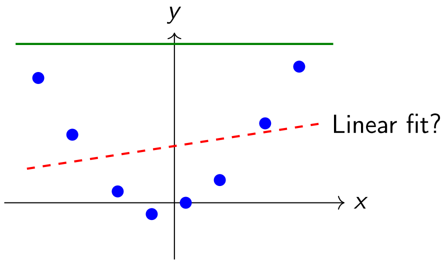
Kernel Methods for Least Squares

Beyond Linearity

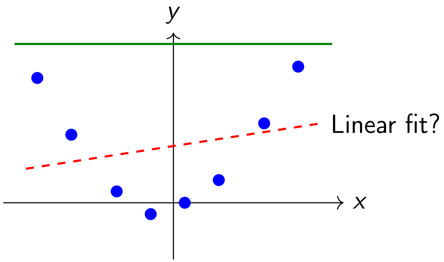
The Limits of Linear Models

Standard Least Squares finds the best linear fit to the data. But what if the underlying relationship is not linear?



The Limits of Linear Models

Standard Least Squares finds the best linear fit to the data. But what if the underlying relationship is not linear?



Problem

A straight line is a poor approximation for this data. We need a way to model non-linear relationships.

The Solution: Feature Maps

The core idea is to project the data into a higher-dimensional **feature space** where the relationship becomes linear.

The Solution: Feature Maps

The core idea is to project the data into a higher-dimensional **feature space** where the relationship becomes linear.

Definition (Feature Map ϕ)

A feature map is a function ϕ that maps each data point $x \in \mathbb{R}^p$ to a higher-dimensional space $\mathbb{R}^D$ (where $D > p$):

$$\phi : \mathbb{R}^p \to \mathbb{R}^D$$

The Solution: Feature Maps

The core idea is to project the data into a higher-dimensional **feature space** where the relationship becomes linear.

Definition (Feature Map ϕ)

A feature map is a function ϕ that maps each data point $x \in \mathbb{R}^p$ to a higher-dimensional space $\mathbb{R}^D$ (where $D > p$):

$$\phi : \mathbb{R}^p \to \mathbb{R}^D$$

Example (1D Input to 2D Feature Space (Polynomial Kernel))

For 1D data x, we can create a 2D feature space with the map:

$$\phi(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix}$$

In this new space, a linear model $w_1(x) + w_2(x^2)$ corresponds to a quadratic model in the original space.

The Solution: Feature Maps I

Example (2D Input to 3D Feature Space (Polynomial Kernel))

- Consider a 2D input $\mathrm{x} = (x_1, x_2)^T \in \mathbb{R}^2$.
- We might encounter data that is not linearly separable in 2D, or has a complex non-linear relationship with y.
- For instance, a decision boundary might be a circle.
- A common feature map for polynomial kernels of degree 2 transforms $\mathrm{x} = (x_1, x_2)^T$ into:

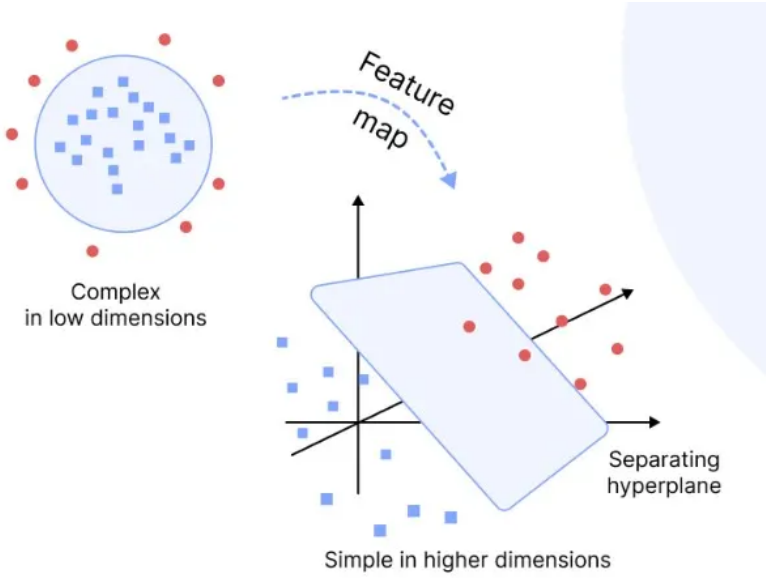
$$\phi(x_1, x_2) = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \in \mathbb{R}^6$$

The Solution: Feature Maps II

Example (2D Input to 3D Feature Space (Polynomial Kernel) cont.)

- For simpler visualization, let's consider a subset of these features, e.g.,
$$\phi(x_1, x_2) = \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \end{pmatrix} \in \mathbb{R}^3.$$
- Now, a circular decision boundary like $x_1^2 + x_2^2 = R^2$ becomes linear in the feature space!
- $1 \cdot (x_1^2) + 1 \cdot (x_2^2) = R^2$ is a linear equation if the features are x_1^2 and x_2^2.

Geometric view



Ridge Regression in Feature Space

We can now perform Ridge Regression in this new, high-dimensional space.

Ridge Regression in Feature Space

We can now perform Ridge Regression in this new, high-dimensional space.

1. Let $\Phi(\mathrm{X})$ be the $n \times D$ matrix where each row is $\phi(\mathrm{x}_i)^T$.

Ridge Regression in Feature Space

We can now perform Ridge Regression in this new, high-dimensional space.

1. Let $\Phi(X)$ be the $n \times D$ matrix where each row is $\phi(x_i)^T$.
2. The Ridge Regression problem becomes:

$$\underset{w \in \mathbb{R}^D}{\text{minimize}} \quad \|\Phi(X)w - y\|_2^2 + \lambda \|w\|_2^2$$

Ridge Regression in Feature Space

We can now perform Ridge Regression in this new, high-dimensional space.

1. Let $\Phi(X)$ be the $n \times D$ matrix where each row is $\phi(x_i)^T$.
2. The Ridge Regression problem becomes:

$$\underset{w \in \mathbb{R}^D}{\text{minimize}} \quad \|\Phi(X)w - y\|_2^2 + \lambda \|w\|_2^2$$

The Computational Problem

The dimension D of the feature space can be very large, or even infinite!

- Explicitly computing and storing the matrix $\Phi(X)$ can be prohibitively expensive or impossible.
- The solution involves inverting a $D \times D$ matrix, which is computationally infeasible.

The Kernel Trick: A Clever Shortcut

It turns out that for many algorithms, including Ridge Regression, we don't need the explicit feature vectors $\phi(x)$. We only need their inner products.

The Kernel Trick: A Clever Shortcut

It turns out that for many algorithms, including Ridge Regression, we don't need the explicit feature vectors $\phi(x)$. We only need their inner products.

Definition (Kernel Function)

A kernel function K computes the inner product of two vectors in the feature space, without ever going into that space:

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = \phi(x_i)^T \phi(x_j)$$

The Kernel Trick: A Clever Shortcut

It turns out that for many algorithms, including Ridge Regression, we don't need the explicit feature vectors $\phi(x)$. We only need their inner products.

Definition (Kernel Function)

A kernel function K computes the inner product of two vectors in the feature space, without ever going into that space:

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = \phi(x_i)^T \phi(x_j)$$

The "Trick"

If we can rewrite our algorithm to only use these inner products, we can avoid the computational cost of high-dimensional feature spaces.

Examples of Kernel Functions

Polynomial Kernel (degree q)

Corresponds to a feature space of all polynomial terms up to degree q.

$$K(\mathrm{x}, \mathrm{z}) = (\mathrm{x}^T \mathrm{z} + c)^q$$

For $c = 0$, it's the homogeneous polynomial kernel. For $c > 0$, it includes lower-order terms.

Examples of Kernel Functions

Polynomial Kernel (degree q)

Corresponds to a feature space of all polynomial terms up to degree q.

$$K(\mathrm{x}, \mathrm{z}) = (\mathrm{x}^T \mathrm{z} + c)^q$$

For $c = 0$, it's the homogeneous polynomial kernel. For $c > 0$, it includes lower-order terms.

Gaussian Kernel (Radial Basis Function - RBF)

Corresponds to an **infinite-dimensional** feature space.

$$K(\mathrm{x}, \mathrm{z}) = \exp\left(-\frac{\|\mathrm{x} - \mathrm{z}\|_2^2}{2\sigma^2}\right) = \exp(-\gamma\|\mathrm{x} - \mathrm{z}\|_2^2)$$

This is one of the most powerful and widely used kernels.

The Representer Theorem

How does the kernel trick help us solve for w? The Representer Theorem provides the key.

The Representer Theorem

How does the kernel trick help us solve for w? The Representer Theorem provides the key.

Theorem (Representer Theorem, simplified)

The solution w^* *to the Kernel Ridge Regression problem can always be written as a linear combination of the feature vectors of the training data:*

$$\mathrm{w}^* = \sum_{i=1}^{n} \alpha_i \phi(\mathrm{x}_i) = \Phi(\mathrm{X})^T \alpha$$

for some coefficient vector $\alpha \in \mathbb{R}^n$.

The Representer Theorem

How does the kernel trick help us solve for w? The Representer Theorem provides the key.

Theorem (Representer Theorem, simplified)

The solution w^* *to the Kernel Ridge Regression problem can always be written as a linear combination of the feature vectors of the training data:*

$$\mathrm{w}^* = \sum_{i=1}^{n} \alpha_i \phi(\mathrm{x}_i) = \Phi(\mathrm{X})^T \alpha$$

for some coefficient vector $\alpha \in \mathbb{R}^n$*.*

Implication

Instead of searching for the potentially infinite-dimensional vector w, we only need to find the *n*-dimensional vector of coefficients α.

Proof of the Representer Theorem I

Let the feature space be $\mathcal{H}$. Any vector $\mathrm{w} \in \mathcal{H}$ can be decomposed into two orthogonal components:

$$\mathrm{w} = \mathrm{w}_{||} + \mathrm{w}_{\perp}$$

where $\mathrm{w}_{||}$ is in the span of the training data $\{\phi(\mathrm{x}_i)\}_{i=1}^{n}$, and $\mathrm{w}_{\perp}$ is in its orthogonal complement.

1. By definition, $\langle \mathrm{w}_{\perp}, \phi(\mathrm{x}_i) \rangle = 0$ for all i. This means the prediction on any training point is unaffected by $\mathrm{w}_{\perp}$:

   $$\langle \mathrm{w}, \phi(\mathrm{x}_i) \rangle = \langle \mathrm{w}_{||}, \phi(\mathrm{x}_i) \rangle + \langle \mathrm{w}_{\perp}, \phi(\mathrm{x}_i) \rangle = \langle \mathrm{w}_{||}, \phi(\mathrm{x}_i) \rangle$$

   So the error term $\|\Phi(\mathrm{X})\mathrm{w} - \mathrm{y}\|_2^2$ only depends on $\mathrm{w}_{||}$.
2. By the Pythagorean theorem, the regularization term becomes:

   $$\|\mathrm{w}\|_2^2 = \|\mathrm{w}_{||}\|_2^2 + \|\mathrm{w}_{\perp}\|_2^2$$

Proof of the Representer Theorem II

The full cost function is:

$$J(\mathrm{w}) = \|\Phi(\mathrm{X})\mathrm{w}_{\|} - \mathrm{y}\|_2^2 + \lambda(\|\mathrm{w}_{\|}\|_2^2 + \|\mathrm{w}_{\perp}\|_2^2)$$

To minimize this function, we must choose $\mathrm{w}_{\perp} = 0$. Therefore, the optimal solution w^* must lie entirely in the span of the training feature vectors.

The Representer Theorem: meaning

- Essentially the Representer Theorem states that **"the solution to a wide class of regularization problems that involve minimizing a loss function plus a regularizer can be written as a linear combination of kernel functions evaluated at the training data points."**

Significance of the Theorem

- It theoretically justifies why we can express the solution purely in terms of kernel evaluations, even when working in infinite-dimensional feature spaces.
- It guarantees that the kernel-based approach finds the correct solution without needing explicit feature mapping.

The Dual Problem and the Kernel Matrix

By substituting $\mathrm{w} = \Phi(\mathrm{X})^T\alpha$ into the original problem, we get a new problem in terms of α.
The solution is found by solving the linear system:

$$(\mathrm{K} + \lambda \mathrm{I}_n)\alpha = \mathrm{y}$$

Definition (The Kernel Matrix K)

K is an $n \times n$ symmetric matrix, also called the Gram matrix, where each entry is the kernel evaluation between two data points:

$$K_{ij} = K(\mathrm{x}_i, \mathrm{x}_j) = \phi(\mathrm{x}_i)^T\phi(\mathrm{x}_j)$$

Note that $\mathrm{K} = \Phi(\mathrm{X})\Phi(\mathrm{X})^T$.

The solution is $\alpha = (\mathrm{K} + \lambda \mathrm{I}_n)^{-1}\mathrm{y}$.

The Starting Point: The Primal Problem I

Solving Ridge Regression in Feature Space

The objective in Kernel Ridge Regression is to minimize a cost function in a high-dimensional feature space. This is called the **primal problem**.

Objective Function

Find the weight vector w that minimizes:

$$J(w) = \underbrace{||\Phi(X)w - y||_2^2}_{\text{Squared Error}} + \underbrace{\lambda||w||_2^2}_{\text{Regularization}}$$

The Starting Point: The Primal Problem II

Solving Ridge Regression in Feature Space

Primal Solution

The solution to this problem is given by the normal equations for Ridge Regression:

$$(\Phi(\mathrm{X})^T \Phi(\mathrm{X}) + \lambda I_D) w = \Phi(\mathrm{X})^T y$$

The challenge is that the matrix $\Phi(\mathrm{X})^T \Phi(\mathrm{X})$ has dimensions $D \times D$. If the feature space dimension D is very large or infinite, this is impossible to compute directly.

The Key Insight: The Representer Theorem

The **Representer Theorem** provides the crucial shortcut needed to solve the problem.

The Substitution

This allows us to make the following critical substitution, moving from the D-dimensional vector w to an n-dimensional vector of coefficients α:

$$w = \Phi(X)^T \alpha$$

We can now derive the final formula by substituting $w = \Phi(X)^T \alpha$ into the primal solution.

The Full Derivation

From the Primal Solution to the Dual Form

Derivation Steps

$$(\Phi(X)^T\Phi(X) + \lambda I_D)w = \Phi(X)^T y$$

$$(\Phi(X)^T\Phi(X) + \lambda I_D)(\Phi(X)^T\alpha) = \Phi(X)^T y$$

$$\Phi(X)^T\Phi(X)\Phi(X)^T\alpha + \lambda\Phi(X)^T\alpha = \Phi(X)^T y$$

$$\Phi(X)^T(\Phi(X)\Phi(X)^T\alpha + \lambda\alpha) = \Phi(X)^T y$$

$$\Phi(X)^T(K\alpha + \lambda I_n\alpha) = \Phi(X)^T y$$

$$\implies (K + \lambda I_n)\alpha = y$$

Conclusion

The Power of the Kernel Trick

The derivation transforms the original, high-dimensional problem into a manageable one.
We are left with a simple linear system in terms of α:

$$(K + \lambda I_n)\alpha = y$$

The solution is found by solving for α:

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

Why This Is Better

This is the **dual solution**. Its key advantage is that it only involves the **Kernel Matrix** K, which is an $n \times n$ matrix. We can solve the entire problem without ever representing the D-dimensional vector w or matrix $\Phi(X)$. The problem's complexity now depends on the number of data points (n), not the dimension of the feature space (D).

The eigendecomposition

The kernel matrix K is symmetric and positive semi-definite. We can use its eigendecomposition to solve the system stably and efficiently.

1. **Decomposition:** Compute the eigendecomposition of K:

   $$\mathsf{K} = \mathsf{U}\Lambda\mathsf{U}^T$$

   where U is an orthogonal matrix of eigenvectors and Λ is a diagonal matrix of non-negative eigenvalues.

2. **Solution for α:** The inverse can be computed stably:

   $$(\mathsf{K} + \lambda\mathsf{I})^{-1} = (\mathsf{U}\Lambda\mathsf{U}^T + \lambda\mathsf{U}\mathsf{U}^T)^{-1} = \mathsf{U}(\Lambda + \lambda\mathsf{I})^{-1}\mathsf{U}^T$$

   The solution is:

   $$\alpha = \mathsf{U}(\Lambda + \lambda\mathsf{I})^{-1}\mathsf{U}^T\mathsf{y}$$

   This avoids issues with ill-conditioned or singular kernel matrices.

Making Predictions

Once we have found the coefficient vector α, how do we predict the value for a new data point x_*?

$$y_* = \hat{f}(x_*) = \langle w^*, \Phi(x_*) \rangle = \left\langle \sum_{i=1}^{n} \alpha_i \Phi(x_i), \Phi(x_*) \right\rangle =$$

$$= \sum_{i=1}^{n} \alpha_i \langle \Phi(x_i), \Phi(x_*) \rangle = \sum_{i=1}^{n} \alpha_i K(x_i, x_*)$$

Conclusion

The entire process—from solving for the coefficients to making new predictions—can be done using only kernel function evaluations. We never need to compute $\Phi(x)$ explicitly. This is the power of the kernel trick.

Example 1: Polynomial Kernel I

From Kernel Function to Explicit Feature Map

This example revisits the calculation for the polynomial kernel $K(x, z) = (xz + 1)^2$. We'll explicitly derive its feature map $\phi(x)$ and show it produces the same result.

Step 1: The Kernel and its Expansion

Given the data points $x_1 = -1$, $x_2 = 0$, $x_3 = 1$ and the kernel function:

$$K(x, z) = (xz + 1)^2$$

Expanding the function gives us:

$$K(x, z) = x^2 z^2 + 2xz + 1$$

Example 1: Polynomial Kernel II

From Kernel Function to Explicit Feature Map

Step 2: Deriving the Implicit Feature Map

We need to find a feature map $\phi(x)$ such that $K(x, z) = \phi(x)^T \phi(z)$. By matching the terms from the expansion, we can write it as a dot product:

$$\phi(x)^T \phi(z) = (x^2)(z^2) + (\sqrt{2}x)(\sqrt{2}z) + (1)(1)$$

This decomposition reveals the feature map:

$$\phi(x) = \begin{pmatrix} x^2 \\ \sqrt{2}x \\ 1 \end{pmatrix} \in \mathbb{R}^3$$

Example 1: Polynomial Kernel III

From Kernel Function to Explicit Feature Map

Step 3: Verification with Mapped Data

First, we map our data points into the 3D feature space:

$$\phi(-1) = \begin{pmatrix} 1 \\ -\sqrt{2} \\ 1 \end{pmatrix}, \quad \phi(0) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \phi(1) = \begin{pmatrix} 1 \\ \sqrt{2} \\ 1 \end{pmatrix}$$

Next, we compute the Gram Matrix K using $K_{ij} = \phi(x_i)^T \phi(x_j)$. For instance:

- $K_{11} = \phi(-1)^T \phi(-1) = (1)(1) + (-\sqrt{2})(-\sqrt{2}) + (1)(1) = 4$
- $K_{13} = \phi(-1)^T \phi(1) = (1)(1) + (-\sqrt{2})(\sqrt{2}) + (1)(1) = 0$

The resulting matrix is the same as the one computed with the kernel trick:

$$K = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 4 \end{pmatrix}$$

Example 2: Homogeneous Quadratic Kernel I

From Feature Map to Kernel Function

Let's start with a feature map for 2D data, derive its kernel, and verify the result.

Step 1: The Feature Map

Let the input data be $x = (x_1, x_2)^T \in \mathbb{R}^2$. We define a feature map to a 3D space:

$$\phi(x) = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \in \mathbb{R}^3$$

Example 2: Homogeneous Quadratic Kernel II

From Feature Map to Kernel Function

Step 2: Deriving the Kernel

The kernel is the dot product $\phi(x)^T\phi(z)$ for $x, z \in \mathbb{R}^2$:

$$K(x,z) = (x_1^2)(z_1^2) + (x_2^2)(z_2^2) + (\sqrt{2}x_1x_2)(\sqrt{2}z_1z_2)$$

$$= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1z_1x_2z_2$$

This expression is a perfect square, which simplifies to the dot product in the original space:

$$K(x,z) = (x_1z_1 + x_2z_2)^2 = (x^Tz)^2$$

Example 2: Homogeneous Quadratic Kernel III

From Feature Map to Kernel Function

Step 3: Calculation and Verification

Let's use sample data: $x_1 = (1,1)^T$ and $x_2 = (1,0)^T$.

- **Method A (Kernel Trick):** $K(x_1, x_1) = ((1,1)^T(1,1))^2 = 2^2 = 4$.
- **Method B (Explicit Mapping):** $\phi(x_1) = (1, 1, \sqrt{2})^T$, so $\phi(x_1)^T\phi(x_1) = 1 + 1 + 2 = 4$.

Both methods yield the same Gram matrix elements, confirming the equivalence.