(Deep) Kernel Mean Embeddings for Representing and Learning on Distributions

Danica J. Sutherland (she/her)
University of British Columbia + Amii
Lifting Inference with Kernel Embeddings (LIKE-23), June 2023

This talk: how to lift inference with kernel embeddings

HTML version at `djsutherland.ml/slides/like23`

(Deep) Kernel Mean Embeddings for Representing and Learning on Distributions

Danica J. Sutherland (she/her)
University of British Columbia + Amii

Lifting Inference with Kernel Embeddings (LIKE-23), June 2023

This talk: how to lift inference with kernel embeddings

HTML version at `djsutherland.ml/slides/like23`

(Deep) Kernel Mean Embeddings for Representing and Learning on Distributions

Danica J. Sutherland (she/her)

University of British Columbia + Amii

Lifting Inference with Kernel Embeddings (LIKE-23), June 2023

This talk: how to lift inference with kernel embeddings

HTML version at `djsutherland.ml/slides/like23`

(Deep) Kernel Mean Embeddings for Representing and Learning on Distributions

Danica J. Sutherland (she/her)
University of British Columbia + Amii

Lifting Inference with Kernel Embeddings (LIKE-23), June 2023

This talk: how to lift inference with kernel embeddings

HTML version at `djsutherland.ml/slides/like23`

Part I: Kernels

Why kernels?

- Machine learning!

Why kernels?

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

Why kernels?

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

Why kernels?

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

Why kernels?

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

Why kernels?

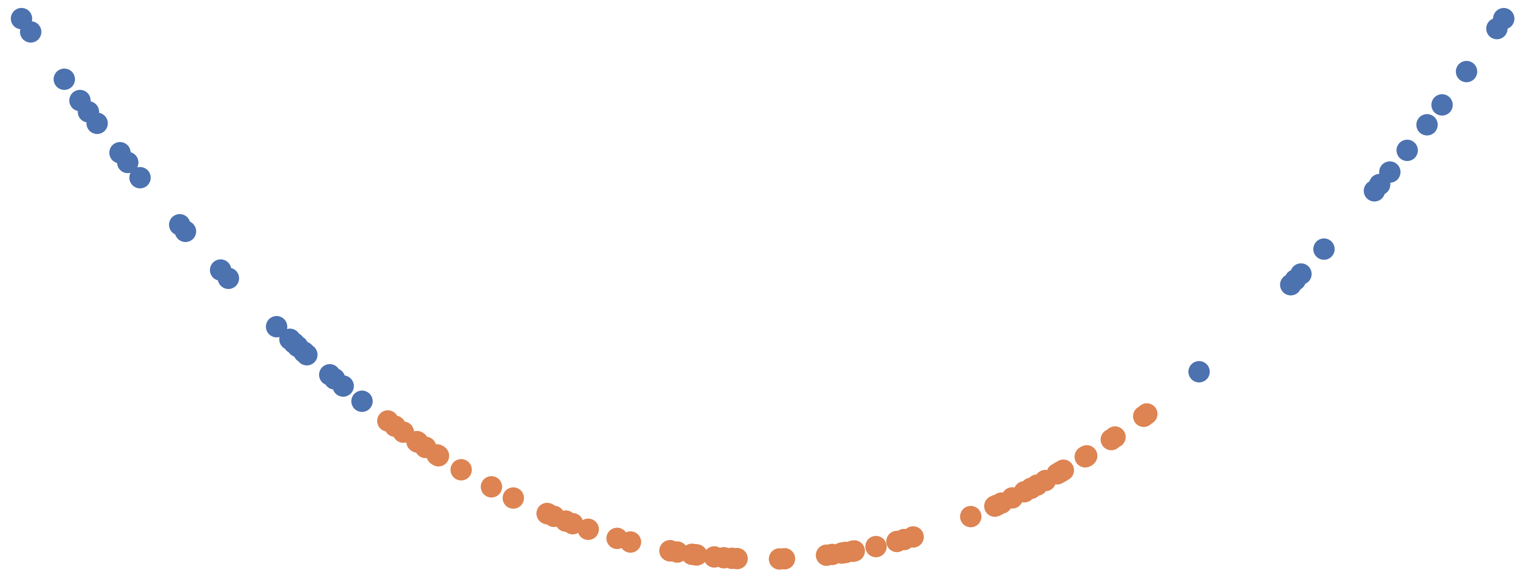
- Machine learning! ...but how do we actually do it?
- Linear models! $f(x) = w_0 + wx$, $\hat{y}(x) = \textbf{sign}(f(x))$
- Extend x...

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

Why kernels?

- 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...

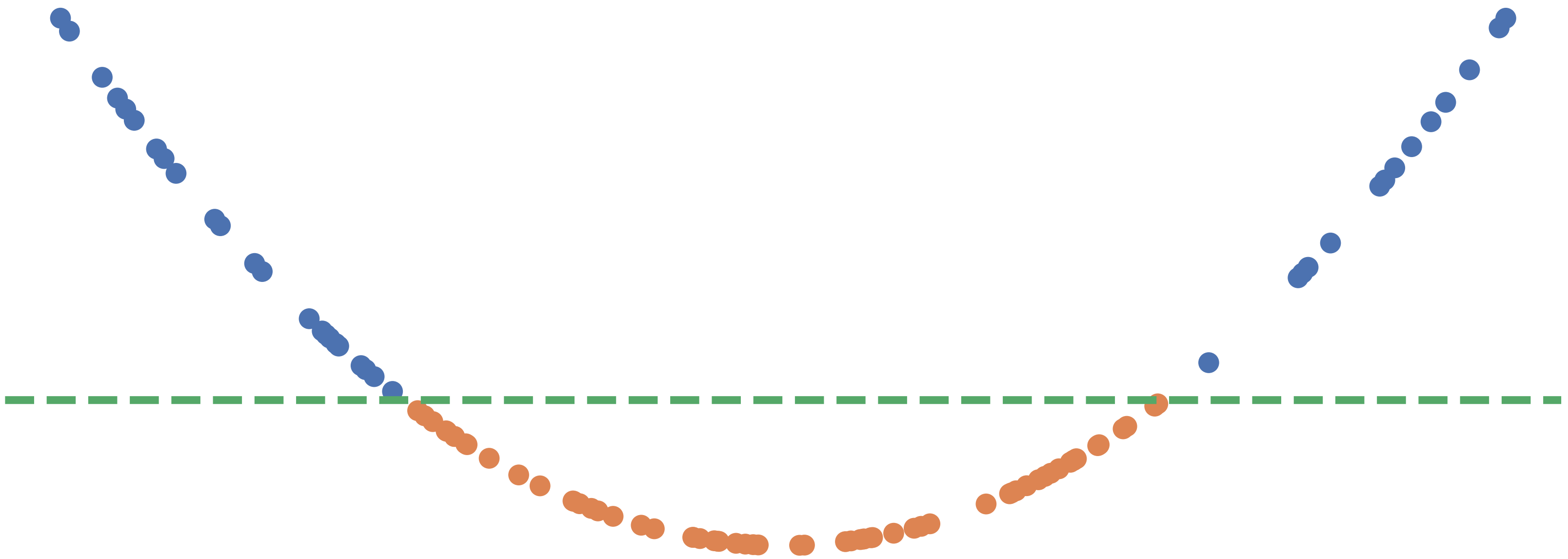
$$f(x) = w^{\mathsf{T}}(1, x, x^2) = w^{\mathsf{T}}\phi(x)$$



Why kernels?

- 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...

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



Why kernels?

- 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

Why kernels?

- 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,
  probability distributions, …

Why kernels?

- 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, probability distributions, ...
- 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"
- Exactly the same: GP covariance function

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Exactly the same: GP covariance function
- Semi-related: kernel density estimation

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Exactly the same: GP covariance function
- 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"
- Exactly the same: GP covariance function
- 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"
- Exactly the same: GP covariance function
- 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"
- Exactly the same: GP covariance function
- 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"
- Exactly the same: GP covariance function
- Semi-related: kernel density estimation
- Unrelated:

Aside: the name “kernel”

- Our concept: "positive semi-definite kernel," "Mercer kernel," "RKHS kernel"
- Exactly the same: GP covariance function
- 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"
- Exactly the same: GP covariance function
- 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"
- Exactly the same: GP covariance function
- 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"
- Exactly the same: GP covariance function
- 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"
- Exactly the same: GP covariance function
- 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"
- Exactly the same: GP covariance function
- 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 $k(x, x) = \|\phi(x)\|^2_{\mathcal{H}} \geq 0$.

Positive definiteness

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

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

- Equivalent: $n \times n$ <u>kernel matrix</u> K is psd (eigenvalues ≥ 0)

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

Positive definiteness

- A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ i.e. $k(x, y) = k(y, x)$
  is *positive semi-definite*
  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. $k(x, y) = k(y, x)$
  is *positive semi-definite*
  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. $k(x, y) = k(y, x)$
  is *positive semi-definite*
  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. $k(x, y) = k(y, x)$
  is *positive semi-definite*
  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

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

$$= \left\| \sum_{i=1}^{n} a_i \phi(x_i) \right\|_{\mathcal{H}}^2$$

Positive definiteness

- A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ i.e. $k(x, y) = k(y, x)$
  is *positive semi-definite*
  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

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

$$= \left\| \sum_{i=1}^{n} a_i \phi(x_i) \right\|_{\mathcal{H}}^2 \geq 0$$

Positive definiteness

- A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ i.e. $k(x, y) = k(y, x)$
  is *positive semi-definite*
  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. $k(x, y) = k(y, x)$
  is *positive semi-definite*
  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
  - $\displaystyle \lim_{m\to\infty} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k_m(x_i, x_j) \geq 0$

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
  - $\operatorname{Cov}(V_i W_i, V_j W_j) = \operatorname{Cov}(V_i, V_j) \operatorname{Cov}(W_i, W_j) = k_\times(x_i, x_j)$
  - 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 $f : X \to \mathbb{R}$, $k_f(x,y) = f(x)k(x,y)f(y)$ 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 $f : X \to \mathbb{R}$, $k_f(x, y) = f(x)k(x, y)f(y)$ 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 $f : X \to \mathbb{R}$, $k_f(x,y) = f(x)k(x,y)f(y)$ 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 $f : X \to \mathbb{R}$, $k_f(x, y) = f(x)k(x, y)f(y)$ 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

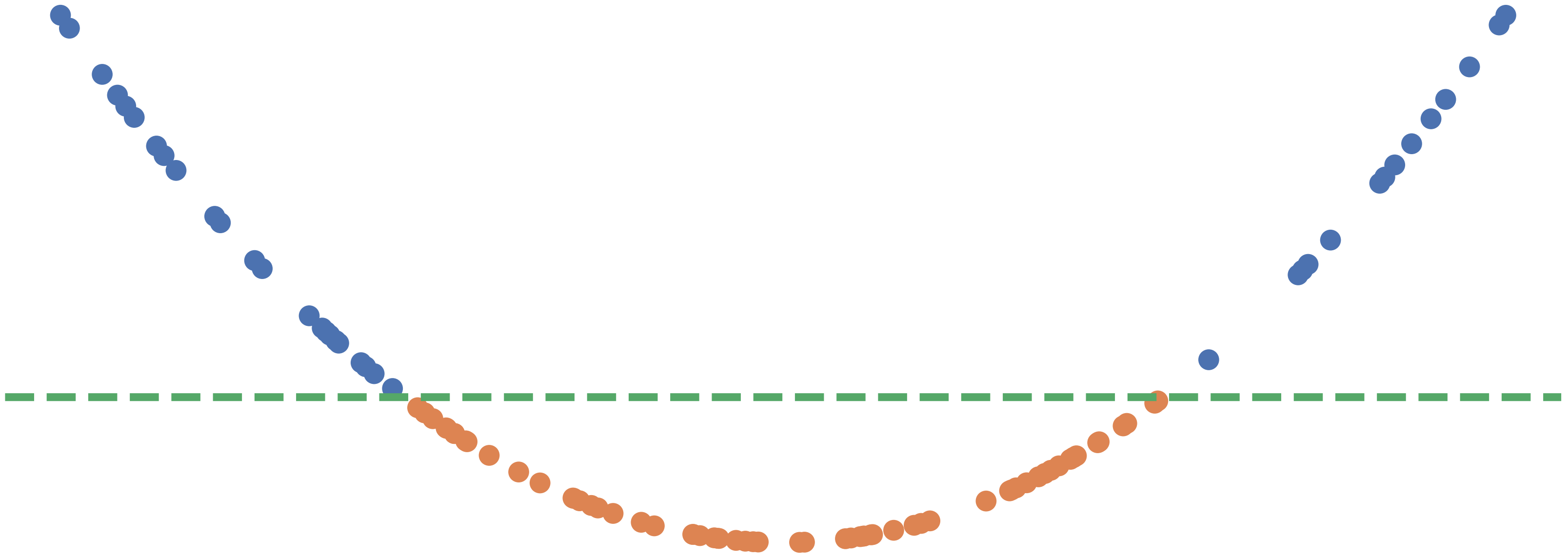
- Recall original motivating example with

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

Reproducing property

- Recall original motivating example with

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

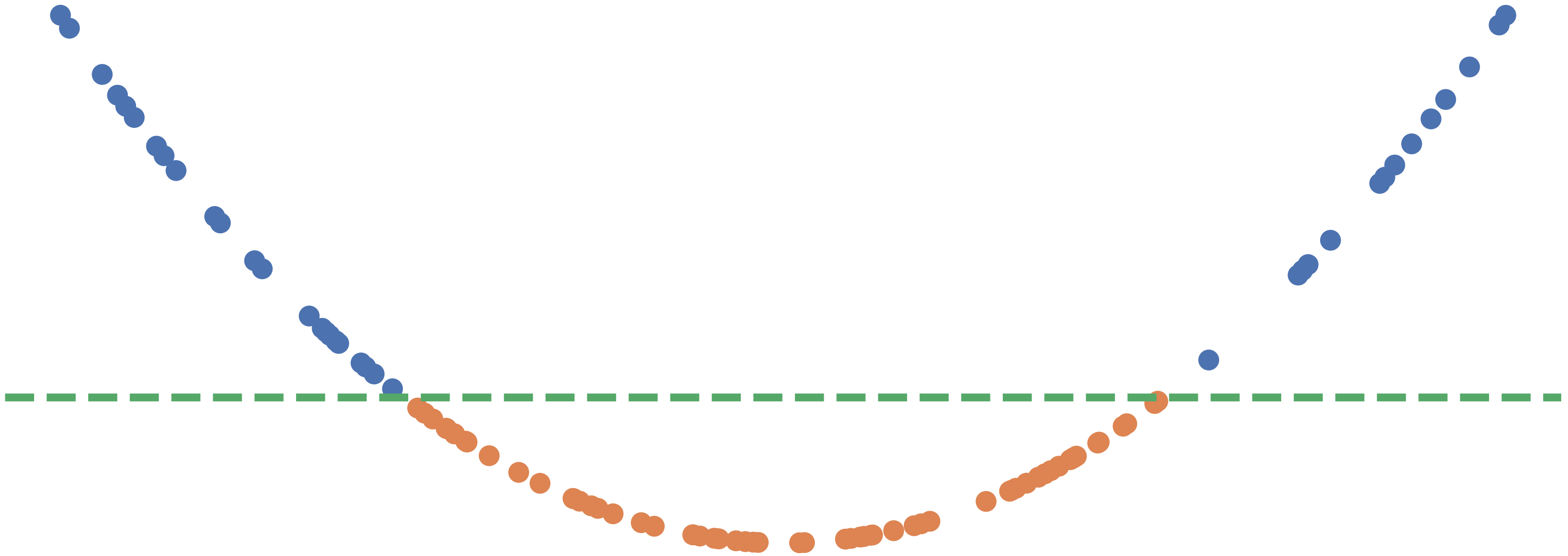


Reproducing property

- Recall original motivating example with

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

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

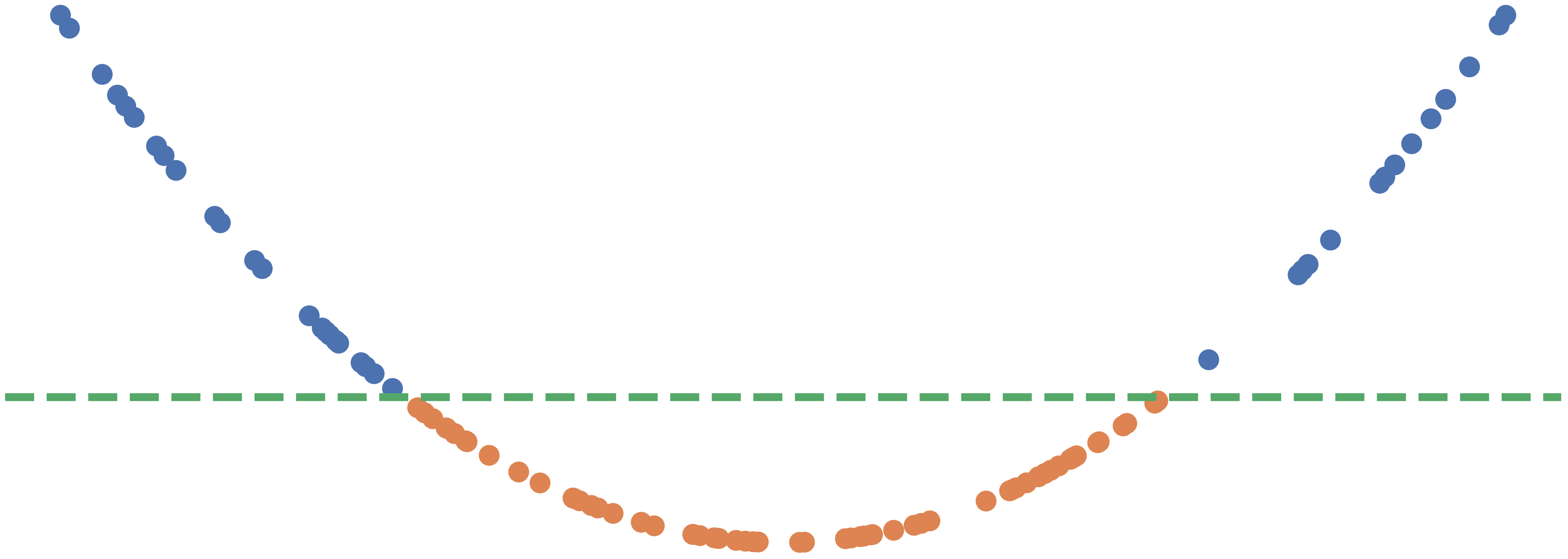


Reproducing property

- 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 $k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}} = 1 + xy + x^2 y^2$
- 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
- 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 property**: $f(x) = \langle f(\cdot), \phi(x) \rangle_{\mathcal{H}}$ 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 $f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}}$

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

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

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 $\mathcal{H}_0 = \textbf{span}(\{k(x, \cdot) : x \in \mathcal{X}\})$
  - 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 $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$

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 $\mathcal{H}$

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 $\mathcal{H}_0 = \mathbf{span}(\{k(x,\cdot) : x \in \mathcal{X}\})$
  - 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 $\mathcal{H}$
  - 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 $\mathcal{X} = \mathbb{R}^d$

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

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 $f(y) = \displaystyle\sum_{i=1}^{n} a_i k(x_i, y)$, then $f(y) = \left[\sum_{i=1}^{n} a_i x_i\right]^\mathsf{T} y$

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 $f(y) = \displaystyle\sum_{i=1}^{n} a_i k(x_i, y)$, 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

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

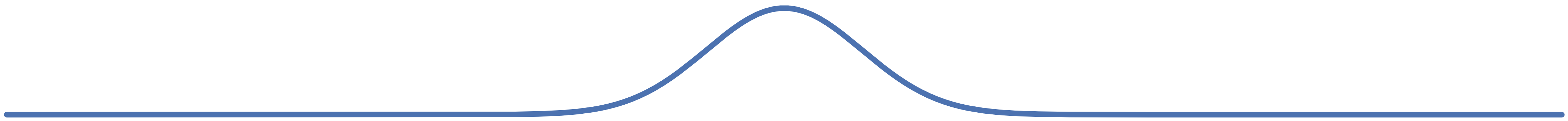
A quick check: linear kernels

- $k(x, y) = x^\mathsf{T} y$ 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

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

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

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:

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

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

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:
  $f(x) = \langle f, k(x, \cdot)\rangle_{\mathcal{H}} \le \sqrt{k(x, x)}\|f\|_{\mathcal{H}} = \|f\|_{\mathcal{H}}$
- Choice of σ controls how fast functions can vary:

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

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

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

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

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

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 $\mathcal{H}_X = \mathrm{span}\{k(x_i, \cdot)\}_{i=1}^n$, and $\mathcal{H}_\perp$ its **orthogonal complement** in $\mathcal{H}$
- Decompose $f = f_X + f_\perp$ with $f_X \in \mathcal{H}_X$, $f_\perp \in \mathcal{H}_\perp$
- $f(x_i) = \langle f_X + f_\perp, k(x_i, \cdot)\rangle_{\mathcal{H}} = \langle f_X, k(x_i, \cdot)\rangle_{\mathcal{H}}$
- $\|f\|_{\mathcal{H}}^2 = \|f_X\|_{\mathcal{H}}^2 + \|f_\perp\|_{\mathcal{H}}^2$
- 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**: $\hat{f} = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, \cdot)$

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

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**: $\hat{f} = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, \cdot)$

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

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

Kernel ridge regression and GP regression

- Compare to regression with $\mathcal{GP}(0, k)$ prior, $\mathcal{N}(0, \sigma^2)$ observation noise

Kernel ridge regression and GP regression

- Compare to regression with $\mathcal{GP}(0, k)$ prior, $\mathcal{N}(0, \sigma^2)$ observation noise
- If we take $\lambda = \sigma^2 / n$, KRR is exactly the GP regression posterior mean

Kernel ridge regression and GP regression

- Compare to regression with $\mathcal{GP}(0, k)$ prior, $\mathcal{N}(0, \sigma^2)$ observation noise
- If we take $\lambda = \sigma^2/n$, KRR is exactly the GP regression posterior mean
- Note that GP posterior samples **are not** in $\mathcal{H}$, but are in a slightly bigger RKHS

Kernel ridge regression and GP regression

- Compare to regression with $\mathcal{GP}(0, k)$ prior, $\mathcal{N}(0, \sigma^2)$ observation noise
- If we take $\lambda = \sigma^2/n$, KRR is exactly the GP regression posterior mean
- Note that GP posterior samples **are not** in $\mathcal{H}$, but are in a slightly bigger RKHS
- Also a connection between posterior variance and KRR worst-case error

Kernel ridge regression and GP regression

- Compare to regression with $\mathcal{GP}(0, k)$ prior, $\mathcal{N}(0, \sigma^2)$ observation noise
- If we take $\lambda = \sigma^2 / n$, KRR is exactly the GP regression posterior mean
- Note that GP posterior samples **are not** in $\mathcal{H}$, but are in a slightly bigger RKHS
- Also a connection between posterior variance and KRR worst-case error
- For many more details:

Gaussian Processes and Kernel Methods:
A Review on Connections and Equivalences

Motonobu Kanagawa[1], Philipp Hennig[1],
Dino Sejdinovic[2], and Bharath K Sriperumbudur[3]

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

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

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

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…

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 very very quick theory

- Generalization: how close is my training set error to the population error?

Some very very quick theory

- Generalization: how close is my training set error to the population error?
  - Say $k(x, x) \leq 1$, consider $\{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$, ρ-Lipschitz loss

Some very very quick theory

- Generalization: how close is my training set error to the population error?
  - Say $k(x, x) \leq 1$, consider $\{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$, ρ-Lipschitz loss
  - Rademacher argument implies expected overfitting $\leq \frac{2\rho B}{\sqrt{n}}$

Some very very quick theory

- Generalization: how close is my training set error to the population error?
  - Say $k(x,x) \leq 1$, consider $\{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$, ρ-Lipschitz loss
  - Rademacher argument implies expected overfitting $\leq \dfrac{2\rho B}{\sqrt{n}}$
  - If "truth" has low RKHS norm, can learn efficiently

Some very very quick theory

- Generalization: how close is my training set error to the population error?
  - Say $k(x, x) \leq 1$, consider $\{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$, ρ-Lipschitz loss
  - Rademacher argument implies expected overfitting $\leq \frac{2\rho B}{\sqrt{n}}$
  - If "truth" has low RKHS norm, can learn efficiently
- Approximation: how big is RKHS norm of target function?

Some very very quick theory

- Generalization: how close is my training set error to the population error?
  - Say $k(x, x) \leq 1$, consider $\{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$, ρ-Lipschitz loss
  - Rademacher argument implies expected overfitting $\leq \frac{2\rho B}{\sqrt{n}}$
  - If “truth” has low RKHS norm, can learn efficiently
- Approximation: how big is RKHS norm of target function?
  - For *universal* kernels, can approximate any target with finite norm

Some very very quick theory

- Generalization: how close is my training set error to the population error?
  - Say $k(x, x) \leq 1$, consider $\{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$, ρ-Lipschitz loss
  - Rademacher argument implies expected overfitting $\leq \frac{2\rho B}{\sqrt{n}}$
  - If "truth" has low RKHS norm, can learn efficiently
- Approximation: how big is RKHS norm of target function?
  - For *universal* kernels, can approximate any target with finite norm
  - Gaussian is universal 💪

Some very very quick theory

- Generalization: how close is my training set error to the population error?
  - Say $k(x,x) \leq 1$, consider $\{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$, ρ-Lipschitz loss
  - Rademacher argument implies expected overfitting $\leq \frac{2\rho B}{\sqrt{n}}$
  - If "truth" has low RKHS norm, can learn efficiently
- Approximation: how big is RKHS norm of target function?
  - For *universal* kernels, can approximate any target with finite norm
  - Gaussian is universal 💪 (nothing finite-dimensional can be)

Some very very quick theory

- Generalization: how close is my training set error to the population error?
  - Say $k(x, x) \leq 1$, consider $\{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$, ρ-Lipschitz loss
  - Rademacher argument implies expected overfitting $\leq \frac{2\rho B}{\sqrt{n}}$
  - If "truth" has low RKHS norm, can learn efficiently
- Approximation: how big is RKHS norm of target function?
  - For *universal* kernels, can approximate any target with finite norm
  - Gaussian is universal 💪 (nothing finite-dimensional can be)
  - But "finite" can be *really really really* big

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, $\max(0, w^\mathsf{T} x)$, 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]
  - Generally apply to learning with *any fixed kernel*

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]
  - Generally apply to learning with *any fixed kernel*
- $\mathcal{O}(n^3)$ computational complexity, $\mathcal{O}(n^2)$ memory
  - Various approximations you can make

Part II: (Deep) Kernel Mean Embeddings

Mean embeddings of distributions

- Represent point $x \in \mathcal{X}$ as $k(x, \cdot)$: $\quad f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}}$

Mean embeddings of distributions

- Represent point $x \in \mathcal{X}$ as $k(x, \cdot)$: $\quad f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}}$
- Represent *distribution* $\mathbb{P}$ as $\mu_{\mathbb{P}}$: $\quad \mathbb{E}_{X \sim \mathbb{P}}\, f(X) = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$

Mean embeddings of distributions

- Represent point $x \in \mathcal{X}$ as $k(x, \cdot)$: $\quad f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}}$
- Represent *distribution* $\mathbb{P}$ as $\mu_{\mathbb{P}}$: $\quad \mathbb{E}_{X \sim \mathbb{P}}\, f(X) = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$

$$\mathbb{E}_{X \sim \mathbb{P}}\, f(X) = \mathbb{E}_{X \sim \mathbb{P}} \langle f, k(X, \cdot) \rangle_{\mathcal{H}}$$

Mean embeddings of distributions

- Represent point $x \in \mathcal{X}$ as $k(x, \cdot)$: $\quad f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}}$
- Represent *distribution* $\mathbb{P}$ as $\mu_{\mathbb{P}}$: $\quad \mathbb{E}_{X \sim \mathbb{P}} f(X) = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$

$$\mathbb{E}_{X \sim \mathbb{P}} f(X) = \mathbb{E}_{X \sim \mathbb{P}} \langle f, k(X, \cdot) \rangle_{\mathcal{H}} = \langle f, \underbrace{\mathbb{E}_{X \sim \mathbb{P}} k(X, \cdot)}_{\mu_{\mathbb{P}}} \rangle_{\mathcal{H}}$$

Mean embeddings of distributions

- Represent point $x \in \mathcal{X}$ as $k(x, \cdot)$: $\quad f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}}$
- Represent *distribution* $\mathbb{P}$ as $\mu_{\mathbb{P}}$: $\quad \mathbb{E}_{X \sim \mathbb{P}}\, f(X) = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$

$$\mathbb{E}_{X \sim \mathbb{P}}\, f(X) = \mathbb{E}_{X \sim \mathbb{P}} \langle f, k(X, \cdot) \rangle_{\mathcal{H}} = \langle f, \underbrace{\mathbb{E}_{X \sim \mathbb{P}}\, k(X, \cdot)}_{\mu_{\mathbb{P}}} \rangle_{\mathcal{H}}$$

  - Last step assumed $\mathbb{E} \sqrt{k(X, X)} < \infty$ *(Bochner integrability)*

Mean embeddings of distributions

- Represent point $x \in \mathcal{X}$ as $k(x, \cdot)$: $\quad f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}}$
- Represent *distribution* $\mathbb{P}$ as $\mu_{\mathbb{P}}$: $\quad \mathbb{E}_{X \sim \mathbb{P}}\, f(X) = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$

$$\mathbb{E}_{X \sim \mathbb{P}}\, f(X) = \mathbb{E}_{X \sim \mathbb{P}} \langle f, k(X, \cdot) \rangle_{\mathcal{H}} = \langle f, \underbrace{\mathbb{E}_{X \sim \mathbb{P}}\, k(X, \cdot)}_{\mu_{\mathbb{P}}} \rangle_{\mathcal{H}}$$

  - Last step assumed $\mathbb{E}\sqrt{k(X, X)} < \infty$ *(Bochner integrability)*
- $\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} = \mathbb{E}_{X \sim \mathbb{P}, Y \sim \mathbb{Q}}\, k(X, Y)$

Mean embeddings of distributions

- Represent point $x \in \mathcal{X}$ as $k(x, \cdot)$: $\quad f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}}$
- Represent *distribution* $\mathbb{P}$ as $\mu_{\mathbb{P}}$: $\quad \mathbb{E}_{X\sim\mathbb{P}}\, f(X) = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$

$$\mathbb{E}_{X\sim\mathbb{P}}\, f(X) = \mathbb{E}_{X\sim\mathbb{P}} \langle f, k(X, \cdot) \rangle_{\mathcal{H}} = \langle f, \underbrace{\mathbb{E}_{X\sim\mathbb{P}}\, k(X, \cdot)}_{\mu_{\mathbb{P}}} \rangle_{\mathcal{H}}$$

  - Last step assumed $\mathbb{E} \sqrt{k(X, X)} < \infty$ *(Bochner integrability)*

- $\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} = \mathbb{E}_{X\sim\mathbb{P}, Y\sim\mathbb{Q}}\, k(X, Y)$
- Okay. Why?

Mean embeddings of distributions

- Represent point $x \in \mathcal{X}$ as $k(x, \cdot)$: $\quad f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}}$
- Represent *distribution* $\mathbb{P}$ as $\mu_{\mathbb{P}}$: $\quad \mathbb{E}_{X \sim \mathbb{P}}\, f(X) = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$

$$\mathbb{E}_{X \sim \mathbb{P}}\, f(X) = \mathbb{E}_{X \sim \mathbb{P}} \langle f, k(X, \cdot) \rangle_{\mathcal{H}} = \langle f, \underbrace{\mathbb{E}_{X \sim \mathbb{P}}\, k(X, \cdot)}_{\mu_{\mathbb{P}}} \rangle_{\mathcal{H}}$$

  - Last step assumed $\mathbb{E} \sqrt{k(X, X)} < \infty$ *(Bochner integrability)*
- $\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} = \mathbb{E}_{X \sim \mathbb{P}, Y \sim \mathbb{Q}}\, k(X, Y)$
- Okay. Why?
  - One reason: ML on distributions [Szabó+ JMLR-16]

Mean embeddings of distributions

- Represent point $x \in \mathcal{X}$ as $k(x, \cdot)$: $\quad f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}}$
- Represent *distribution* $\mathbb{P}$ as $\mu_{\mathbb{P}}$: $\quad \mathbb{E}_{X \sim \mathbb{P}} f(X) = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$

$$\mathbb{E}_{X \sim \mathbb{P}} f(X) = \mathbb{E}_{X \sim \mathbb{P}} \langle f, k(X, \cdot) \rangle_{\mathcal{H}} = \langle f, \underbrace{\mathbb{E}_{X \sim \mathbb{P}} k(X, \cdot)}_{\mu_{\mathbb{P}}} \rangle_{\mathcal{H}}$$

  - Last step assumed $\mathbb{E} \sqrt{k(X, X)} < \infty$ *(Bochner integrability)*
- $\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} = \mathbb{E}_{X \sim \mathbb{P}, Y \sim \mathbb{Q}} k(X, Y)$
- Okay. Why?
  - One reason: ML on distributions [Szabó+ JMLR-16]
  - More common reason: comparing distributions

Maximum Mean Discrepancy

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}$$

Maximum Mean Discrepancy

$$\begin{aligned}\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}} \leq 1} \langle f, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}} \rangle_{\mathcal{H}}\end{aligned}$$

Maximum Mean Discrepancy

$$
\begin{aligned}
\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} \\
&= \sup_{\|f\|_{\mathcal{H}} \leq 1} \langle f, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \\
&= \sup_{\|f\|_{\mathcal{H}} \leq 1} \mathbb{E}_{X \sim \mathbb{P}}\, f(X) - \mathbb{E}_{Y \sim \mathbb{Q}}\, f(Y)
\end{aligned}
$$

- Last line is Integral Probability Metric (IPM) form

Maximum Mean Discrepancy

$$\begin{aligned}\mathrm{MMD}(\mathbb{P},\mathbb{Q}) &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}}\le 1} \langle f, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\rangle_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}}\le 1} \mathbb{E}_{X\sim\mathbb{P}}\, f(X) - \mathbb{E}_{Y\sim\mathbb{Q}}\, f(Y)\end{aligned}$$

- Last line is Integral Probability Metric (IPM) form
- f is called “witness function” or “critic”: high on $\mathbb{P}$, low on $\mathbb{Q}$

$$f^*(t) \propto \langle \mu_{\mathbb{P}} - \mu_{\mathbb{Q}}, k(t,\cdot)\rangle_{\mathcal{H}} = \mathbb{E}_{\mathbb{P}}\, k(t,X) - \mathbb{E}_{\mathbb{Q}}\, k(t,Y)$$

Maximum Mean Discrepancy

$$\begin{aligned}\mathrm{MMD}(\mathbb{P},\mathbb{Q}) &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}}\le 1} \langle f, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\rangle_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}}\le 1} \mathbb{E}_{X\sim\mathbb{P}}\, f(X) - \mathbb{E}_{Y\sim\mathbb{Q}}\, f(Y)\end{aligned}$$

- Last line is Integral Probability Metric (IPM) form
- f is called “witness function” or “critic”: high on $\mathbb{P}$, low on $\mathbb{Q}$

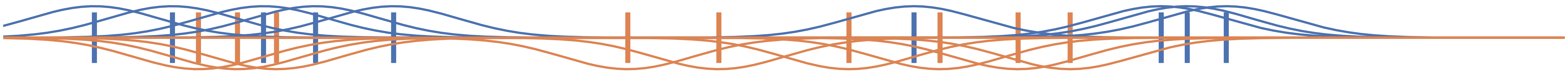
$$f^*(t) \propto \langle \mu_{\mathbb{P}} - \mu_{\mathbb{Q}}, k(t,\cdot)\rangle_{\mathcal{H}} = \mathbb{E}_{\mathbb{P}}\, k(t,X) - \mathbb{E}_{\mathbb{Q}}\, k(t,Y)$$

Maximum Mean Discrepancy

$$\begin{aligned} \mathrm{MMD}(\mathbb{P}, \mathbb{Q}) &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}} \le 1} \langle f, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}} \le 1} \mathbb{E}_{X \sim \mathbb{P}} f(X) - \mathbb{E}_{Y \sim \mathbb{Q}} f(Y) \end{aligned}$$

- Last line is Integral Probability Metric (IPM) form
- f is called “witness function” or “critic”: high on $\mathbb{P}$, low on $\mathbb{Q}$

$$f^*(t) \propto \langle \mu_{\mathbb{P}} - \mu_{\mathbb{Q}}, k(t, \cdot) \rangle_{\mathcal{H}} = \mathbb{E}_{\mathbb{P}}\, k(t, X) - \mathbb{E}_{\mathbb{Q}}\, k(t, Y)$$

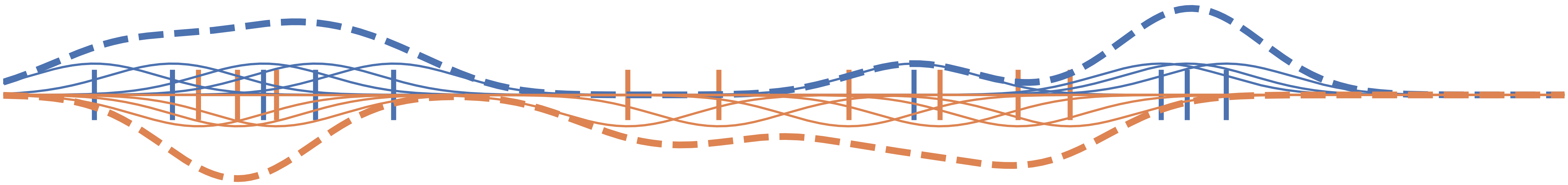


Maximum Mean Discrepancy

$$\begin{aligned}\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}} \le 1} \langle f, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}} \le 1} \mathbb{E}_{X \sim \mathbb{P}}\, f(X) - \mathbb{E}_{Y \sim \mathbb{Q}}\, f(Y)\end{aligned}$$

- Last line is Integral Probability Metric (IPM) form
- f is called "witness function" or "critic": high on $\mathbb{P}$, low on $\mathbb{Q}$

$$f^*(t) \propto \langle \mu_{\mathbb{P}} - \mu_{\mathbb{Q}}, k(t, \cdot) \rangle_{\mathcal{H}} = \mathbb{E}_{\mathbb{P}}\, k(t, X) - \mathbb{E}_{\mathbb{Q}}\, k(t, Y)$$

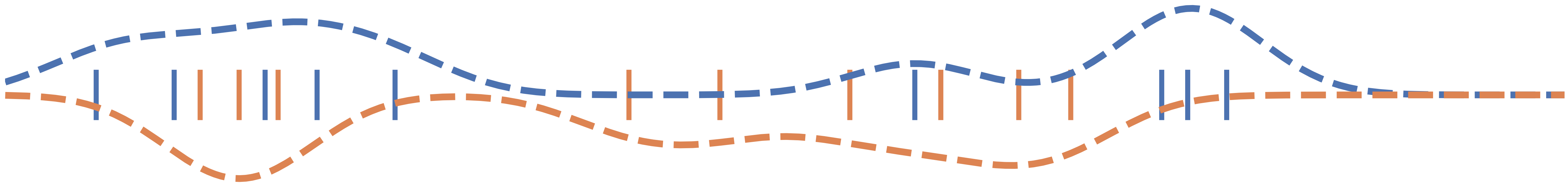


Maximum Mean Discrepancy

$$\begin{aligned}\mathrm{MMD}(\mathbb{P},\mathbb{Q}) &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}}\le 1} \langle f, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\rangle_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}}\le 1} \mathbb{E}_{X\sim\mathbb{P}}\, f(X) - \mathbb{E}_{Y\sim\mathbb{Q}}\, f(Y)\end{aligned}$$

- Last line is Integral Probability Metric (IPM) form
- f is called “witness function” or “critic”: high on $\mathbb{P}$, low on $\mathbb{Q}$

$$f^*(t) \propto \langle \mu_{\mathbb{P}} - \mu_{\mathbb{Q}}, k(t,\cdot)\rangle_{\mathcal{H}} = \mathbb{E}_{\mathbb{P}}\, k(t,X) - \mathbb{E}_{\mathbb{Q}}\, k(t,Y)$$

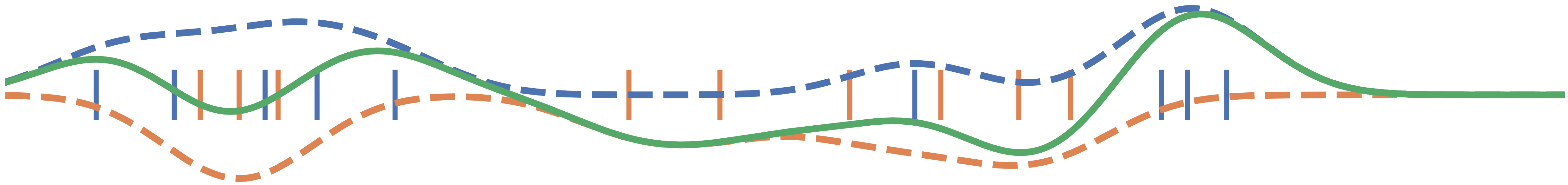


Maximum Mean Discrepancy

$$
\begin{aligned}
\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} \\
&= \sup_{\|f\|_{\mathcal{H}} \le 1} \langle f, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \\
&= \sup_{\|f\|_{\mathcal{H}} \le 1} \mathbb{E}_{X \sim \mathbb{P}}\, f(X) - \mathbb{E}_{Y \sim \mathbb{Q}}\, f(Y)
\end{aligned}
$$

- Last line is Integral Probability Metric (IPM) form
- f is called "witness function" or "critic": high on $\mathbb{P}$, low on $\mathbb{Q}$

$$f^*(t) \propto \langle \mu_{\mathbb{P}} - \mu_{\mathbb{Q}}, k(t, \cdot) \rangle_{\mathcal{H}} = \mathbb{E}_{\mathbb{P}}\, k(t, X) - \mathbb{E}_{\mathbb{Q}}\, k(t, Y)$$

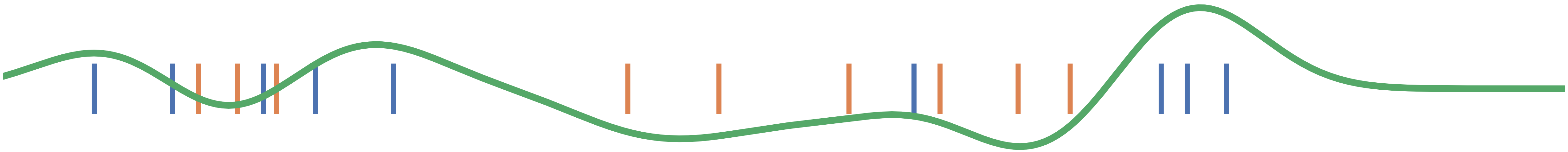


Maximum Mean Discrepancy

$$
\begin{aligned}
\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} \\
&= \sup_{\|f\|_{\mathcal{H}} \le 1} \langle f, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \\
&= \sup_{\|f\|_{\mathcal{H}} \le 1} \mathbb{E}_{X \sim \mathbb{P}}\, f(X) - \mathbb{E}_{Y \sim \mathbb{Q}}\, f(Y)
\end{aligned}
$$

- Last line is Integral Probability Metric (IPM) form
- f is called "witness function" or "critic": high on $\mathbb{P}$, low on $\mathbb{Q}$

$$f^*(t) \propto \langle \mu_{\mathbb{P}} - \mu_{\mathbb{Q}}, k(t, \cdot) \rangle_{\mathcal{H}} = \mathbb{E}_{\mathbb{P}}\, k(t, X) - \mathbb{E}_{\mathbb{Q}}\, k(t, Y)$$

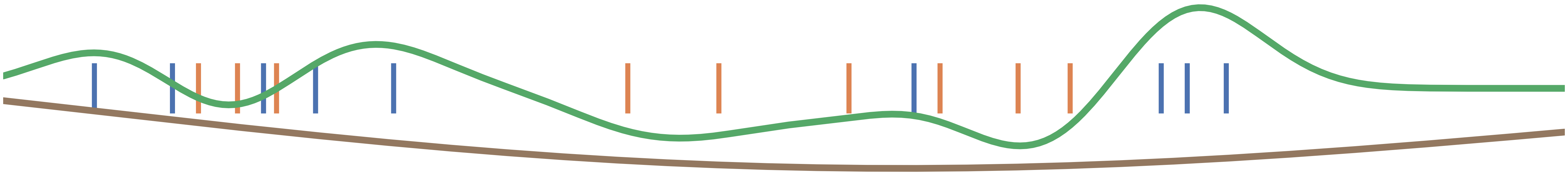


Maximum Mean Discrepancy

$$\begin{aligned}\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}} \le 1} \langle f, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}} \le 1} \mathbb{E}_{X \sim \mathbb{P}}\, f(X) - \mathbb{E}_{Y \sim \mathbb{Q}}\, f(Y)\end{aligned}$$

- Last line is Integral Probability Metric (IPM) form
- f is called “witness function” or “critic”: high on $\mathbb{P}$, low on $\mathbb{Q}$

$$f^*(t) \propto \langle \mu_{\mathbb{P}} - \mu_{\mathbb{Q}}, k(t, \cdot) \rangle_{\mathcal{H}} = \mathbb{E}_{\mathbb{P}}\, k(t, X) - \mathbb{E}_{\mathbb{Q}}\, k(t, Y)$$

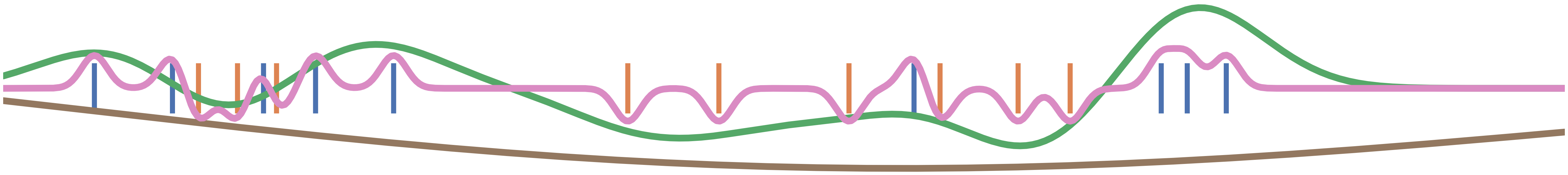


Maximum Mean Discrepancy

$$\begin{aligned}\mathrm{MMD}(\mathbb{P},\mathbb{Q}) &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}}\le 1} \langle f, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\rangle_{\mathcal{H}} \\ &= \sup_{\|f\|_{\mathcal{H}}\le 1} \mathbb{E}_{X\sim\mathbb{P}}\, f(X) - \mathbb{E}_{Y\sim\mathbb{Q}}\, f(Y)\end{aligned}$$

- Last line is Integral Probability Metric (IPM) form
- f is called “witness function” or “critic”: high on $\mathbb{P}$, low on $\mathbb{Q}$

$$f^*(t) \propto \langle \mu_{\mathbb{P}} - \mu_{\mathbb{Q}}, k(t,\cdot)\rangle_{\mathcal{H}} = \mathbb{E}_{\mathbb{P}}\, k(t,X) - \mathbb{E}_{\mathbb{Q}}\, k(t,Y)$$



MMD properties

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}$$

- $\mathrm{MMD}(\mathbb{P}, \mathbb{P}) = 0$, symmetry, triangle inequality

MMD properties

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}$$

- $\mathrm{MMD}(\mathbb{P}, \mathbb{P}) = 0$, symmetry, triangle inequality
- If $\boldsymbol{k}$ is *characteristic*, then $\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = 0$ iff $\mathbb{P} = \mathbb{Q}$
  - i.e. $\mathbb{P} \mapsto \mu_{\mathbb{P}}$ is injective

MMD properties

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}$$

- $\mathrm{MMD}(\mathbb{P}, \mathbb{P}) = 0$, symmetry, triangle inequality
- If k is *characteristic*, then $\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = 0$ iff $\mathbb{P} = \mathbb{Q}$
  - i.e. $\mathbb{P} \mapsto \mu_{\mathbb{P}}$ is injective
  - Makes MMD a metric on probability distributions

MMD properties

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}$$

- $\mathrm{MMD}(\mathbb{P}, \mathbb{P}) = 0$, symmetry, triangle inequality
- If k is *characteristic*, then $\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = 0$ iff $\mathbb{P} = \mathbb{Q}$
    - i.e. $\mathbb{P} \mapsto \mu_{\mathbb{P}}$ is injective
    - Makes MMD a metric on probability distributions
    - Universal $\implies$ characteristic

MMD properties

$$\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}$$

- $\mathrm{MMD}(\mathbb{P}, \mathbb{P}) = 0$, symmetry, triangle inequality
- If k is *characteristic*, then $\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = 0$ iff $\mathbb{P} = \mathbb{Q}$
  - i.e. $\mathbb{P} \mapsto \mu_{\mathbb{P}}$ is injective
  - Makes MMD a metric on probability distributions
  - Universal $\implies$ characteristic
- If we use a linear kernel:
  - $\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}$ just Euclidean distance between means

MMD properties

$$\operatorname{MMD}(\mathbb{P}, \mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}$$

- $\operatorname{MMD}(\mathbb{P}, \mathbb{P}) = 0$, symmetry, triangle inequality
- If k is *characteristic*, then $\operatorname{MMD}(\mathbb{P}, \mathbb{Q}) = 0$ iff $\mathbb{P} = \mathbb{Q}$
  - i.e. $\mathbb{P} \mapsto \mu_{\mathbb{P}}$ is injective
  - Makes MMD a metric on probability distributions
  - Universal $\implies$ characteristic
- If we use a linear kernel:
  - $\operatorname{MMD}(\mathbb{P}, \mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}$ just Euclidean distance between means
- If we use $k(x, y) = d(x, 0) + d(y, 0) - d(x, y)$, the squared MMD becomes the *energy distance* [Sejdinovic+ Annals-13]

Application: Kernel Herding

- Want a "super-sample" from $\mathbb{P}$: $\mathbb{E}\, f(X) \approx \frac{1}{n} \sum_j f(Y_j)$ for all f

Application: Kernel Herding

- Want a "super-sample" from $\mathbb{P}$: $\mathbb{E}\, f(X) \approx \frac{1}{n} \sum_j f(Y_j)$ for all f
  - Letting $\mathbb{Q} = \frac{1}{T} \sum_{j=1}^{T} \delta_{Y_j}$, want $\langle f, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \approx \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$

Application: Kernel Herding

- Want a "super-sample" from $\mathbb{P}$: $\mathbb{E}\, f(X) \approx \frac{1}{n} \sum_j f(Y_j)$ for all f
  - Letting $\mathbb{Q} = \frac{1}{T} \sum_{j=1}^{T} \delta_{Y_j}$, want $\langle f, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \approx \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$
  - Error $\leq \|f\|_{\mathcal{H}} \operatorname{MMD}(\mathbb{P}, \mathbb{Q})$

Application: Kernel Herding

- Want a "super-sample" from $\mathbb{P}$: $\mathbb{E}\, f(X) \approx \frac{1}{n} \sum_j f(Y_j)$ for all f
  - Letting $\mathbb{Q} = \frac{1}{T} \sum_{j=1}^{T} \delta_{Y_j}$, want $\langle f, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \approx \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$
  - Error $\leq \|f\|_{\mathcal{H}} \operatorname{MMD}(\mathbb{P}, \mathbb{Q})$
- Greedily minimize the MMD:

$$Y_{T+1} \in \operatorname*{arg\,min}_{Y \in \mathcal{X}} \mathbb{E}_{X' \sim \mathbb{P}}\, k(Y, X') - \frac{1}{T+1} \sum_{j=1}^{T} k(Y, Y_j)$$

Application: Kernel Herding

- Want a "super-sample" from $\mathbb{P}$: $\mathbb{E}\, f(X) \approx \frac{1}{n} \sum_j f(Y_j)$ for all f
  - Letting $\mathbb{Q} = \frac{1}{T} \sum_{j=1}^{T} \delta_{Y_j}$, want $\langle f, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \approx \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$
  - Error $\leq \|f\|_{\mathcal{H}} \operatorname{MMD}(\mathbb{P}, \mathbb{Q})$
- Greedily minimize the MMD:

$$Y_{T+1} \in \operatorname*{arg\,min}_{Y \in \mathcal{X}} \mathbb{E}_{X' \sim \mathbb{P}}\, k(Y, X') - \frac{1}{T+1} \sum_{j=1}^{T} k(Y, Y_j)$$

- Get $\mathcal{O}(1/T)$ approximation instead of $\mathcal{O}(1/\sqrt{T})$ with random samples

- Want a "super-s ... for all f
  - Letting $\mathbb{Q}$ ... $\rangle_{\mathcal{H}}$ for all $f \in \mathcal{H}$
  - Error $\leq \|$ ...
- Greedily minim ...

$$Y_{T+1} \in \quad \cdots \quad k(Y, Y_j)$$

- Get $\mathcal{O}(1/T)$ ap ... ndom samples

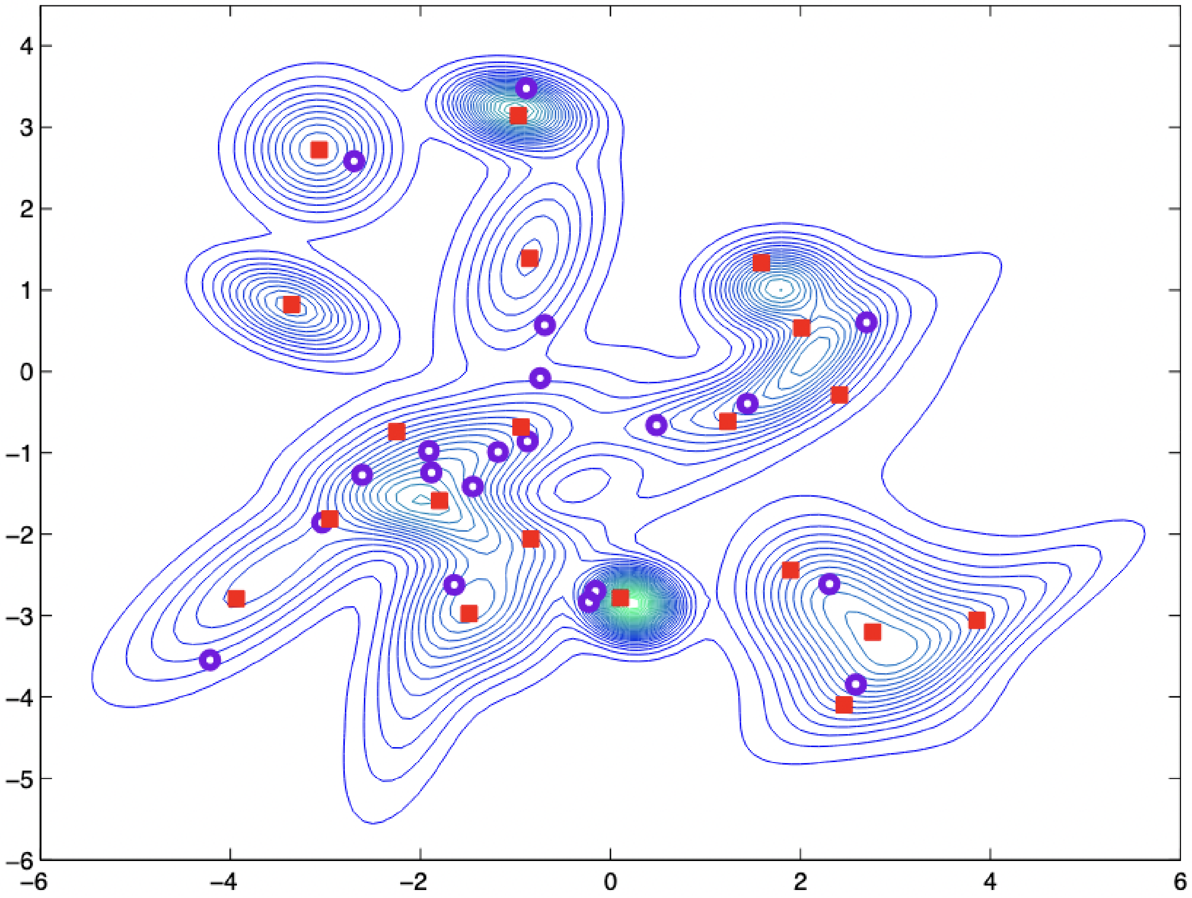


Figure 1: First 20 samples form herding (red squares) versus *i.i.d.* random sampling (purple circles).

Estimating MMD from samples

$$\mathrm{MMD}_k^2(\mathbb{P}, \mathbb{Q}) = \langle \mu_{\mathbb{P}}, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} + \langle \mu_{\mathbb{Q}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}}$$

Estimating MMD from samples

$$
\begin{aligned}
\mathrm{MMD}_k^2(\mathbb{P}, \mathbb{Q}) &= \langle \mu_{\mathbb{P}}, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} + \langle \mu_{\mathbb{Q}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \\
&= \mathbb{E}_{\substack{X, X' \sim \mathbb{P} \\ Y, Y' \sim \mathbb{Q}}} \left[k(X, X') - 2k(X, Y) + k(Y, Y') \right]
\end{aligned}
$$

Estimating MMD from samples

$$
\begin{aligned}
\mathrm{MMD}_k^2(\mathbb{P}, \mathbb{Q}) &= \langle \mu_{\mathbb{P}}, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} + \langle \mu_{\mathbb{Q}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \\
&= \mathbb{E}_{\substack{X, X' \sim \mathbb{P} \\ Y, Y' \sim \mathbb{Q}}} \left[k(X, X') - 2k(X, Y) + k(Y, Y') \right]
\end{aligned}
$$

$$
\widehat{\mathrm{MMD}}_k^2(X, Y) = \mathrm{mean}(K_{XX}) + \mathrm{mean}(K_{YY}) - 2\,\mathrm{mean}(K_{XY})
$$

Estimating MMD from samples

$$\mathrm{MMD}_k^2(\mathbb{P}, \mathbb{Q}) = \langle \mu_{\mathbb{P}}, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} + \langle \mu_{\mathbb{Q}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}}$$

$$= \mathbb{E}_{\substack{X, X' \sim \mathbb{P} \\ Y, Y' \sim \mathbb{Q}}} \left[k(X, X') - 2k(X, Y) + k(Y, Y') \right]$$

$$\widehat{\mathrm{MMD}}_k^2(X, Y) = \mathrm{mean}(K_{XX}) + \mathrm{mean}(K_{YY}) - 2\,\mathrm{mean}(K_{XY})$$

K_{XX}

			
	1.0	0.2	0.6
	0.2	1.0	0.5
	0.6	0.5	1.0

Estimating MMD from samples

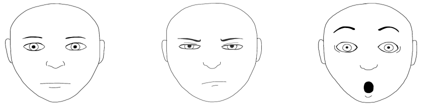
$$\begin{aligned}\mathrm{MMD}_k^2(\mathbb{P}, \mathbb{Q}) &= \langle \mu_{\mathbb{P}}, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} + \langle \mu_{\mathbb{Q}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} \\ &= \mathbb{E}_{\substack{X, X' \sim \mathbb{P} \\ Y, Y' \sim \mathbb{Q}}} [k(X, X') - 2k(X, Y) + k(Y, Y')]\end{aligned}$$

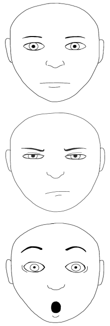
$$\widehat{\mathrm{MMD}}_k^2(X, Y) = \mathrm{mean}(K_{XX}) + \mathrm{mean}(K_{YY}) - 2\,\mathrm{mean}(K_{XY})$$

K_{XX}

	1.0	0.2	0.6
	0.2	1.0	0.5
	0.6	0.5	1.0

K_{YY}





	1.0	0.8	0.7
	0.8	1.0	0.6
	0.7	0.6	1.0

Estimating MMD from samples

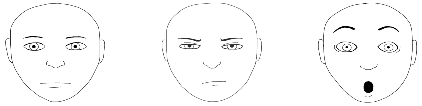
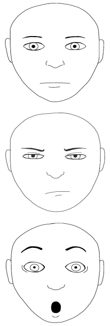
$$\mathrm{MMD}_k^2(\mathbb{P},\mathbb{Q}) = \langle \mu_{\mathbb{P}}, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} + \langle \mu_{\mathbb{Q}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}}$$
$$= \mathbb{E}_{\substack{X,X' \sim \mathbb{P} \\ Y,Y' \sim \mathbb{Q}}} [k(X,X') - 2k(X,Y) + k(Y,Y')]$$

$$\widehat{\mathrm{MMD}}_k^2(X,Y) = \mathrm{mean}(K_{XX}) + \mathrm{mean}(K_{YY}) - 2\,\mathrm{mean}(K_{XY})$$

K_{XX}

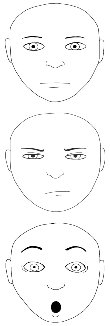

	1.0	0.2	0.6
	0.2	1.0	0.5
	0.6	0.5	1.0

K_{YY}

	1.0	0.8	0.7
	0.8	1.0	0.6
	0.7	0.6	1.0

K_{XY}

	0.3	0.1	0.2
	0.2	0.3	0.3
	0.2	0.1	0.4

MMD vs other distances

- MMD has easy $\mathcal{O}(n^2)$ estimator
  - *block* or *incomplete* estimators are $\mathcal{O}(n^\alpha)$ for $\alpha \in [1, 2]$, but noisier

MMD vs other distances

- MMD has easy $\mathcal{O}(n^2)$ estimator
  - *block* or *incomplete* estimators are $\mathcal{O}(n^\alpha)$ for $\alpha \in [1, 2]$, but noisier
- For bounded kernel, $\mathcal{O}_p(1/\sqrt{n})$ estimation error

MMD vs other distances

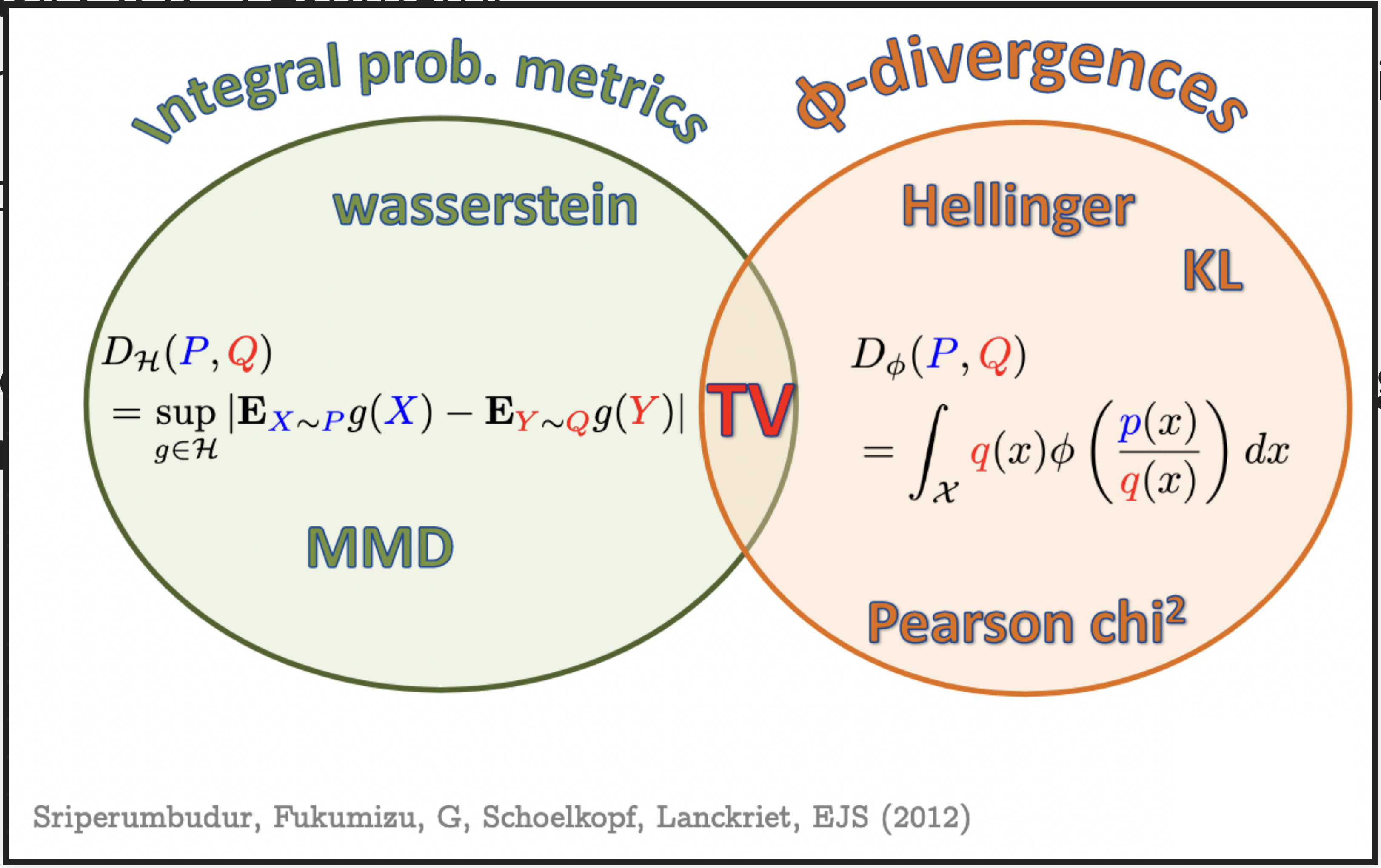
- MMD has easy $\mathcal{O}(n^2)$ estimator
  - *block* or *incomplete* estimators are $\mathcal{O}(n^\alpha)$ for $\alpha \in [1, 2]$, but noisier
- For bounded kernel, $\mathcal{O}_p(1/\sqrt{n})$ estimation error
  - Independent of data dimension!

MMD vs other distances

- MMD has easy $\mathcal{O}(n^2)$ estimator
  - *block* or *incomplete* estimators are $\mathcal{O}(n^\alpha)$ for $\alpha \in [1, 2]$, but noisier
- For bounded kernel, $\mathcal{O}_p(1/\sqrt{n})$ estimation error
  - Independent of data dimension!
  - But, no free lunch…the *value* of the MMD generally shrinks with growing dimension, so constant $\mathcal{O}_p(1/\sqrt{n})$ error gets worse relatively

MMD vs other distances

- MMD has easy $\mathcal{O}(n^2)$ estimator
  - *block* or *in*... isier
- For bounded...
  - Independ...
  - But, no fr... growing dimensio...



GP view of MMD

$$\begin{aligned}\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) &= \left(\sup_{f:\|f\|_{\mathcal{H}}\le 1} \mathbb{E}_{X\sim\mathbb{P}}\, f(X) - \mathbb{E}_{Y\sim\mathbb{Q}}\, f(Y)\right)^2 \\ &= \mathrm{Var}_{f\sim\mathcal{GP}(0,k)}\left[\mathbb{E}_{X\sim\mathbb{P}}\, f(X) - \mathbb{E}_{Y\sim\mathbb{Q}}\, f(Y)\right]\end{aligned}$$

GP view of MMD

$$\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}) = \left(\sup_{f : \|f\|_{\mathcal{H}} \leq 1} \mathbb{E}_{X \sim \mathbb{P}}\, f(X) - \mathbb{E}_{Y \sim \mathbb{Q}}\, f(Y) \right)^2$$

$$= \mathrm{Var}_{f \sim \mathcal{GP}(0,k)} \left[\mathbb{E}_{X \sim \mathbb{P}}\, f(X) - \mathbb{E}_{Y \sim \mathbb{Q}}\, f(Y) \right]$$

- Optimizing the gap in $\mathcal{H} \leftrightarrow$ average-case gap sampled from GP

GP view of MMD

$$\begin{aligned}\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}) &= \left(\sup_{f : \|f\|_{\mathcal{H}} \le 1} \mathbb{E}_{X \sim \mathbb{P}}\, f(X) - \mathbb{E}_{Y \sim \mathbb{Q}}\, f(Y) \right)^2 \\ &= \mathrm{Var}_{f \sim \mathcal{GP}(0,k)} \left[\mathbb{E}_{X \sim \mathbb{P}}\, f(X) - \mathbb{E}_{Y \sim \mathbb{Q}}\, f(Y) \right]\end{aligned}$$

- Optimizing the gap in $\mathcal{H}$ $\leftrightarrow$ average-case gap sampled from GP
- Six-line proof [Kanagawa+ 18, Proposition 6.1]

Application: Two-sample testing

- Given samples from two unknown distributions

$$X \sim \mathbb{P} \qquad\qquad Y \sim \mathbb{Q}$$

- Question: is $\mathbb{P} = \mathbb{Q}$?

Application: Two-sample testing

- Given samples from two unknown distributions

$$X \sim \mathbb{P} \qquad Y \sim \mathbb{Q}$$

- Do smokers/non-smokers get different cancers?

Application: Two-sample testing

- Given samples from two unknown distributions

$$X \sim \mathbb{P} \qquad Y \sim \mathbb{Q}$$

- Do smokers/non-smokers get different cancers?
- Do Brits have the same friend network types as Americans?

Application: Two-sample testing

- Given samples from two unknown distributions

$$X \sim \mathbb{P} \qquad Y \sim \mathbb{Q}$$

- Do smokers/non-smokers get different cancers?
- Do Brits have the same friend network types as Americans?
- When does my laser agree with the one on Mars?

Application: Two-sample testing

- Given samples from two unknown distributions

$$X \sim \mathbb{P} \qquad Y \sim \mathbb{Q}$$

- Do smokers/non-smokers get different cancers?
- Do Brits have the same friend network types as Americans?
- When does my laser agree with the one on Mars?
- Are storms in the 2000s different from storms in the 1800s?

Application: Two-sample testing

- Given samples from two unknown distributions

$$X \sim \mathbb{P} \qquad Y \sim \mathbb{Q}$$

- Do smokers/non-smokers get different cancers?
- Do Brits have the same friend network types as Americans?
- When does my laser agree with the one on Mars?
- Are storms in the 2000s different from storms in the 1800s?
- Does presence of this protein affect DNA binding? [MMDiff2]

Application: Two-sample testing

- Given samples from two unknown distributions

$$X \sim \mathbb{P} \qquad Y \sim \mathbb{Q}$$

- Do smokers/non-smokers get different cancers?
- Do Brits have the same friend network types as Americans?
- When does my laser agree with the one on Mars?
- Are storms in the 2000s different from storms in the 1800s?
- Does presence of this protein affect DNA binding? [`MMDiff2`]
- Do these `dob` and `birthday` columns mean the same thing?

Application: Two-sample testing

- Given samples from two unknown distributions

$$X \sim \mathbb{P} \qquad Y \sim \mathbb{Q}$$

- Do smokers/non-smokers get different cancers?
- Do Brits have the same friend network types as Americans?
- When does my laser agree with the one on Mars?
- Are storms in the 2000s different from storms in the 1800s?
- Does presence of this protein affect DNA binding? [`MMDiff2`]
- Do these `dob` and `birthday` columns mean the same thing?
- Does my generative model $\mathbb{Q}_\theta$ match $\mathbb{P}_{\text{data}}$?

Application: Two-sample testing

- Given samples from two unknown distributions

$$\boldsymbol{X} \sim \mathbb{P} \qquad \boldsymbol{Y} \sim \mathbb{Q}$$

- Question: is $\mathbb{P} = \mathbb{Q}$?

Application: Two-sample testing

- Given samples from two unknown distributions

$$\boldsymbol{X} \sim \mathbb{P} \qquad \boldsymbol{Y} \sim \mathbb{Q}$$

- Question: is $\mathbb{P} = \mathbb{Q}$?
- Hypothesis testing approach:

$$H_0 : \mathbb{P} = \mathbb{Q} \qquad H_1 : \mathbb{P} \neq \mathbb{Q}$$

Application: Two-sample testing

- Given samples from two unknown distributions

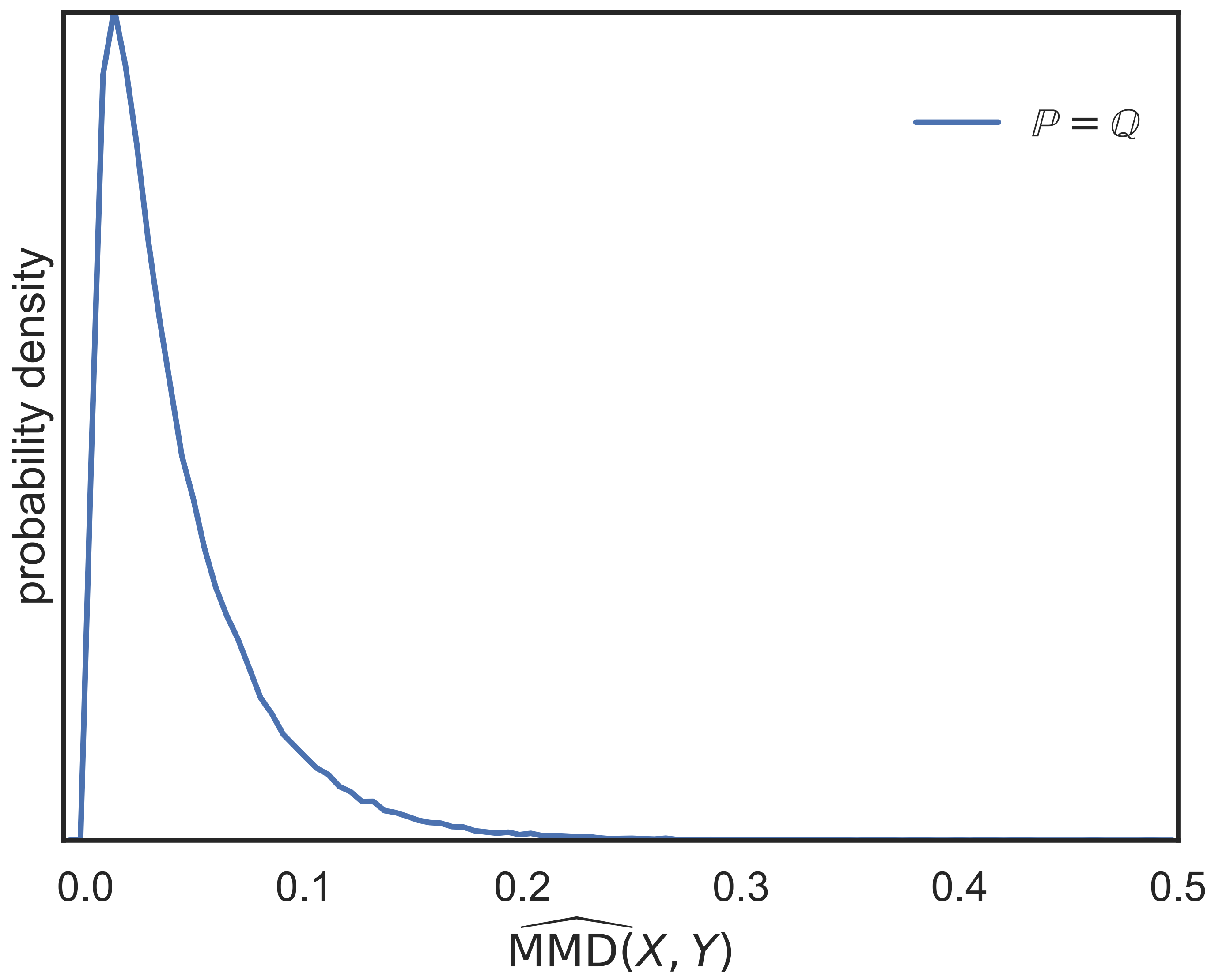
$$X \sim \mathbb{P} \qquad Y \sim \mathbb{Q}$$

- Question: is $\mathbb{P} = \mathbb{Q}$?
- Hypothesis testing approach:

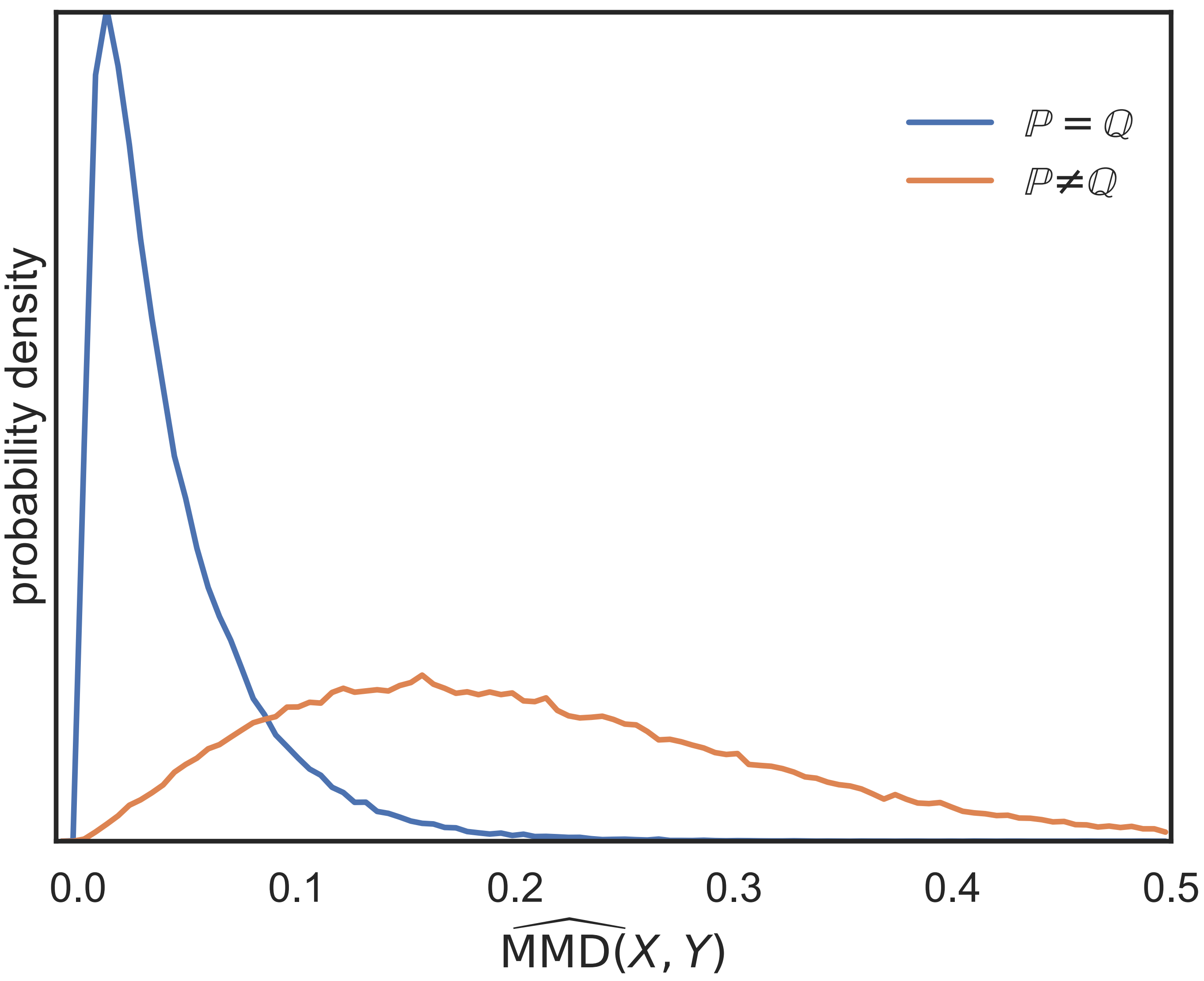
$$H_0 : \mathbb{P} = \mathbb{Q} \qquad H_1 : \mathbb{P} \neq \mathbb{Q}$$

- Reject H_0 if $\widehat{\mathrm{MMD}}(X, Y) > c_\alpha$

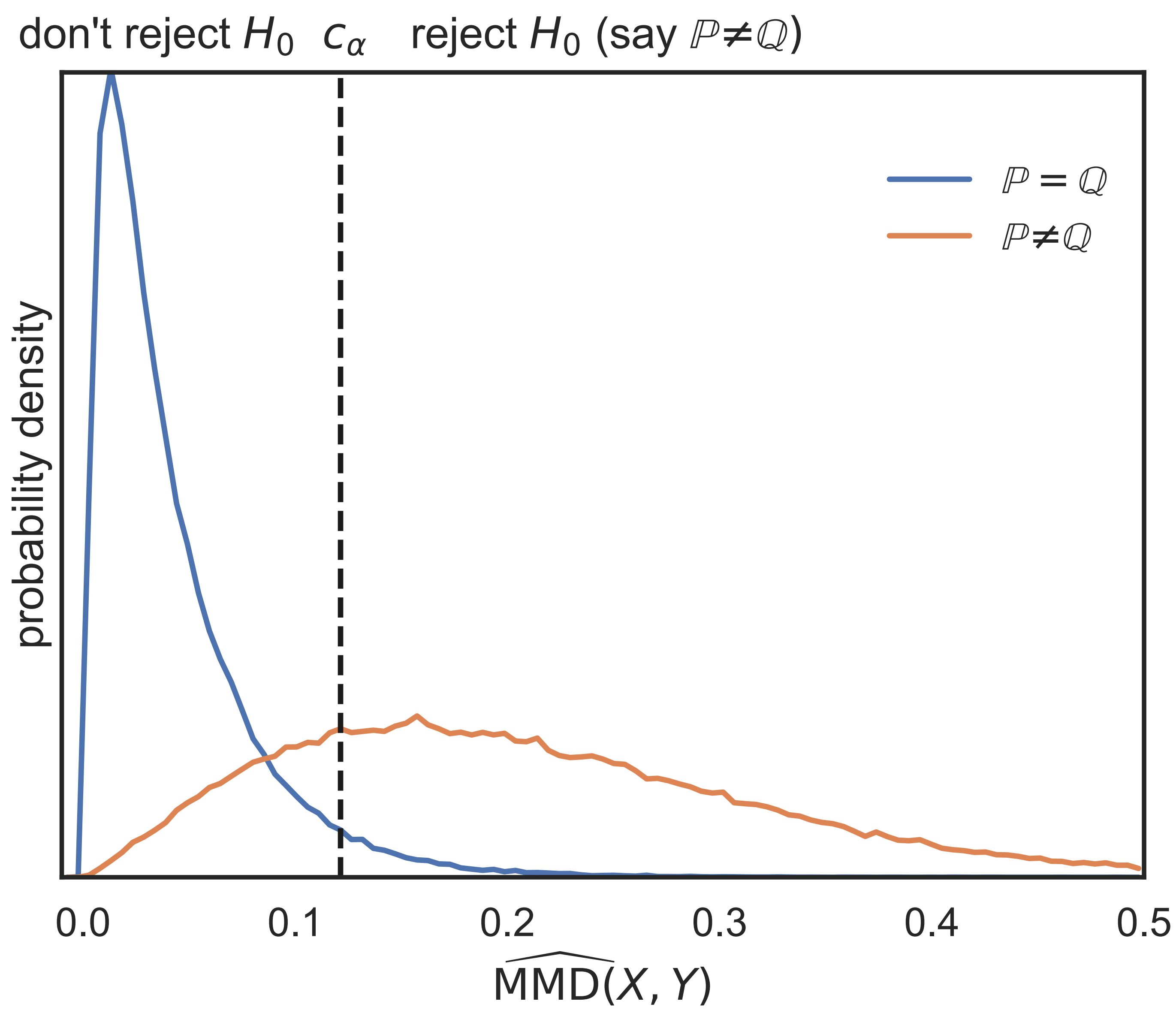
What's a hypothesis test again?



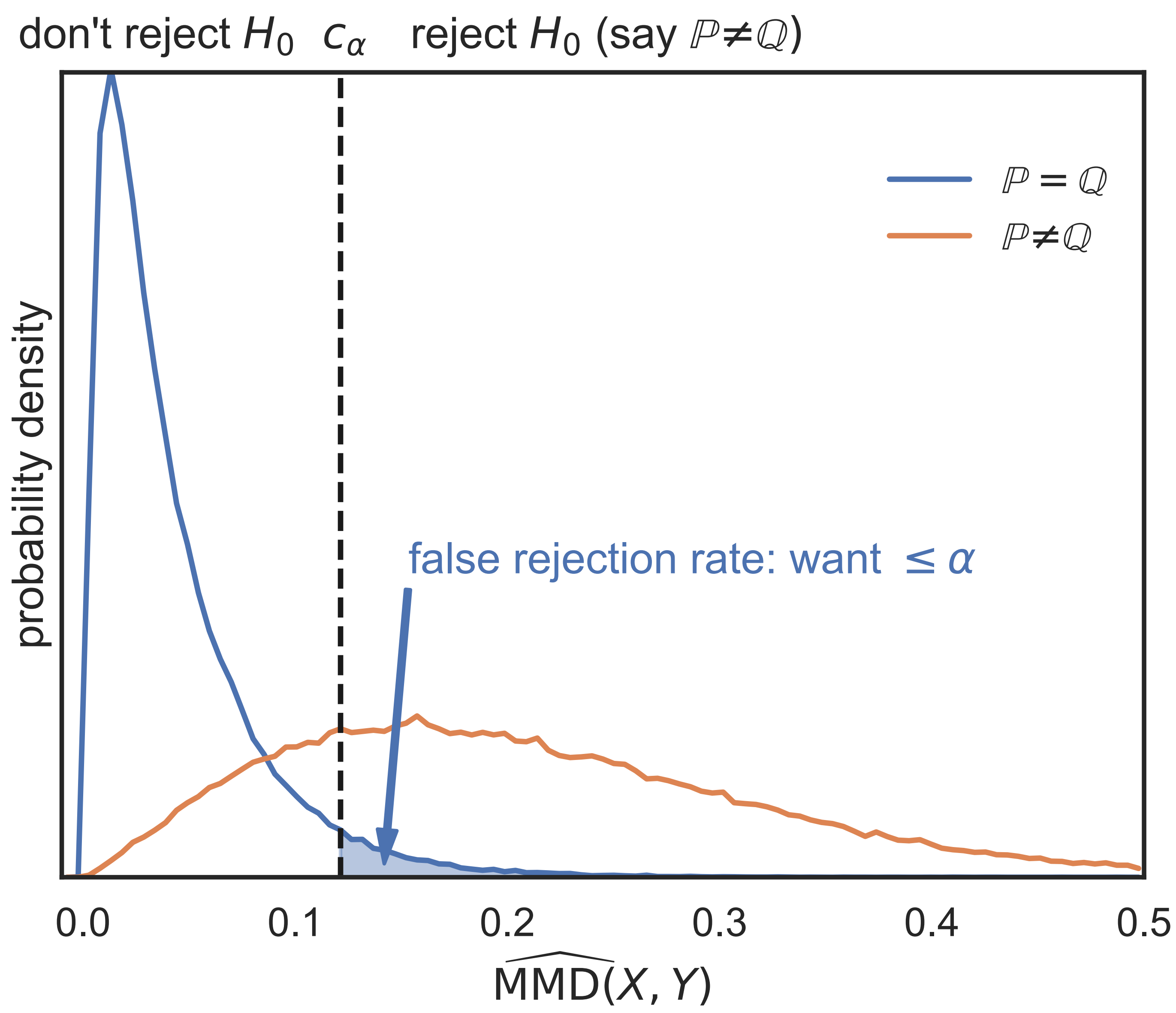
What's a hypothesis test again?



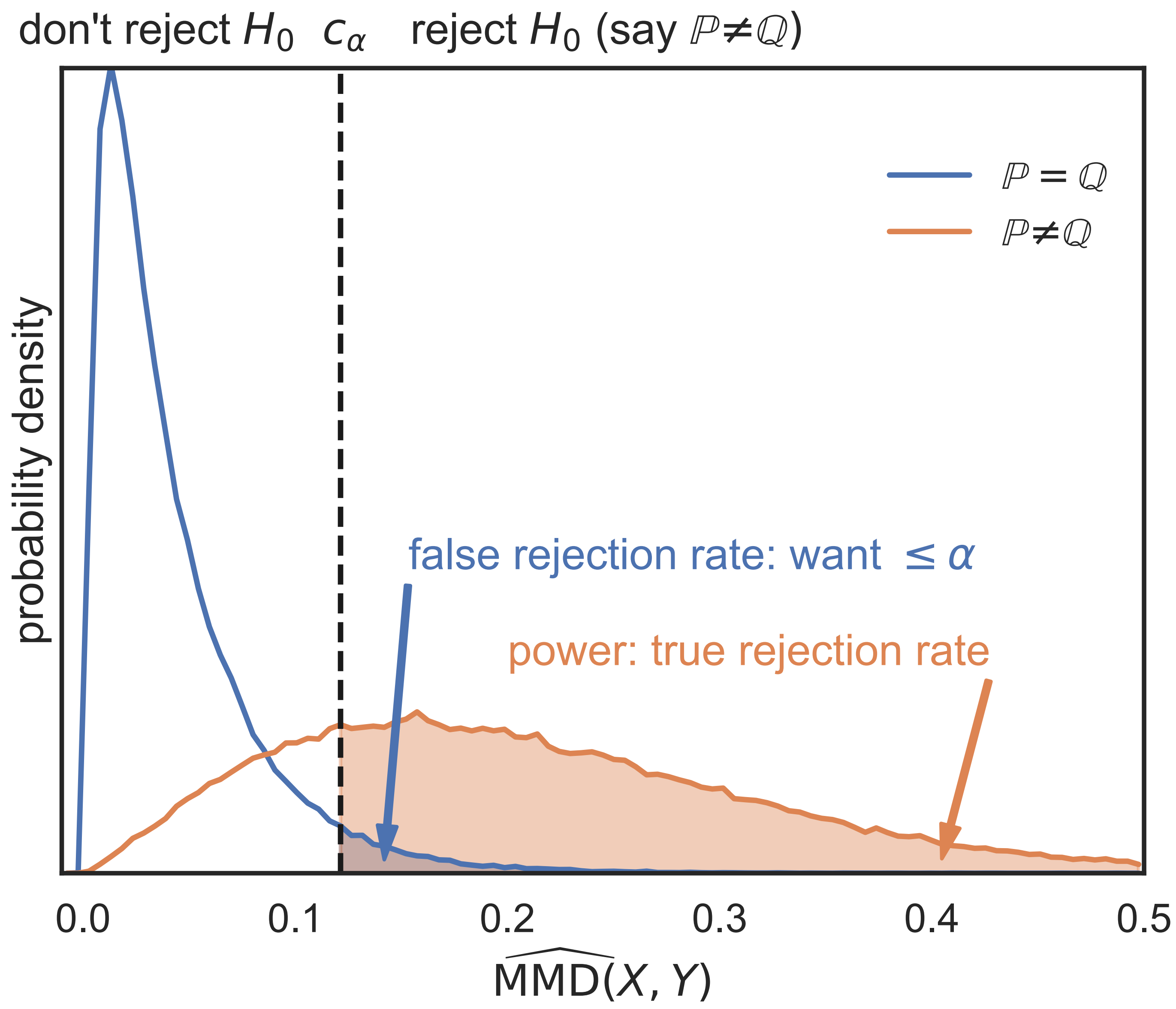
What's a hypothesis test again?



What's a hypothesis test again?



What's a hypothesis test again?



MMD-based testing

- H_0: $n\widehat{\mathrm{MMD}}^2$ converges in distribution to…something
  - Infinite mixture of χ^2s, params depend on $\mathbb{P}$ and k

MMD-based testing

- $H_0 : n\widehat{\mathrm{MMD}}^2$ converges in distribution to…something
  - Infinite mixture of χ^2s, params depend on $\mathbb{P}$ and k
  - Can estimate threshold with *permutation testing*

MMD-based testing

- H_0: $n\widehat{\mathrm{MMD}}^2$ converges in distribution to…something
  - Infinite mixture of χ^2s, params depend on $\mathbb{P}$ and k
  - Can estimate threshold with *permutation testing*
- H_1: $\sqrt{n}(\widehat{\mathrm{MMD}}^2 - \mathrm{MMD}^2) \xrightarrow{d}$ asymptotically normal

MMD-based testing

- H_0: $n\widehat{\mathrm{MMD}}^2$ converges in distribution to…something
  - Infinite mixture of χ^2s, params depend on $\mathbb{P}$ and k
  - Can estimate threshold with *permutation testing*
- H_1: $\sqrt{n}(\widehat{\mathrm{MMD}}^2 - \mathrm{MMD}^2) \xrightarrow{d}$ asymptotically normal
- Any characteristic kernel gives consistent test

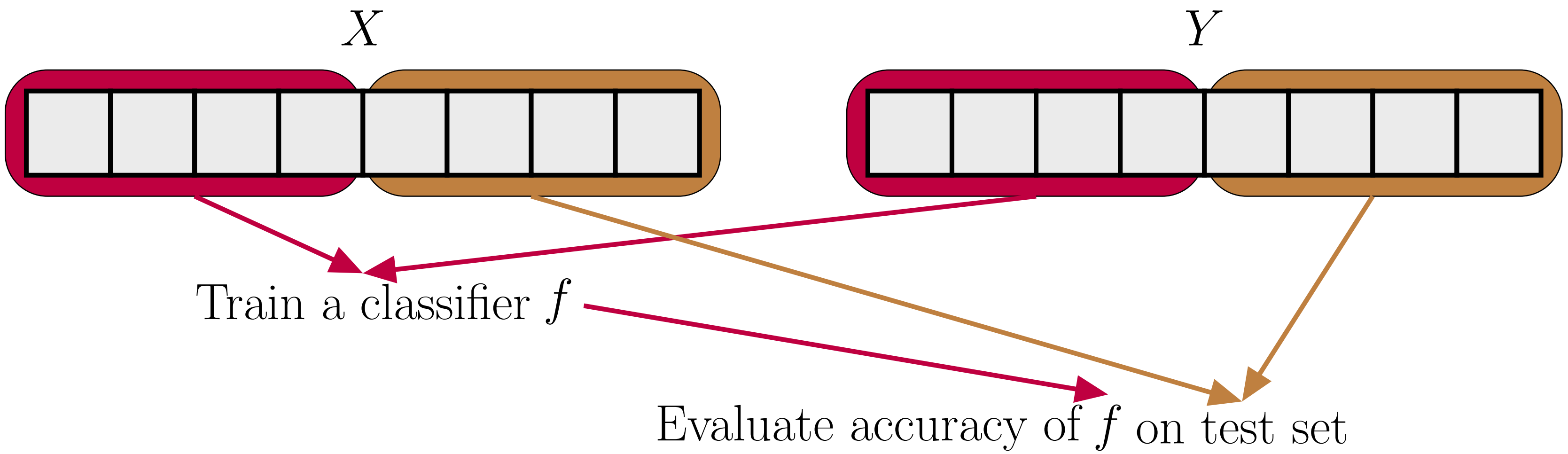
MMD-based testing

- H_0: $n\widehat{\mathrm{MMD}}^2$ converges in distribution to…something
  - Infinite mixture of χ^2s, params depend on $\mathbb{P}$ and $\boldsymbol{k}$
  - Can estimate threshold with *permutation testing*
- H_1: $\sqrt{n}(\widehat{\mathrm{MMD}}^2 - \mathrm{MMD}^2) \xrightarrow{d}$ asymptotically normal
- Any characteristic kernel gives consistent test…eventually

MMD-based testing

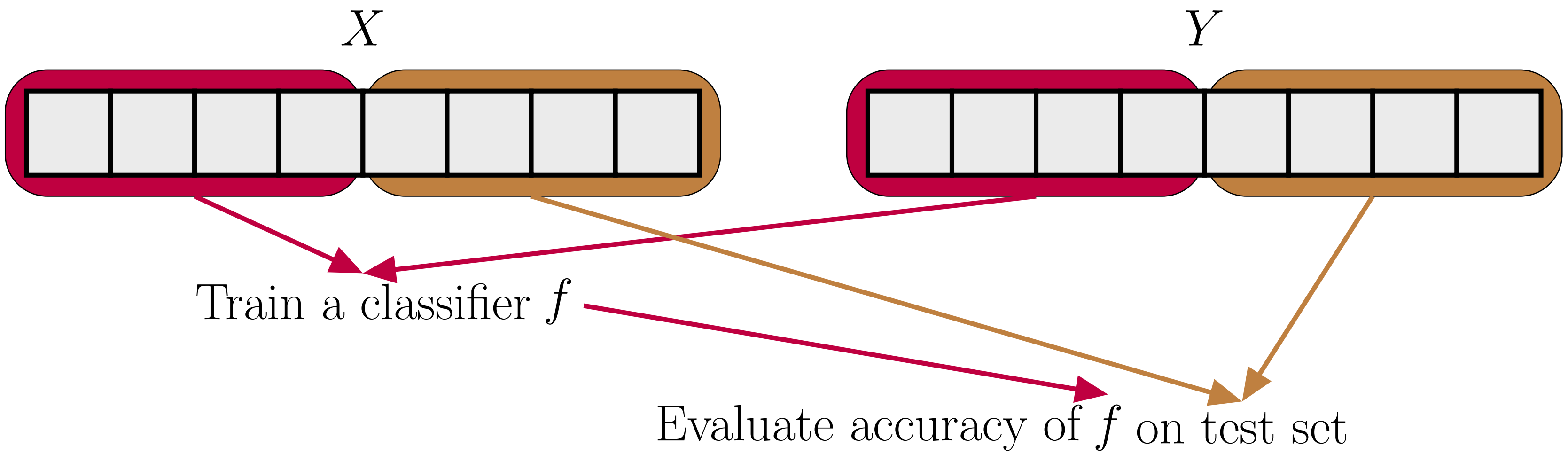
- H_0: $n\widehat{\mathrm{MMD}}^2$ converges in distribution to…something
  - Infinite mixture of χ^2s, params depend on $\mathbb{P}$ and $\boldsymbol{k}$
  - Can estimate threshold with *permutation testing*
- H_1: $\sqrt{n}(\widehat{\mathrm{MMD}}^2 - \mathrm{MMD}^2) \xrightarrow{d}$ asymptotically normal
- Any characteristic kernel gives consistent test…eventually
- Need enormous n if kernel is bad for problem

Classifier two-sample tests



- $\hat{T}(X, Y)$ is the accuracy of f on the test set
- Under H_0, classification impossible: $\hat{T} \sim \mathrm{Binomial}(n, \frac{1}{2})$

Classifier two-sample tests



- $\hat{T}(X, Y)$ is the accuracy of f on the test set
- Under H_0, classification impossible: $\hat{T} \sim \text{Binomial}(n, \frac{1}{2})$
- With $k(x, y) = \frac{1}{4} f(x) f(y)$ where $f(x) \in \{-1, 1\}$, get $\widehat{\text{MMD}}(X, Y) = \left| \hat{T}(X, Y) - \frac{1}{2} \right|$

Deep learning and deep kernels

- $k(x, y) = \frac{1}{4} f(x) f(y)$ is one form of *deep kernel*

Deep learning and deep kernels

- $k(x, y) = \frac{1}{4} f(x) f(y)$ is one form of *deep kernel*
- Deep models are usually of the form $f(x) = w^\mathsf{T} \phi_\psi(x)$
  - With a *learned* $\phi_\psi(x) : \mathcal{X} \to \mathbb{R}^D$

Deep learning and deep kernels

- $k(x, y) = \frac{1}{4} f(x) f(y)$ is one form of *deep kernel*
- Deep models are usually of the form $f(x) = w^\mathsf{T} \phi_\psi(x)$
  - With a *learned* $\phi_\psi(x) : \mathcal{X} \to \mathbb{R}^D$
- If we fix ψ, have $f \in \mathcal{H}_\psi$ with $k_\psi(x, y) = \phi_\psi(x)^\mathsf{T} \phi_\psi(y)$

Deep learning and deep kernels

- $k(x, y) = \frac{1}{4} f(x) f(y)$ is one form of *deep kernel*
- Deep models are usually of the form $f(x) = w^\mathsf{T} \phi_\psi(x)$
  - With a *learned* $\phi_\psi(x) : \mathcal{X} \to \mathbb{R}^D$
- If we fix ψ, have $f \in \mathcal{H}_\psi$ with $k_\psi(x, y) = \phi_\psi(x)^\mathsf{T} \phi_\psi(y)$
  - Same idea as NNGP approximation

Deep learning and deep kernels

- $k(x, y) = \frac{1}{4} f(x) f(y)$ is one form of *deep kernel*
- Deep models are usually of the form $f(x) = w^\mathsf{T} \phi_\psi(x)$
  - With a *learned* $\phi_\psi(x) : \mathcal{X} \to \mathbb{R}^D$
- If we fix ψ, have $f \in \mathcal{H}_\psi$ with $k_\psi(x, y) = \phi_\psi(x)^\mathsf{T} \phi_\psi(y)$
  - Same idea as NNGP approximation
- Generalize to a **deep kernel**:

$$k_\psi(x, y) = \kappa\left(\phi_\psi(x), \phi_\psi(y)\right)$$

Normal deep learning $\subset$ deep kernels

- Take $k_\psi(x, y) = \frac{1}{4} f_\psi(x) f_\psi(y)$
- Final function in $\mathcal{H}_\psi$ will be $a f_\psi(x)$

Normal deep learning $\subset$ deep kernels

- Take $k_\psi(x, y) = \frac{1}{4} f_\psi(x) f_\psi(y) + 1$
- Final function in $\mathcal{H}_\psi$ will be $a f_\psi(x) + b$

Normal deep learning $\subset$ deep kernels

- Take $k_\psi(x, y) = \frac{1}{4} f_\psi(x) f_\psi(y) + 1$
- Final function in $\mathcal{H}_\psi$ will be $a f_\psi(x) + b$
- With logistic loss: this is Platt scaling

Normal deep learning $\subset$ deep kernels

- Take $k_\psi(x, y) = \frac{1}{4} f_\psi(x) f_\psi(y) + 1$
- Final function in $\mathcal{H}_\psi$ will be $a f_\psi(x) + b$
- With logistic loss: this is Platt scaling

On Calibration of Modern Neural Networks

Chuan Guo [*1] **Geoff Pleiss** [*1] **Yu Sun** [*1] **Kilian Q. Weinberger** [1]

“Normal deep learning $\subset$ deep kernels” – so?

- This does *not* say that deep learning is (even approximately) a kernel method

“Normal deep learning ⊂ deep kernels” – so?

- This does *not* say that deep learning is (even approximately) a kernel method
- …despite what some people might want you to think

Computer Science > Machine Learning

[Submitted on 30 Nov 2020]

Every Model Learned by Gradient Descent Is Approximately a Kernel Machine

Pedro Domingos

“Normal deep learning $\subset$ deep kernels” – so?

- This does *not* say that deep learning is (even approximately) a kernel method
- …despite what some people might want you to think

Computer Science > Machine Learning

[Submitted on 30 Nov 2020]

Every Model Learned by Gradient Descent Is Approximately a Kernel Machine

Pedro Domingos

- We know theoretically deep learning can learn some things faster than any kernel method [see Malach+ ICML-21 + refs]

“Normal deep learning $\subset$ deep kernels” – so?

- This does *not* say that deep learning is (even approximately) a kernel method
- …despite what some people might want you to think

> **Computer Science > Machine Learning**
>
> *[Submitted on 30 Nov 2020]*
>
> **Every Model Learned by Gradient Descent Is Approximately a Kernel Machine**
>
> Pedro Domingos

- We know theoretically deep learning can learn some things faster than any kernel method [see Malach+ ICML-21 + refs]
- But deep kernel learning ≠ traditional kernel models
    - exactly like how usual deep learning ≠ linear models

Optimizing power of MMD tests

- Asymptotics of $\widehat{\mathrm{MMD}}^2$ give us immediately that

$$\Pr_{H_1}\left(n\widehat{\mathrm{MMD}}^2 > c_\alpha\right) \approx \Phi\left(\frac{\sqrt{n}\,\mathrm{MMD}^2}{\sigma_{H_1}} - \frac{c_\alpha}{\sqrt{n}\sigma_{H_1}}\right)$$

MMD, σ_{H_1}, c_α are constants: first term usually dominates

Optimizing power of MMD tests

- Asymptotics of $\widehat{\mathrm{MMD}}^2$ give us immediately that

$$\Pr_{H_1}\left(n\widehat{\mathrm{MMD}}^2 > c_\alpha\right) \approx \Phi\left(\frac{\sqrt{n}\,\mathrm{MMD}^2}{\sigma_{H_1}} - \frac{c_\alpha}{\sqrt{n}\sigma_{H_1}}\right)$$

  MMD, σ_{H_1}, c_α are constants: first term usually dominates

- Pick k to maximize an estimate of $\mathrm{MMD}^2 / \sigma_{H_1}$

Optimizing power of MMD tests

- Asymptotics of $\widehat{\mathrm{MMD}}^2$ give us immediately that

$$\Pr_{H_1}\left(n\widehat{\mathrm{MMD}}^2 > c_\alpha\right) \approx \Phi\left(\frac{\sqrt{n}\,\mathrm{MMD}^2}{\sigma_{H_1}} - \frac{c_\alpha}{\sqrt{n}\sigma_{H_1}}\right)$$

  MMD, σ_{H_1}, c_α are constants: first term usually dominates

- Pick k to maximize an estimate of $\mathrm{MMD}^2 / \sigma_{H_1}$

- Use $\widehat{\mathrm{MMD}}$ from before, get $\hat{\sigma}_{H_1}$ from U-statistic theory

Optimizing power of MMD tests

- Asymptotics of $\widehat{\mathrm{MMD}}^2$ give us immediately that

$$\Pr_{H_1}\left(n\widehat{\mathrm{MMD}}^2 > c_\alpha\right) \approx \Phi\left(\frac{\sqrt{n}\,\mathrm{MMD}^2}{\sigma_{H_1}} - \frac{c_\alpha}{\sqrt{n}\sigma_{H_1}}\right)$$

  MMD, σ_{H_1}, c_α are constants: first term usually dominates

- Pick k to maximize an estimate of $\mathrm{MMD}^2 / \sigma_{H_1}$
- Use $\widehat{\mathrm{MMD}}$ from before, get $\hat{\sigma}_{H_1}$ from U-statistic theory
- Can show uniform $\mathcal{O}_P(n^{-\frac{1}{3}})$ convergence of estimator

Optimizing power of MMD tests

- Asymptotics of $\widehat{\mathrm{MMD}}^2$ give us immediately that

$$\Pr_{H_1}\left(n\widehat{\mathrm{MMD}}^2 > c_\alpha\right) \approx \Phi\left(\frac{\sqrt{n}\,\mathrm{MMD}^2}{\sigma_{H_1}} - \frac{c_\alpha}{\sqrt{n}\sigma_{H_1}}\right)$$

  MMD, σ_{H_1}, c_α are constants: first term usually dominates

- Pick k to maximize an estimate of $\mathrm{MMD}^2 / \sigma_{H_1}$
- Use $\widehat{\mathrm{MMD}}$ from before, get $\hat{\sigma}_{H_1}$ from U-statistic theory
- Can show uniform $\mathcal{O}_P(n^{-\frac{1}{3}})$ convergence of estimator
- Get better tests (even after data splitting)

Application: (S)MMD GANs

- An implicit generative model:
  - A generator net outputs samples from $\mathbb{Q}_\theta$

Application: (S)MMD GANs

- An implicit generative model:
  - A generator net outputs samples from $\mathbb{Q}_\theta$
  - Minimize estimate of $\operatorname{MMD}_\psi(\mathbb{P}^m, \mathbb{Q}_\theta^n)$ on a minibatch

Application: (S)MMD GANs

- An implicit generative model:
  - A generator net outputs samples from $\mathbb{Q}_\theta$
  - Minimize estimate of $\operatorname{MMD}_\psi(\mathbb{P}^m, \mathbb{Q}_\theta^n)$ on a minibatch
- MMD GAN: $\min_\theta \left[\max_\psi \operatorname{MMD}_\psi(\mathbb{P}, \mathbb{Q}_\theta)\right]$

Application: (S)MMD GANs

- An implicit generative model:
  - A generator net outputs samples from $\mathbb{Q}_\theta$
  - Minimize estimate of $\operatorname{MMD}_\psi(\mathbb{P}^m, \mathbb{Q}_\theta^n)$ on a minibatch
- MMD GAN: $\min_\theta \left[\max_\psi \operatorname{MMD}_\psi(\mathbb{P}, \mathbb{Q}_\theta)\right]$
- SMMD GAN: $\min_\theta \left[\max_\psi \operatorname{SMMD}_\psi(\mathbb{P}, \mathbb{Q}_\theta)\right]$
  - Scaled MMD uses kernel properties to ensure smooth loss for θ by making witness function smooth [Arbel+ NeurIPS-18]

Application: (S)MMD GANs

- An implicit generative model:
  - A generator net outputs samples from $\mathbb{Q}_\theta$
  - Minimize estimate of $\operatorname{MMD}_\psi(\mathbb{P}^m, \mathbb{Q}_\theta^n)$ on a minibatch
- MMD GAN: $\min_\theta \left[\max_\psi \operatorname{MMD}_\psi(\mathbb{P}, \mathbb{Q}_\theta)\right]$
- SMMD GAN: $\min_\theta \left[\max_\psi \operatorname{SMMD}_\psi(\mathbb{P}, \mathbb{Q}_\theta)\right]$
  - Scaled MMD uses kernel properties to ensure smooth loss for θ by making witness function smooth [Arbel+ NeurIPS-18]
  - Uses $\langle f, \partial_{x_1} k(x, \cdot)\rangle_{\mathcal{H}} = \partial_{x_1} f(x)$

Application: (S)MMD GANs

- An implicit generative model:
  - A generator net outputs samples from $\mathbb{Q}_\theta$
  - Minimize estimate of $\mathrm{MMD}_\psi(\mathbb{P}^m, \mathbb{Q}_\theta^n)$ on a minibatch
- MMD GAN: $\min_\theta \left[\max_\psi \mathrm{MMD}_\psi(\mathbb{P}, \mathbb{Q}_\theta)\right]$
- SMMD GAN: $\min_\theta \left[\max_\psi \mathrm{SMMD}_\psi(\mathbb{P}, \mathbb{Q}_\theta)\right]$
  - Scaled MMD uses kernel properties to ensure smooth loss for θ by making witness function smooth [Arbel+ NeurIPS-18]
  - Uses $\langle f, \partial_{x_1} k(x, \cdot)\rangle_{\mathcal{H}} = \partial_{x_1} f(x)$
  - Standard WGAN-GP better thought of in kernel framework

Application: fair representation learning (MMD-B-FAIR)

[Deka/Sutherland AISTATS-23]

- Want to find a representation where
    - We can tell whether an applicant is “creditworthy”
    - We can't distinguish applicants by race

Application: fair representation learning (MMD-B-FAIR)

[Deka/Sutherland AISTATS-23]

- Want to find a representation where
  - We can tell whether an applicant is “creditworthy”
  - We can't distinguish applicants by race
- Find a good classifier with near-zero test power for race

Application: fair representation learning (MMD-B-FAIR)

[Deka/Sutherland AISTATS-23]

- Want to find a representation where
  - We can tell whether an applicant is "creditworthy"
  - We can't distinguish applicants by race
- Find a good classifier with near-zero test power for race
- *Minimizing* the test power criterion turns out to be hard
  - Workaround: minimize test power of a (theoretical) *block* test

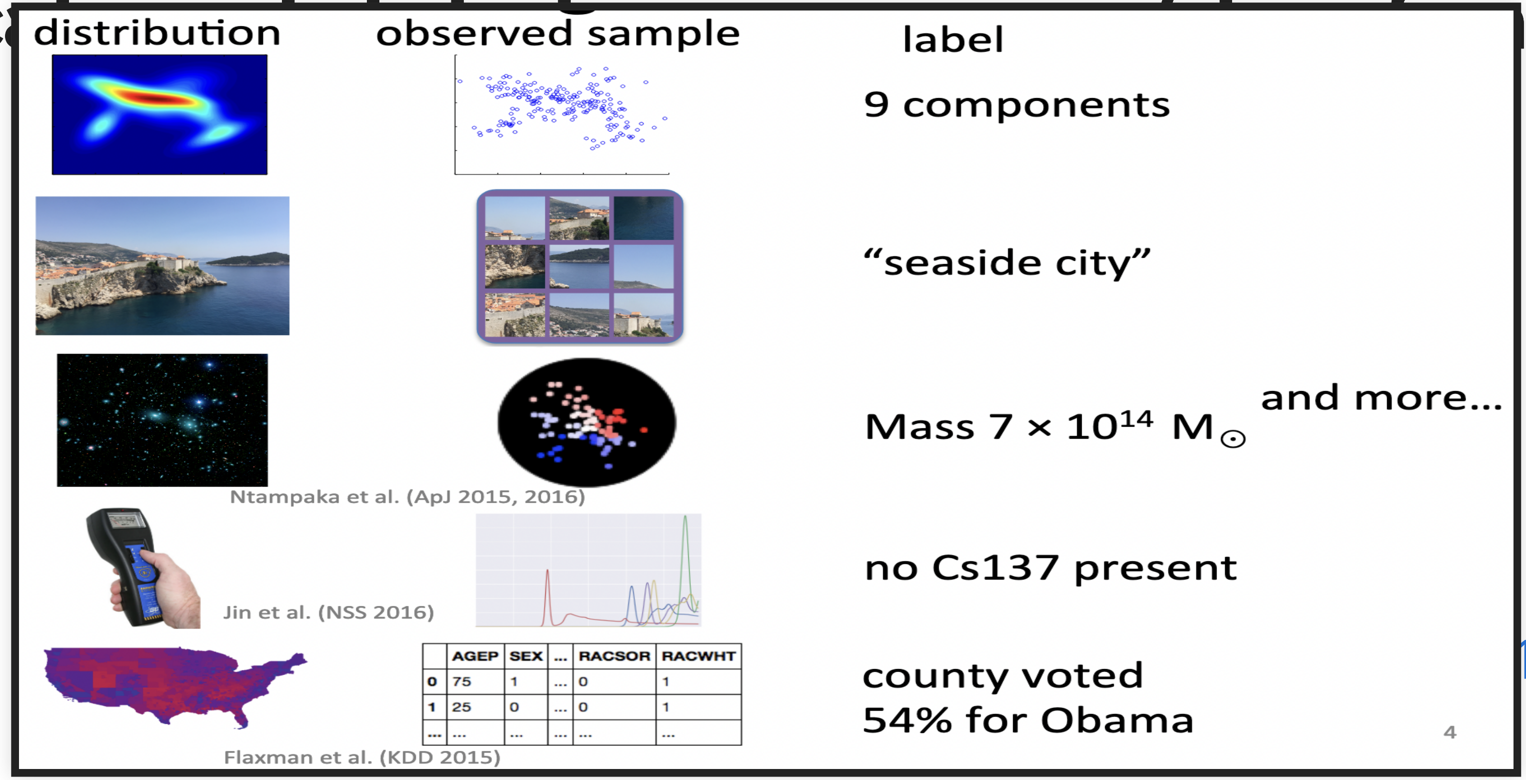
Application: distribution regression/classification/...

- We can define a kernel on distributions by, e.g.,

$$k(\mathbb{P}, \mathbb{Q}) = \exp\left(-\frac{1}{2\sigma^2}\mathrm{MMD}^2(\mathbb{P}, \mathbb{Q})\right)$$

- Some pointers:
  [Muandet+ NeurIPS-12] [Sutherland 2016] [Szabó+ JMLR-16]

Applic… tion/...



16]

Example: age from face images [Law+ AISTATS-18]

Bayesian distribution regression: incorporate $\mu_{\mathbb{P}}$ uncertainty

$\Bigg\{$ , ,  $\Bigg\} \to 35$

Example: age from face images [Law+ AISTATS-18]

Bayesian distribution regression: incorporate $\mu_{\mathbb{P}}$ uncertainty

$\{$ , ,  $\} \to 35$

IMDb database [Rothe+ 2015]: 400k images of 20k celebrities

Example: age from face images [Law+ AISTATS-18]

Bayesian distribution regression: incorporate $\mu_{\mathbb{P}}$ uncertainty

$\{$ , ,  $\} \to 35$

IMDb database [Rothe+ 2015]: 400k images of 20k celebrities

Example: age from face images [Law+ AISTATS-18]

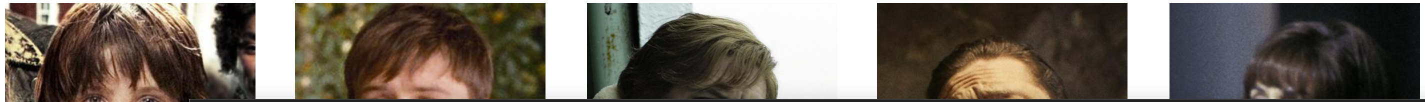
Bayesian distribution regression: incorporate $\mu_{\mathbb{P}}$ uncertainty

$\{$ , , , ,  $\} \to 35$

IMDb database [Rothe+ 2015]: 400k images of 20k celebrities

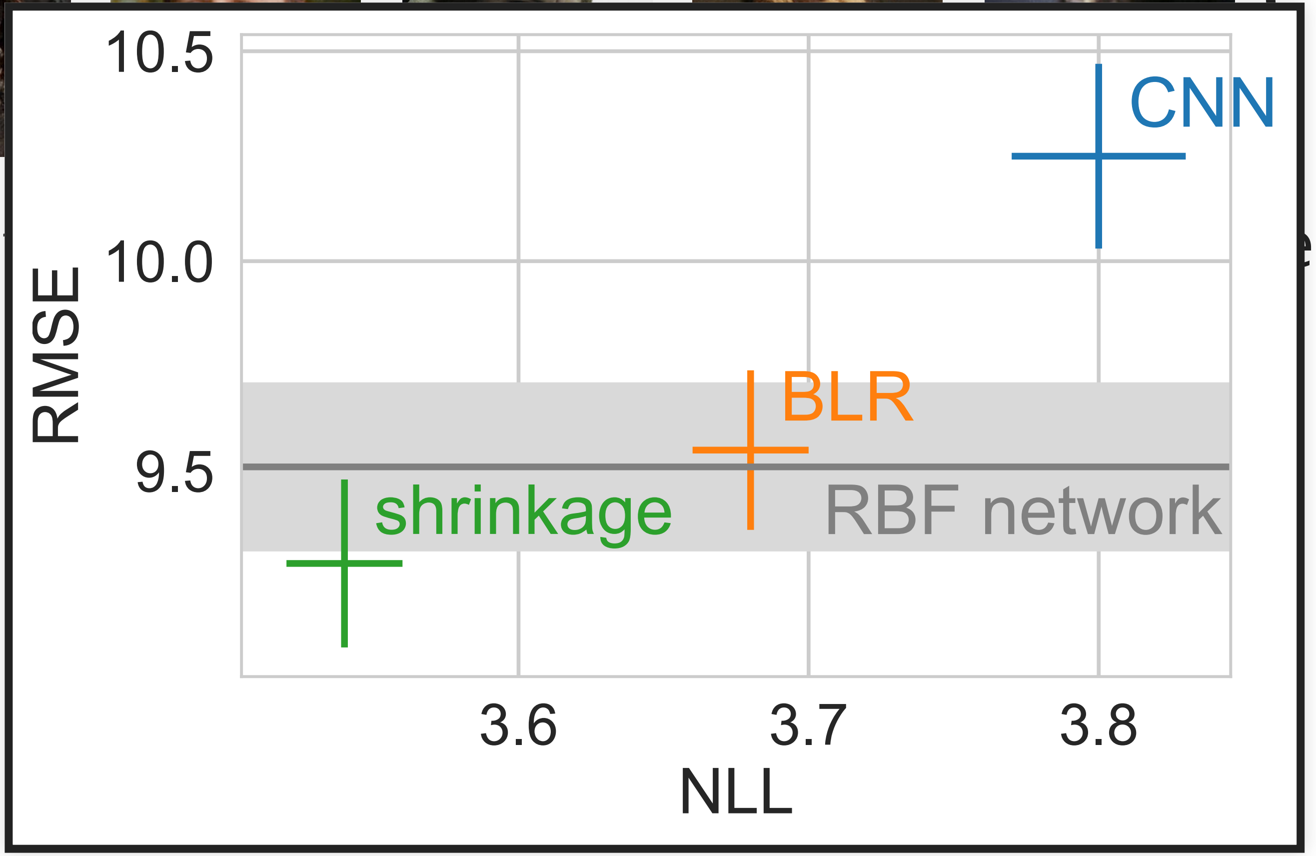
Example: age from face images [Law+ AISTATS-18]

Bayesian distribution regression: incorporate $\mu_{\mathbb{P}}$ uncertainty



$\to 35$

IMDb da ... ebrities



Independence

- $X \perp\!\!\!\perp Y$ iff $\mathrm{Cov}(f(X), g(Y)) = 0$ for all square-integrable f, g

Independence

- $X \perp\!\!\!\perp Y$ iff $\mathrm{Cov}(f(X), g(Y)) = 0$ for all square-integrable f, g
- Let's implement for RKHS functions $f \in \mathcal{H}_x$, $g \in \mathcal{H}_y$:

$$\mathbb{E}[f(X)]\,\mathbb{E}[g(Y)]$$

Independence

- $X \perp\!\!\!\perp Y$ iff $\mathrm{Cov}(f(X), g(Y)) = 0$ for all square-integrable f, g
- Let's implement for RKHS functions $f \in \mathcal{H}_x$, $g \in \mathcal{H}_y$:

$$\mathbb{E}[f(X)]\,\mathbb{E}[g(Y)] = \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}_x} \langle \mu_{\mathbb{Q}}, g \rangle_{\mathcal{H}_y}$$

Independence

- $X \perp\!\!\!\perp Y$ iff $\mathrm{Cov}(f(X), g(Y)) = 0$ for all square-integrable f, g
- Let's implement for RKHS functions $f \in \mathcal{H}_x$, $g \in \mathcal{H}_y$:

$$
\begin{aligned}
\mathbb{E}[f(X)]\,\mathbb{E}[g(Y)] &= \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}_x} \langle \mu_{\mathbb{Q}}, g \rangle_{\mathcal{H}_y} \\
&= \langle f, (\mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}) g \rangle_{\mathcal{H}_x}
\end{aligned}
$$

Independence

- $X \perp\!\!\!\perp Y$ iff $\mathrm{Cov}(f(X), g(Y)) = 0$ for all square-integrable f, g
- Let's implement for RKHS functions $f \in \mathcal{H}_x$, $g \in \mathcal{H}_y$:

$$
\begin{aligned}
\mathbb{E}[f(X)]\,\mathbb{E}[g(Y)] &= \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}_x} \langle \mu_{\mathbb{Q}}, g \rangle_{\mathcal{H}_y} \\
&= \langle f, (\mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}) g \rangle_{\mathcal{H}_x} \\
\mathbb{E}[f(X) g(Y)]
\end{aligned}
$$

Independence

- $X \perp\!\!\!\perp Y$ iff $\mathrm{Cov}(f(X), g(Y)) = 0$ for all square-integrable f, g
- Let's implement for RKHS functions $f \in \mathcal{H}_x$, $g \in \mathcal{H}_y$:

$$\begin{aligned}
\mathbb{E}[f(X)]\,\mathbb{E}[g(Y)] &= \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}_x} \langle \mu_{\mathbb{Q}}, g \rangle_{\mathcal{H}_y} \\
&= \langle f, (\mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}) g \rangle_{\mathcal{H}_x} \\
\mathbb{E}[f(X) g(Y)] &= \mathbb{E}[\langle f, k_x(X, \cdot) \rangle_{\mathcal{H}_x} \langle k_y(Y, \cdot), g \rangle_{\mathcal{H}_y}]
\end{aligned}$$

Independence

- $X \perp\!\!\!\perp Y$ iff $\mathrm{Cov}(f(X), g(Y)) = 0$ for all square-integrable f, g
- Let's implement for RKHS functions $f \in \mathcal{H}_x, g \in \mathcal{H}_y$:

$$
\begin{aligned}
\mathbb{E}[f(X)]\,\mathbb{E}[g(Y)] &= \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}_x} \langle \mu_{\mathbb{Q}}, g \rangle_{\mathcal{H}_y} \\
&= \langle f, (\mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}) g \rangle_{\mathcal{H}_x} \\
\mathbb{E}[f(X) g(Y)] &= \mathbb{E}[\langle f, k_x(X, \cdot) \rangle_{\mathcal{H}_x} \langle k_y(Y, \cdot), g \rangle_{\mathcal{H}_y}] \\
&= \langle f, \mathbb{E}[k_x(X, \cdot) \otimes k_y(Y, \cdot)]\, g \rangle_{\mathcal{H}_x}
\end{aligned}
$$

Independence

- $X \perp\!\!\!\perp Y$ iff $\mathrm{Cov}(f(X), g(Y)) = 0$ for all square-integrable f, g
- Let's implement for RKHS functions $f \in \mathcal{H}_x$, $g \in \mathcal{H}_y$:

$$\begin{aligned}
\mathbb{E}[f(X)]\,\mathbb{E}[g(Y)] &= \langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}_x} \langle \mu_{\mathbb{Q}}, g \rangle_{\mathcal{H}_y} \\
&= \langle f, (\mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}) g \rangle_{\mathcal{H}_x} \\
\mathbb{E}[f(X) g(Y)] &= \mathbb{E}[\langle f, k_x(X, \cdot) \rangle_{\mathcal{H}_x} \langle k_y(Y, \cdot), g \rangle_{\mathcal{H}_y}] \\
&= \langle f, \mathbb{E}[k_x(X, \cdot) \otimes k_y(Y, \cdot)]\, g \rangle_{\mathcal{H}_x} \\
\mathrm{Cov}(f(X), g(Y)) &= \langle f, C_{XY}\, g \rangle_{\mathcal{H}_x}
\end{aligned}$$

where $C_{XY} : \mathcal{H}_y \to \mathcal{H}_x$ is

$$\mathbb{E}[k_x(X, \cdot) \otimes k_y(Y, \cdot)] - \mathbb{E}[k_x(X, \cdot)] \otimes \mathbb{E}[k_y(Y, \cdot)]$$

Cross-covariance operator and independence

- $\mathrm{Cov}(f(X), g(Y)) = \langle f, C_{XY}\, g \rangle_{\mathcal{H}_x}$
- $C_{XY} = \mathbb{E}[k_x(X, \cdot) \otimes k_y(Y, \cdot)] - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}$

Cross-covariance operator and independence

- $\mathrm{Cov}(f(X), g(Y)) = \langle f, C_{XY}\, g\rangle_{\mathcal{H}_x}$
- $C_{XY} = \mathbb{E}[k_x(X,\cdot) \otimes k_y(Y,\cdot)] - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}$
- If $X \perp\!\!\!\perp Y$, then $C_{XY} = 0$

Cross-covariance operator and independence

- $\mathrm{Cov}(f(X), g(Y)) = \langle f, C_{XY}\, g \rangle_{\mathcal{H}_x}$
- $C_{XY} = \mathbb{E}[k_x(X, \cdot) \otimes k_y(Y, \cdot)] - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}$
- If $X \perp\!\!\!\perp Y$, then $C_{XY} = 0$
- If $C_{XY} = 0$, $\mathrm{Cov}(f(X), g(Y)) = 0 \quad \forall f \in \mathcal{H}_x, g \in \mathcal{H}_y$

Cross-covariance operator and independence

- $\mathrm{Cov}(f(X), g(Y)) = \langle f, C_{XY}\, g\rangle_{\mathcal{H}_x}$
- $C_{XY} = \mathbb{E}[k_x(X,\cdot) \otimes k_y(Y,\cdot)] - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}$
- If $X \perp\!\!\!\perp Y$, then $C_{XY} = 0$
- If $C_{XY} = 0$, $\mathrm{Cov}(f(X), g(Y)) = 0 \quad \forall f \in \mathcal{H}_x, g \in \mathcal{H}_y$
- If k_x, k_y are characteristic:
  - $C_{XY} = 0$ implies $X \perp\!\!\!\perp Y$ [Szabó/Sriperumbudur JMLR-18]

Cross-covariance operator and independence

- $\mathrm{Cov}(f(X), g(Y)) = \langle f, C_{XY}\, g\rangle_{\mathcal{H}_x}$
- $C_{XY} = \mathbb{E}[k_x(X, \cdot) \otimes k_y(Y, \cdot)] - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}$
- If $X \perp\!\!\!\perp Y$, then $C_{XY} = 0$
- If $C_{XY} = 0$, $\mathrm{Cov}(f(X), g(Y)) = 0 \quad \forall f \in \mathcal{H}_x, g \in \mathcal{H}_y$
- If k_x, k_y are characteristic:
  - $C_{XY} = 0$ implies $X \perp\!\!\!\perp Y$ [Szabó/Sriperumbudur JMLR-18]
  - $X \perp\!\!\!\perp Y$ iff $C_{XY} = 0$

Cross-covariance operator and independence

- $\mathrm{Cov}(f(X), g(Y)) = \langle f, C_{XY}\, g\rangle_{\mathcal{H}_x}$
- $C_{XY} = \mathbb{E}[k_x(X, \cdot) \otimes k_y(Y, \cdot)] - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}$
- If $X \perp\!\!\!\perp Y$, then $C_{XY} = 0$
- If $C_{XY} = 0$, $\mathrm{Cov}(f(X), g(Y)) = 0 \quad \forall f \in \mathcal{H}_x, g \in \mathcal{H}_y$
- If k_x, k_y are characteristic:
  - $C_{XY} = 0$ implies $X \perp\!\!\!\perp Y$ [Szabó/Sriperumbudur JMLR-18]
  - $X \perp\!\!\!\perp Y$ iff $C_{XY} = 0$
  - $X \perp\!\!\!\perp Y$ iff $0 = \|C_{XY}\|^2_{\mathrm{HS}}$ (sum squared singular values)
    - HSIC: "Hilbert-Schmidt Independence Criterion"

HSIC

$$C_{XY} = \mathbb{E}[k_x(X,\cdot) \otimes k_y(Y,\cdot)] - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}$$

$$\|C_{XY}\|_{\mathrm{HS}}^2 = \|\mu_{\mathbb{P}_{XY}} - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}\|_{\mathcal{H}_x \otimes \mathcal{H}_y}^2$$

HSIC

$$
\begin{aligned}
C_{XY} &= \mathbb{E}[k_x(X,\cdot) \otimes k_y(Y,\cdot)] - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}} \\
\|C_{XY}\|_{\mathrm{HS}}^2 &= \|\mu_{\mathbb{P}_{XY}} - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}\|_{\mathcal{H}_x \otimes \mathcal{H}_y}^2 \\
&= \mathrm{MMD}(\mathbb{P}_{XY}, \mathbb{P} \times \mathbb{Q})^2
\end{aligned}
$$

HSIC

$$\begin{aligned}
C_{XY} &= \mathbb{E}[k_x(X,\cdot) \otimes k_y(Y,\cdot)] - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}} \\
\|C_{XY}\|^2_{\mathrm{HS}} &= \|\mu_{\mathbb{P}_{XY}} - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}\|^2_{\mathcal{H}_x \otimes \mathcal{H}_y} \\
&= \mathrm{MMD}(\mathbb{P}_{XY}, \mathbb{P} \times \mathbb{Q})^2 \\
&= \mathbb{E}[k_x(X,X')k_y(Y,Y')] \\
&\quad - 2\,\mathbb{E}[k_x(X,X')k_x(Y,Y'')] \\
&\quad + \mathbb{E}[k_x(X,X')]\,\mathbb{E}[k_y(Y,Y')]
\end{aligned}$$

HSIC

$$
\begin{aligned}
C_{XY} &= \mathbb{E}[k_x(X,\cdot) \otimes k_y(Y,\cdot)] - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}} \\
\|C_{XY}\|_{\mathrm{HS}}^2 &= \|\mu_{\mathbb{P}_{XY}} - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}\|_{\mathcal{H}_x \otimes \mathcal{H}_y}^2 \\
&= \mathrm{MMD}(\mathbb{P}_{XY}, \mathbb{P} \times \mathbb{Q})^2 \\
&= \mathbb{E}[k_x(X,X')k_y(Y,Y')] \\
&\quad - 2\,\mathbb{E}[k_x(X,X')k_x(Y,Y'')] \\
&\quad + \mathbb{E}[k_x(X,X')]\,\mathbb{E}[k_y(Y,Y')]
\end{aligned}
$$

- Linear case: C_{XY} is cross-covariance matrix, HSIC is squared Frobenius norm

HSIC

$$
\begin{aligned}
C_{XY} &= \mathbb{E}[k_x(X,\cdot) \otimes k_y(Y,\cdot)] - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}} \\
\|C_{XY}\|_{\mathrm{HS}}^2 &= \|\mu_{\mathbb{P}_{XY}} - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}\|_{\mathcal{H}_x \otimes \mathcal{H}_y}^2 \\
&= \mathrm{MMD}(\mathbb{P}_{XY}, \mathbb{P} \times \mathbb{Q})^2 \\
&= \mathbb{E}[k_x(X,X')k_y(Y,Y')] \\
&\quad - 2\,\mathbb{E}[k_x(X,X')k_x(Y,Y'')] \\
&\quad + \mathbb{E}[k_x(X,X')]\,\mathbb{E}[k_y(Y,Y')]
\end{aligned}
$$

- Linear case: C_{XY} is cross-covariance matrix, HSIC is squared Frobenius norm
- Default estimator (biased, but simple): $\langle HK_XH, K_Y\rangle_F$, $H = I - \mathbf{1}\mathbf{1}^\mathsf{T}$

HSIC

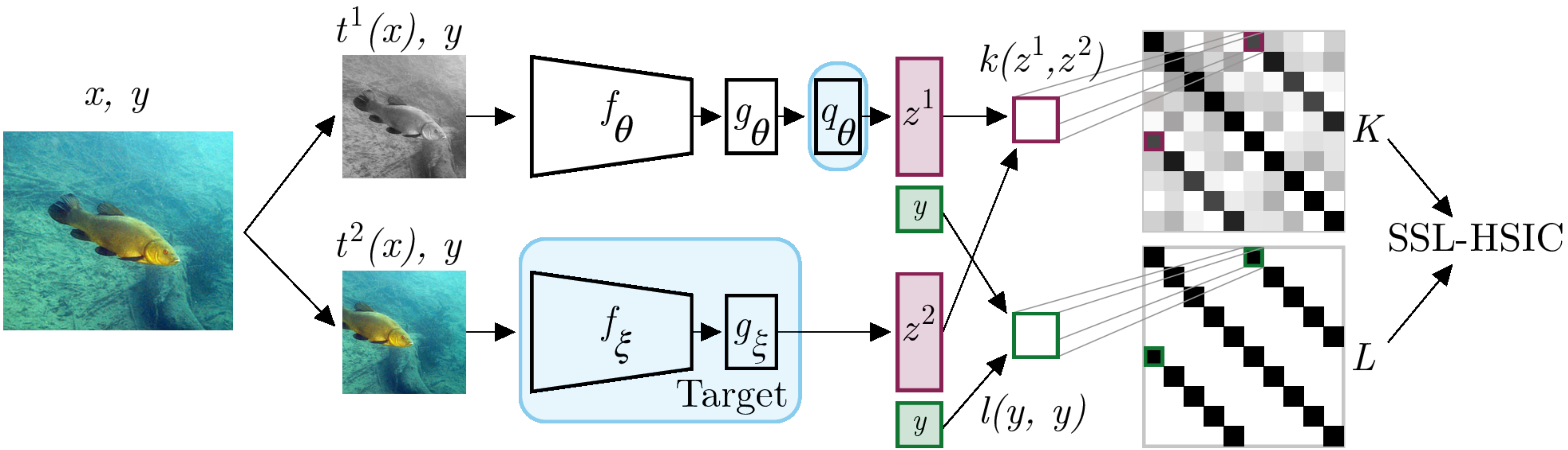
$$
\begin{aligned}
C_{XY} &= \mathbb{E}[k_x(X,\cdot) \otimes k_y(Y,\cdot)] - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}} \\
\|C_{XY}\|_{\mathrm{HS}}^2 &= \|\mu_{\mathbb{P}_{XY}} - \mu_{\mathbb{P}} \otimes \mu_{\mathbb{Q}}\|_{\mathcal{H}_x \otimes \mathcal{H}_y}^2 \\
&= \mathrm{MMD}(\mathbb{P}_{XY}, \mathbb{P} \times \mathbb{Q})^2 \\
&= \mathbb{E}[k_x(X,X')k_y(Y,Y')] \\
&\quad - 2\,\mathbb{E}[k_x(X,X')k_x(Y,Y'')] \\
&\quad + \mathbb{E}[k_x(X,X')]\,\mathbb{E}[k_y(Y,Y')] \\
&= \mathbb{E}_{\substack{f \sim \mathcal{GP}(0,k_x) \\ g \sim \mathcal{GP}(0,k_y)}} \left[\mathrm{Cov}(f(X), g(Y))^2\right]
\end{aligned}
$$

- Linear case: C_{XY} is cross-covariance matrix, HSIC is squared Frobenius norm
- Default estimator (biased, but simple): $\langle HK_XH, K_Y\rangle_F$, $H = I - \mathbf{1}\mathbf{1}^\mathsf{T}$

HSIC applications

- Independence testing [Gretton+ NeurIPS-07]
- Clustering [Song+ ICML-07]
- Feature selection [Song+ JMLR-12]
- HSIC Bottleneck: alternative to backprop [Ma+ AAAI-20]
    - biologically plausible(ish) [Pogodin+ NeurIPS-20]
    - more robust [Wang+ NeurIPS-21]
- Self-supervised learning [Li+ NeurIPS-21]
    - maybe better explanation of why InfoNCE/etc work
- $\vdots$
- Broadly: easier-to-estimate, sometimes-nicer version of mutual information

Example: SSL-HSIC [Li+ NeurIPS-21]



- Maximizes dependence between image features f and its identity on a minibatch
- Using a learned deep kernel based on g

Recap

- Point embedding $k(X, \cdot)$: if $f \in \mathcal{H}$ then $\langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} = \mathbb{E}_{X \sim \mathbb{P}}\, f(X)$
- Mean embedding $\mu_{\mathbb{P}} = \mathbb{E}\, k(X, \cdot)$: if $f \in \mathcal{H}$ then $\langle f, \mu_{\mathbb{P}} \rangle_{\mathcal{H}} = \mathbb{E}_{X \sim \mathbb{P}}\, f(X)$

- $\mathrm{MMD}(\mathbb{P}, \mathbb{Q}) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}$ is 0 iff $\mathbb{P} = \mathbb{Q}$ (for characteristic kernels)
- $\mathrm{HSIC}(X, Y) = \|C_{XY}\|_{\mathrm{HS}} = \mathrm{MMD}(\mathbb{P}_{XY}, \mathbb{P} \times \mathbb{Q})^2$ is 0 iff $X \perp\!\!\!\perp Y$

  (for characteristic k_x, k_y...or slightly weaker)

- Often need to **learn a kernel** for good performance on complicated data
  - Can often do end-to-end for downstream loss, asymptotic test power, ...

More resources

- Berlinet and Thomas-Agnan, *RKHS in Probability and Statistics*
  - kernels in general + mean embedding basics
- Steinwart and Christmann, *Support Vector Machines*
  - kernels in general, learning theory
- Course slides by Julien Mairal + Jean-Philippe Vert
  - kernels in general, learning theory
- Course materials by Arthur Gretton
  - kernels in general, mean embeddings, MMD/HSIC
- Connections to Gaussian processes [Kanagawa+ 'GPs and Kernel Methods' 2018]
- Mean embeddings: survey [Muandet+ 'Kernel Mean Embedding of Distributions']
- These slides are at `djsutherland.ml/slides/like23`