|^2)$$

- $\mathcal{H}$ is *infinite-dimensional*

More complicated: Gaussian kernels

$$k(x, y) = \exp(\frac{1}{2\sigma^2}\|x - y\|^2)$$

- $\mathcal{H}$ is *infinite-dimensional*

More complicated: Gaussian kernels

$$k(x,y) = \exp\left(\frac{1}{2\sigma^2}\|x-y\|^2\right)$$

- $\mathcal{H}$ is *infinite-dimensional*
- Functions in $\mathcal{H}$ are bounded:
  $f(x) = \langle f, k(x,\cdot)\rangle_{\mathcal{H}} \leq \sqrt{k(x,x)}\|f\|_{\mathcal{H}} = \|f\|_{\mathcal{H}}$

More complicated: Gaussian kernels

$$k(x, y) = \exp(\frac{1}{2\sigma^2} \|x - y\|^2)$$

- $\mathcal{H}$ is *infinite-dimensional*
- Functions in $\mathcal{H}$ are bounded:
  $f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}} \leq \sqrt{k(x, x)} \|f\|_{\mathcal{H}} = \|f\|_{\mathcal{H}}$
- Choice of σ controls how fast functions can vary:

$$f(x + t) - f(x) \leq \|k(x + t, \cdot) - k(x', \cdot)\|_{\mathcal{H}} \|f\|_{\mathcal{H}}$$

$$\|k(x + t, \cdot) - k(x, \cdot)\|_{\mathcal{H}}^2 = 2 - 2k(x, x + t) = 2 - 2\exp\left(-\frac{\|t\|^2}{2\sigma^2}\right)$$

More complicated: Gaussian kernels

$$k(x, y) = \exp(\frac{1}{2\sigma^2}\|x - y\|^2)$$

- $\mathcal{H}$ is *infinite-dimensional*
- Functions in $\mathcal{H}$ are bounded:
  $f(x) = \langle f, k(x, \cdot)\rangle_{\mathcal{H}} \leq \sqrt{k(x, x)}\|f\|_{\mathcal{H}} = \|f\|_{\mathcal{H}}$
- Choice of σ controls how fast functions can vary:

$$f(x + t) - f(x) \leq \|k(x + t, \cdot) - k(x', \cdot)\|_{\mathcal{H}}\|f\|_{\mathcal{H}}$$

$$\|k(x + t, \cdot) - k(x, \cdot)\|^2_{\mathcal{H}} = 2 - 2k(x, x + t) = 2 - 2\exp\left(-\frac{\|t\|^2}{2\sigma^2}\right)$$

- Can say lots more with Fourier properties

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

Kernel ridge regression

$$\hat{f} = \operatorname*{arg\,min}_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

Linear kernel gives normal ridge regression:

$$\hat{f}(x) = \hat{w}^\mathsf{T} x; \quad \hat{w} = \operatorname*{arg\,min}_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (w^\mathsf{T} x_i - y_i)^2 + \lambda \|w\|^2$$

Nonlinear kernels will give nonlinear regression!

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$?

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**

- Let $\mathcal{H}_X = \mathrm{span}\{k(x_i, \cdot)\}_{i=1}^{n}$
  $\mathcal{H}_\perp$ its orthogonal complement in $\mathcal{H}$

Kernel ridge regression

$$\hat{f} = \arg\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2 + \lambda\|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**

- Let $\mathcal{H}_X = \text{span}\{k(x_i, \cdot)\}_{i=1}^{n}$
  $\mathcal{H}_\perp$ its orthogonal complement in $\mathcal{H}$
- Decompose $f = f_X + f_\perp$ with $f_\mathcal{X} \in \mathcal{H}_X$, $f_\perp \in \mathcal{H}_\perp$

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**

- Let $\mathcal{H}_X = \mathrm{span}\{k(x_i, \cdot)\}_{i=1}^{n}$
  $\mathcal{H}_\perp$ its orthogonal complement in $\mathcal{H}$
- Decompose $f = f_X + f_\perp$ with $f_{\mathcal{X}} \in \mathcal{H}_X$, $f_\perp \in \mathcal{H}_\perp$
- $f(x_i) = \langle f_X + f_\perp, k(x_i, \cdot) \rangle_{\mathcal{H}} = \langle f_X, k(x_i, \cdot) \rangle_{\mathcal{H}}$

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**

- Let $\mathcal{H}_X = \text{span}\{k(x_i, \cdot)\}_{i=1}^n$
  $\mathcal{H}_\perp$ its orthogonal complement in $\mathcal{H}$
- Decompose $f = f_X + f_\perp$ with $f_{\mathcal{X}} \in \mathcal{H}_X$, $f_\perp \in \mathcal{H}_\perp$
- $f(x_i) = \langle f_X + f_\perp, k(x_i, \cdot) \rangle_{\mathcal{H}} = \langle f_X, k(x_i, \cdot) \rangle_{\mathcal{H}}$
- $\|f\|_{\mathcal{H}}^2 = \|f_X\|_{\mathcal{H}}^2 + \|f_\perp\|_{\mathcal{H}}^2$

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**

- Let $\mathcal{H}_X = \mathrm{span}\{k(x_i, \cdot)\}_{i=1}^n$
  $\mathcal{H}_\perp$ its orthogonal complement in $\mathcal{H}$
- Decompose $f = f_X + f_\perp$ with $f_\mathcal{X} \in \mathcal{H}_X$, $f_\perp \in \mathcal{H}_\perp$
- $f(x_i) = \langle f_X + f_\perp, k(x_i, \cdot) \rangle_\mathcal{H} = \langle f_X, k(x_i, \cdot) \rangle_\mathcal{H}$
- $\|f\|_\mathcal{H}^2 = \|f_X\|_\mathcal{H}^2 + \|f_\perp\|_\mathcal{H}^2$
- Minimizer needs $f_\perp = 0$, and so $\hat{f} = \sum_{i=1}^n \alpha_i k(x_i, \cdot)$

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**: $\hat{f} = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, \cdot)$

Kernel ridge regression

$$\hat{f} = \arg\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2 + \lambda\|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**: $\hat{f} = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, \cdot)$

$$\sum_{i=1}^{n}\left(\sum_{j=1}^{n}\alpha_j k(x_i, x_j) - y_i\right)^2 = \sum_{i=1}^{n}([K\alpha]_i - y_i)^2$$

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**: $\hat{f} = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, \cdot)$

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{n} \alpha_j k(x_i, x_j) - y_i \right)^2 = \sum_{i=1}^{n} ([K\alpha]_i - y_i)^2 = \|K\alpha - y\|^2$$

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**: $\hat{f} = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, \cdot)$

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{n} \alpha_j k(x_i, x_j) - y_i \right)^2 = \sum_{i=1}^{n} ([K\alpha]_i - y_i)^2 = \|K\alpha - y\|^2$$

$$= \alpha^\top K^2 \alpha - 2y^\top K\alpha + y^\top y$$

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**: $\hat{f} = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, \cdot)$

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{n} \alpha_j k(x_i, x_j) - y_i \right)^2 = \sum_{i=1}^{n} ([K\alpha]_i - y_i)^2 = \|K\alpha - y\|^2$$

$$= \alpha^\top K^2 \alpha - 2y^\top K\alpha + y^\top y$$

$$\left\| \sum_{i=1}^{n} \alpha_i k(x_i, \cdot) \right\|_{\mathcal{H}}^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i k(x_i, x_j) \alpha_j$$

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**: $\hat{f} = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, \cdot)$

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{n} \alpha_j k(x_i, x_j) - y_i \right)^2 = \sum_{i=1}^{n} ([K\alpha]_i - y_i)^2 = \|K\alpha - y\|^2$$

$$= \alpha^\top K^2 \alpha - 2y^\top K\alpha + y^\top y$$

$$\left\| \sum_{i=1}^{n} \alpha_i k(x_i, \cdot) \right\|_{\mathcal{H}}^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i k(x_i, x_j) \alpha_j = \alpha^\top K \alpha$$

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**: $\hat{f} = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, \cdot)$

$$\hat{\alpha} = \arg\min_{\alpha \in \mathbb{R}^n} \alpha^\top K^2 \alpha - 2y^\top K\alpha + y^\top y + n\lambda\alpha^\top K\alpha$$

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**: $\hat{f} = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, \cdot)$

$$\hat{\alpha} = \arg\min_{\alpha \in \mathbb{R}^n} \alpha^\top K^2 \alpha - 2y^\top K\alpha + y^\top y + n\lambda\alpha^\top K\alpha$$

$$= \arg\min_{\alpha \in \mathbb{R}^n} \alpha^\top K(K + n\lambda I)\alpha - 2y^\top K\alpha$$

Kernel ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2$$

How to find $\hat{f}$? **Representer Theorem**: $\hat{f} = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, \cdot)$

$$\begin{aligned} \hat{\alpha} &= \arg\min_{\alpha \in \mathbb{R}^n} \alpha^\top K^2 \alpha - 2y^\top K\alpha + y^\top y + n\lambda\alpha^\top K\alpha \\ &= \arg\min_{\alpha \in \mathbb{R}^n} \alpha^\top K(K + n\lambda I)\alpha - 2y^\top K\alpha \end{aligned}$$

Setting derivative to zero gives $K(K + n\lambda I)\hat{\alpha} = Ky$,
satisfied by $\hat{\alpha} = (K + n\lambda I)^{-1} y$

Other kernel algorithms

- Representer theorem applies if R is strictly increasing in

$$\min_{f \in \mathcal{H}} L(f(x_1), \cdots, f(x_n)) + R(\|f\|_{\mathcal{H}})$$

- Kernel methods can then train based on kernel matrix K
- Classification algorithms:
    - Support vector machines: L is hinge loss
    - Kernel logistic regression: L is logistic loss
- Principal component analysis, canonical correlation analysis
- Many, many more...
- But *not everything* works...e.g. Lasso $\|w\|_1$ regularizer

Some theory: generalization

- Rademacher complexity of $\{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$ is upper-bounded by $B/\sqrt{n}$ if $k(x,x) \leq 1$

Some theory: generalization

- Rademacher complexity of $\{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$ is upper-bounded by $B/\sqrt{n}$ if $k(x,x) \leq 1$
- Implies for L-Lipschitz losses $\ell(\cdot, y)$ that

$$\sup_{f:\|f\|_{\mathcal{H}} \leq B} \mathbb{E}[\ell(f(x), y)] - \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i) \leq \frac{2LB}{\sqrt{n}}$$

Some theory: generalization

- Rademacher complexity of $\{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$ is upper-bounded by $B/\sqrt{n}$ if $k(x,x) \leq 1$
- Implies for L-Lipschitz losses $\ell(\cdot, y)$ that

$$\sup_{f:\|f\|_{\mathcal{H}} \leq B} \mathbb{E}[\ell(f(x), y)] - \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i) \leq \frac{2LB}{\sqrt{n}}$$

- Same kind of rates with stability-based analyses

Some theory: generalization

- Rademacher complexity of $\{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$ is upper-bounded by $B/\sqrt{n}$ if $k(x,x) \leq 1$
- Implies for L-Lipschitz losses $\ell(\cdot, y)$ that

$$\sup_{f:\|f\|_{\mathcal{H}} \leq B} \mathbb{E}[\ell(f(x), y)] - \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i) \leq \frac{2LB}{\sqrt{n}}$$

- Same kind of rates with stability-based analyses
- Implies that, if the "truth" is low-norm, most kernel methods are $\tilde{\mathcal{O}}(1/\sqrt{n})$ suboptimal

Some theory: generalization

- Rademacher complexity of $\{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$ is upper-bounded by $B/\sqrt{n}$ if $k(x,x) \leq 1$
- Implies for L-Lipschitz losses $\ell(\cdot, y)$ that

$$\sup_{f:\|f\|_{\mathcal{H}} \leq B} \mathbb{E}[\ell(f(x), y)] - \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i) \leq \frac{2LB}{\sqrt{n}}$$

- Same kind of rates with stability-based analyses
- Implies that, if the "truth" is low-norm, most kernel methods are $\tilde{\mathcal{O}}(1/\sqrt{n})$ suboptimal
- Difficulty of learning is controlled by RKHS norm of target

Some theory: universality

- One definition: a continuous kernel on a compact metric space $\mathcal{X}$ is **universal** if $\mathcal{H}$ is L_∞-dense in $C(\mathcal{X})$:
  for every continuous $g : \mathcal{X} \to \mathbb{R}$, for every $\varepsilon > 0$, there is an $f \in \mathcal{H}$ with $\|f - g\|_\infty = \sup_{x \in \mathcal{X}} |f(x) - g(x)| \le \varepsilon$

Some theory: universality

- One definition: a continuous kernel on a compact metric space $\mathcal{X}$ is **universal** if $\mathcal{H}$ is L_∞-dense in $C(\mathcal{X})$:
  for every continuous $g : \mathcal{X} \to \mathbb{R}$, for every $\varepsilon > 0$, there is an $f \in \mathcal{H}$ with $\|f - g\|_\infty = \sup_{x \in \mathcal{X}} |f(x) - g(x)| \le \varepsilon$
- Implies that, on compact $\mathcal{X}$, $\mathcal{H}$ can separate compact sets
  - $\exists f \in \mathcal{H}$ with $f(x) > 0$ for $x \in X_1$, $f(x) < 0$ for $x \in X_2$
  - Which implies there are $f \in \mathcal{H}$ with arbitrarily small loss
  - Might take **arbitrarily large** norm: approximation/estimation tradeoff

Some theory: universality

- One definition: a continuous kernel on a compact metric space $\mathcal{X}$ is **universal** if $\mathcal{H}$ is L_∞-dense in $C(\mathcal{X})$:
  for every continuous $g : \mathcal{X} \to \mathbb{R}$, for every $\varepsilon > 0$, there is an $f \in \mathcal{H}$ with $\|f - g\|_\infty = \sup_{x \in \mathcal{X}} |f(x) - g(x)| \le \varepsilon$
- Implies that, on compact $\mathcal{X}$, $\mathcal{H}$ can separate compact sets
  - $\exists f \in \mathcal{H}$ with $f(x) > 0$ for $x \in X_1$, $f(x) < 0$ for $x \in X_2$
  - Which implies there are $f \in \mathcal{H}$ with arbitrarily small loss
  - Might take **arbitrarily large** norm: approximation/estimation tradeoff
- Can prove via Stone-Weierstrass or Fourier properties

Some theory: universality

- One definition: a continuous kernel on a compact metric space $\mathcal{X}$ is **universal** if $\mathcal{H}$ is L_∞-dense in $C(\mathcal{X})$:
  for every continuous $g : \mathcal{X} \to \mathbb{R}$, for every $\varepsilon > 0$, there is an $f \in \mathcal{H}$ with $\|f - g\|_\infty = \sup_{x \in \mathcal{X}} |f(x) - g(x)| \leq \varepsilon$
- Implies that, on compact $\mathcal{X}$, $\mathcal{H}$ can separate compact sets
  - $\exists f \in \mathcal{H}$ with $f(x) > 0$ for $x \in X_1$, $f(x) < 0$ for $x \in X_2$
  - Which implies there are $f \in \mathcal{H}$ with arbitrarily small loss
  - Might take **arbitrarily large** norm: approximation/estimation tradeoff
- Can prove via Stone-Weierstrass or Fourier properties
- Never true for finite-dim kernels: need $\mathbf{rank}(K) = n$

Translation-invariant kernels on $\mathbb{R}^d$

- Assume k is bounded, continuous, and *translation invariant*
  - $k(x, y) = \psi(x - y)$

Translation-invariant kernels on $\mathbb{R}^d$

- Assume k is bounded, continuous, and *translation invariant*
  - $k(x, y) = \psi(x - y)$
- Then ψ is proportional to the Fourier transform of a probability measure (Bochner's theorem)

Translation-invariant kernels on $\mathbb{R}^d$

- Assume k is bounded, continuous, and *translation invariant*
  - $k(x, y) = \psi(x - y)$
- Then ψ is proportional to the Fourier transform of a probability measure (Bochner's theorem)
- If $\psi \in L_1$, the measure has a density

Translation-invariant kernels on $\mathbb{R}^d$

- Assume k is bounded, continuous, and *translation invariant*
  - $k(x, y) = \psi(x - y)$
- Then ψ is proportional to the Fourier transform of a probability measure (Bochner's theorem)
- If $\psi \in L_1$, the measure has a density
- If that density is positive everywhere, k is universal

Translation-invariant kernels on $\mathbb{R}^d$

- Assume k is bounded, continuous, and *translation invariant*
  - $k(x, y) = \psi(x - y)$
- Then ψ is proportional to the Fourier transform of a probability measure (Bochner's theorem)
- If $\psi \in L_1$, the measure has a density
- If that density is positive everywhere, k is universal
- For all nonzero finite signed measures μ, $\int_{\mathcal{X}} \int_{\mathcal{X}} k(x, y) \, \mathrm{d}\mu(x) \, \mathrm{d}\mu(y) > 0$

Translation-invariant kernels on $\mathbb{R}^d$

- Assume k is bounded, continuous, and *translation invariant*
  - $k(x, y) = \psi(x - y)$
- Then ψ is proportional to the Fourier transform of a probability measure (Bochner's theorem)
- If $\psi \in L_1$, the measure has a density
- If that density is positive everywhere, k is universal
- For all nonzero finite signed measures μ, $\int_{\mathcal{X}} \int_{\mathcal{X}} k(x, y) \, \mathrm{d}\mu(x) \, \mathrm{d}\mu(y) > 0$
- True for Gaussian $\exp(-\frac{1}{2\sigma^2} \|x - y\|^2)$

Translation-invariant kernels on $\mathbb{R}^d$

- Assume k is bounded, continuous, and *translation invariant*
  - $k(x, y) = \psi(x - y)$
- Then ψ is proportional to the Fourier transform of a probability measure (Bochner's theorem)
- If $\psi \in L_1$, the measure has a density
- If that density is positive everywhere, k is universal
- For all nonzero finite signed measures μ, $\int_{\mathcal{X}} \int_{\mathcal{X}} k(x, y) \, \mathrm{d}\mu(x) \, \mathrm{d}\mu(y) > 0$
- True for Gaussian $\exp(-\frac{1}{2\sigma^2} \|x - y\|^2)$
- and Laplace $\exp(-\frac{1}{\sigma} \|x - y\|)$

Limitations of kernel-based learning

- Generally bad at learning *sparsity*
  - e.g. $f(x_1, \ldots, x_d) = 3x_2 - 5x_{17}$ for large d

Limitations of kernel-based learning

- Generally bad at learning *sparsity*
  - e.g. $f(x_1, \ldots, x_d) = 3x_2 - 5x_{17}$ for large d
- Provably statistically slower than deep learning for a few problems
  - e.g. to learn a single ReLU, $\max(0, w^\top x)$, need norm exponential in d [Yehudai/Shamir NeurIPS-19]
  - Also some hierarchical problems, etc [Kamath+ COLT-20]

Limitations of kernel-based learning

- Generally bad at learning *sparsity*
  - e.g. $f(x_1, \ldots, x_d) = 3x_2 - 5x_{17}$ for large d
- Provably statistically slower than deep learning for a few problems
  - e.g. to learn a single ReLU, $\max(0, w^\mathsf{T} x)$, need norm exponential in d [Yehudai/Shamir NeurIPS-19]
  - Also some hierarchical problems, etc [Kamath+ COLT-20]
- $\mathcal{O}(n^3)$ computational complexity, $\mathcal{O}(n^2)$ memory
  - Various approximations you can make

Relationship to deep learning

- Deep models usually end as $f_L(x) = w_L^\top f_{L-1}(x)$

Relationship to deep learning

- Deep models usually end as $f_L(x) = w_L^\mathsf{T} f_{L-1}(x)$
- Can think of as *learned* kernel, $k(x, y) = f_{L-1}(x) f_{L-1}(y)$

Relationship to deep learning

- Deep models usually end as $f_L(x) = w_L^\intercal f_{L-1}(x)$
- Can think of as *learned* kernel, $k(x, y) = f_{L-1}(x) f_{L-1}(y)$
- Does this gain us anything?

Relationship to deep learning

- Deep models usually end as $f_L(x) = w_L^\mathsf{T} f_{L-1}(x)$
- Can think of as *learned* kernel, $k(x, y) = f_{L-1}(x) f_{L-1}(y)$
- Does this gain us anything?
  - Random nets with trained last layer (NNGP) can be decent

Relationship to deep learning

- Deep models usually end as $f_L(x) = w_L^\mathsf{T} f_{L-1}(x)$
- Can think of as *learned* kernel, $k(x, y) = f_{L-1}(x) f_{L-1}(y)$
- Does this gain us anything?
  - Random nets with trained last layer (NNGP) can be decent
  - As width $\to \infty$, nets become neural tangent kernel
    - Widely used theoretical analysis...more tomorrow
    - SVMs with NTK can be great on small data

Relationship to deep learning

- Deep models usually end as $f_L(x) = w_L^\mathsf{T} f_{L-1}(x)$
- Can think of as *learned* kernel, $k(x, y) = f_{L-1}(x) f_{L-1}(y)$
- Does this gain us anything?
  - Random nets with trained last layer (NNGP) can be decent
  - As width $\to \infty$, nets become neural tangent kernel
    - Widely used theoretical analysis...more tomorrow
    - SVMs with NTK can be great on small data
  - Inspiration: learn the kernel model end-to-end
    - Ongoing area; good results in two-sample testing, GANs, density estimation, meta-learning, semi-supervised learning, ...
    - Explored a bit in interactive session!

What's next

- After break: interactive session exploring w/ ridge regression
- Tomorrow: a subset of
  - Representing distributions
    - Uses for statistical testing + generative models
  - Connections to Gaussian processes, probabilistic numerics
  - Approximation methods for faster computation
  - Deeper connection to deep learning
- More details on basics:
  - Berlinet and Thomas-Agnan, *RKHS in Probability and Statistics*
  - Steinwart and Christmann, *Support Vector Machines*