S&DS 365 / 665

**Intermediate Machine Learning**

# Mercer Kernels

September 11

Yale

# Please note

- Materials posted to `http://interml.ydata123.org`
- Readings from “Probabilistic Machine Learning: An Introduction”
- `https://probml.github.io/pml-book/book1.html`

# Some reminders

- Assn 1 posted today
- Due at midnight, September 25 (two weeks)
- Topics: Lasso, smoothing, Mercer kernels, leave-one-out

# Topics for today

- Calculation from last class
- Mercer kernels

# Bias-variance for density estimation

Recall from last class: We derived an expression for the squared bias of kernel density estimation using a Taylor expansion

The calculation is very similar for the variance

# Bias

Recall:

$$
\begin{aligned}
\mathbb{E}\widehat{f}(x) &= \frac{1}{nh^p}\sum_{i=1}^{n}\mathbb{E}K\left(\frac{X_i - x}{h}\right) \\
&= \frac{1}{h^p}\int K\left(\frac{u-x}{h}\right) f(u)\,du \\
&= \int K(v)\, f(x+hv)\,dv \\
&= \int K(v)\left(f(x) + hv^T\nabla f(x) + \frac{1}{2}h^2 v^T\nabla^2 f(x)v + o(h^2)\right) dv \\
&= f(x) + C(x)h^2 + o(h^2)
\end{aligned}
$$

using $\int K(u)\,du = 1$ and $\int uK(u)\,du = 0$

# Variance

By a similar argument, using $\mathrm{Var}(X) \leq \mathbb{E}X^2$

$$\begin{aligned}
\mathbb{E}\widehat{f}(x)^2 &\leq C_2 \frac{1}{n^2 h^{2p}} \sum_{i=1}^{n} \mathbb{E} K\left(\frac{X_i - x}{h}\right)^2 \\
&= C_2 \frac{1}{n h^{2p}} \int K\left(\frac{u - x}{h}\right)^2 f(u)\, du \\
&= C_2 \frac{f(x)}{n h^p} \int K(v)^2\, dv + o\left(\frac{1}{n h^p}\right) \\
&= C_2 \frac{f(x)}{n h^p} + o\left(\frac{1}{n h^p}\right)
\end{aligned}$$

assuming $nh^p \to \infty$.

# Risk

This gives

$$\text{bias}^2 \approx h^4$$

$$\text{var} \approx \frac{1}{nh^p}$$

On the assignment, you'll work with these expressions to reason about the smallest possible risk and the curse of dimensionality

# Generative models

- The KDE is a *generative model*
- We can sample from the density to “generate” a new data point
- What is an algorithm for sampling from the estimated distribution?

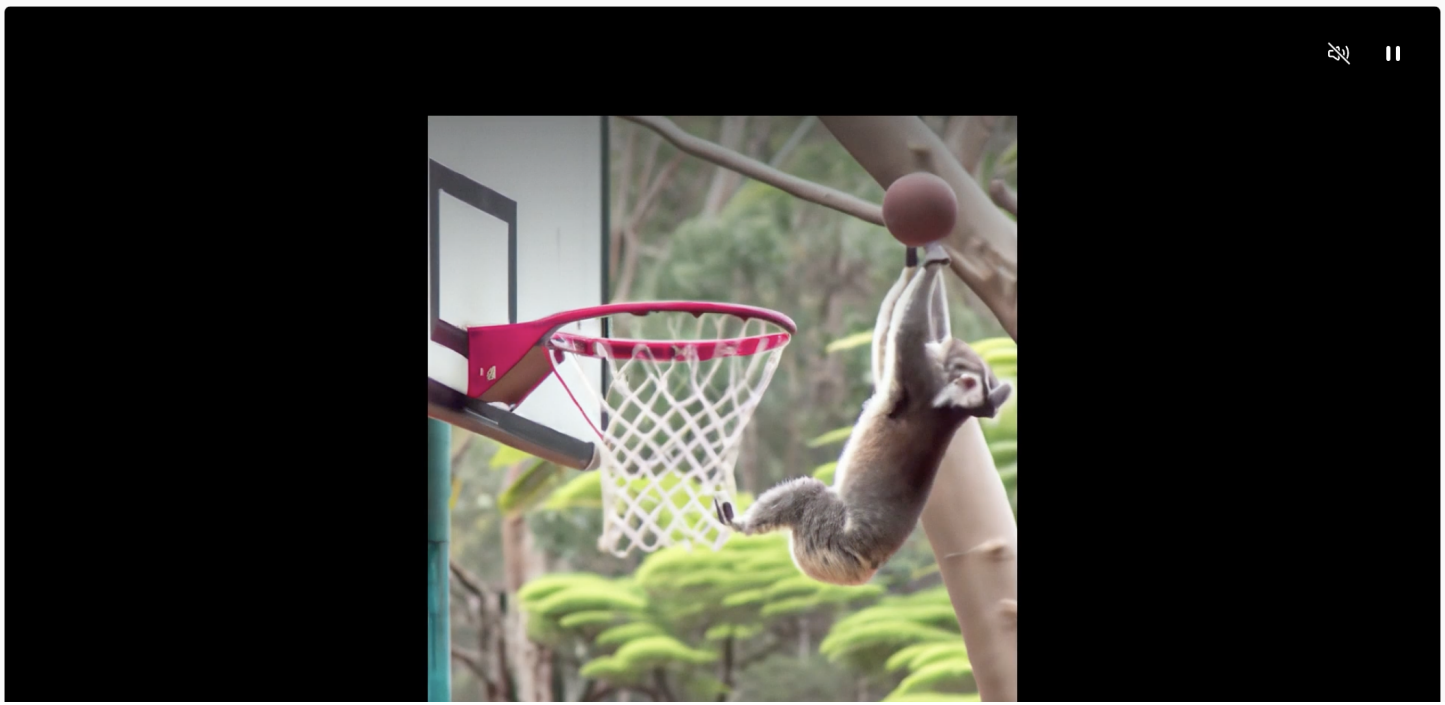
# Generative models

1. Sample an index $i$ uniformly from 1 to $n$
2. Sample a point $x$ from a Gaussian with mean $X_i$ and variance $h^2$

# Generative models

DALL·E 2 is an AI system that can create realistic images and art from a description in natural language.

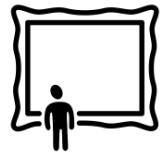
Try DALL·E ↗ Follow on Instagram ↗



# Generative models

As we'll see later in the course, Transformers can be naturally seen as a form of kernel smoothing and kernel density estimation.

# Mercer Kernels: The big picture



Instead of using local smoothing, we can optimize the fit to the data subject to regularization (penalization). Choose $\widehat{m}$ to minimize

$$\sum_i (Y_i - \widehat{m}(X_i))^2 + \lambda \text{ penalty}(\widehat{m})$$

where penalty($\widehat{m}$) is a *roughness penalty*.

$\lambda$ is a parameter that controls the amount of smoothing.

How do we construct a penalty that measures roughness? One approach is: *Mercer Kernels* and *RKHS = Reproducing Kernel Hilbert Spaces.*

# What is a Mercer Kernel?

A kernel is a bivariate function $K(x, x')$. We think of this as a measure of "similarity" between points $x$ and $x'$.

# What is a Mercer Kernel?

A kernel is a bivariate function $K(x, x')$. We think of this as a measure of "similarity" between points $x$ and $x'$.

A Mercer kernel has a special property: For any set of points $x_1, \ldots, x_n$ the $n \times n$ matrix

$$\mathbb{K} = \left[K(x_i, x_j)\right]$$

is positive semidefinite (no negative eigenvalues)

# What is a Mercer Kernel?

A kernel is a bivariate function $K(x, x')$. We think of this as a measure of "similarity" between points $x$ and $x'$.

A Mercer kernel has a special property: For any set of points $x_1, \ldots, x_n$ the $n \times n$ matrix

$$\mathbb{K} = \left[K(x_i, x_j)\right]$$

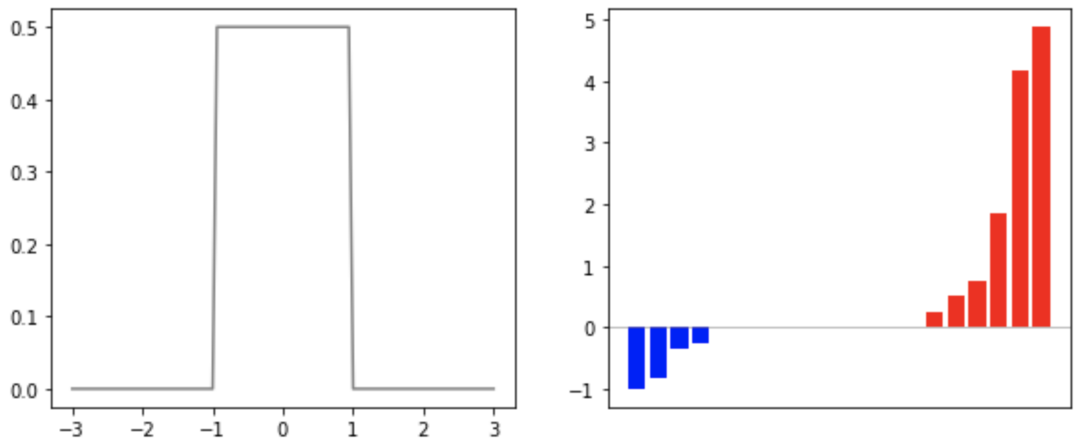
is positive semidefinite (no negative eigenvalues)

This property has many important (and beautiful!) mathematical consequences. It is a characterization of Mercer kernels.

Which of the kernels we used for smoothing are Mercer?

(demo)

```
In [8]: plot_eigenvalues(boxcar, 20)
```



# Mercer Kernels: Key example

A Gaussian gives us a Mercer kernel:

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

Note: Here we fix the bandwidth $h$.

# What is a Mercer Kernel?

A *Mercer kernel* $K(x, x')$ is symmetric and positive semidefinite bivariate function:

$$\int \int f(x) f(x') K(x, x') \, dx \, dx' \geq 0$$

for all (univariate) functions $f$.

# Basis functions

We can create a set of *basis functions* based on $K$.

Fix $z$ and think of $K(z, x)$ as a function of $x$. That is,

$$K(z, x) = K_z(x)$$

is a function of the second argument, with the first argument fixed.

# Defining a norm from the kernel

Because of the positive semidefinite property, we can create an *inner product* and *norm* over the span of these functions

If $f(x) = \sum_r \alpha_r K_{z_r}(x)$, $g(x) = \sum_s \beta_s K_{y_s}(x)$, the inner product is

$$\begin{aligned}\langle f, g \rangle_K &= \sum_r \sum_s \alpha_r \beta_s K(z_r, y_s) \\ &= \alpha^T \mathbb{K} \beta\end{aligned}$$

where $\mathbb{K} = [K(z_r, y_s)]$

# Defining a norm from the kernel

Because of the positive semidefinite property, we can create an *inner product* and *norm* over the span of these functions

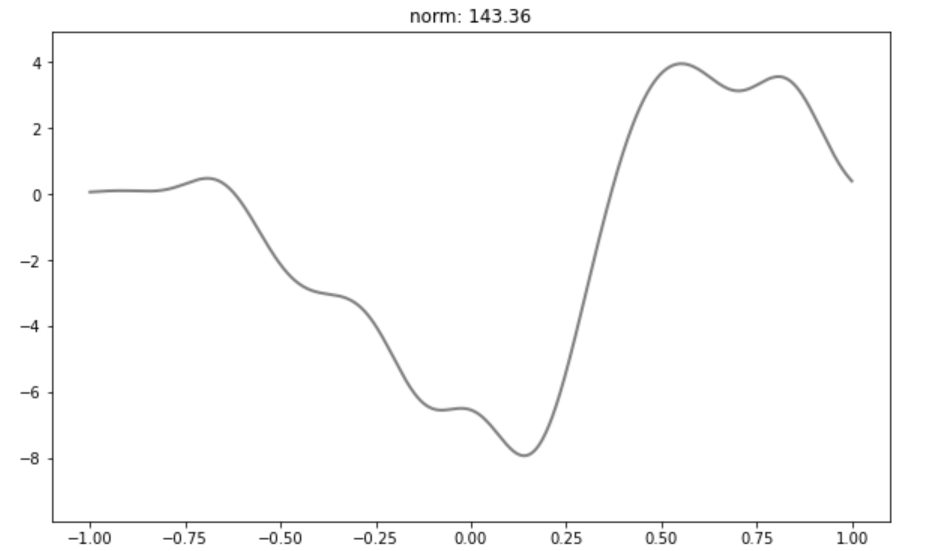
The norm is

$$\begin{aligned}\|f\|_K^2 = \langle f, f\rangle_K &= \sum_r \sum_s \alpha_r \alpha_s K(z_r, z_s) \\ &= \alpha^T \mathbb{K} \alpha \geq 0\end{aligned}$$

# What do the functions look like?

# (demo)

```
plot_function(x, f, norm, sleeptime=1)
```



# Defining a Hilbert space from the kernel

This gives us an infinite dimensional space of functions with a geometry — a notion of angle from the inner product $\langle \cdot, \cdot \rangle_K$

Technically speaking, we define the Hilbert space by "completing" the functions to include the limits of all Cauchy sequences with respect to the norm.

# Defining a Hilbert space from the kernel

This gives us an infinite dimensional space of functions with a geometry — a notion of angle from the inner product $\langle \cdot, \cdot \rangle_K$

It is called a *Reproducing Kernel Hilbert Space* (RKHS) because

$$\langle f, K_x(\cdot) \rangle_K = f(x)$$

That is, the kernel "reproduces" the values of the functions through the inner products

---

Technically speaking, we define the Hilbert space by "completing" the functions to include the limits of all Cauchy sequences with respect to the norm.

# Defining a Hilbert space from the kernel

This gives us an infinite dimensional space of functions with a geometry — a notion of angle from the inner product $\langle \cdot, \cdot \rangle_K$

It is called a *Reproducing Kernel Hilbert Space* (RKHS) because

$$\langle f, K_x(\cdot) \rangle_K = f(x)$$

That is, the kernel "reproduces" the values of the functions through the inner products

Exercise: Verify this identity!

---

Technically speaking, we define the Hilbert space by "completing" the functions to include the limits of all Cauchy sequences with respect to the norm.

# Nonparametric regression using Mercer kernels

The norm gives us a way to penalize functions for being too complex.

We carry out least squares regression subject to this penalty:

Minimize

$$\sum_{i=1}^{n}(Y_i - m(X_i))^2 + \lambda \|m\|_K^2.$$

over the RKHS of functions

# Dilemma?

How do we carry out this penalized regression? It looks complicated!

Or maybe intractable...

# Linear algebra to the rescue!

**Representer Theorem**

Let $\widehat{m}$ minimize

$$J(m) = \sum_{i=1}^{n} (Y_i - m(X_i))^2 + \lambda \|m\|_K^2.$$

Then

$$\widehat{m}(x) = \sum_{i=1}^{n} \alpha_i K(X_i, x)$$

for some $\alpha_1, \ldots, \alpha_n$.

So, we only need to find the coefficients

$$\alpha = (\alpha_1, \ldots, \alpha_n).$$

# Mercer kernel regression

Plug $\widehat{m}(x) = \sum_{i=1}^{n} \alpha_i K(X_i, x)$ into $J$:

$$J(\alpha) = \|Y - \mathbb{K}\alpha\|^2 + \lambda \alpha^T \mathbb{K} \alpha$$

where $\mathbb{K}_{jk} = K(X_j, X_k)$

Now we find $\alpha$ to minimize $J$. We get (Assn 1):

$$\widehat{\alpha} = (\mathbb{K} + \lambda I)^{-1} Y$$

$$\widehat{m}(x) = \sum_i \widehat{\alpha}_i K(X_i, x)$$

# Mercer kernel regression

The estimator depends on the amount of regularization $\lambda$.

Again, there is a bias-variance tradeoff.

We choose $\lambda$ by cross-validation. This is like the bandwidth in smoothing kernel regression.

# Takeaways

- Mercer kernels have a special property: When restricted to a finite sample they give positive semidefinite matrices
- This allows us to define an inner product and a norm
- We use the norm to *penalize* functions for being too rough

The underlying math is rich—see the notes if you want to learn more!

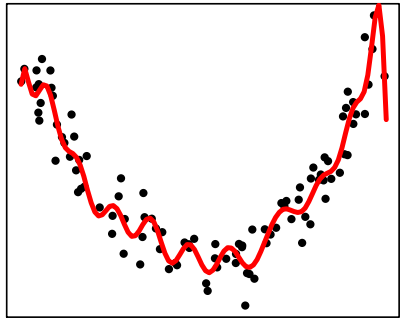
# Smoothing Kernels *Versus* Mercer Kernels

*Smoothing kernels*: bandwidth $h$ controls the amount of smoothing.

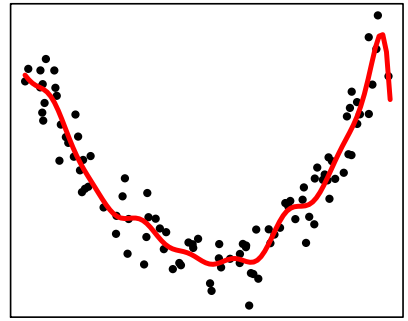
*Mercer kernels*: norm $\|f\|_K$ controls the amount of smoothing.

*In practice these two methods give answers that are very similar.*

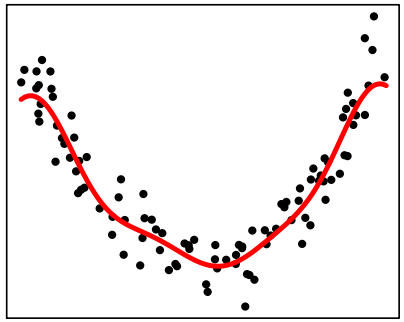
# Mercer Kernels: Examples



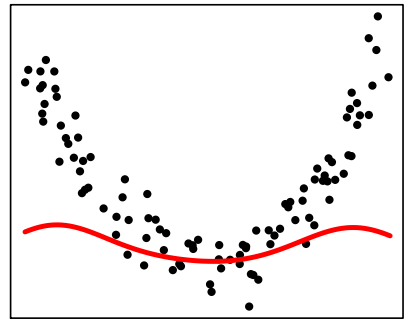
very small lambda



small lambda



medium lambda



large lambda

# Kernels from features—and vice-versa

If $x \to \varphi(x) \in \mathbb{R}^d$ is a feature mapping, we can define a Mercer kernel by

$$K(x, x') = \varphi(x)^T \varphi(x')$$

Conversely, for any Mercer kernel we can derive the corresponding feature map (from the spectral theorem)

---

See the slides at the end

# The importance of being Kernelist

- Mercer kernels play a central role in machine learning
  - Can define similarity functions that are kernels for all kinds of data — graphs, molecules, text documents
  - Gaussian processes
  - Modern understanding of deep neural networks

# Summary for today

- Smoothing methods compute local averages, weighting points by a kernel. The details of the kernel don't matter much
- Mercer kernels using penalization rather than smoothing
- Defining property: Matrix $\mathbb{K}$ is always positive semidefinite
- Equivalent to a type of ridge regression in function space
- The curse of dimensionality limits use of both approaches

# Some technical details (optional)

# Defining the inner product

Check that it is well defined:

If $f = \sum_r \alpha_r K(z_r, \cdot)$, $g = \sum_s \beta_s K(y_s, \cdot)$, the inner product is

$$\begin{aligned}\langle f, g \rangle_K &= \sum_r \sum_s \alpha_r \beta_s K(z_r, y_s) \\ &= \sum_r \alpha_r g(z_r) \\ &= \sum_s \beta_s f(y_s)\end{aligned}$$

using the reproducing property $\langle f, K(x, \cdot) \rangle = f(x)$

# Representer theorem: Proof sketch

We can write any $f \in \mathcal{H}_K$ as

$$f(x) = \sum_i \alpha_i K(X_i, x) + v(x)$$

where $v$ is orthogonal to the span of the functions $K(X_i, \cdot)$

By the reproducing property, $f(X_i)$ does not depend on $v$, and

$$\|f\|_K^2 = \alpha^T \mathbb{K} \alpha + \|v\|_K^2.$$

So, it must be that the minimizing function has $v = 0$

# Feature maps

If $M$ is symmetric, positive semidefinite matrix, can write

$$M = U^T \Lambda U$$

where $U$ is an orthogonal matrix. Can rewrite this as

$$M = \Phi^T \Phi$$

where

$$\Phi = \sqrt{\Lambda} U$$

This transformation allows us to define *features* or *feature maps* for Mercer kernels

# Features for Mercer kernels

Eigen-decomposition: $\{\psi_j\}$, $\{\lambda_j\}$ where

$$\int K(x,y)\psi_j(y)dy = \lambda_j\psi_j(x) \quad (K\psi_j = \lambda_j\psi_j)$$

The spectral theorem (see previous slide for finite dimensional case) tells us that

$$K(x,y) = \sum_{j=1}^{\infty} \lambda_j\psi_j(x)\psi_j(y)$$

We can think of the kernel in terms of the *feature map*

$$x \longrightarrow \Phi(x) = (\sqrt{\lambda_1}\psi_1(x), \sqrt{\lambda_2}\psi_2(x), \ldots)$$

# Features for Mercer kernels (continued)

Since $\psi_j$ forms an orthonormal basis, can write any function $f$ as

$$f(x) = \sum_{r=1}^{\infty} a_r \psi_r(x)$$

By construction of the RKHS, can also write it as

$$f(x) = \sum_{j} \alpha_j K(x_j, x)$$

It follows that

$$\|f\|_K^2 = \sum_{r=1}^{\infty} \frac{a_r^2}{\lambda_r}$$

*The functions that are smooth in the RKHS assign small weight to eigenfunctions with small eigenvalues*