

Lecture 6: Kernel Methods

Readings: Bach (Ch. 7), ESL (Ch. 5.8); code

Soon Hoe Lim

September 11, 2025

Outline

- ➊ Connecting the Dots
- ➋ Kernel Methods: Introduction
- ➌ Properties of Kernels
- ➍ Kernel Ridge Regression
- ➎ The Representer Theorem
- ➏ Kernel Methods in Practice
- ➐ Approximations with Random Features
- ➑ Exercises
- ➒ Appendix: Optional Materials

Connecting the Dots: The Big Picture

We've encountered several powerful, but seemingly distinct, ideas:

- ▶ In our study of SVMs, we saw a clever **“kernel trick”** that created complex non-linear boundaries by implicitly mapping data to a new feature space.
- ▶ In our lecture on non-linear modeling, we saw that **smoothing splines** find a flexible function by minimizing a combined “loss + smoothness penalty”.
- ▶ Both the ridge regression and SVC also use a “loss + regularization penalty” to control complexity.

This raises some deep questions:

- ▶ Is the “kernel trick” a one-off gimmick just for SVMs, or is it a more general recipe?
- ▶ Is there a formal connection between a penalty on weights like $\|\beta\|^2$ and a penalty on a function's “wiggleness”?
- ▶ Can we build a **single, unified theory** that explains all of these?

In this lecture, we will reveal the beautiful and powerful theory of **kernel methods** that connects all these dots.

Kernel Methods: Introduction

- ▶ General linear basis models (see Lecture 5) work by transforming inputs $\mathbf{x}_i \in \mathbb{R}^d$ via a class of functions $\{\phi_j\}$ into \mathbb{R} . We call these functions **feature maps**.
- ▶ They extract useful features from the input data and allow us to use a linear model on the resulting space.
- ▶ We will develop this idea in greater generality in the context of kernel methods.

Least Squares Revisited

We begin by revisiting the ℓ_2 -regularized least squares problem (ridge regression) with respect to feature maps $\{\phi_j\}_{j=0}^{M-1}$.

The objective is to minimize:

$$R_{\text{emp}}(\beta) = \frac{1}{2N} \|\Phi\beta - y\|^2 + \lambda \|\beta\|^2$$

where $\Phi_{ij} = \phi_j(\mathbf{x}_i)$ and the regularization parameter $\lambda > 0$.

The solution is given by:

$$\hat{\beta} = (\Phi^T \Phi + \lambda N \mathbf{I}_M)^{-1} \Phi^T y$$

The prediction on a new sample $\mathbf{x} \in \mathbb{R}^d$ is:

$$\hat{f}(\mathbf{x}) = \phi(\mathbf{x})^T \hat{\beta}$$

where $\phi(\mathbf{x}) = (\phi_0(\mathbf{x}), \dots, \phi_{M-1}(\mathbf{x}))$.

Rewriting the Solution: The Dual Form

We can rewrite the regularized least squares solution in another way.

Proposition 1: Matrix Inversion Lemma

The solution $\hat{\beta}$ can be expressed as:

$$\hat{\beta} = (\Phi^T \Phi + \lambda N \mathbf{I}_M)^{-1} \Phi^T y = \Phi^T (\Phi \Phi^T + \lambda N \mathbf{I}_N)^{-1} y$$

Proof.

We show that $(\Phi^T \Phi + \lambda N \mathbf{I}_M) \Phi^T (\Phi \Phi^T + \lambda N \mathbf{I}_N)^{-1} y = \Phi^T y$.

$$\begin{aligned} & (\Phi^T \Phi + \lambda N \mathbf{I}_M) \Phi^T (\Phi \Phi^T + \lambda N \mathbf{I}_N)^{-1} y \\ &= (\Phi^T \Phi \Phi^T + \lambda N \Phi^T) (\Phi \Phi^T + \lambda N \mathbf{I}_N)^{-1} y \\ &= \Phi^T (\Phi \Phi^T + \lambda N \mathbf{I}_N) (\Phi \Phi^T + \lambda N \mathbf{I}_N)^{-1} y \\ &= \Phi^T y \end{aligned}$$

Multiplying both sides by $(\Phi^T \Phi + \lambda N \mathbf{I}_M)^{-1}$ gives the result. □

The Predictor in Dual Form

Using the dual form of $\hat{\beta}$, the predictor function $\hat{f}(\mathbf{x})$ can be written as:

$$\hat{f}(\mathbf{x}) = \phi(\mathbf{x})^T \hat{\beta} = \phi(\mathbf{x})^T \Phi^T (\Phi \Phi^T + \lambda N \mathbf{I}_N)^{-1} \mathbf{y}$$

By defining a vector of coefficients $\in \mathbb{R}^N$ as:

$$\alpha = (\Phi \Phi^T + \lambda N \mathbf{I}_N)^{-1} \mathbf{y}$$

We can rewrite the predictor as:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^N \alpha_i \phi(\mathbf{x})^T \phi(\mathbf{x}_i)$$

What have we achieved?

- ▶ The $N \times N$ matrix $\Phi \Phi^T$ has components $(\Phi \Phi^T)_{ij} = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$.
- ▶ The predictor formula now only depends on the feature map ϕ through the function $(\mathbf{x}, \mathbf{x}') \mapsto \phi(\mathbf{x})^T \phi(\mathbf{x}')$.

The Kernel Trick

We can define a kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ by:

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$$

The predictor \hat{f} only depends on the feature maps through this kernel k . Once k is known, we never need to compute the feature maps $\mathbf{x} \mapsto \phi(\mathbf{x})$ directly to make a prediction.

The predictor can be rewritten as:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^N [(G + \lambda N \mathbf{I}_N)^{-1} \mathbf{y}]_i k(\mathbf{x}, \mathbf{x}_i)$$

where G is the $N \times N$ Gram matrix with $G_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Advantage: If M is very large (even infinite!), as long as we can compute the kernel k efficiently, this is a huge saving. This is the key idea of kernel methods.

Basic Properties of Kernels

For any set of feature maps $\{\phi_j\}$, we can construct a kernel:

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') = \sum_{j=0}^{M-1} \phi_j(\mathbf{x}) \phi_j(\mathbf{x}')$$

Example: Kernels and Feature Maps

1. Let $d = 1, M = 2$ and $\phi(x) = (1, x)$. This is 1D simple linear regression. The kernel is:

$$k(x, x') = 1 + xx'$$

2. Consider $d = 2$ and the function $k(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^T \mathbf{x}')^2$. This corresponds to a feature map.

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= (1 + x_1 x'_1 + x_2 x'_2)^2 \\ &= [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1^2, x_2^2]^T [1, \sqrt{2}x'_1, \dots, x_2'^2] \end{aligned}$$

The feature map is $\phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1^2, x_2^2)$.

What is a Valid Kernel?

Can any function $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ be a kernel? **No.**

- ▶ A kernel must be symmetric in its arguments: $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$.
- ▶ It must satisfy $k(\mathbf{x}, \mathbf{x}) = \|\phi(\mathbf{x})\|^2 \geq 0$.
- ▶ For example, $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}' - 1$ cannot be a kernel since $k(\mathbf{0}, \mathbf{0}) = -1 < 0$.

This motivates a general restriction on what functions can be valid kernels.

Definition 1: Symmetric Positive Definite (SPD) Kernels

A function $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is called a symmetric positive definite (SPD) kernel if for any collection $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, the Gram matrix G with elements $G_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is symmetric and positive semi-definite.

(Recall: G is symmetric if $G_{ij} = G_{ji}$ and positive semi-definite if $c^T G c \geq 0$ for any $c \in \mathbb{R}^n$.)

Mercer's Theorem

If $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$, one can check that k is an SPD kernel. What is interesting is that the reverse is also true.

Theorem 1: Mercer's Theorem

For any SPD kernel k , there exists a Hilbert space (called a feature space) \mathcal{H} and a mapping $\phi : \mathbb{R}^d \rightarrow \mathcal{H}$ such that

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

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the inner product in \mathcal{H} . In general, \mathcal{H} may be infinite-dimensional.

This result follows from Mercer's theorem on representations of symmetric positive definite functions, and gives rise to the theory of reproducing kernel Hilbert spaces (RKHS) – not covered in this course, see Appendix for a brief introduction.

What is a Hilbert Space? An Intuition

For our purposes, a Hilbert space \mathcal{H} is a vector space that generalizes the familiar Euclidean space \mathbb{R}^M to potentially infinite dimensions.

- ▶ It is a **vector space**, so we can add vectors and scale them by constants.
- ▶ It has an **inner product** $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ (also called a dot product). This allows us to define geometric notions like angles and lengths.
- ▶ The inner product defines a **norm** (length) for any vector $\beta \in \mathcal{H}$ as $\|\beta\|_{\mathcal{H}} = \sqrt{\langle \beta, \beta \rangle_{\mathcal{H}}}$.
- ▶ (Technically, it must also be "complete," which ensures that limits and calculus work as we expect.)

We will mainly work with models where the feature space \mathcal{H} is \mathbb{R}^M . In this case, the inner product and norm are the familiar ones:

- ▶ **Inner Product:** $\langle \beta, \mathbf{z} \rangle = \beta^{\top} \mathbf{z} = \sum_{j=1}^M \beta_j z_j$
- ▶ **Euclidean Norm:** $\|\beta\| = \sqrt{\beta^{\top} \beta} = \sqrt{\sum_{j=1}^M \beta_j^2}$

The goal of kernel methods is to handle cases where M is very large or even infinite, without ever working in that space directly.

Examples of SPD Kernels

1. **Linear kernel:** $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$.
2. **Polynomial kernel:** $k(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^T \mathbf{x}')^m, m > 0$.
3. **Gaussian (RBF) kernel:** $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right), \sigma > 0$.
4. **Laplacian kernel:** $k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|), \gamma > 0$.
5. **Matérn Kernels¹:** $k_{\nu, \ell}(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\ell}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\ell}\right)$, for parameters $\nu, \ell > 0$, where Γ is the Gamma function and K_ν is a modified Bessel function of the second kind. For $\nu = 1/2$, this simplifies to the exponential kernel. More generally, a smaller ν results in less smooth functions, and as $\nu \rightarrow \infty$, the Matérn kernel converges to the Gaussian kernel. Popular choices are $\nu = 3/2$ and $\nu = 5/2$.
6. **Kernel on Sets:** For a finite set Ω , let $A, A' \subseteq \Omega$. $k(A, A') = 2^{|A \cap A'|}$ is an SPD kernel. This shows kernel methods can be applied to inputs other than \mathbb{R}^d .

¹Matérn kernels are frequently used in the setting of spatial statistics and for Gaussian processes.

Building New Kernels from Old

Proposition 2: Closure Properties of SPD Kernels

Suppose k_1, k_2, \dots is a collection of SPD kernels. Then the following are also SPD kernels:

1. **Scaling:** $k(\mathbf{x}, \mathbf{x}') = \alpha k_1(\mathbf{x}, \mathbf{x}')$ for any $\alpha > 0$.
2. **Addition:** $k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$.
3. **Normalization:** $k(\mathbf{x}, \mathbf{x}') = g(\mathbf{x})k_1(\mathbf{x}, \mathbf{x}')g(\mathbf{x}')$ for any function g .
4. **Limit:** $k(\mathbf{x}, \mathbf{x}') = \lim_{n \rightarrow \infty} k_n(\mathbf{x}, \mathbf{x}')$, if the limit exists.
5. **Product:** $k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}')$.

Proof.

See blackboard. □

Building Complex Kernels:

- ▶ $K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}') + K_2(\mathbf{x}, \mathbf{x}')$ combines different types of similarity
- ▶ $K(\mathbf{x}, \mathbf{x}') = \exp(K_1(\mathbf{x}, \mathbf{x}'))$ where K_1 is PSD creates "exponential" variants
- ▶ Polynomial kernels: $(\mathbf{x}^T \mathbf{x}' + c)^d$

⚠ $K(\mathbf{x}, \mathbf{x}') = \tanh(\mathbf{x}^T \mathbf{x}')$ is *not* always PSD, despite being used in practice!

The Kernel Trick Illustrated

Let's derive the explicit feature map for the Gaussian (RBF) kernel in 1D to see why using kernels is so powerful.

$$\begin{aligned}k(x, x') &= \exp\left(-\frac{1}{2\sigma^2}(x - x')^2\right) \\&= \exp\left(-\frac{x^2}{2\sigma^2}\right) \exp\left(\frac{xx'}{\sigma^2}\right) \exp\left(-\frac{x'^2}{2\sigma^2}\right) \\&= \exp\left(-\frac{x^2}{2\sigma^2}\right) \left(\sum_{m=0}^{\infty} \frac{(xx')^m}{\sigma^{2m} m!}\right) \exp\left(-\frac{x'^2}{2\sigma^2}\right) \\&= \phi(x)^T \phi(x')\end{aligned}$$

The corresponding feature map $\phi(x)$ is:

$$\phi(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \left[1, \sqrt{\frac{1}{\sigma^2 1!}}x, \sqrt{\frac{1}{\sigma^4 2!}}x^2, \dots\right]^T$$

💡 The feature map $\phi(x)$ is infinite-dimensional. Computing it explicitly is impossible, but the kernel $k(x, x')$ can be computed easily. This is the kernel trick.

Kernel Ridge Regression

Given any SPD kernel k , we have a corresponding hypothesis space:

$$\mathcal{H}(k) = \left\{ f : f(x) = \sum_{j=0}^{\infty} w_j \phi_j(x), \text{ where } k(x, x') = \sum_{j=0}^{\infty} \phi_j(x) \phi_j(x') \right\}$$

In this space, the regularized empirical risk minimization has the solution we derived earlier, now expressed purely in terms of the kernel.

Definition 2: Kernel Ridge Regression

The predictor is given by:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^N [(G + \lambda N \mathbf{I}_N)^{-1} \mathbf{y}]_i k(\mathbf{x}, \mathbf{x}_i)$$

where G is the Gram matrix with $G_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. This can be computed without any explicit knowledge of the feature maps $\{\phi_j\}$.

Motivation: The Primal Problem

Dealing with infinite-dimensional models initially seems impossible because algorithms cannot be run in infinite dimensions. The kernel function plays a crucial role in making such problems computationally tractable.

Learning with Linear Models in a Hilbert Space \mathcal{H}

Given data $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, for $i = 1, \dots, N$, we consider the optimization problem:

$$\min_{\beta \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^N \ell(y_i, \langle \phi(x_i), \beta \rangle) + \frac{\lambda}{2} \|\beta\|^2,$$

where $\phi : \mathcal{X} \rightarrow \mathcal{H}$ is a (potentially infinite-dimensional) feature map.

The loss function ℓ can be the hinge loss, logistic loss, least-squares, etc.

The Representer Theorem

The key property of the objective function is that it only accesses inputs through dot products $\langle \beta, \phi(x_i) \rangle$ and penalizes the norm $\|\beta\|$.

Theorem 2: Representer Theorem

Let $\phi : \mathcal{X} \rightarrow \mathcal{H}$ be a feature map and $\Psi : \mathbb{R}^{N+1} \rightarrow \mathbb{R}$ be a functional that is strictly increasing in its last argument. Then any minimizer β^* of

$$\Psi(\langle \beta, \phi(x_1) \rangle, \dots, \langle \beta, \phi(x_N) \rangle, \|\beta\|^2)$$

must lie in the span of the feature vectors of the training data. That is, β^* must be of the form:

$$\beta^* = \sum_{i=1}^N \alpha_i \phi(x_i), \quad \text{for some } \alpha \in \mathbb{R}^N$$

Proof.

See blackboard.



Representer Theorem for Supervised Learning

Theorem 3: Representer Theorem for Supervised Learning

For any $\lambda > 0$, any minimizer β^* of the regularized risk

$$\frac{1}{N} \sum_{i=1}^N \ell(y_i, \langle \beta, \phi(x_i) \rangle) + \frac{\lambda}{2} \|\beta\|^2$$

can be expressed as a linear combination of the feature vectors:

$$\beta^* = \sum_{i=1}^N \alpha_i \phi(x_i), \quad \text{for some } \alpha \in \mathbb{R}^N.$$

Proof.

We deduce this straightforwardly from previous theorem. □

💡 There is no assumption on the loss function ℓ . In particular, no convexity is assumed. This result is very general.

Reformulation using Kernels

Given the theorem, we can reformulate the learning problem by substituting $\beta = \sum_{i=1}^N \alpha_i \phi(x_i)$.

- Define the **kernel function** as the inner product of feature vectors:

$$k(x, x') = \langle \phi(x), \phi(x') \rangle.$$

- The model's predictions on the training data become:

$$\langle \beta, \phi(x_j) \rangle = \sum_{i=1}^N \alpha_i \langle \phi(x_i), \phi(x_j) \rangle = \sum_{i=1}^N \alpha_i k(x_i, x_j) = (\mathbf{K}\alpha)_j$$

where $\mathbf{K} \in \mathbb{R}^{N \times N}$ is the kernel (Gram) matrix.

- The regularization term becomes a quadratic form in α :

$$\|\beta\|^2 = \left\langle \sum_i \alpha_i \phi(x_i), \sum_j \alpha_j \phi(x_j) \right\rangle = \sum_{i,j} \alpha_i \alpha_j K_{ij} = \alpha^\top \mathbf{K} \alpha$$

The Dual Problem and Prediction

The original infinite-dimensional problem in $\beta \in \mathcal{H}$ is now a finite-dimensional problem in $\alpha \in \mathbb{R}^N$:

$$\min_{\alpha \in \mathbb{R}^N} \frac{1}{N} \sum_{i=1}^N \ell(y_i, (\mathbf{K}\alpha)_i) + \frac{\lambda}{2} \alpha^\top \mathbf{K} \alpha.$$

For any test point $x \in \mathcal{X}$, the prediction function also only requires kernel evaluations:

$$f(x) = \langle \beta, \phi(x) \rangle = \sum_{i=1}^N \alpha_i \langle \phi(x_i), \phi(x) \rangle = \sum_{i=1}^N \alpha_i k(x, x_i).$$

The Kernel Trick: Summary

The input observations are summarized in the $N \times N$ kernel matrix \mathbf{K} . We never need to explicitly compute the feature vector $\phi(x)$.

This is the **kernel trick**, which allows us to:

- ▶ **Solve in \mathbb{R}^N :** Replace the potentially infinite-dimensional search space \mathcal{H} with the finite-dimensional space \mathbb{R}^N .
- ▶ **Separate Concerns:** Separate the representation problem (designing powerful kernels $k(\cdot, \cdot)$ for many data types) from the algorithmic problem (which only uses the Gram matrix \mathbf{K}).

Computational Implications

The infinite-dimensional optimization problem reduces to the finite-dimensional problem:

$$\min_{\alpha \in \mathbb{R}^n} \left[\sum_{i=1}^N L \left(y_i, \sum_{j=1}^N \alpha_j K(x_i, x_j) \right) + \lambda \sum_{i,j=1}^N \alpha_i \alpha_j K(x_i, x_j) \right],$$

which can be written as: $\min_{\alpha \in \mathbb{R}^N} \left[\sum_{i=1}^N L(y_i, (\mathbf{K}\alpha)_i) + \lambda \alpha^T \mathbf{K} \alpha \right]$.

- ▶ **Finite computation:** Only need to work with $N \times N$ Gram matrix \mathbf{K}
- ▶ **No explicit features:** Never compute $\phi(x)$, only kernel evaluations
- ▶ **General applicability:** Works for any loss function L
- ▶ **Scales with data:** Complexity is $O(N)$, not dimension of feature space

The Computational Bottleneck of Kernel Methods

The Representer Theorem is powerful, but it leads to a significant computational challenge for large datasets.

The Problem: The Gram Matrix

The solution for kernel methods like Kernel Ridge Regression or SVMs involves solving a linear system with the $N \times N$ Gram matrix \mathbf{K} .

$$\alpha = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y} \quad (\text{for Kernel Ridge Regression}).$$

- ▶ Storage Cost: Storing the Gram matrix requires $O(N^2)$ memory.
- ▶ Computational Cost: Solving the linear system (e.g., via matrix inversion or Cholesky decomposition) costs $O(N^3)$ time. This is infeasible for large N .

Kernel methods in their standard form **do not scale well** with the number of samples N . We need approximation techniques.

Approximation via Random Fourier Features

Powerful way to approximate a kernel machine with a simple linear model, trading a controllable amount of accuracy for massive gain in compute efficiency.

Proposition 3: Bochner's Theorem

A continuous kernel $K(x, x')$ is positive semi-definite if and only if it is the Fourier transform of a non-negative measure. For a shift-invariant kernel $K(x, x') = K(x - x')$, this means:

$$K(x - x') = \int_{\mathbb{R}^d} p(\omega) e^{i\omega^T(x-x')} d\omega$$

where $p(\omega)$ is a non-negative probability density.

Proof.

See Bach Ch. 7.3.3. □

💡 We can approximate this integral with a Monte Carlo average. By sampling D frequencies ω_j from the density $p(\omega)$, we can construct an explicit, low-dimensional feature map that approximates the infinite-dimensional map.

The Random Fourier Features Algorithm

Random Fourier Features for Gaussian Kernels

- 1: **Goal:** Approximate the Gaussian kernel $K(x, x') = \exp(-\frac{\|x-x'\|^2}{2\sigma^2})$. Its Fourier transform is a Gaussian density.
- 2: **Sampling:** Draw D random vectors $\omega_1, \dots, \omega_D$ from $\mathcal{N}(0, \frac{1}{\sigma^2}\mathbf{I})$.
- 3: **Feature Map:** Create the new feature map $z : \mathbb{R}^d \rightarrow \mathbb{R}^{2D}$: $z(x) =$

$$\frac{1}{\sqrt{D}} \begin{bmatrix} \cos(\omega_1^T x) \\ \sin(\omega_1^T x) \\ \vdots \\ \cos(\omega_D^T x) \\ \sin(\omega_D^T x) \end{bmatrix} \in \mathbb{R}^{2D}.$$

- 4: **Linear Model:** Train a standard linear model (e.g., Ridge Regression or a linear SVC) on the new data $(z(x_i), y_i)$.

The inner product of these new features, $z(x)^T z(x')$, is an unbiased estimator of the true kernel value $K(x, x')$.

Theoretical Guarantees for Random Features

The performance of the linear model trained on random Fourier features converges to the performance of the full kernel machine.

Theorem 4: Approximation Guarantee (Informal)

Let \hat{f}_D be the solution of Ridge Regression on D random Fourier features, and let $\hat{f}_{\mathcal{H}}$ be the solution of the full Kernel Ridge Regression. Then, for the squared loss case,

$$\mathbb{E}[R(\hat{f}_D)] - \mathbb{E}[R(\hat{f}_{\mathcal{H}})] = O\left(\frac{1}{\sqrt{D}}\right).$$

Proof.

See blackboard. □

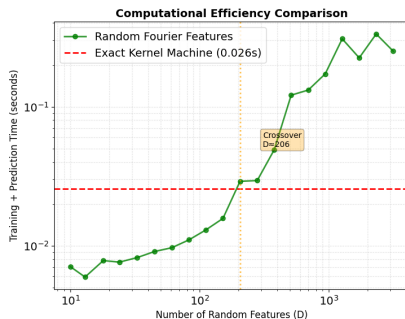
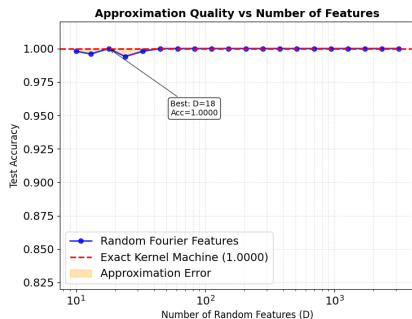
💡 A powerful result: by increasing D , we can get arbitrarily close to the performance of the exact kernel method. The computational cost is now only $O(ND^2 + D^3)$, which is much better than $O(N^3)$ if $D \ll N$.

Example: Approximating a Kernel Machine

We can run a simulation to see how well the random features approximation works in practice.

- ▶ **Data:** A non-linearly separable "two moons" dataset with $N = 500$ points.
- ▶ **Models:**
 1. An exact Kernel Ridge Regression classifier (our gold standard).
 2. Several linear Ridge Regression classifiers trained on Random Fourier Features, with an increasing number of features D .
- ▶ **Analysis:** We will plot the test accuracy of the approximate models as a function of the number of random features D and compare it to the accuracy of the exact kernel machine.

Example: Random Features in Action



Random features can achieve nearly the same predictive power as a full kernel machine at a fraction of the computational cost, making kernel methods practical for large datasets!

💡 The seminal work by Rahimi and Recht on random features won the 2017 NIPS Test of Time Award:

<https://eecs.berkeley.edu/news/ben-recht-wins-nips-test-time-award/>

Exercise 6

- (a) Show that if $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a positive-definite kernel, then so is the function $(x, x') \mapsto e^{k(x, x')}$.

(b) Write down the explicit expressions for the Matérn kernel when $\nu = 3/2$ and $\nu = 5/2$. Verify that the resulting kernels for these two cases are positive definite.
- Minimum Norm Interpolation.** Let \mathcal{H} be a Hilbert space with feature map $\phi : \mathcal{X} \rightarrow \mathcal{H}$. Given data $(x_1, y_1), \dots, (x_N, y_N)$ where $x_i \in \mathcal{X}$ and $\mathbf{y} \in \mathbb{R}^N$. Assume that the set of interpolating solutions is non-empty, i.e., there exists at least one $\beta \in \mathcal{H}$ such that $y_i = \langle \beta, \phi(x_i) \rangle_{\mathcal{H}}$ for all $i \in \{1, \dots, N\}$. Show that, among all such interpolating solutions, the unique solution β^* with the minimum norm $\|\beta^*\|_{\mathcal{H}}$ can be expressed as: $\beta^* = \sum_{i=1}^N \alpha_i \phi(x_i)$, where the coefficient vector $\alpha \in \mathbb{R}^N$ is a solution to the linear system $\mathbf{y} = K\alpha$. Here, K is the $N \times N$ Gram matrix with entries $K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$.

Exercise 6

3. The primal problem for Kernel Ridge Regression (KRR) is to find the vector $\beta \in \mathbb{R}^M$ in a feature space of dimension M that minimizes:

$$L_P(\beta) = \frac{1}{2} \|\mathbf{y} - \Phi\beta\|_2^2 + \frac{\lambda}{2} \|\beta\|_2^2,$$

where $\Phi \in \mathbb{R}^{N \times M}$ is the design matrix of feature vectors. This leads to solving a linear system involving the $M \times M$ primal matrix $H_P = \Phi^\top \Phi + \lambda I_M$.

(a) Derive the dual problem for KRR. Show that it involves solving a linear system for a dual variable $\alpha \in \mathbb{R}^N$ with the $N \times N$ dual matrix $H_D = K + \lambda I_N$, where $K = \Phi\Phi^\top$ is the kernel Gram matrix.

(b) The condition number of a matrix measures the numerical stability of a linear system, with lower values being better. Compare the condition number of the primal matrix H_P with that of the dual matrix H_D . Analyze the two cases: (i) $N > M$, (ii) $M > N$ (the typical scenario in KRR).

(c) Compare the two formulations to the use of normal equations as in Lecture 2, and relate the two using the matrix inversion lemma.

Exercise 6

4. Show that the polynomial kernel $k(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\top \mathbf{x}')^p$ for $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ corresponds to a feature space spanned by all monomials of the form $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d}$ such that the total degree $\sum_{i=1}^d \alpha_i \leq p$. Also, show that the dimension of this feature space is given by the binomial coefficient: $\binom{d+p}{p}$.
5. **Random Features.** Consider the Random Fourier Features map for the Gaussian kernel $z(\mathbf{x}) \in \mathbb{R}^{2D}$, i.e.,

$$z(\mathbf{x}) = \frac{1}{\sqrt{D}} \begin{bmatrix} \cos(\omega_1^\top \mathbf{x}) \\ \sin(\omega_1^\top \mathbf{x}) \\ \vdots \\ \cos(\omega_D^\top \mathbf{x}) \\ \sin(\omega_D^\top \mathbf{x}) \end{bmatrix}.$$

Show that the expected inner product $\mathbb{E}_\omega[z(\mathbf{x})^\top z(\mathbf{x}')] is an unbiased estimator of the true kernel value $K(\mathbf{x}, \mathbf{x}') = \exp(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2})$, where the expectation is over the random draws of $\omega \sim \mathcal{N}(0, \frac{1}{\sigma^2} \mathbf{I})$.$

Exercise 6 (Experimental)

How does a kernel's performance depend on whether its implicit smoothness assumption matches the regularity of the target function?

- ▶ **Setup:** You will perform Kernel Ridge Regression (KRR) for two functions defined on $[0, 1]$.

- ▶ **Smooth Function (Sine Wave):** $f_{\text{smooth}}(x) = \sin(4\pi x)$

- ▶ **Non-smooth Function (Piecewise Constant, Square Wave):**

$$f_{\text{nonsmooth}}(x) = \begin{cases} 1 & \text{if } 0 \leq x < 0.25, 0.5 \leq x < 0.75 \\ -1 & \text{otherwise} \end{cases}$$

- ▶ **Kernels to Compare:** You will compare three kernels with different smoothness properties.

- ▶ **Low Smoothness (Exponential):** $k(x, x') = \exp\left(-\frac{|x-x'|}{\ell}\right)$

- ▶ **Medium Smoothness (Matérn 5/2):**

$$k(x, x') = \left(1 + \frac{\sqrt{5}|x-x'|}{\ell} + \frac{5(x-x')^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}|x-x'|}{\ell}\right)$$

- ▶ **High Smoothness (Gaussian/RBF):** $k(x, x') = \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)$

Exercise 6 (Experimental)

- ▶ **Procedure:** For *each* of the two target functions:
 - ▶ Generate a training set of $N = 40$ points by sampling $x_i \sim \text{Uniform}[0, 1]$ and setting $y_i = f(x_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, 0.1^2)$.
 - ▶ For *each* of the three kernels, train a KRR model.
 - ▶ **Crucially**, tune the hyperparameters (length-scale ℓ and regularization λ) for each model using 5-fold cross-validation on the training set to ensure a fair comparison.
- ▶ **Evaluation and Analysis:**
 - ▶ For each of the six final models (2 functions \times 3 tuned kernels), compute the mean squared error (MSE) on a set of 1000 uniformly spaced points spanning $[0, 1]$. Create a summary table for the six test MSE values.
 - ▶ Create two plots (one for the sine wave, one for the square wave) showing the true function, the noisy data, and the three fitted KRR curves.
 - ▶ Discuss the results. Which kernel performed best on the smooth sine function and which on the non-smooth square wave?

Appendix: From Kernels to Function Spaces

Theorem 5: Mercer's Theorem (Finite Version)

A function $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a positive semi-definite kernel if and only if there exists a Hilbert space \mathcal{H} and a mapping $\phi : \mathcal{X} \rightarrow \mathcal{H}$ such that:

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

💡 The PSD condition is both necessary and sufficient. It guarantees that the kernel trick is mathematically valid.

Proof.

See blackboard.



Reproducing Kernel Hilbert Space (RKHS)

We can understand kernels as defining infinite-dimensional function spaces.

Definition 3: Reproducing Kernel Hilbert Space

Given a PSD kernel K on domain \mathcal{X} , the **Reproducing Kernel Hilbert Space** \mathcal{H}_K is a Hilbert space of functions $f : \mathcal{X} \rightarrow \mathbb{R}$ with two key properties:

1. **Kernel functions are in the space:** For each $x \in \mathcal{X}$, the function $K_x(\cdot) := K(\cdot, x)$ belongs to \mathcal{H}_K .
2. **Reproducing property:** For any $f \in \mathcal{H}_K$ and $x \in \mathcal{X}$:

$$f(x) = \langle f, K_x \rangle_{\mathcal{H}_K} = \langle f, K(\cdot, x) \rangle_{\mathcal{H}_K}.$$

💡 The RKHS is the "natural" function space associated with kernel K . Functions in \mathcal{H}_K can be evaluated at any point via inner product with the kernel.

The Moore-Aronszajn Theorem

Theorem 6: Existence and Uniqueness of RKHS

For every positive semi-definite kernel K on \mathcal{X} , there exists a unique Reproducing Kernel Hilbert Space \mathcal{H}_K having K as its reproducing kernel.

Proof.

See blackboard. □

Proposition 4: Key Properties

- ▶ $K(x, y) = \langle K(\cdot, x), K(\cdot, y) \rangle_{\mathcal{H}_K}$.
- ▶ For $f = \sum_i \alpha_i K(\cdot, x_i)$: $\|f\|_{\mathcal{H}_K}^2 = \sum_i \sum_j \alpha_i \alpha_j K(x_i, x_j)$.

Proof.

See blackboard. □

RKHS Examples

Linear Kernel: $K(x, x') = x^T x'$

$\mathcal{H}_K = \{f(x) = w^T x : w \in \mathbb{R}^p\}$ with $\|f\|_{\mathcal{H}_K}^2 = \|w\|^2$
This recovers standard linear regression/classification.

Polynomial Kernel: $K(x, x') = (1 + x^T x')^d$

\mathcal{H}_K consists of polynomials up to degree d with appropriate norm.

Gaussian Kernel: $K(x, x') = \exp(-\|x - x'\|^2 / (2\sigma^2))$

\mathcal{H}_K is infinite-dimensional and very rich:

- ▶ Contains smooth functions
- ▶ Universal approximation: dense in $C(\mathcal{X})$ for compact \mathcal{X}
- ▶ Functions decay appropriately at infinity

💡 Different kernels encode different notions of function complexity via their RKHS norms.

The Key Result for Kernel Methods

Theorem 7: The Representer Theorem


Consider the regularized empirical risk minimization problem:

$$\min_{f \in \mathcal{H}_K} \left[\frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda \|f\|_{\mathcal{H}_K}^2 \right]$$

where L is any loss function, $\lambda > 0$, and \mathcal{H}_K is the RKHS associated with PSD kernel K . Then the minimizer \hat{f} has the finite representation: $\hat{f}(x) = \sum_{i=1}^N \alpha_i K(x, x_i)$ for some coefficients $\alpha_1, \dots, \alpha_N \in \mathbb{R}$.

Proof.

See blackboard. □

 **Remarkable:** Even though we're optimizing over an infinite-dimensional function space, the solution lives in an n -dimensional subspace spanned by kernel evaluations at the training points!

Recap: SVM as a Penalization Method

Definition 4: Hinge Loss + Penalty Formulation

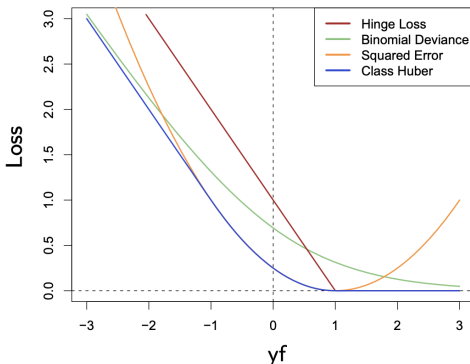
The SVM solution is equivalent to minimizing:

$$\min_{f \in \mathcal{H}_K} \left(\sum_{i=1}^N [1 - y_i f(x_i)]_+ + \lambda \|f\|_{\mathcal{H}_K}^2 \right), \quad (1)$$

- ▶ $[1 - u]_+ = \max(0, 1 - u)$ is the **hinge loss function**.
- ▶ \mathcal{H}_K is a Reproducing Kernel Hilbert Space defined by the kernel K .
- ▶ $\|f\|_{\mathcal{H}_K}^2$ is a penalty on the complexity (smoothness) of the function f .

Connection to Logistic Regression

This formulation is very similar to penalized logistic regression, which minimizes: $\sum_{i=1}^N \log(1 + e^{-y_i f(x_i)}) + \lambda \|f\|_{\mathcal{H}_K}^2$. The hinge loss and binomial deviance (logistic loss) are both convex surrogates for the 0-1 loss and produce similar classifiers.



Summary: The Power of the Kernel Viewpoint

- ▶ **Generality:** The kernel trick provides a powerful recipe for converting linear techniques that depend on inner products into non-linear methods capable of learning complex decision boundaries or regression functions.
- ▶ **Computational Efficiency:** By working with the Gram matrix \mathbf{K} , the complexity of the algorithm depends on the number of samples, not the (potentially infinite) dimension of the feature space.
- ▶ **Theoretical Foundation:** The theory of RKHS provides a rigorous mathematical framework for understanding why these methods work and for designing new kernel functions.
- ▶ **Unifying Framework:** The "Loss + Penalty in an RKHS" formulation, justified by the Representer Theorem, unifies many disparate-seeming methods (SVMs, Ridge Regression, Smoothing Splines) under a single elegant theory.