Kernel Methods in Statistics
Past, Present and Future

Soham Sarkar

LIKE 2022

10 January 2022

The Regression Problem

Observation pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.

Find the relationship between $\mathbf{x}$ and y.

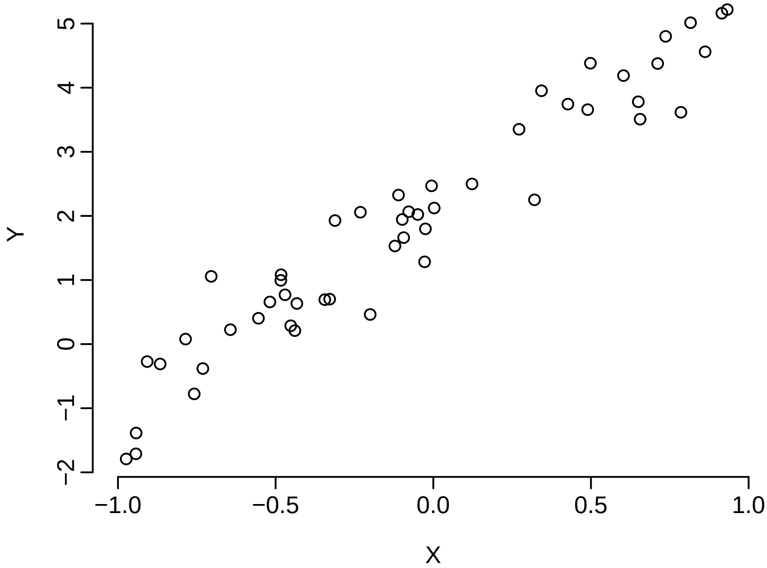
— a method to predict y using $\mathbf{x}$.

The Regression Problem

Observation pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.

Find the relationship between $\mathbf{x}$ and y.

— a method to predict y using $\mathbf{x}$.

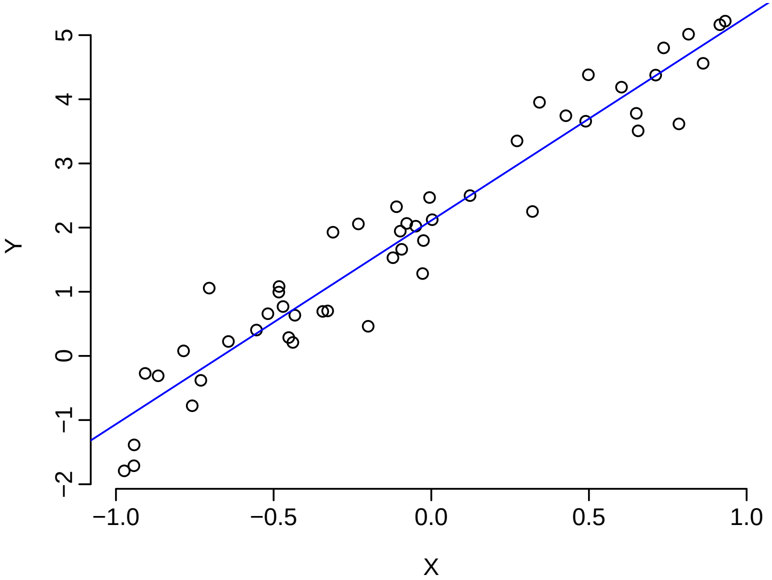


The Regression Problem

Observation pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.

Find the relationship between $\mathbf{x}$ and y.

— a method to predict y using $\mathbf{x}$.

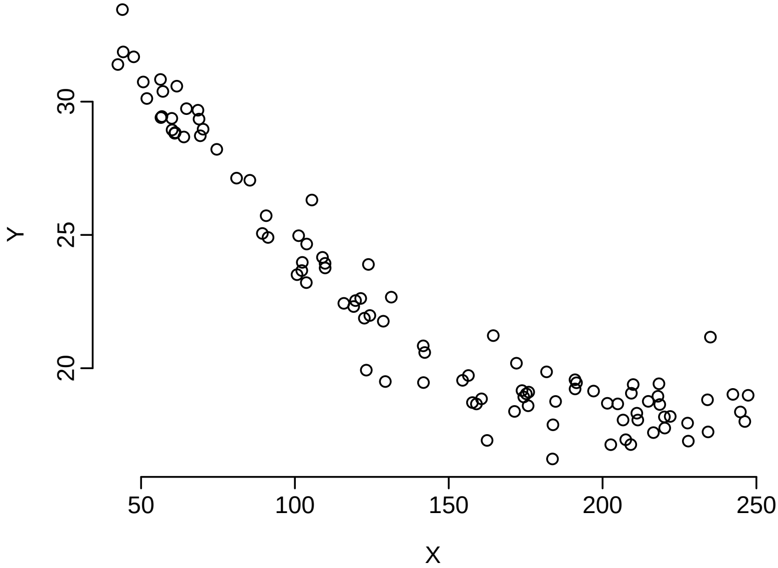


The Regression Problem

Observation pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.

Find the relationship between $\mathbf{x}$ and y.

— a method to predict y using $\mathbf{x}$.

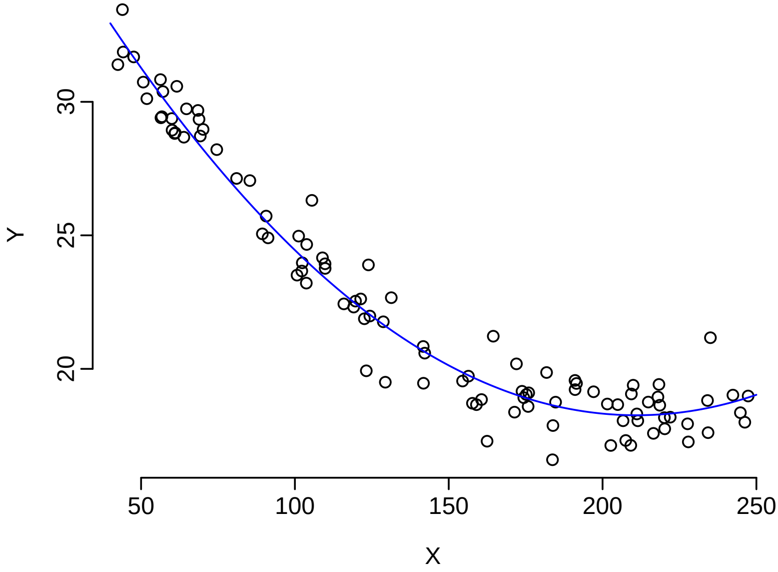


The Regression Problem

Observation pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.

Find the relationship between $\mathbf{x}$ and y.

— a method to predict y using $\mathbf{x}$.

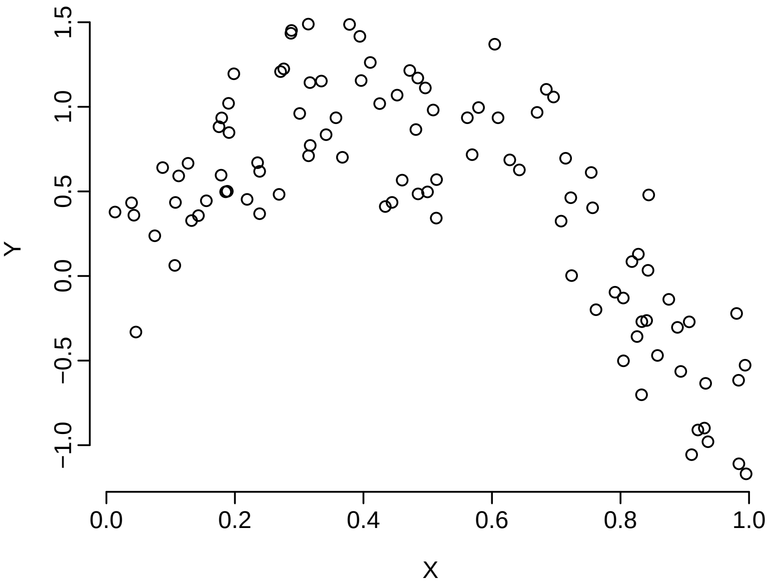


The Regression Problem

Observation pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.

Find the relationship between $\mathbf{x}$ and y.

— a method to predict y using $\mathbf{x}$.

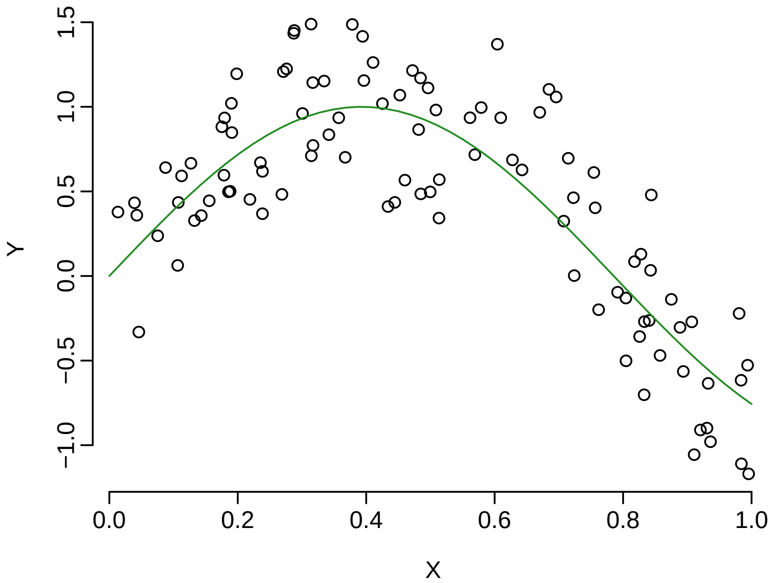


The Regression Problem

Observation pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.

Find the relationship between $\mathbf{x}$ and y.

— a method to predict y using $\mathbf{x}$.

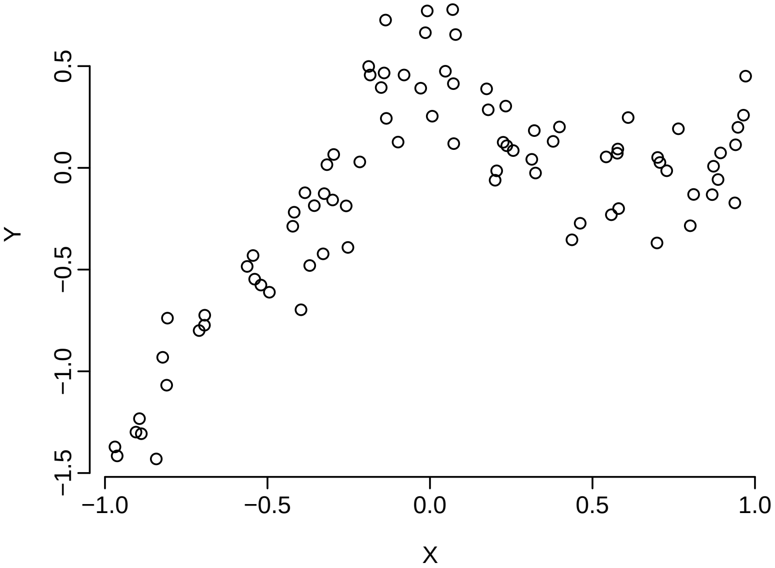


The Regression Problem

Observation pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.

Find the relationship between $\mathbf{x}$ and y.

— a method to predict y using $\mathbf{x}$.

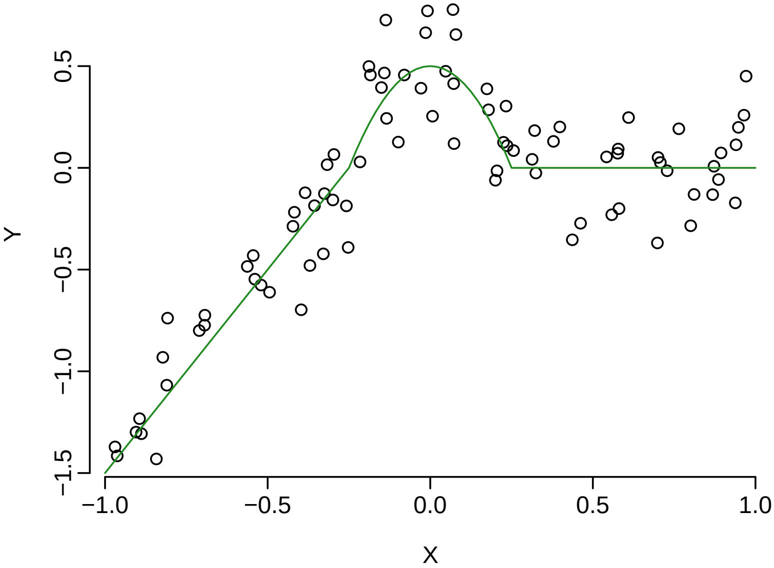


The Regression Problem

Observation pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.

Find the relationship between $\mathbf{x}$ and y.

— a method to predict y using $\mathbf{x}$.



Nonparametric Regression

Obtain a regression estimate with very few assumptions.

Nonparametric Regression

Obtain a regression estimate with very few assumptions.

⋆ The assumption involves infinite number of parameters.

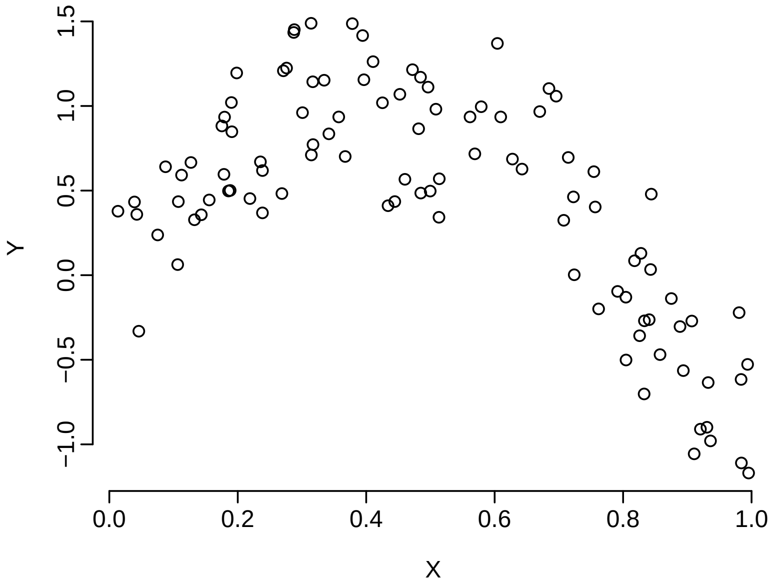
- e.g., continuity, smoothness etc.

Nonparametric Regression

Obtain a regression estimate with very few assumptions.

⋆ The assumption involves infinite number of parameters.

– e.g., continuity, smoothness etc.

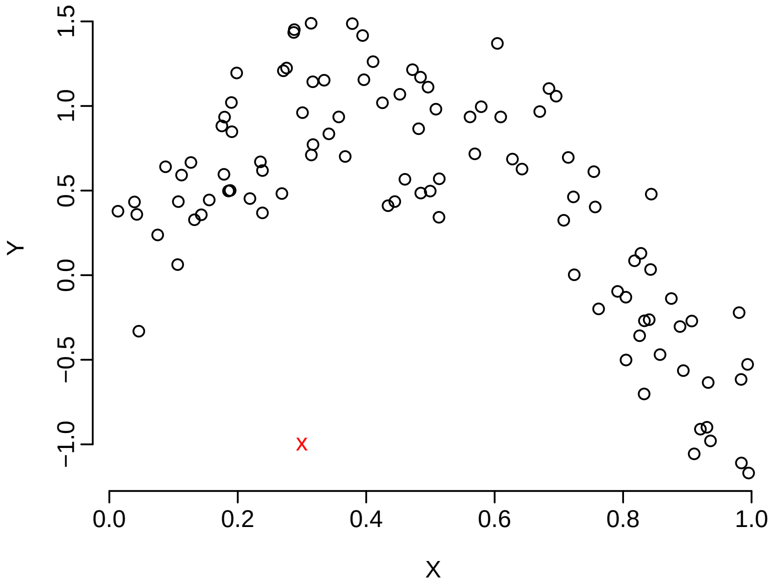


Nonparametric Regression

Obtain a regression estimate with very few assumptions.

⋆ The assumption involves infinite number of parameters.

– e.g., continuity, smoothness etc.

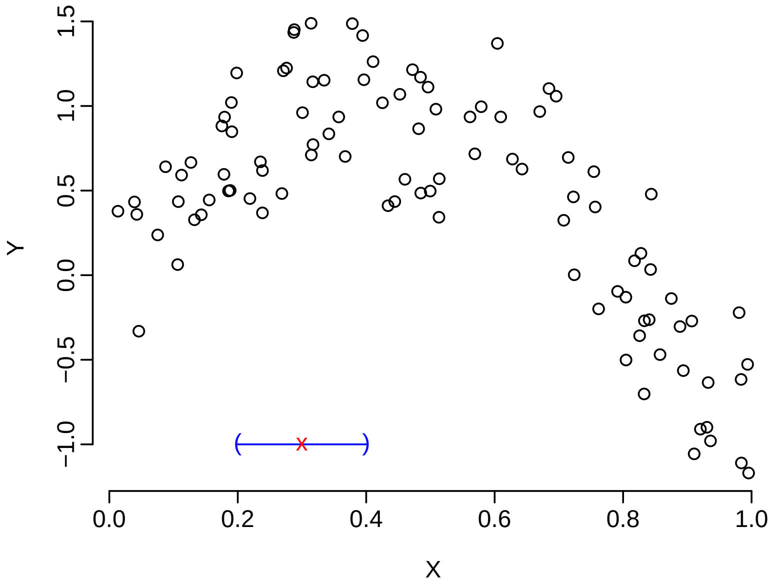


Nonparametric Regression

Obtain a regression estimate with very few assumptions.

⋆ The assumption involves infinite number of parameters.

– e.g., continuity, smoothness etc.

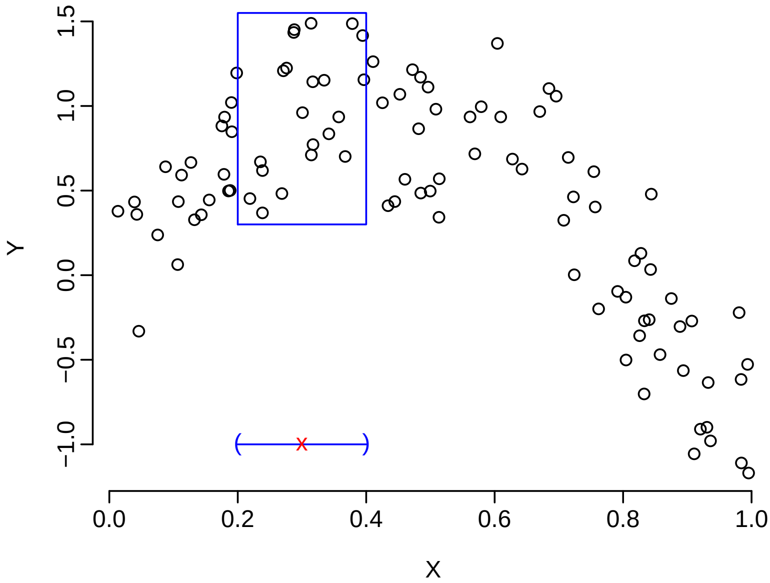


Nonparametric Regression

Obtain a regression estimate with very few assumptions.

⋆ The assumption involves infinite number of parameters.

– e.g., continuity, smoothness etc.

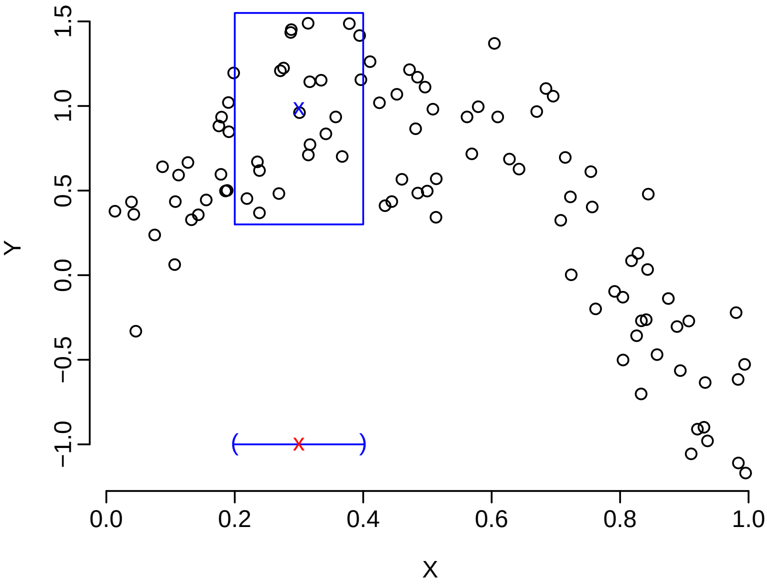


Nonparametric Regression

Obtain a regression estimate with very few assumptions.

⋆ The assumption involves infinite number of parameters.

– e.g., continuity, smoothness etc.

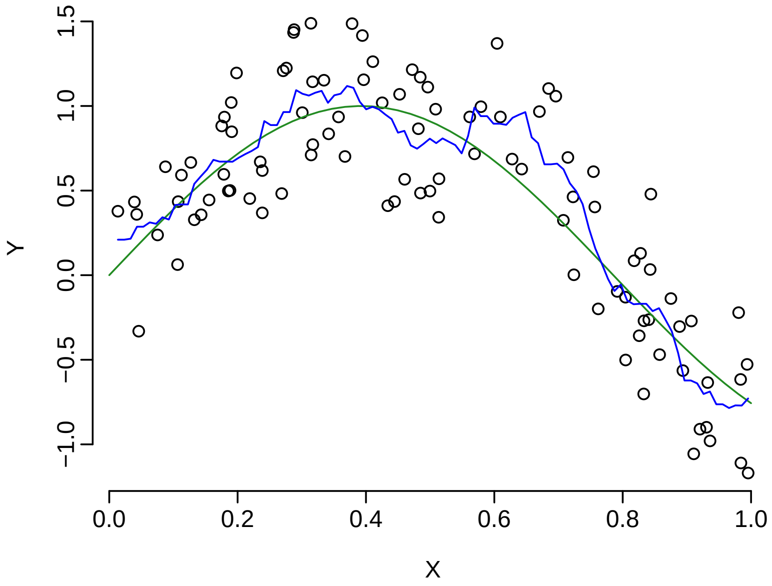


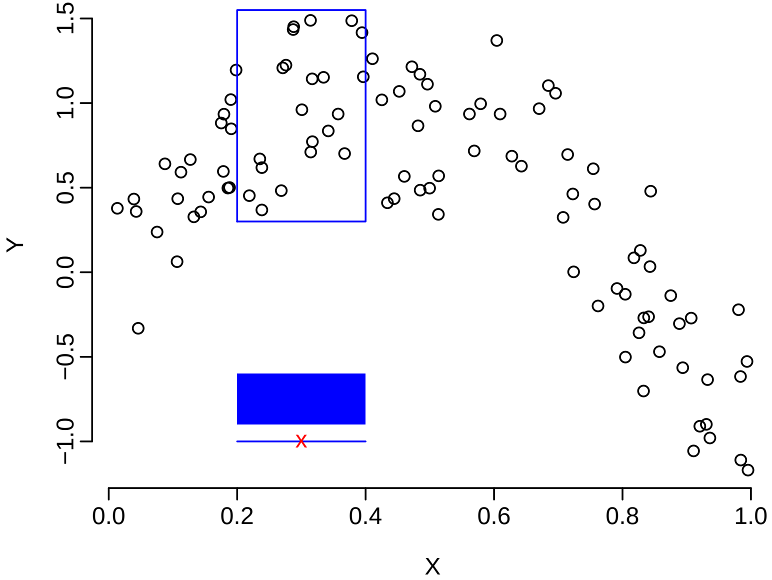
Nonparametric Regression

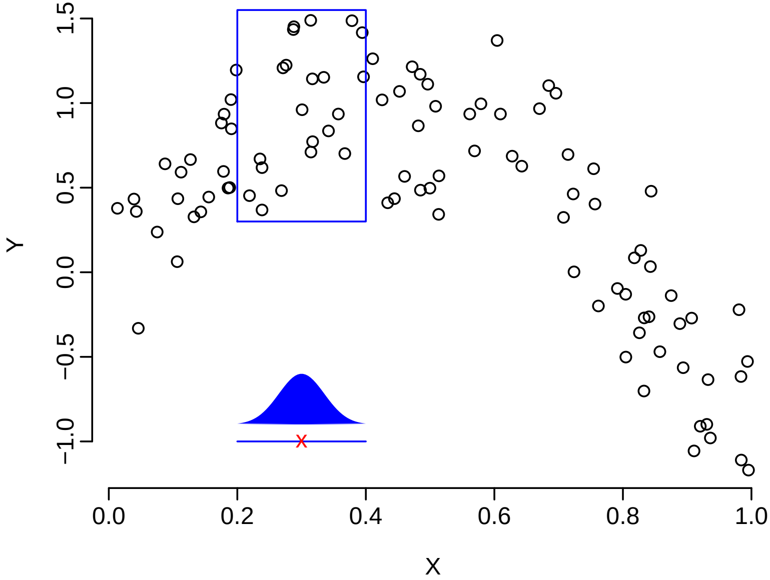
Obtain a regression estimate with very few assumptions.

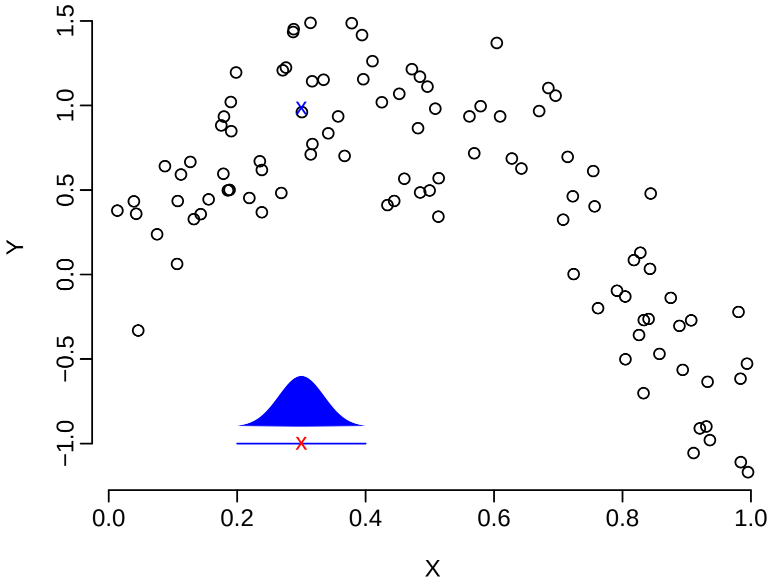
⋆ The assumption involves infinite number of parameters.

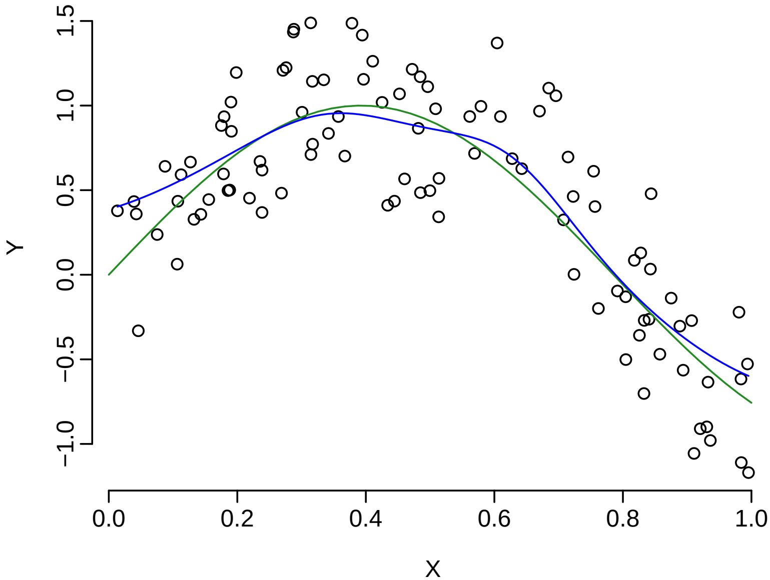
– e.g., continuity, smoothness etc.

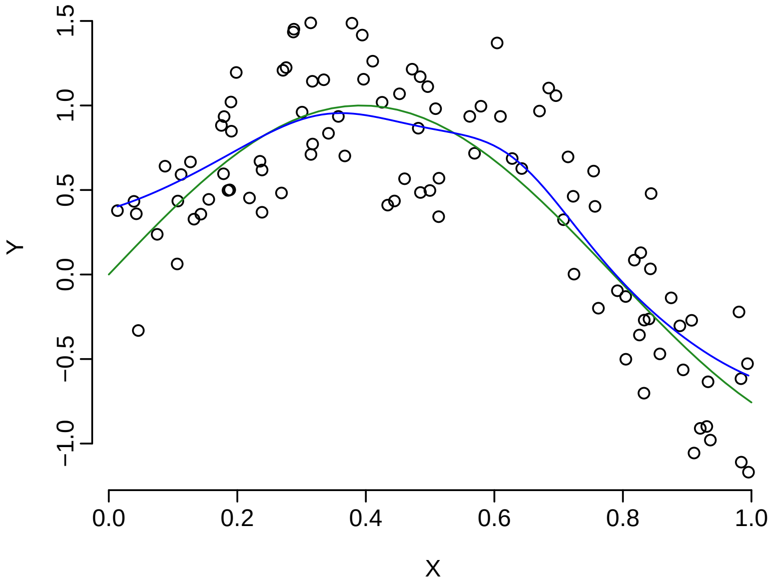



1.5
1.0
0.5
0.0
−0.5
−1.0
Y
x
0.0
0.2
0.4
0.6
0.8
1.0
X


1.5
1.0
0.5
0.0
−0.5
−1.0
Y
x
0.0
0.2
0.4
0.6
0.8
1.0
X


1.5
1.0
0.5
0.0
−0.5
−1.0
Y
0.0
0.2
0.4
0.6
0.8
1.0
X


1.5
1.0
0.5
0.0
−0.5
−1.0
Y
0.0
0.2
0.4
0.6
0.8
1.0
X



$$\widehat{f}(x) = \frac{\sum_{i=1}^{n} y_i K\left(\frac{x-x_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)}$$ — the Nadaraya-Watson estimator (1964).

K is a kernel. Usually: $\int K(u)\,\mathrm{d}u = 1, \int uK(u)\,\mathrm{d}u = 0, \int u^2 K(u)\,\mathrm{d}u < \infty.$

Density estimation

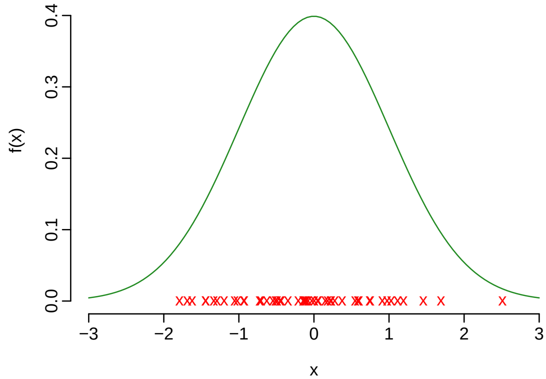
$x_1, \dots, x_n \overset{i.i.d.}{\sim} f$ (density)

Want to estimate f from $x_1, \dots, x_n$

Density estimation

$x_1, \ldots, x_n \overset{i.i.d.}{\sim} f$ (density)

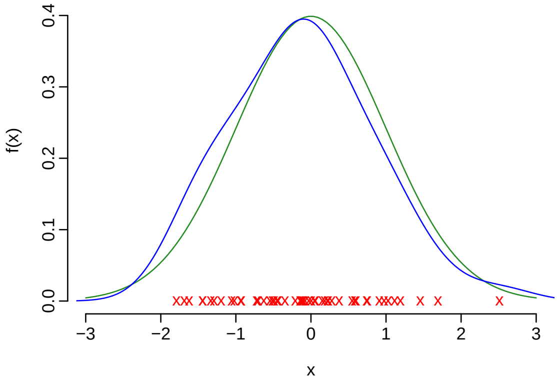
Want to estimate f from $x_1, \ldots, x_n$



Density estimation

$x_1, \ldots, x_n \overset{i.i.d.}{\sim} f$ (density)

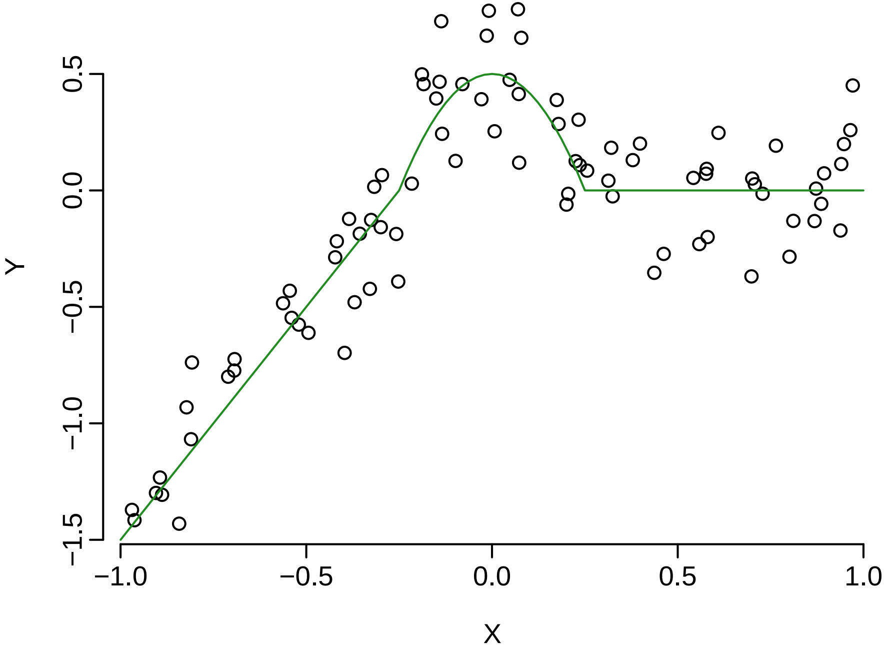
Want to estimate f from $x_1, \ldots, x_n$



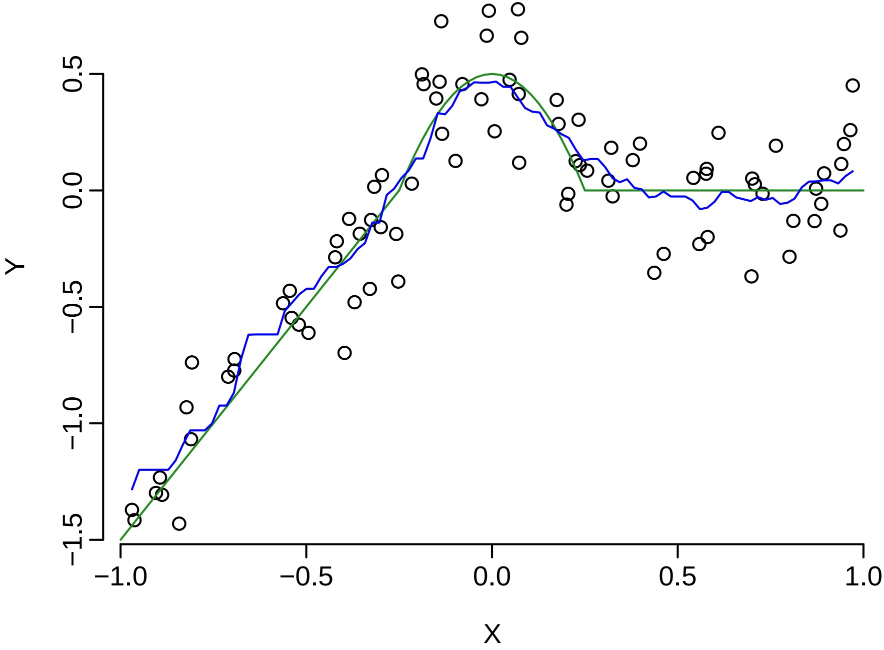
Kernel density estimator: $$\widehat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

— Rosenblatt (1956), Parzen (1962)

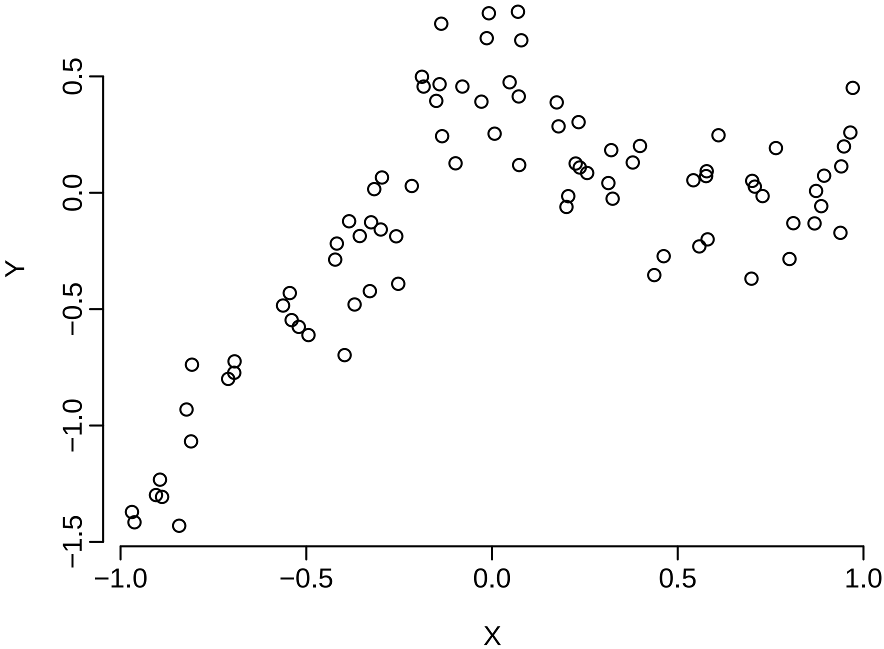
Back to regression



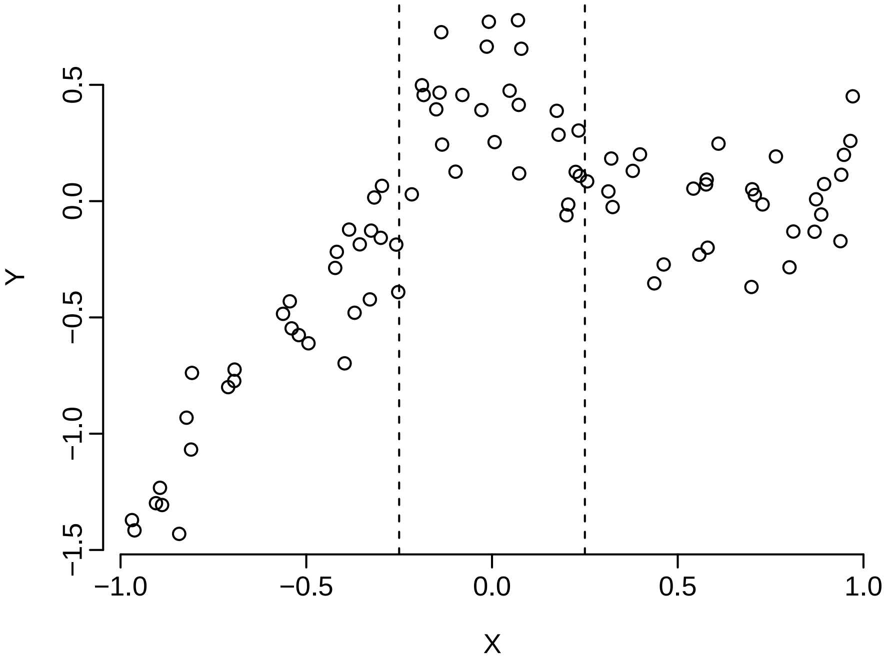
Back to regression



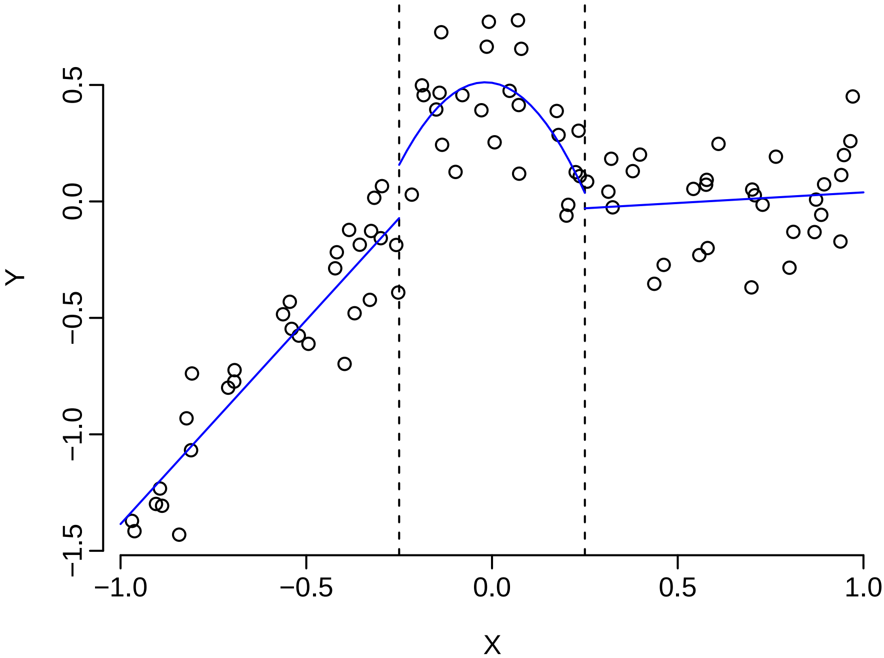
Back to regression



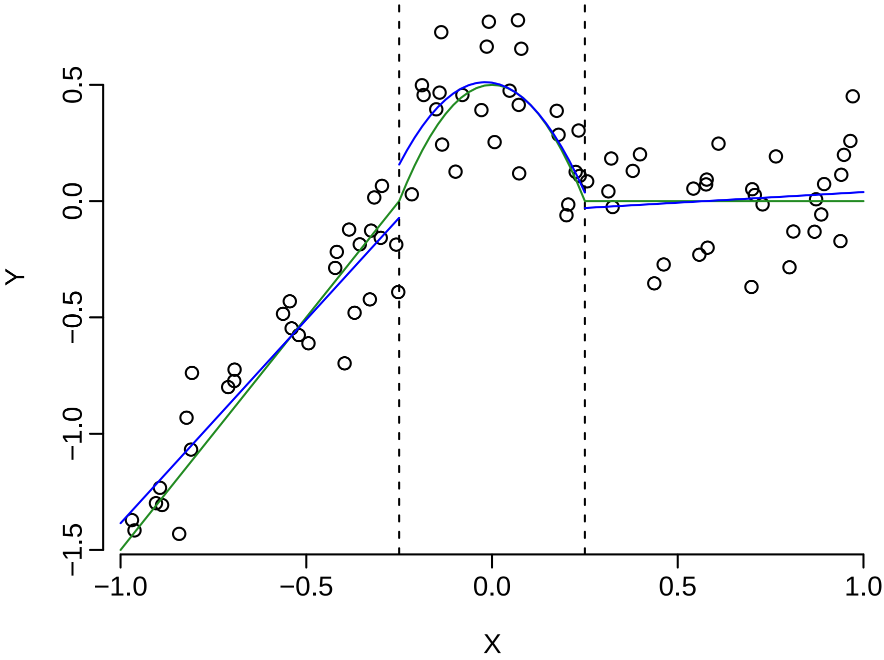
Back to regression



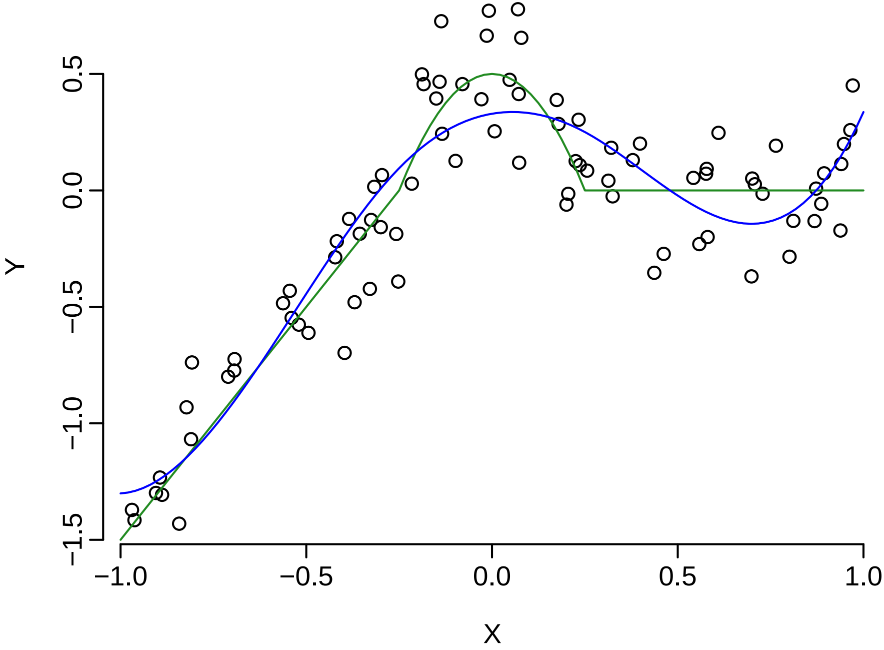
Back to regression



Back to regression



Back to regression



Spline: Fit local polynomials while enforcing continuity of f and its derivatives.

Cubic spline f with knots $\xi_1, \ldots, \xi_K$: f continuous, f', f'' continuous at $\xi_1, \ldots, \xi_K$

Spline: Fit local polynomials while enforcing continuity of f and its derivatives.

Cubic spline f with knots $\xi_1, \ldots, \xi_K$: f continuous, f', f'' continuous at $\xi_1, \ldots, \xi_K$

$$f(x) = \sum_{j=0}^{K+2} \beta_j h_j(x) \qquad h_0(x) \equiv 1, h_1(x) = x, h_2(x) = x^2, h_{j+2}(x) = (x - \xi_j)_+^3$$

Spline: Fit local polynomials while enforcing continuity of f and its derivatives.

Cubic spline f with knots $\xi_1, \ldots, \xi_K$: f continuous, f', f'' continuous at $\xi_1, \ldots, \xi_K$

$$f(x) = \sum_{j=0}^{K+2} \beta_j h_j(x) \qquad h_0(x) \equiv 1, h_1(x) = x, h_2(x) = x^2, h_{j+2}(x) = (x - \xi_j)_+^3$$

Issues: How many knots? Where to place them?

Spline: Fit local polynomials while enforcing continuity of f and its derivatives.

Cubic spline f with knots $\xi_1, \dots, \xi_K$: f continuous, f', f'' continuous at $\xi_1, \dots, \xi_K$

$$f(x) = \sum_{j=0}^{K+2} \beta_j h_j(x) \qquad h_0(x) \equiv 1, h_1(x) = x, h_2(x) = x^2, h_{j+2}(x) = (x - \xi_j)_+^3$$

Issues: How many knots? Where to place them?

Smoothing spline: Find $\widehat{f}$ that minimizes

$$\sum_{i=1}^{n} (y_i - f(X_i))^2 + \lambda \int \{f''(t)\}^2 \, \mathrm{d}t.$$

Spline: Fit local polynomials while enforcing continuity of f and its derivatives.

Cubic spline f with knots $\xi_1, \ldots, \xi_K$: f continuous, f', f'' continuous at $\xi_1, \ldots, \xi_K$

$$f(x) = \sum_{j=0}^{K+2} \beta_j h_j(x) \qquad h_0(x) \equiv 1, h_1(x) = x, h_2(x) = x^2, h_{j+2}(x) = (x - \xi_j)_+^3$$

Issues: How many knots? Where to place them?

Smoothing spline: Find $\widehat{f}$ that minimizes

$$\sum_{i=1}^{n} (y_i - f(X_i))^2 + \lambda \int \{f''(t)\}^2 \, \mathrm{d}t.$$

A plausible f should be twice-differentiable and f'' should be in $\mathcal{L}_2$.

Spline: Fit local polynomials while enforcing continuity of f and its derivatives.

Cubic spline f with knots $\xi_1, \ldots, \xi_K$: f continuous, f', f'' continuous at $\xi_1, \ldots, \xi_K$

$$f(x) = \sum_{j=0}^{K+2} \beta_j h_j(x) \qquad h_0(x) \equiv 1, h_1(x) = x, h_2(x) = x^2, h_{j+2}(x) = (x - \xi_j)_+^3$$

Issues: How many knots? Where to place them?

Smoothing spline: Find $\widehat{f}$ that minimizes

$$\sum_{i=1}^{n} (y_i - f(X_i))^2 + \lambda \int \{f''(t)\}^2 \,\mathrm{d}t.$$

A plausible f should be twice-differentiable and f'' should be in $\mathcal{L}_2$.

⋆ f resides in a Sobolev space — an RKHS. (More later)

Spline: Fit local polynomials while enforcing continuity of f and its derivatives.

Cubic spline f with knots $\xi_1, \ldots, \xi_K$: f continuous, f', f'' continuous at $\xi_1, \ldots, \xi_K$

$$f(x) = \sum_{j=0}^{K+2} \beta_j h_j(x) \qquad h_0(x) \equiv 1, h_1(x) = x, h_2(x) = x^2, h_{j+2}(x) = (x - \xi_j)_+^3$$

Issues: How many knots? Where to place them?

Smoothing spline: Find $\widehat{f}$ that minimizes

$$\sum_{i=1}^{n} (y_i - f(X_i))^2 + \lambda \int \{f''(t)\}^2 \, \mathrm{d}t.$$

A plausible f should be twice-differentiable and f'' should be in $\mathcal{L}_2$.

$\star$ f resides in a Sobolev space — an RKHS. (More later)

Remarkably, the solution exists and is given by a cubic spline with knots at $x_1, \ldots, x_n$.

— Representer theorem. Kimeldorf & Wahba (1971)

The solution can be obtained in a closed form.

Modern kernels

Modern kernels

Definition

A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive definite (p.d.) kernel if for every $n \geq 1$, $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ and $c_1, \ldots, c_n \in \mathbb{R}$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0.$$

K is called strictly positive definite if for distinct $\mathbf{x}_1, \ldots, \mathbf{x}_n$, "=" above holds if and only if $c_1 = \cdots = c_n = 0$.

Modern kernels

Definition

A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive definite (p.d.) kernel if for every $n \geq 1$, $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ and $c_1, \ldots, c_n \in \mathbb{R}$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0.$$

K is called strictly positive definite if for distinct $\mathbf{x}_1, \ldots, \mathbf{x}_n$, "=" above holds if and only if $c_1 = \cdots = c_n = 0$.

The $n \times n$ matrix $\big(\big(K(\mathbf{x}_i, \mathbf{x}_j)\big)\big)_{1 \leq i,j \leq n}$ is positive semi-definite (or positive definite).

Modern kernels

Definition

A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive definite (p.d.) kernel if for every $n \geq 1$, $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ and $c_1, \ldots, c_n \in \mathbb{R}$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0.$$

K is called strictly positive definite if for distinct $\mathbf{x}_1, \ldots, \mathbf{x}_n$, "=" above holds if and only if $c_1 = \cdots = c_n = 0$.

The $n \times n$ matrix $\big((K(\mathbf{x}_i, \mathbf{x}_j))\big)_{1 \leq i,j \leq n}$ is positive semi-definite (or positive definite).

⋆ Sometimes, K is called non-negative definite and positive definite. Always check the definition.

Modern kernels

Definition

A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive definite (p.d.) kernel if for every $n \geq 1$, $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ and $c_1, \ldots, c_n \in \mathbb{R}$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0.$$

K is called strictly positive definite if for distinct $\mathbf{x}_1, \ldots, \mathbf{x}_n$, "=" above holds if and only if $c_1 = \cdots = c_n = 0$.

The $n \times n$ matrix $\big((K(\mathbf{x}_i, \mathbf{x}_j))\big)_{1 \leq i,j \leq n}$ is positive semi-definite (or positive definite).

- ⋆ Sometimes, K is called non-negative definite and positive definite. Always check the definition.
- ⋆ K is called a Mercer kernel if it is p.d. and continuous.

Modern kernels

Definition

A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive definite (p.d.) kernel if for every $n \geq 1$, $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ and $c_1, \ldots, c_n \in \mathbb{R}$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0.$$

K is called strictly positive definite if for distinct $\mathbf{x}_1, \ldots, \mathbf{x}_n$, "=" above holds if and only if $c_1 = \cdots = c_n = 0$.

The $n \times n$ matrix $\big((K(\mathbf{x}_i, \mathbf{x}_j))\big)_{1 \leq i,j \leq n}$ is positive semi-definite (or positive definite).

- ⋆ Sometimes, K is called non-negative definite and positive definite. Always check the definition.
- ⋆ K is called a Mercer kernel if it is p.d. and continuous.

Ex. Inner-product. $K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$

Modern kernels

Definition

A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive definite (p.d.) kernel if for every $n \geq 1$, $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ and $c_1, \ldots, c_n \in \mathbb{R}$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0.$$

K is called strictly positive definite if for distinct $\mathbf{x}_1, \ldots, \mathbf{x}_n$, "=" above holds if and only if $c_1 = \cdots = c_n = 0$.

The $n \times n$ matrix $\big(\big(K(\mathbf{x}_i, \mathbf{x}_j)\big)\big)_{1 \leq i,j \leq n}$ is positive semi-definite (or positive definite).

- ⋆ Sometimes, K is called non-negative definite and positive definite. Always check the definition.
- ⋆ K is called a Mercer kernel if it is p.d. and continuous.

Ex. Inner-product. $K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$

Lemma

$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a p.d. kernel if and only if there exists a Hilbert space $\mathcal{H}$ and a map $\phi : \mathcal{X} \to \mathcal{H}$ such that

$$K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}, \qquad \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}.$$

Modern kernels

Definition

A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive definite (p.d.) kernel if for every $n \geq 1$, $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathcal{X}$ and $c_1, \dots, c_n \in \mathbb{R}$,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0.$$

K is called strictly positive definite if for distinct $\mathbf{x}_1, \dots, \mathbf{x}_n$, "=" above holds if and only if $c_1 = \cdots = c_n = 0$.

The $n \times n$ matrix $\big((K(\mathbf{x}_i, \mathbf{x}_j))\big)_{1 \leq i,j \leq n}$ is positive semi-definite (or positive definite).

- ⋆ Sometimes, K is called non-negative definite and positive definite. Always check the definition.
- ⋆ K is called a Mercer kernel if it is p.d. and continuous.

Ex. Inner-product. $K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$

Lemma

$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a p.d. kernel if and only if there exists a Hilbert space $\mathcal{H}$ and a map $\phi : \mathcal{X} \to \mathcal{H}$ such that

$$K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}, \qquad \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}.$$

- A kernel induces and is induced by an inner-product. Has important consequences.

Some Properties of Kernels

From now on, by kernel we mean p.d. kernel

Some Properties of Kernels

From now on, by kernel we mean p.d. kernel

- K_1, K_2 are kernels on $\mathcal{X} \quad \Rightarrow \quad K_1 + K_2$ is a kernel on $\mathcal{X}$.

$$\left[(K_1 + K_2)(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y}) + K_2(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X} \right]$$

Some Properties of Kernels

From now on, by kernel we mean p.d. kernel

- K_1, K_2 are kernels on $\mathcal{X} \quad \Rightarrow \quad K_1 + K_2$ is a kernel on $\mathcal{X}$.

$$\Big[(K_1 + K_2)(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y}) + K_2(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}\Big]$$

- K_1, K_2 are kernels on $\mathcal{X} \quad \Rightarrow \quad K_1 \times K_2$ is a kernel on $\mathcal{X}$.

$$\Big[(K_1 \times K_2)(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y}) \times K_2(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}\Big]$$

Some Properties of Kernels

From now on, by kernel we mean p.d. kernel

- K_1, K_2 are kernels on $\mathcal{X} \quad \Rightarrow \quad K_1 + K_2$ is a kernel on $\mathcal{X}$.

  $$\left[(K_1 + K_2)(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y}) + K_2(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}\right]$$

- K_1, K_2 are kernels on $\mathcal{X} \quad \Rightarrow \quad K_1 \times K_2$ is a kernel on $\mathcal{X}$.

  $$\left[(K_1 \times K_2)(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y}) \times K_2(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}\right]$$

- K is a kernel on $\mathcal{X}$, $A : \widetilde{\mathcal{X}} \to \mathcal{X} \quad \Rightarrow \quad \widetilde{K}(\mathbf{x}, \mathbf{y}) = K(A(\mathbf{x}), A(\mathbf{y}))$ is a kernel on $\widetilde{\mathcal{X}}$

Some Properties of Kernels

From now on, by kernel we mean p.d. kernel

- K_1, K_2 are kernels on $\mathcal{X} \quad \Rightarrow \quad K_1 + K_2$ is a kernel on $\mathcal{X}$.

$$\left[(K_1 + K_2)(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y}) + K_2(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}\right]$$

- K_1, K_2 are kernels on $\mathcal{X} \quad \Rightarrow \quad K_1 \times K_2$ is a kernel on $\mathcal{X}$.

$$\left[(K_1 \times K_2)(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y}) \times K_2(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}\right]$$

- K is a kernel on $\mathcal{X}$, $A : \widetilde{\mathcal{X}} \to \mathcal{X} \quad \Rightarrow \quad \widetilde{K}(\mathbf{x}, \mathbf{y}) = K(A(\mathbf{x}), A(\mathbf{y}))$ is a kernel on $\widetilde{\mathcal{X}}$

Using these, we can show that the following are kernels:

Some Properties of Kernels

From now on, by kernel we mean p.d. kernel

- K_1, K_2 are kernels on $\mathcal{X} \quad \Rightarrow \quad K_1 + K_2$ is a kernel on $\mathcal{X}$.

$$\Big[(K_1 + K_2)(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y}) + K_2(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}\Big]$$

- K_1, K_2 are kernels on $\mathcal{X} \quad \Rightarrow \quad K_1 \times K_2$ is a kernel on $\mathcal{X}$.

$$\Big[(K_1 \times K_2)(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y}) \times K_2(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}\Big]$$

- K is a kernel on $\mathcal{X}$, $A : \widetilde{\mathcal{X}} \to \mathcal{X} \quad \Rightarrow \quad \widetilde{K}(\mathbf{x}, \mathbf{y}) = K(A(\mathbf{x}), A(\mathbf{y}))$ is a kernel on $\widetilde{\mathcal{X}}$

Using these, we can show that the following are kernels:

  - $K(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + c)^m, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d. \quad (c \geq 0,\ m \in \mathbb{N})$ Polynomial kernel.

Some Properties of Kernels

From now on, by kernel we mean p.d. kernel

- K_1, K_2 are kernels on $\mathcal{X} \quad \Rightarrow \quad K_1 + K_2$ is a kernel on $\mathcal{X}$.

  $$\Big[(K_1 + K_2)(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y}) + K_2(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}\Big]$$

- K_1, K_2 are kernels on $\mathcal{X} \quad \Rightarrow \quad K_1 \times K_2$ is a kernel on $\mathcal{X}$.

  $$\Big[(K_1 \times K_2)(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y}) \times K_2(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}\Big]$$

- K is a kernel on $\mathcal{X}$, $A : \widetilde{\mathcal{X}} \to \mathcal{X} \quad \Rightarrow \quad \widetilde{K}(\mathbf{x}, \mathbf{y}) = K(A(\mathbf{x}), A(\mathbf{y}))$ is a kernel on $\widetilde{\mathcal{X}}$

Using these, we can show that the following are kernels:

- $K(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + c)^m, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d. \quad (c \geq 0,\ m \in \mathbb{N})$ Polynomial kernel.
- $K(\mathbf{x}, \mathbf{y}) = e^{\langle \mathbf{x}, \mathbf{y} \rangle}, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d.$ Exponential kernel.

Some Properties of Kernels

From now on, by kernel we mean p.d. kernel

- K_1, K_2 are kernels on $\mathcal{X} \quad \Rightarrow \quad K_1 + K_2$ is a kernel on $\mathcal{X}$.

  $$\left[(K_1 + K_2)(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y}) + K_2(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}\right]$$

- K_1, K_2 are kernels on $\mathcal{X} \quad \Rightarrow \quad K_1 \times K_2$ is a kernel on $\mathcal{X}$.

  $$\left[(K_1 \times K_2)(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y}) \times K_2(\mathbf{x}, \mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{X}\right]$$

- K is a kernel on $\mathcal{X}$, $A : \widetilde{\mathcal{X}} \to \mathcal{X} \quad \Rightarrow \quad \widetilde{K}(\mathbf{x}, \mathbf{y}) = K(A(\mathbf{x}), A(\mathbf{y}))$ is a kernel on $\widetilde{\mathcal{X}}$

Using these, we can show that the following are kernels:

- $K(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + c)^m, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d. \quad (c \geq 0,\ m \in \mathbb{N})$ Polynomial kernel.
- $K(\mathbf{x}, \mathbf{y}) = e^{\langle \mathbf{x}, \mathbf{y} \rangle}, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d.$ Exponential kernel.
- $K(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2}}, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d.$ Gaussian/RBF kernel

Reproducing Kernel Hilbert Space

Definition

Let $\mathcal{X}$ be a non-empty set. Let $\mathscr{H}$ be a Hilbert space of real functions on $\mathcal{X}$. $\mathscr{H}$ is called a reproducing kernel Hilbert space (RKHS) if there exists a p.d. kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that

- $K(\cdot, \mathbf{x}) \in \mathscr{H} \quad \forall \mathbf{x} \in \mathcal{X}$.
- $f(\mathbf{x}) = \langle f, K(\cdot, \mathbf{x}) \rangle_{\mathscr{H}} \quad \forall f \in \mathscr{H}, \mathbf{x} \in \mathcal{X}$.

In this case, K is called the reproducing kernel (rk) associated with $\mathscr{H}$.

Reproducing Kernel Hilbert Space

Definition

Let $\mathcal{X}$ be a non-empty set. Let $\mathscr{H}$ be a Hilbert space of real functions on $\mathcal{X}$. $\mathscr{H}$ is called a reproducing kernel Hilbert space (RKHS) if there exists a p.d. kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that

- $K(\cdot, \mathbf{x}) \in \mathscr{H} \quad \forall \mathbf{x} \in \mathcal{X}$.
- $f(\mathbf{x}) = \langle f, K(\cdot, \mathbf{x}) \rangle_{\mathscr{H}} \quad \forall f \in \mathscr{H}, \mathbf{x} \in \mathcal{X}$.

In this case, K is called the reproducing kernel (rk) associated with $\mathscr{H}$.

Evaluation of a function $f \in \mathscr{H}$ can be obtained via inner-product with K

— the reproducing property.

Reproducing Kernel Hilbert Space

Definition

Let $\mathcal{X}$ be a non-empty set. Let $\mathscr{H}$ be a Hilbert space of real functions on $\mathcal{X}$. $\mathscr{H}$ is called a reproducing kernel Hilbert space (RKHS) if there exists a p.d. kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that

- $K(\cdot, \mathbf{x}) \in \mathscr{H} \quad \forall \mathbf{x} \in \mathcal{X}$.
- $f(\mathbf{x}) = \langle f, K(\cdot, \mathbf{x}) \rangle_{\mathscr{H}} \quad \forall f \in \mathscr{H}, \mathbf{x} \in \mathcal{X}$.

In this case, K is called the reproducing kernel (rk) associated with $\mathscr{H}$.

Evaluation of a function $f \in \mathscr{H}$ can be obtained via inner-product with K

— the reproducing property.

⋆ An RKHS is equivalent to its associated rk. (Moore-Aronszajn Theorem)

Reproducing Kernel Hilbert Space

Definition

Let $\mathcal{X}$ be a non-empty set. Let $\mathscr{H}$ be a Hilbert space of real functions on $\mathcal{X}$. $\mathscr{H}$ is called a reproducing kernel Hilbert space (RKHS) if there exists a p.d. kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that

- $K(\cdot, \mathbf{x}) \in \mathscr{H} \quad \forall \mathbf{x} \in \mathcal{X}$.
- $f(\mathbf{x}) = \langle f, K(\cdot, \mathbf{x}) \rangle_{\mathscr{H}} \quad \forall f \in \mathscr{H}, \mathbf{x} \in \mathcal{X}$.

In this case, K is called the reproducing kernel (rk) associated with $\mathscr{H}$.

Evaluation of a function $f \in \mathscr{H}$ can be obtained via inner-product with K

— the reproducing property.

⋆ An RKHS is equivalent to its associated rk. (Moore-Aronszajn Theorem)

Some properties:

Reproducing Kernel Hilbert Space

Definition

Let $\mathcal{X}$ be a non-empty set. Let $\mathscr{H}$ be a Hilbert space of real functions on $\mathcal{X}$. $\mathscr{H}$ is called a reproducing kernel Hilbert space (RKHS) if there exists a p.d. kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that

- $K(\cdot, \mathbf{x}) \in \mathscr{H} \quad \forall \mathbf{x} \in \mathcal{X}$.
- $f(\mathbf{x}) = \langle f, K(\cdot, \mathbf{x}) \rangle_{\mathscr{H}} \quad \forall f \in \mathscr{H}, \mathbf{x} \in \mathcal{X}$.

In this case, K is called the reproducing kernel (rk) associated with $\mathscr{H}$.

Evaluation of a function $f \in \mathscr{H}$ can be obtained via inner-product with K

— the reproducing property.

⋆ An RKHS is equivalent to its associated rk. (Moore-Aronszajn Theorem)

Some properties:

- $K(\mathbf{x}, \mathbf{y}) = \langle K(\cdot, \mathbf{x}), K(\cdot, \mathbf{y}) \rangle_{\mathscr{H}} \quad \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$.

Reproducing Kernel Hilbert Space

Definition

Let $\mathcal{X}$ be a non-empty set. Let $\mathscr{H}$ be a Hilbert space of real functions on $\mathcal{X}$. $\mathscr{H}$ is called a reproducing kernel Hilbert space (RKHS) if there exists a p.d. kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that

- $K(\cdot, \mathbf{x}) \in \mathscr{H} \quad \forall \mathbf{x} \in \mathcal{X}$.
- $f(\mathbf{x}) = \langle f, K(\cdot, \mathbf{x}) \rangle_{\mathscr{H}} \quad \forall f \in \mathscr{H}, \mathbf{x} \in \mathcal{X}$.

In this case, K is called the reproducing kernel (rk) associated with $\mathscr{H}$.

Evaluation of a function $f \in \mathscr{H}$ can be obtained via inner-product with K

— the reproducing property.

⋆ An RKHS is equivalent to its associated rk. (Moore-Aronszajn Theorem)

Some properties:

- $K(\mathbf{x}, \mathbf{y}) = \langle K(\cdot, \mathbf{x}), K(\cdot, \mathbf{y}) \rangle_{\mathscr{H}} \quad \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$.
- An RKHS consists of functions of the form $\sum_{i=1}^{m} \alpha_i K(\cdot, \mathbf{x}_i)$ and their limits (w.r.t. the inner-product)

Reproducing Kernel Hilbert Space

Definition

Let $\mathcal{X}$ be a non-empty set. Let $\mathscr{H}$ be a Hilbert space of real functions on $\mathcal{X}$. $\mathscr{H}$ is called a reproducing kernel Hilbert space (RKHS) if there exists a p.d. kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that

- $K(\cdot, \mathbf{x}) \in \mathscr{H} \quad \forall \mathbf{x} \in \mathcal{X}$.
- $f(\mathbf{x}) = \langle f, K(\cdot, \mathbf{x}) \rangle_{\mathscr{H}} \quad \forall f \in \mathscr{H}, \mathbf{x} \in \mathcal{X}$.

In this case, K is called the reproducing kernel (rk) associated with $\mathscr{H}$.

Evaluation of a function $f \in \mathscr{H}$ can be obtained via inner-product with K

— the reproducing property.

⋆ An RKHS is equivalent to its associated rk. (Moore-Aronszajn Theorem)

Some properties:

- $K(\mathbf{x}, \mathbf{y}) = \langle K(\cdot, \mathbf{x}), K(\cdot, \mathbf{y}) \rangle_{\mathscr{H}} \quad \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$.
- An RKHS consists of functions of the form $\sum_{i=1}^{m} \alpha_i K(\cdot, \mathbf{x}_i)$ and their limits (w.r.t. the inner-product)
- For $f = \sum_{i=1}^{m} \alpha_i K(\cdot, \mathbf{x}_i)$ and $g = \sum_{j=1}^{n} \beta_j K(\cdot, \mathbf{y}_j)$, $\langle f, g \rangle_{\mathscr{H}} = \sum_{i=1}^{m} \sum_{j=1}^{n} \alpha_i \beta_j K(\mathbf{x}_i, \mathbf{y}_j)$.

The Representer Theorem

Let $\mathscr{H}$ be an RKHS with kernel K. Consider the following problem:

$$\text{minimize} \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_{\mathscr{H}}^2 \quad \text{w.r.t. } f \in \mathscr{H}.$$

The Representer Theorem

Let $\mathscr{H}$ be an RKHS with kernel K. Consider the following problem:

$$\text{minimize } \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_{\mathscr{H}}^2 \quad \text{w.r.t. } f \in \mathscr{H}.$$

We are trying to find/estimate a function. An infinite-dimensional problem!

The Representer Theorem

Let $\mathscr{H}$ be an RKHS with kernel K. Consider the following problem:

$$\text{minimize } \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_{\mathscr{H}}^2 \quad \text{w.r.t. } f \in \mathscr{H}.$$

We are trying to find/estimate a function. An infinite-dimensional problem!

Remarkably, the minimizer has the form

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i K(\cdot, \mathbf{x}_i).$$

The Representer Theorem

Let $\mathscr{H}$ be an RKHS with kernel K. Consider the following problem:

$$\text{minimize} \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_{\mathscr{H}}^2 \quad \text{w.r.t. } f \in \mathscr{H}.$$

We are trying to find/estimate a function. An infinite-dimensional problem!

Remarkably, the minimizer has the form

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i K(\cdot, \mathbf{x}_i).$$

- The infinite-dimensional problem reduces to finding coefficients $\alpha_1, \ldots, \alpha_n$.

The Representer Theorem

Let $\mathscr{H}$ be an RKHS with kernel K. Consider the following problem:

$$\text{minimize } \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_{\mathscr{H}}^2 \quad \text{w.r.t. } f \in \mathscr{H}.$$

We are trying to find/estimate a function. An infinite-dimensional problem!

Remarkably, the minimizer has the form

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i K(\cdot, \mathbf{x}_i).$$

- The infinite-dimensional problem reduces to finding coefficients $\alpha_1, \ldots, \alpha_n$.

In the spline problem, we wanted to:

$$\text{minimize } \sum_{i=1}^{n} (Y_i - f(X_i))^2 + \lambda \int (f''(t))^2 \, \mathrm{d}t.$$

The Representer Theorem

Let $\mathscr{H}$ be an RKHS with kernel K. Consider the following problem:

$$\text{minimize} \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_{\mathscr{H}}^2 \quad \text{w.r.t. } f \in \mathscr{H}.$$

We are trying to find/estimate a function. An infinite-dimensional problem!

Remarkably, the minimizer has the form

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i K(\cdot, \mathbf{x}_i).$$

- The infinite-dimensional problem reduces to finding coefficients $\alpha_1, \dots, \alpha_n$.

In the spline problem, we wanted to:

$$\text{minimize} \sum_{i=1}^{n} (Y_i - f(X_i))^2 + \lambda \int (f''(t))^2 \, \mathrm{d}t.$$

Can show: $\displaystyle\int (f''(t))^2 \, \mathrm{d}t = \|f\|_{\mathscr{H}}^2$ in an appropriate RKHS $\mathscr{H}$. This leads to our previous result.

Kernel As A Tool for Dimension Augmentation

In regression spline, we did the following:

- From the given feature $x \in \mathbb{R}$, derive new features $x, x^2, (x - x_1)_+^3, \dots, (x - x_n)_+^3$.
- Perform linear regression with the derived features.

In essence, we augment the feature dimension and use linear methods in the augmented space!

— A recurring theme in modern kernel methods.

Kernel As A Tool for Dimension Augmentation

In regression spline, we did the following:

- From the given feature $x \in \mathbb{R}$, derive new features $x, x^2, (x - x_1)_+^3, \ldots, (x - x_n)_+^3$.
- Perform linear regression with the derived features.

In essence, we augment the feature dimension and use linear methods in the augmented space!

— A recurring theme in modern kernel methods.

Let $\mathcal{H}$ be a Hilbert space and $\phi : \mathbb{R}^d \to \mathcal{H}$ be a feature map.

$$\|\phi(\mathbf{x}) - \phi(\mathbf{y})\|_{\mathcal{H}}^2$$

Kernel As A Tool for Dimension Augmentation

In regression spline, we did the following:

- From the given feature $x \in \mathbb{R}$, derive new features $x, x^2, (x - x_1)_+^3, \ldots, (x - x_n)_+^3$.
- Perform linear regression with the derived features.

In essence, we augment the feature dimension and use linear methods in the augmented space!

— A recurring theme in modern kernel methods.

Let $\mathcal{H}$ be a Hilbert space and $\phi : \mathbb{R}^d \to \mathcal{H}$ be a feature map.

$$\|\phi(\mathbf{x}) - \phi(\mathbf{y})\|_{\mathcal{H}}^2 = \|\phi(\mathbf{x})\|_{\mathcal{H}}^2 + \|\phi(\mathbf{y})\|_{\mathcal{H}}^2 - 2\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}$$

Kernel As A Tool for Dimension Augmentation

In regression spline, we did the following:

- From the given feature $x \in \mathbb{R}$, derive new features $x, x^2, (x - x_1)_+^3, \ldots, (x - x_n)_+^3$.
- Perform linear regression with the derived features.

In essence, we augment the feature dimension and use linear methods in the augmented space!

— A recurring theme in modern kernel methods.

Let $\mathcal{H}$ be a Hilbert space and $\phi : \mathbb{R}^d \to \mathcal{H}$ be a feature map.

$$\|\phi(\mathbf{x}) - \phi(\mathbf{y})\|_{\mathcal{H}}^2 = \|\phi(\mathbf{x})\|_{\mathcal{H}}^2 + \|\phi(\mathbf{y})\|_{\mathcal{H}}^2 - 2\underbrace{\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}}_{K(\mathbf{x},\mathbf{y})}$$

Kernel As A Tool for Dimension Augmentation

In regression spline, we did the following:

- From the given feature $x \in \mathbb{R}$, derive new features $x, x^2, (x - x_1)_+^3, \ldots, (x - x_n)_+^3$.
- Perform linear regression with the derived features.

In essence, we augment the feature dimension and use linear methods in the augmented space!

— A recurring theme in modern kernel methods.

Let $\mathcal{H}$ be a Hilbert space and $\phi : \mathbb{R}^d \to \mathcal{H}$ be a feature map.

$$\begin{aligned} \|\phi(\mathbf{x}) - \phi(\mathbf{y})\|_{\mathcal{H}}^2 &= \|\phi(\mathbf{x})\|_{\mathcal{H}}^2 + \|\phi(\mathbf{y})\|_{\mathcal{H}}^2 - 2 \underbrace{\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}}_{K(\mathbf{x},\mathbf{y})} \\ &= K(\mathbf{x}, \mathbf{x}) + K(\mathbf{y}, \mathbf{y}) - 2K(\mathbf{x}, \mathbf{y}) \end{aligned}$$

Kernel As A Tool for Dimension Augmentation

In regression spline, we did the following:

- From the given feature $x \in \mathbb{R}$, derive new features $x, x^2, (x - x_1)_+^3, \ldots, (x - x_n)_+^3$.
- Perform linear regression with the derived features.

In essence, we augment the feature dimension and use linear methods in the augmented space!

— A recurring theme in modern kernel methods.

Let $\mathcal{H}$ be a Hilbert space and $\phi : \mathbb{R}^d \to \mathcal{H}$ be a feature map.

$$
\begin{aligned}
\|\phi(\mathbf{x}) - \phi(\mathbf{y})\|_{\mathcal{H}}^2 &= \|\phi(\mathbf{x})\|_{\mathcal{H}}^2 + \|\phi(\mathbf{y})\|_{\mathcal{H}}^2 - 2\underbrace{\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}}_{K(\mathbf{x},\mathbf{y})} \\
&= K(\mathbf{x}, \mathbf{x}) + K(\mathbf{y}, \mathbf{y}) - 2K(\mathbf{x}, \mathbf{y})
\end{aligned}
$$

— No need to form the features. Calculations based on the kernel.

An instance of the kernel trick.

Kernel As A Tool for Dimension Augmentation

In regression spline, we did the following:

- From the given feature $x \in \mathbb{R}$, derive new features $x, x^2, (x - x_1)_+^3, \ldots, (x - x_n)_+^3$.
- Perform linear regression with the derived features.

In essence, we augment the feature dimension and use linear methods in the augmented space!

— A recurring theme in modern kernel methods.

Let $\mathcal{H}$ be a Hilbert space and $\phi : \mathbb{R}^d \to \mathcal{H}$ be a feature map.

$$\begin{aligned}\|\phi(\mathbf{x}) - \phi(\mathbf{y})\|_{\mathcal{H}}^2 &= \|\phi(\mathbf{x})\|_{\mathcal{H}}^2 + \|\phi(\mathbf{y})\|_{\mathcal{H}}^2 - 2\underbrace{\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}}_{K(\mathbf{x},\mathbf{y})} \\ &= K(\mathbf{x}, \mathbf{x}) + K(\mathbf{y}, \mathbf{y}) - 2K(\mathbf{x}, \mathbf{y})\end{aligned}$$

— No need to form the features. Calculations based on the kernel.

An instance of the kernel trick.

Why do this?

Kernel As A Tool for Dimension Augmentation

In regression spline, we did the following:

- From the given feature $x \in \mathbb{R}$, derive new features $x, x^2, (x - x_1)_+^3, \dots, (x - x_n)_+^3$.
- Perform linear regression with the derived features.

In essence, we augment the feature dimension and use linear methods in the augmented space!

— A recurring theme in modern kernel methods.

Let $\mathcal{H}$ be a Hilbert space and $\phi : \mathbb{R}^d \to \mathcal{H}$ be a feature map.

$$\begin{aligned}\|\phi(\mathbf{x}) - \phi(\mathbf{y})\|_{\mathcal{H}}^2 &= \|\phi(\mathbf{x})\|_{\mathcal{H}}^2 + \|\phi(\mathbf{y})\|_{\mathcal{H}}^2 - 2\,\underbrace{\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}}_{K(\mathbf{x},\mathbf{y})} \\ &= K(\mathbf{x}, \mathbf{x}) + K(\mathbf{y}, \mathbf{y}) - 2K(\mathbf{x}, \mathbf{y})\end{aligned}$$

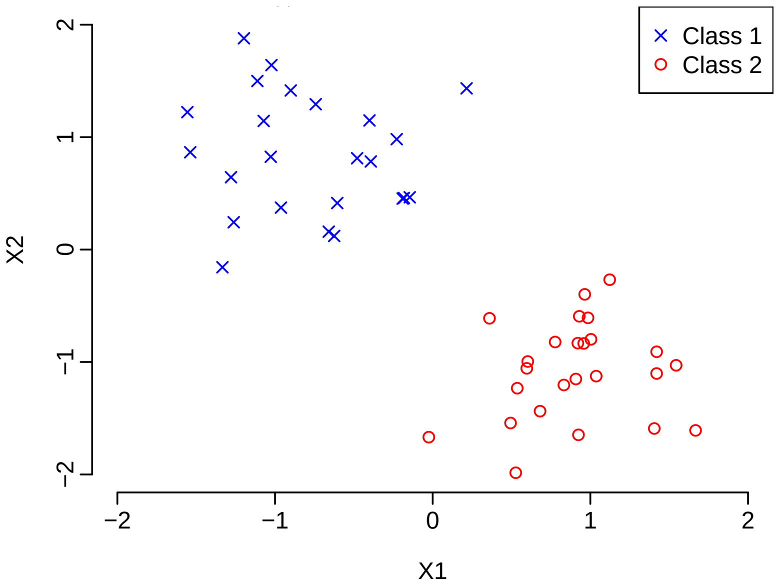
— No need to form the features. Calculations based on the kernel.

An instance of the kernel trick.

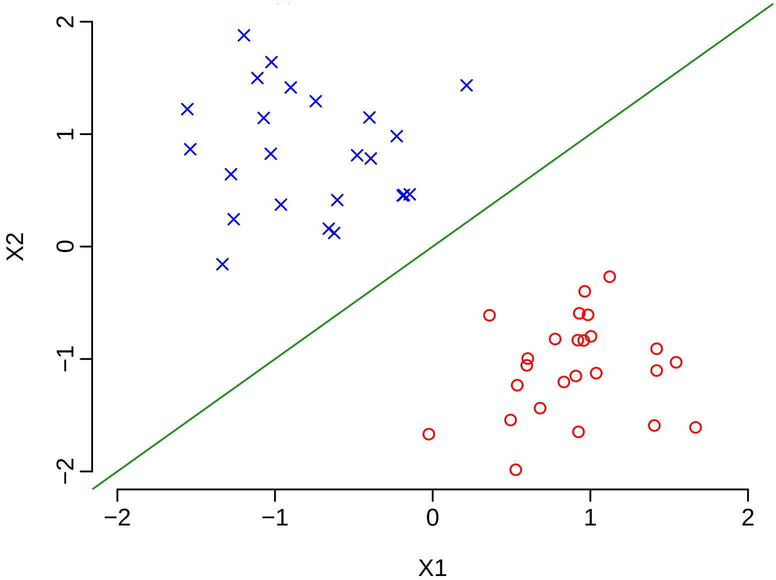
Why do this?

The problem may not be linear at the **x**-level ... but may be linear at a higher dimension!!

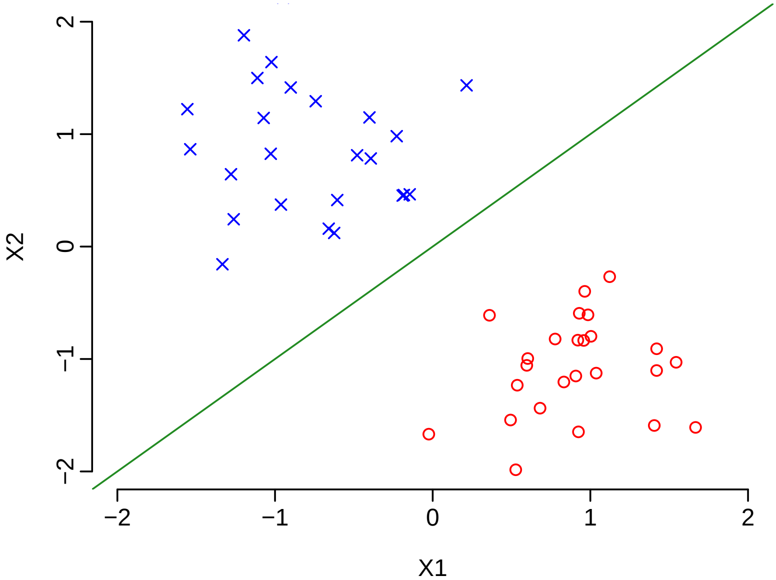
Classification



Classification

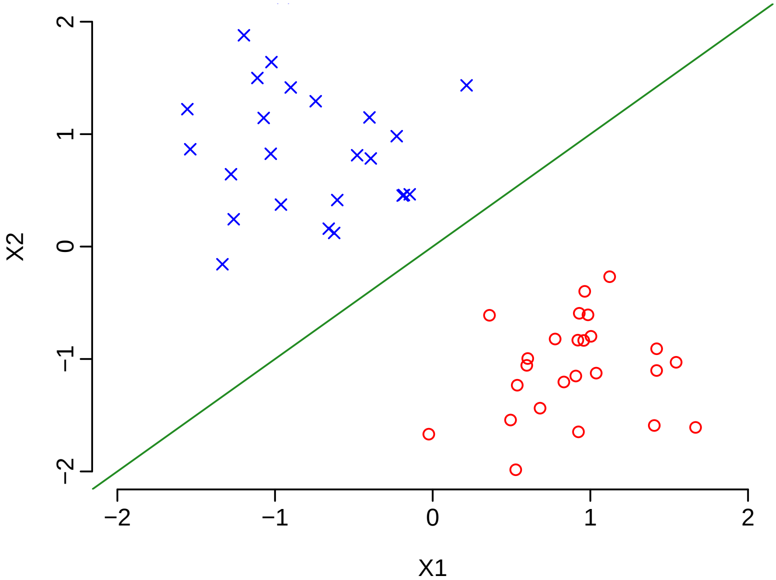


Classification



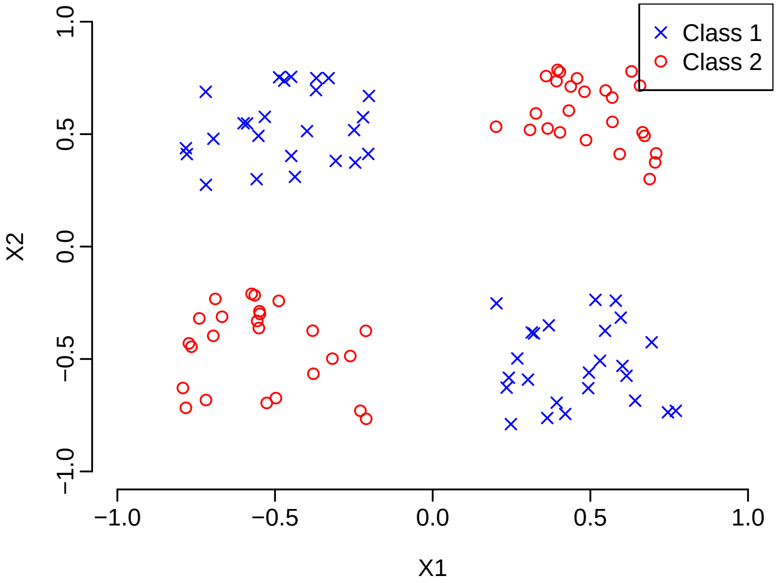
Linear classifier: $\delta(x_1, x_2) = \begin{cases} 1 & \text{if } \beta_1 x_1 + \beta_2 x_2 > \alpha \\ 2 & \text{otherwise} \end{cases}$

Classification

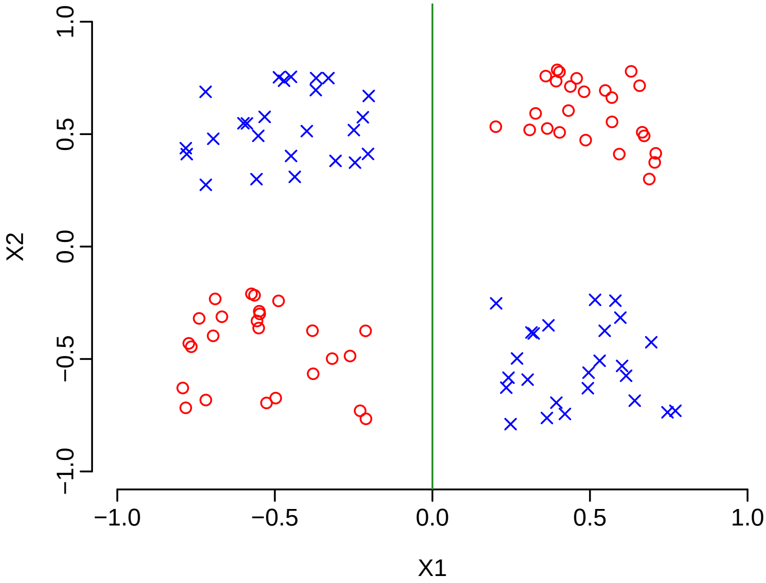


In general, Linear classifier: $\delta(\mathbf{x}) = \begin{cases} 1 & \text{if } \boldsymbol{\beta}^\top \mathbf{x} > \alpha \\ 2 & \text{otherwise} \end{cases}$

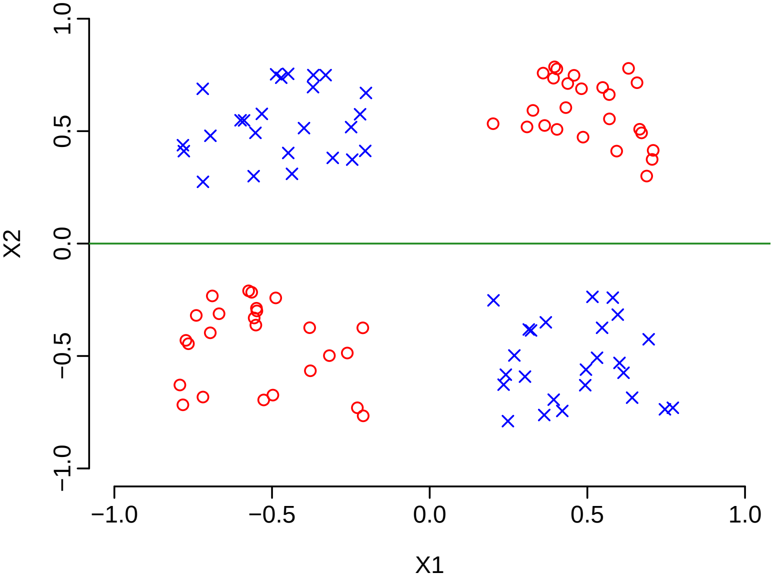
Another Example



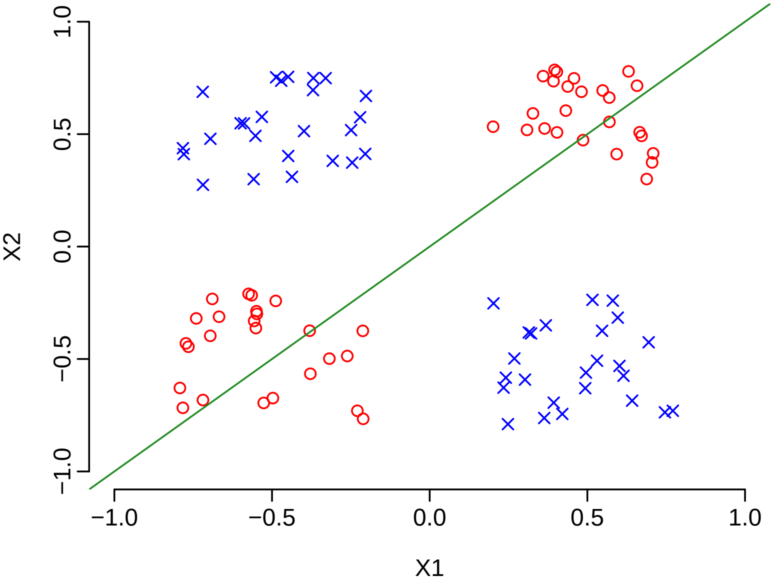
Another Example



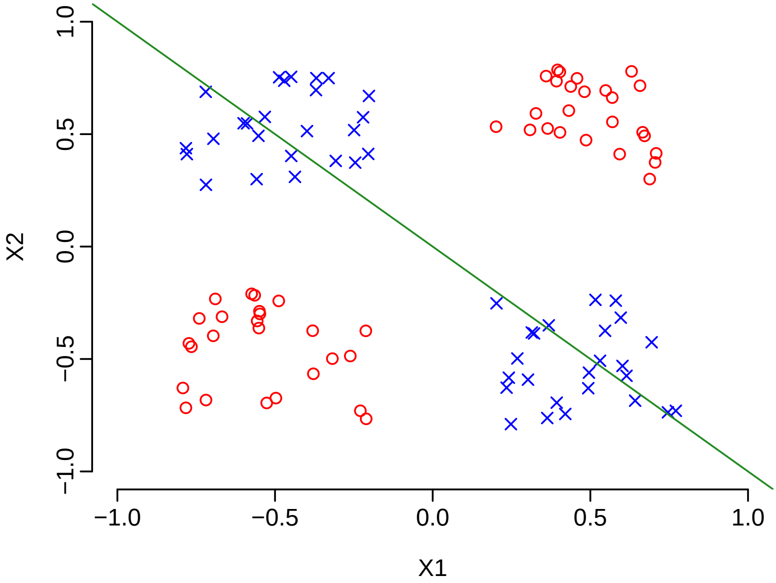
Another Example



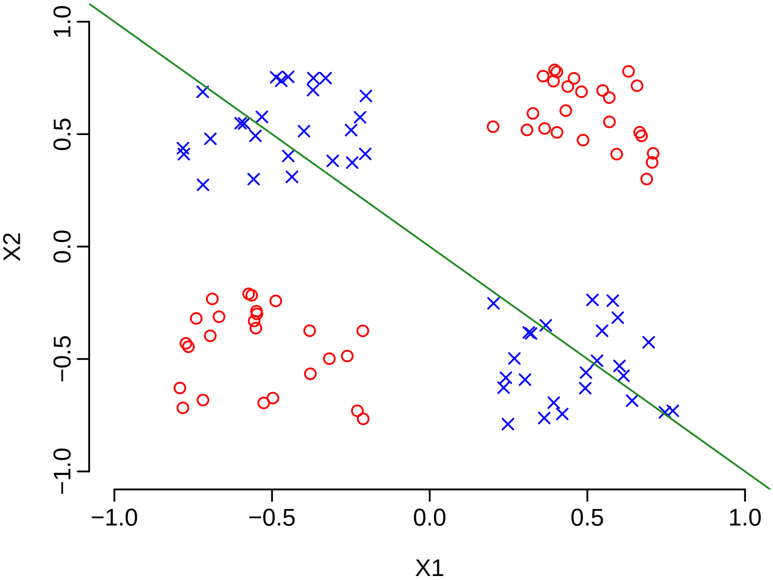
Another Example



Another Example

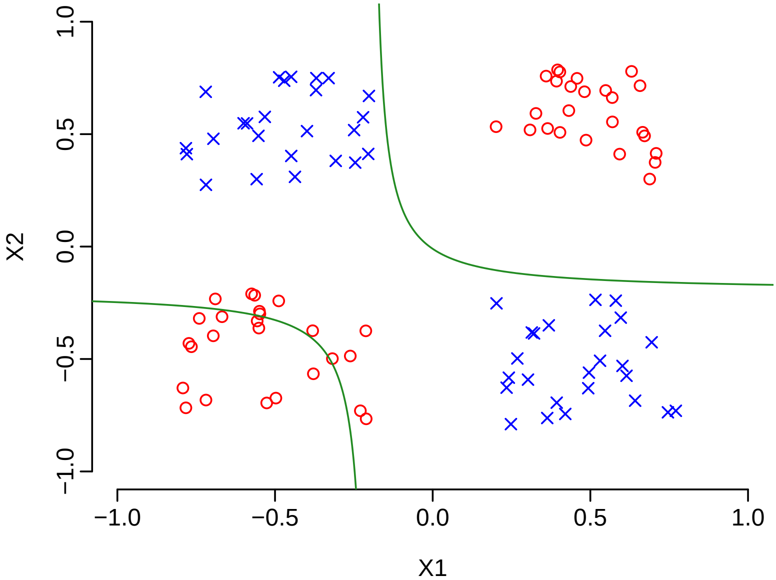


Another Example



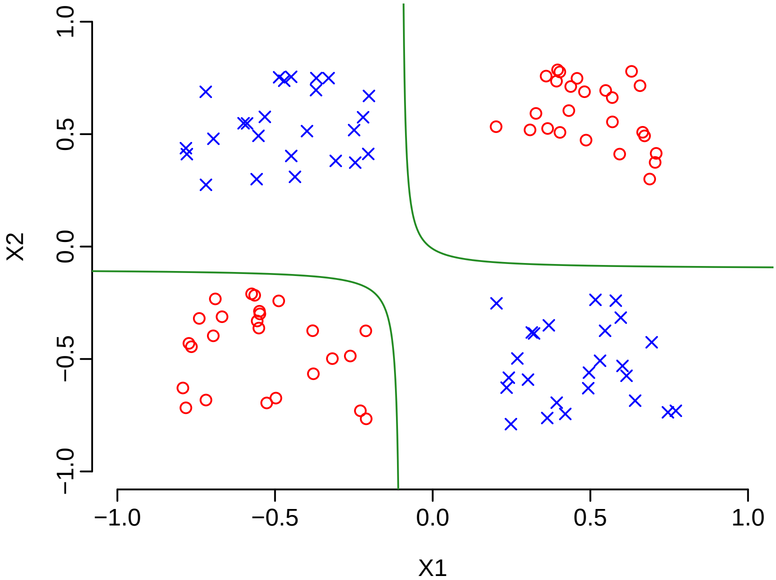
Linear classifier does not work in this case.

Another Example



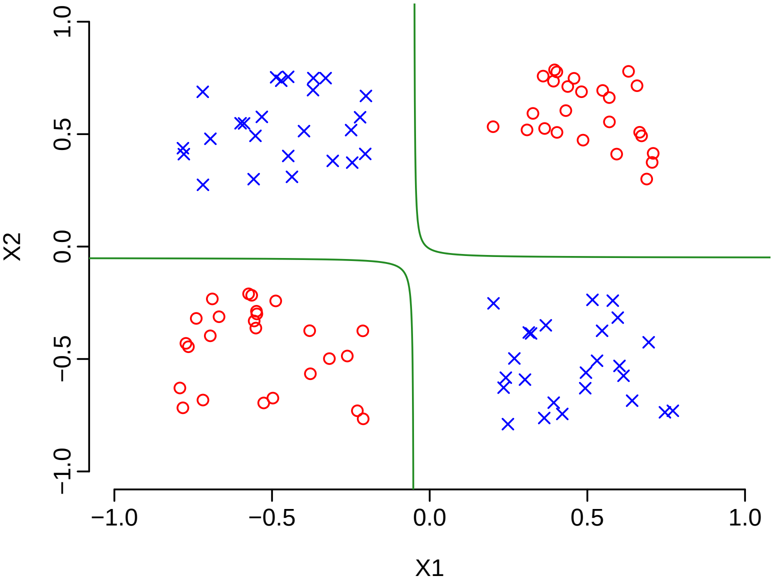
$$\delta(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 + x_2 + 5x_1x_2 > 0.01 \\ 2 & \text{otherwise} \end{cases}$$

Another Example



$$\delta(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 + x_2 + 10x_1x_2 > 0.01 \\ 2 & \text{otherwise} \end{cases}$$

Another Example



$$\delta(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 + x_2 + 20x_1x_2 > 0.01 \\ 2 & \text{otherwise} \end{cases}$$

The problem is not linearly solvable in (x_1, x_2).

The problem is not linearly solvable in (x_1, x_2).

Transform: $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mapsto \begin{pmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{pmatrix}$

The problem is not linearly solvable in (x_1, x_2).

Transform: $\underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{\mathbf{x}} \mapsto \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{pmatrix}}_{\phi(\mathbf{x})}$

The problem is not linearly solvable in (x_1, x_2).

Transform: $\underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{\mathbf{x}} \mapsto \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{pmatrix}}_{\phi(\mathbf{x})}$

The problem is linearly solvable in the transformed variable $\phi(\mathbf{x})$.

The problem is not linearly solvable in (x_1, x_2).

Transform: $\underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{\mathbf{x}} \mapsto \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{pmatrix}}_{\phi(\mathbf{x})}$

The problem is linearly solvable in the transformed variable $\phi(\mathbf{x})$.

But how does that help?

The problem is not linearly solvable in (x_1, x_2).

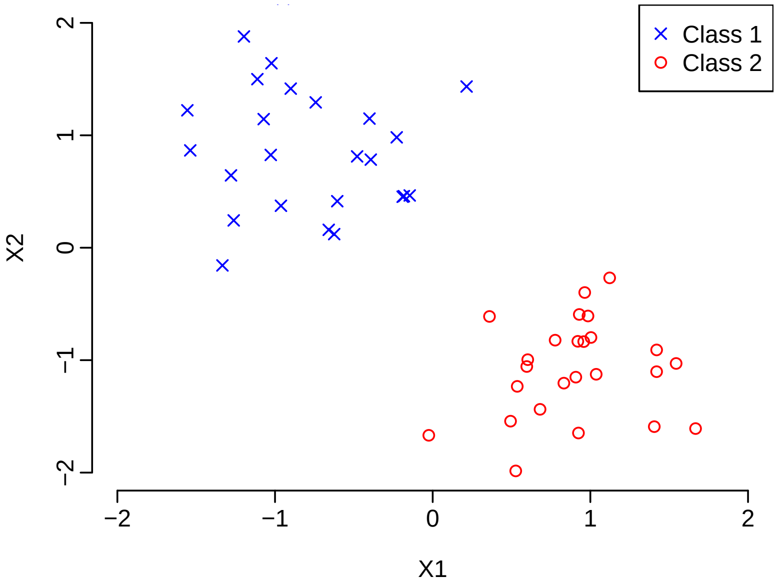
Transform: $\underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{\mathbf{x}} \mapsto \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{pmatrix}}_{\phi(\mathbf{x})}$

The problem is linearly solvable in the transformed variable $\phi(\mathbf{x})$.

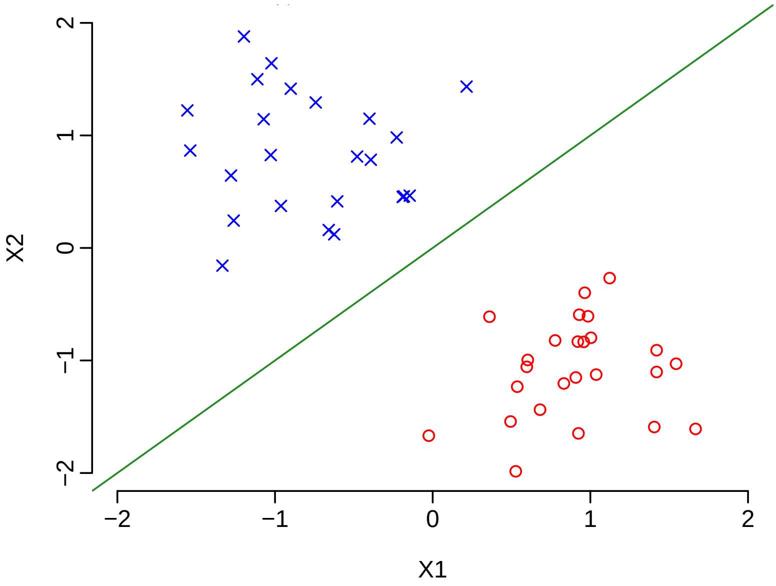
But how does that help?

For that, need to see how linear classifiers work.

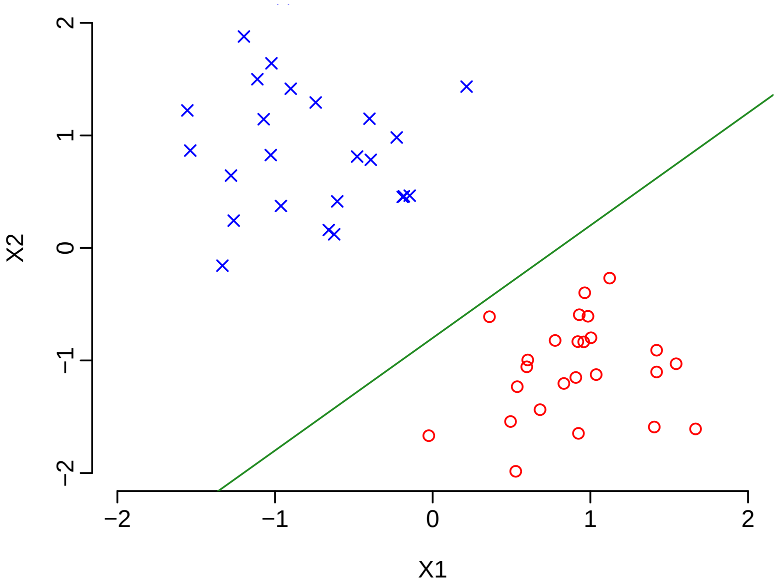
Back to the Linear Problem



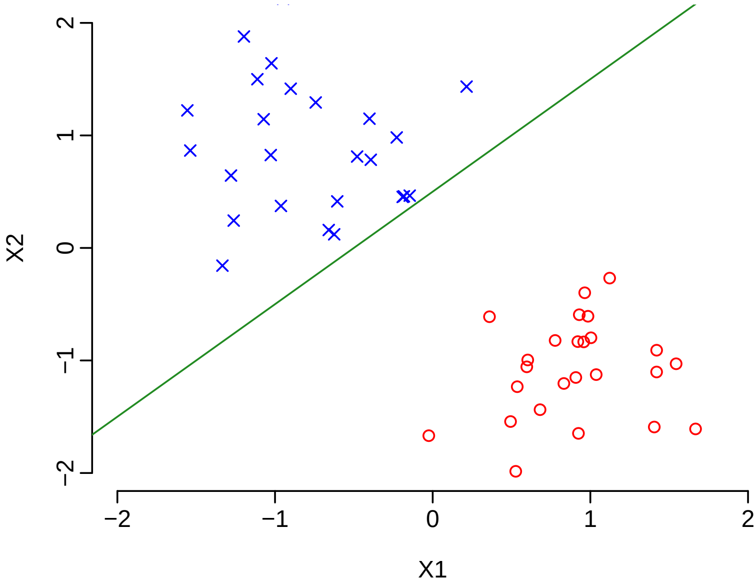
Back to the Linear Problem



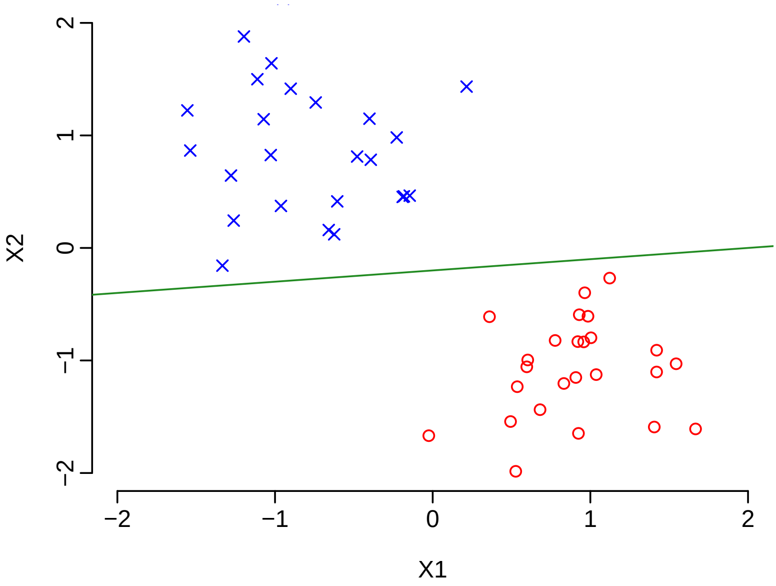
Back to the Linear Problem



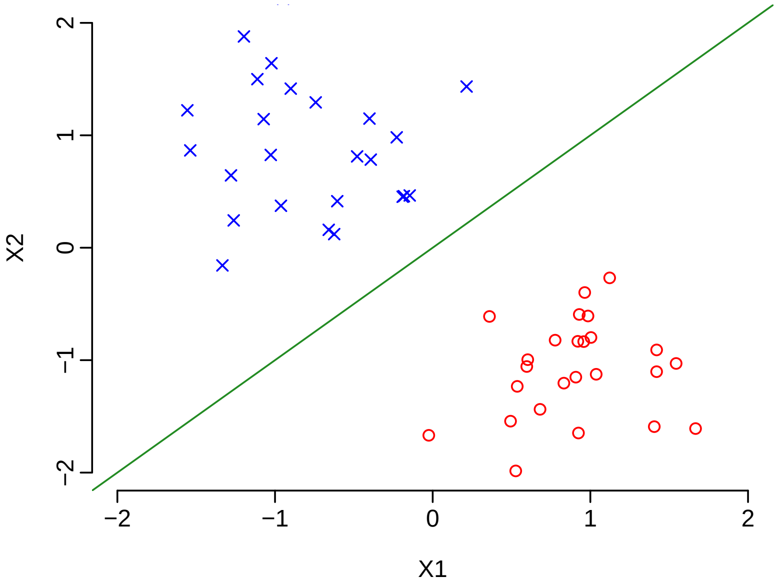
Back to the Linear Problem



Back to the Linear Problem

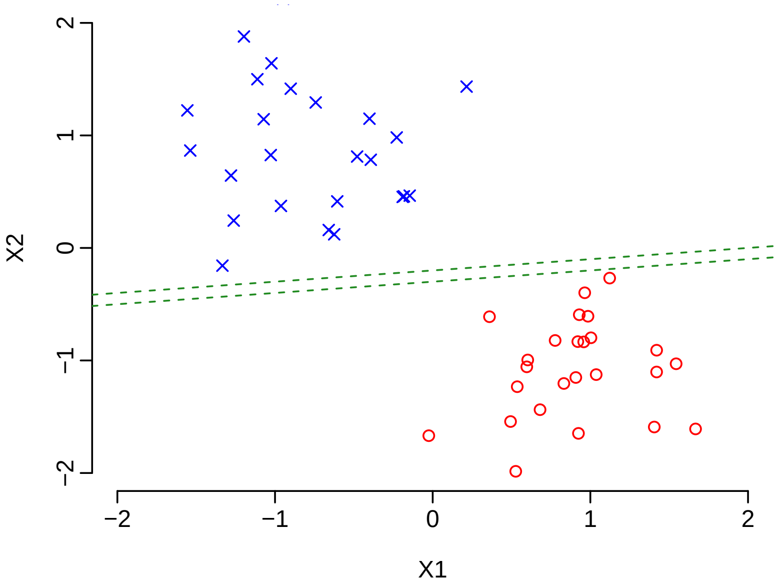


Back to the Linear Problem

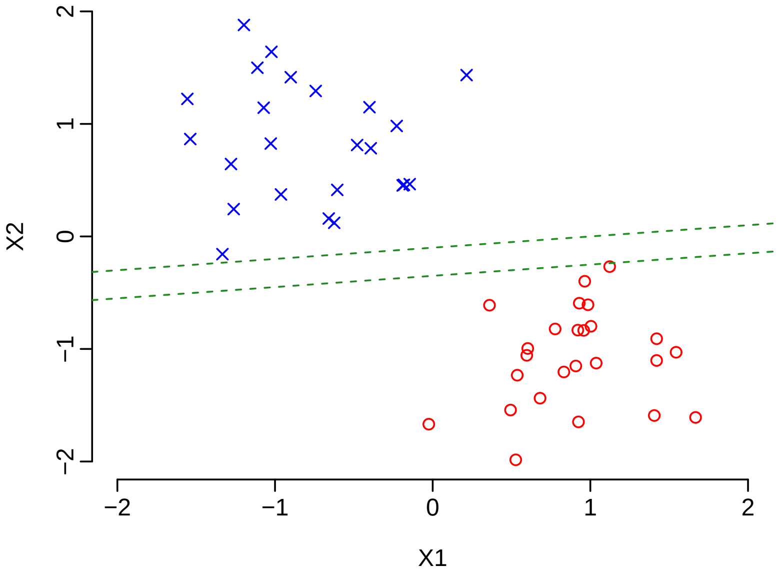


Margin

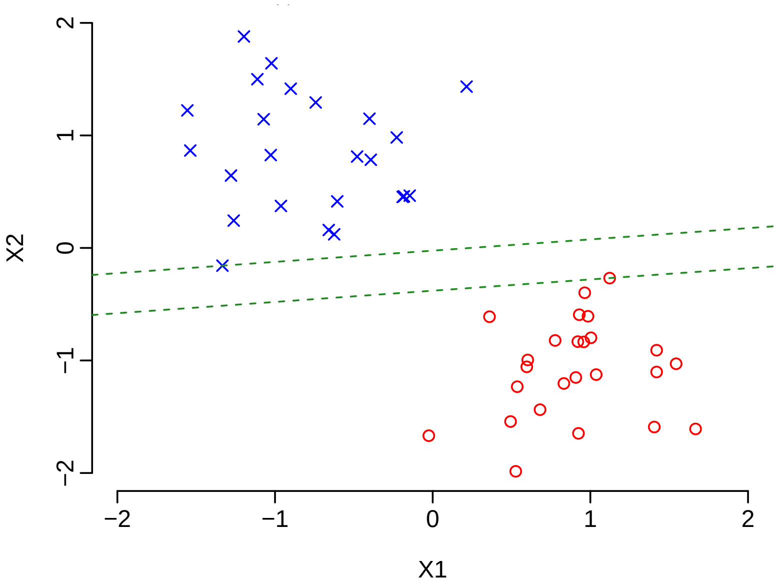
Margin



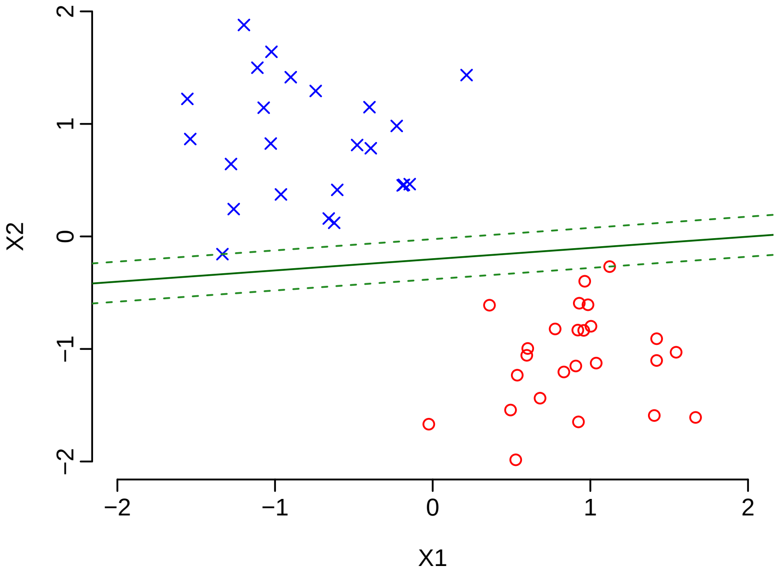
Margin



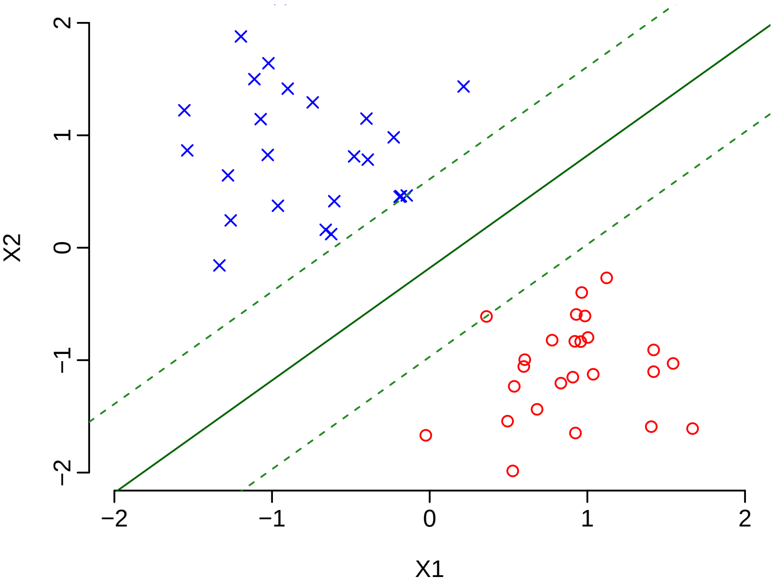
Margin



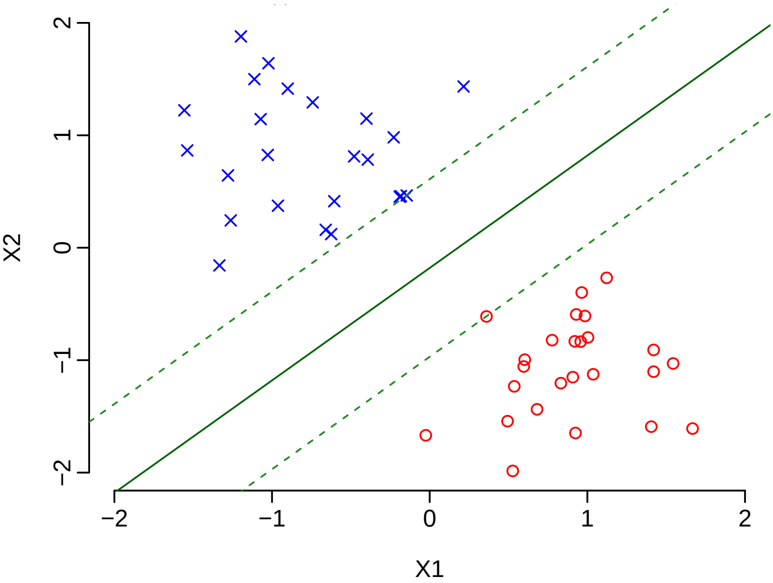
Margin



Margin



Margin



We prefer the direction which leads to the largest margin.

Mathematically ...

Let $y = \begin{cases} +1 & \text{if from Class 1} \\ -1 & \text{if from Class 2} \end{cases}$

Our classifier is: $\delta(\mathbf{x}) = \begin{cases} \text{Class 1} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 \\ \text{Class 2} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 \end{cases}$

Mathematically ...

Let $y = \begin{cases} +1 & \text{if from Class 1} \\ -1 & \text{if from Class 2} \end{cases}$

Our classifier is: $\delta(\mathbf{x}) = \begin{cases} \text{Class 1} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 \\ \text{Class 2} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 \end{cases}$

Correct decision if: $\begin{cases} \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 & \text{when } \mathbf{x} \text{ comes from Class 1} \\ \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 & \text{when } \mathbf{x} \text{ comes from Class 2} \end{cases}$

Mathematically ...

Let $y = \begin{cases} +1 & \text{if from Class 1} \\ -1 & \text{if from Class 2} \end{cases}$

Our classifier is: $\delta(\mathbf{x}) = \begin{cases} \text{Class 1} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 \\ \text{Class 2} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 \end{cases}$

Correct decision if: $\begin{cases} \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 & \text{when } y = +1 \\ \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 & \text{when } y = -1 \end{cases}$

Mathematically ...

Let $y = \begin{cases} +1 & \text{if from Class 1} \\ -1 & \text{if from Class 2} \end{cases}$

Our classifier is: $\delta(\mathbf{x}) = \begin{cases} \text{Class 1} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 \\ \text{Class 2} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 \end{cases}$

Correct decision if: $\begin{cases} \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 & \text{when } y = +1 \\ \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 & \text{when } y = -1 \end{cases} \quad \Leftrightarrow \quad y(\boldsymbol{\beta}^\top \mathbf{x} + \beta_0) > 0$

Mathematically ...

Let $y = \begin{cases} +1 & \text{if from Class 1} \\ -1 & \text{if from Class 2} \end{cases}$

Our classifier is: $\delta(\mathbf{x}) = \begin{cases} \text{Class 1} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 \\ \text{Class 2} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 \end{cases}$

Correct decision if: $\begin{cases} \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 & \text{when } y = +1 \\ \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 & \text{when } y = -1 \end{cases} \quad \Leftrightarrow \quad y(\boldsymbol{\beta}^\top \mathbf{x} + \beta_0) > 0$

Find $\boldsymbol{\beta}, \beta_0$ such that

$$y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) > 0 \; \forall\, i = 1, \ldots, n.$$

Mathematically ...

Let $y = \begin{cases} +1 & \text{if from Class 1} \\ -1 & \text{if from Class 2} \end{cases}$

Our classifier is: $\delta(\mathbf{x}) = \begin{cases} \text{Class 1} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 \\ \text{Class 2} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 \end{cases}$

Correct decision if: $\begin{cases} \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 & \text{when } y = +1 \\ \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 & \text{when } y = -1 \end{cases} \quad \Leftrightarrow \quad y(\boldsymbol{\beta}^\top \mathbf{x} + \beta_0) > 0$

Find $\boldsymbol{\beta}, \beta_0$ such that

$$y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) > 0 \;\forall\, i = 1, \ldots, n.$$

Too many possibilities!!

Mathematically ...

Let $y = \begin{cases} +1 & \text{if from Class 1} \\ -1 & \text{if from Class 2} \end{cases}$

Our classifier is: $\delta(\mathbf{x}) = \begin{cases} \text{Class 1} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 \\ \text{Class 2} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 \end{cases}$

Correct decision if: $\begin{cases} \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 & \text{when } y = +1 \\ \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 & \text{when } y = -1 \end{cases} \quad \Leftrightarrow \quad y(\boldsymbol{\beta}^\top \mathbf{x} + \beta_0) > 0$

Find $\boldsymbol{\beta}, \beta_0$ such that

$$y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) > 0 \;\forall\, i = 1, \ldots, n.$$

Too many possibilities!!

Make the gap as large as possible.

Mathematically ...

Let $y = \begin{cases} +1 & \text{if from Class 1} \\ -1 & \text{if from Class 2} \end{cases}$

Our classifier is: $\delta(\mathbf{x}) = \begin{cases} \text{Class 1} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 \\ \text{Class 2} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 \end{cases}$

Correct decision if: $\begin{cases} \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 & \text{when } y = +1 \\ \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 & \text{when } y = -1 \end{cases} \quad \Leftrightarrow \quad y(\boldsymbol{\beta}^\top \mathbf{x} + \beta_0) > 0$

Find $\boldsymbol{\beta}, \beta_0$ such that

$$y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) > 0 \; \forall \, i = 1, \ldots, n.$$

Too many possibilities!!

Impose $y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M$ and make M as large as possible.

Mathematically ...

Let $y = \begin{cases} +1 & \text{if from Class 1} \\ -1 & \text{if from Class 2} \end{cases}$

Our classifier is: $\delta(\mathbf{x}) = \begin{cases} \text{Class 1} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 \\ \text{Class 2} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 \end{cases}$

Correct decision if: $\begin{cases} \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 & \text{when } y = +1 \\ \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 & \text{when } y = -1 \end{cases} \quad \Leftrightarrow \quad y(\boldsymbol{\beta}^\top \mathbf{x} + \beta_0) > 0$

Find $\boldsymbol{\beta}, \beta_0$ such that

$$y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) > 0 \; \forall \, i = 1, \ldots, n.$$

Too many possibilities!!

Impose $y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M$ and make M as large as possible.

Find $\boldsymbol{\beta}, \beta_0$ such that

$$y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M \; \forall \, i = 1, \ldots, n \quad \text{and} \quad M \text{ is as large as possible.}$$

Mathematically ...

Let $y = \begin{cases} +1 & \text{if from Class 1} \\ -1 & \text{if from Class 2} \end{cases}$

Our classifier is: $\delta(\mathbf{x}) = \begin{cases} \text{Class 1} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 \\ \text{Class 2} & \text{if } \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 \end{cases}$

Correct decision if: $\begin{cases} \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 > 0 & \text{when } y = +1 \\ \boldsymbol{\beta}^\top \mathbf{x} + \beta_0 < 0 & \text{when } y = -1 \end{cases} \quad \Leftrightarrow \quad y(\boldsymbol{\beta}^\top \mathbf{x} + \beta_0) > 0$

Find $\boldsymbol{\beta}, \beta_0$ such that

$$y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) > 0 \ \forall\, i = 1, \ldots, n.$$

Too many possibilities!!

Impose $y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M$ and make M as large as possible.

Find $\boldsymbol{\beta}, \beta_0$ such that

$$y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M \ \forall\, i = 1, \ldots, n \quad \text{and} \quad \underbrace{M \text{ is as large as possible}}_{\text{maximizing the margin}}.$$

$$\underset{\boldsymbol{\beta},\beta_0,\|\boldsymbol{\beta}\|=1}{\text{maximize}} \; M \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M \; \forall \, i = 1, \ldots, n$$

$$
\begin{aligned}
&\underset{\boldsymbol{\beta},\beta_0,\|\boldsymbol{\beta}\|=1}{\text{maximize}} \;\; M \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M \;\forall\, i = 1, \ldots, n \\
\Leftrightarrow \quad &\underset{\boldsymbol{\beta},\beta_0}{\text{maximize}} \;\; M \quad \text{s.t.} \quad \frac{1}{\|\boldsymbol{\beta}\|} y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M \;\forall\, i = 1, \ldots, n
\end{aligned}
$$

$$\underset{\boldsymbol{\beta},\beta_0,\|\boldsymbol{\beta}\|=1}{\text{maximize}} \; M \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M \; \forall\, i = 1, \ldots, n$$

$$\Leftrightarrow \quad \underset{\boldsymbol{\beta},\beta_0}{\text{maximize}} \; M \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M\|\boldsymbol{\beta}\| \; \forall\, i = 1, \ldots, n$$

$$
\begin{aligned}
& \underset{\boldsymbol{\beta},\beta_0,\|\boldsymbol{\beta}\|=1}{\text{maximize}} \; M \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M \;\forall\, i = 1,\dots,n \\
\Leftrightarrow \quad & \underset{\boldsymbol{\beta},\beta_0}{\text{maximize}} \; M \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M\|\boldsymbol{\beta}\| \;\forall\, i = 1,\dots,n \\
\Leftrightarrow \quad & \underset{\boldsymbol{\beta},\beta_0}{\text{maximize}} \; \frac{1}{\|\boldsymbol{\beta}\|} \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \;\forall\, i = 1,\dots,n \qquad \big(\text{rescale by setting } M\|\beta\| = 1\big)
\end{aligned}
$$

$$
\begin{aligned}
& \underset{\boldsymbol{\beta},\beta_0,\|\boldsymbol{\beta}\|=1}{\text{maximize}} \;\; M \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M \; \forall\, i = 1, \ldots, n \\
\Leftrightarrow \quad & \underset{\boldsymbol{\beta},\beta_0}{\text{maximize}} \;\; M \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M\|\boldsymbol{\beta}\| \; \forall\, i = 1, \ldots, n \\
\Leftrightarrow \quad & \underset{\boldsymbol{\beta},\beta_0}{\text{maximize}} \;\; \frac{1}{\|\boldsymbol{\beta}\|} \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \; \forall\, i = 1, \ldots, n \qquad \big(\text{rescale by setting } M\|\beta\| = 1\big) \\
\Leftrightarrow \quad & \underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \;\; \|\boldsymbol{\beta}\| \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \; \forall\, i = 1, \ldots, n
\end{aligned}
$$

$$
\begin{aligned}
& \underset{\boldsymbol{\beta},\beta_0,\|\boldsymbol{\beta}\|=1}{\text{maximize}} \;\; M \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M \; \forall\, i = 1,\ldots,n \\
\Leftrightarrow \;\; & \underset{\boldsymbol{\beta},\beta_0}{\text{maximize}} \;\; M \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq M\|\boldsymbol{\beta}\| \; \forall\, i = 1,\ldots,n \\
\Leftrightarrow \;\; & \underset{\boldsymbol{\beta},\beta_0}{\text{maximize}} \;\; \frac{1}{\|\boldsymbol{\beta}\|} \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \; \forall\, i = 1,\ldots,n \qquad \big(\text{rescale by setting } M\|\beta\| = 1\big) \\
\Leftrightarrow \;\; & \underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \;\; \|\boldsymbol{\beta}\| \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \; \forall\, i = 1,\ldots,n \\
\Leftrightarrow \;\; & \underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \;\; \frac{1}{2}\|\boldsymbol{\beta}\|^2 \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \; \forall\, i = 1,\ldots,n
\end{aligned}
$$

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \;\; \frac{1}{2}\|\boldsymbol{\beta}\|^2 \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \;\forall\, i = 1, \ldots, n.$$

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \ \forall\, i = 1, \ldots, n.$$

Primal:

$$L_P(\boldsymbol{\beta}, \beta_0) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_{i=1}^{n} \lambda_i \{y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - 1\}.$$

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \;\forall\, i = 1, \ldots, n.$$

Primal:

$$L_P(\boldsymbol{\beta}, \beta_0) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_{i=1}^{n} \lambda_i \{y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - 1\}.$$

Set the derivatives to zero:

$$\frac{\partial L_P}{\partial \boldsymbol{\beta}} = 0 \Rightarrow \boldsymbol{\beta} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i, \qquad \frac{\partial L_P}{\partial \beta_0} = 0 \Rightarrow \sum_{i=1}^{n} \lambda_i y_i = 0.$$

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \ \forall \, i = 1, \ldots, n.$$

Primal:

$$L_P(\boldsymbol{\beta}, \beta_0) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_{i=1}^{n} \lambda_i \{ y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - 1 \}.$$

Set the derivatives to zero:

$$\frac{\partial L_P}{\partial \boldsymbol{\beta}} = 0 \Rightarrow \boldsymbol{\beta} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i, \qquad \frac{\partial L_P}{\partial \beta_0} = 0 \Rightarrow \sum_{i=1}^{n} \lambda_i y_i = 0.$$

Substitute to get the dual:

$$L_D(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \quad \text{with} \quad \lambda_i \geq 0, \sum_{i=1}^{n} \lambda_i y_i = 0$$

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \;\forall\, i = 1, \ldots, n.$$

Primal:

$$L_P(\boldsymbol{\beta}, \beta_0) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_{i=1}^n \lambda_i \big\{ y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - 1 \big\}.$$

Set the derivatives to zero:

$$\frac{\partial L_P}{\partial \boldsymbol{\beta}} = 0 \Rightarrow \boldsymbol{\beta} = \sum_{i=1}^n \lambda_i y_i \mathbf{x}_i, \qquad \frac{\partial L_P}{\partial \beta_0} = 0 \Rightarrow \sum_{i=1}^n \lambda_i y_i = 0.$$

Substitute to get the dual:

$$L_D(\boldsymbol{\lambda}) = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \quad \text{with} \quad \lambda_i \geq 0, \sum_{i=1}^n \lambda_i y_i = 0$$

$$\text{and the KKT condition } \lambda_i \big\{ y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - 1 \big\} = 0 \;\forall i$$

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{\beta}\|^2 \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \ \forall\, i = 1, \ldots, n.$$

Primal:

$$L_P(\boldsymbol{\beta}, \beta_0) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_{i=1}^{n} \lambda_i \big\{ y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - 1 \big\}.$$

Set the derivatives to zero:

$$\frac{\partial L_P}{\partial \boldsymbol{\beta}} = 0 \Rightarrow \boldsymbol{\beta} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i, \qquad \frac{\partial L_P}{\partial \beta_0} = 0 \Rightarrow \sum_{i=1}^{n} \lambda_i y_i = 0.$$

Substitute to get the dual:

$$L_D(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \quad \text{with} \quad \lambda_i \geq 0, \sum_{i=1}^{n} \lambda_i y_i = 0$$

$$\text{and the KKT condition } \lambda_i \big\{ y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - 1 \big\} = 0 \, \forall i$$

- $\boldsymbol{\beta} = \sum_{i \in \mathcal{S}} \lambda_i y_i \mathbf{x}_i, \quad \mathcal{S} = \{ i : y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) = 1 \}$ — the support vectors.

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \;\forall\, i = 1, \ldots, n.$$

Primal:

$$L_P(\boldsymbol{\beta}, \beta_0) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_{i=1}^{n} \lambda_i \big\{ y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - 1 \big\}.$$

Set the derivatives to zero:

$$\frac{\partial L_P}{\partial \boldsymbol{\beta}} = 0 \Rightarrow \boldsymbol{\beta} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i, \qquad \frac{\partial L_P}{\partial \beta_0} = 0 \Rightarrow \sum_{i=1}^{n} \lambda_i y_i = 0.$$

Substitute to get the dual:

$$L_D(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \quad \text{with} \quad \lambda_i \geq 0, \sum_{i=1}^{n} \lambda_i y_i = 0$$

and the KKT condition $\lambda_i \big\{ y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - 1 \big\} = 0 \,\forall i$

- $\boldsymbol{\beta} = \sum_{i \in \mathcal{S}} \lambda_i y_i \mathbf{x}_i, \quad \mathcal{S} = \{ i : y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) = 1 \}$ — the support vectors.
- $\boldsymbol{\beta}^\top \mathbf{x} = \sum_{i \in \mathcal{S}} \lambda_i y_i \mathbf{x}_i^\top \mathbf{x}$

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \ \forall\, i = 1, \ldots, n.$$

Primal:

$$L_P(\boldsymbol{\beta}, \beta_0) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_{i=1}^{n} \lambda_i \big\{ y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - 1 \big\}.$$

Set the derivatives to zero:

$$\frac{\partial L_P}{\partial \boldsymbol{\beta}} = 0 \Rightarrow \boldsymbol{\beta} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i, \qquad \frac{\partial L_P}{\partial \beta_0} = 0 \Rightarrow \sum_{i=1}^{n} \lambda_i y_i = 0.$$

Substitute to get the dual:

$$L_D(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \quad \text{with} \quad \lambda_i \geq 0, \sum_{i=1}^{n} \lambda_i y_i = 0$$

$$\text{and the KKT condition } \lambda_i \big\{ y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - 1 \big\} = 0 \,\forall i$$

- $\boldsymbol{\beta} = \sum_{i \in \mathcal{S}} \lambda_i y_i \mathbf{x}_i, \quad \mathcal{S} = \{ i : y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) = 1 \}$ — the support vectors.
- $\boldsymbol{\beta}^\top \mathbf{x} = \sum_{i \in \mathcal{S}} \lambda_i y_i \mathbf{x}_i^\top \mathbf{x}$
- The method depends on the feature $\mathbf{x}$ only via inner-products

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) \geq 1 \; \forall\, i = 1, \ldots, n.$$

Primal:

$$L_P(\boldsymbol{\beta}, \beta_0) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_{i=1}^{n} \lambda_i \big\{ y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - 1 \big\}.$$

Set the derivatives to zero:

$$\frac{\partial L_P}{\partial \boldsymbol{\beta}} = 0 \Rightarrow \boldsymbol{\beta} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i, \qquad \frac{\partial L_P}{\partial \beta_0} = 0 \Rightarrow \sum_{i=1}^{n} \lambda_i y_i = 0.$$

Substitute to get the dual:

$$L_D(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \quad \text{with} \quad \lambda_i \geq 0, \sum_{i=1}^{n} \lambda_i y_i = 0$$

$$\text{and the KKT condition } \lambda_i \big\{ y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - 1 \big\} = 0 \, \forall i$$

- $\boldsymbol{\beta} = \sum_{i \in \mathcal{S}} \lambda_i y_i \mathbf{x}_i, \quad \mathcal{S} = \{ i : y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) = 1 \}$ — the support vectors.
- $\boldsymbol{\beta}^\top \mathbf{x} = \sum_{i \in \mathcal{S}} \lambda_i y_i \mathbf{x}_i^\top \mathbf{x}$
- The method depends on the feature $\mathbf{x}$ only via inner-products

— Optimal separating hyperplane. Vapnik & Chervonenkis (1974)

Transform: $\mathbf{x} \to \phi(\mathbf{x})$

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{\beta}\|^2 \quad \text{s.t.} \quad y_i\big(\langle\boldsymbol{\beta}, \phi(\mathbf{x}_i)\rangle + \beta_0\big) \geq 1 \ \forall\, i = 1, \ldots, n.$$

Primal:

$$L_P(\boldsymbol{\beta}, \beta_0) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_{i=1}^{n} \lambda_i\big\{y_i\big(\langle\boldsymbol{\beta}, \phi(\mathbf{x}_i)\rangle + \beta_0\big) - 1\big\}.$$

Set the derivatives to zero:

$$\frac{\partial L_P}{\partial \boldsymbol{\beta}} = 0 \Rightarrow \boldsymbol{\beta} = \sum_{i=1}^{n} \lambda_i y_i \phi(\mathbf{x}_i), \qquad \frac{\partial L_P}{\partial \beta_0} = 0 \Rightarrow \sum_{i=1}^{n} \lambda_i y_i = 0.$$

Substitute to get the dual:

$$L_D(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i\lambda_j y_i y_j \underbrace{\langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle}_{K(\mathbf{x}_i,\mathbf{x}_j)} \quad \text{with} \quad \lambda_i \geq 0, \sum_{i=1}^{n} \lambda_i y_i = 0$$

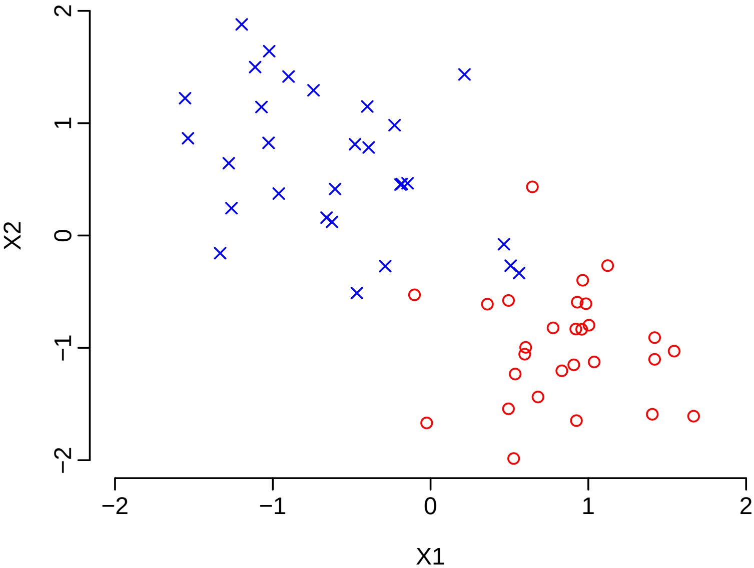
and the KKT condition $\lambda_i\big\{y_i\big(\langle\boldsymbol{\beta}, \phi(\mathbf{x}_i)\rangle + \beta_0\big) - 1\big\} = 0 \,\forall i$

- $\boldsymbol{\beta} = \sum_{i\in\mathcal{S}} \lambda_i y_i \phi(\mathbf{x}_i), \quad \mathcal{S} = \big\{i : y_i\big(\langle\boldsymbol{\beta}, \phi(\mathbf{x}_i)\rangle + \beta_0\big) = 1\big\}$ — the support vectors.
- $\langle\boldsymbol{\beta}, \phi(\mathbf{x})\rangle = \sum_{i\in\mathcal{S}} \lambda_i y_i \underbrace{\langle\phi(\mathbf{x}_i), \phi(\mathbf{x})\rangle}_{K(\mathbf{x}_i,\mathbf{x})}$
- The method depends on the feature $\mathbf{x}$ only via the kernel K.

— Boser, Guyon & Vapnik (1992)

In general, problems are not exactly solvable.

In general, problems are not exactly solvable.



In general, problems are not exactly solvable. A slightly different formulation is used:

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \ \ \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}\zeta_i \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x} + \beta_0) \geq 1 - \zeta_i \ \forall\, i = 1, \ldots, n, \quad \zeta_i \geq 0.$$

In general, problems are not exactly solvable. A slightly different formulation is used:

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \;\; \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}\zeta_i \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x} + \beta_0) \geq 1 - \zeta_i \;\forall\, i = 1,\ldots,n, \quad \zeta_i \geq 0.$$

Leads to a similar solution: $\displaystyle \boldsymbol{\beta} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i$

$\lambda_1,\ldots,\lambda_n$ obtained by maximizing $\displaystyle L_D = \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle$

s.t. $0 \leq \lambda_i \leq C, \quad \sum_{i=1}^{n}\lambda_i y_i = 0, \quad \lambda_i\big\{y_i(\boldsymbol{\beta}^\top\mathbf{x}_i + \beta_0) - (1-\zeta_i)\big\} = 0$

— Support Vector Machine. Cortes & Vapnik (1995)

In general, problems are not exactly solvable. A slightly different formulation is used:

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}\zeta_i \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x} + \beta_0) \geq 1 - \zeta_i \ \forall\, i = 1, \ldots, n, \quad \zeta_i \geq 0.$$

Leads to a similar solution: $\boldsymbol{\beta} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i$

$\lambda_1, \ldots, \lambda_n$ obtained by maximizing $L_D = \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$

s.t. $0 \leq \lambda_i \leq C, \quad \sum_{i=1}^{n} \lambda_i y_i = 0, \quad \lambda_i\big\{y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - (1 - \zeta_i)\big\} = 0$

— Support Vector Machine. Cortes & Vapnik (1995)

The Kernel Trick:

1. Transform the feature: $\mathbf{x} \to \phi(\mathbf{x})$
2. Identify a linear method that depends on the feature only via inner-products $\mathbf{x}^\top \widetilde{\mathbf{x}}$.
3. In the transformed feature $\phi(\mathbf{x})$, the method depends on inner-products $\langle \phi(\mathbf{x}), \phi(\widetilde{\mathbf{x}}) \rangle$.
   Use kernel to compute this inner-product $\langle \phi(\mathbf{x}), \phi(\widetilde{\mathbf{x}}) \rangle = K(\mathbf{x}, \widetilde{\mathbf{x}})$.

In general, problems are not exactly solvable. A slightly different formulation is used:

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \;\; \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}\zeta_i \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x} + \beta_0) \geq 1 - \zeta_i \; \forall\, i = 1,\ldots,n, \quad \zeta_i \geq 0.$$

Leads to a similar solution: $\boldsymbol{\beta} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i$

$$\lambda_1,\ldots,\lambda_n \text{ obtained by maximizing } L_D = \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle$$

$$\text{s.t.} \quad 0 \leq \lambda_i \leq C, \quad \textstyle\sum_{i=1}^{n}\lambda_i y_i = 0, \quad \lambda_i\big\{y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - (1-\zeta_i)\big\} = 0$$

— Support Vector Machine. Cortes & Vapnik (1995)

The Kernel Trick:

1. Transform the feature: $\mathbf{x} \to \phi(\mathbf{x})$
2. Identify a linear method that depends on the feature only via inner-products $\mathbf{x}^\top\widetilde{\mathbf{x}}$.
3. In the transformed feature $\phi(\mathbf{x})$, the method depends on inner-products $\langle\phi(\mathbf{x}), \phi(\widetilde{\mathbf{x}})\rangle$.
   Use kernel to compute this inner-product $\langle\phi(\mathbf{x}), \phi(\widetilde{\mathbf{x}})\rangle = K(\mathbf{x}, \widetilde{\mathbf{x}})$.

Linear method in $\phi(\mathbf{x})$-space $\quad\Rightarrow\quad$ Non-linear structure in the $\mathbf{x}$-space

In general, problems are not exactly solvable. A slightly different formulation is used:

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \;\; \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}\zeta_i \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x} + \beta_0) \geq 1 - \zeta_i \;\forall\, i = 1, \ldots, n, \quad \zeta_i \geq 0.$$

Leads to a similar solution: $\quad \boldsymbol{\beta} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i$

$$\lambda_1, \ldots, \lambda_n \text{ obtained by maximizing } L_D = \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle$$

$$\text{s.t.} \quad 0 \leq \lambda_i \leq C, \quad \textstyle\sum_{i=1}^{n}\lambda_i y_i = 0, \quad \lambda_i\big\{y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - (1 - \zeta_i)\big\} = 0$$

— Support Vector Machine. Cortes & Vapnik (1995)

The Kernel Trick:

1. Transform the feature: $\mathbf{x} \to \phi(\mathbf{x})$
2. Identify a linear method that depends on the feature only via inner-products $\mathbf{x}^\top\widetilde{\mathbf{x}}$.
3. In the transformed feature $\phi(\mathbf{x})$, the method depends on inner-products $\langle\phi(\mathbf{x}), \phi(\widetilde{\mathbf{x}})\rangle$.
   Use kernel to compute this inner-product $\langle\phi(\mathbf{x}), \phi(\widetilde{\mathbf{x}})\rangle = K(\mathbf{x}, \widetilde{\mathbf{x}})$.

Linear method in $\phi(\mathbf{x})$-space $\quad\Rightarrow\quad$ Non-linear structure in the $\mathbf{x}$-space

- The feature map can be arbitrary/non-unique. Even infinite dimensional!! But, we don't care!

  As long as we have a valid (and appropriate!) kernel, we are good.

In general, problems are not exactly solvable. A slightly different formulation is used:

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimize}} \;\; \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}\zeta_i \quad \text{s.t.} \quad y_i(\boldsymbol{\beta}^\top \mathbf{x} + \beta_0) \geq 1 - \zeta_i \;\forall\, i = 1,\ldots,n, \quad \zeta_i \geq 0.$$

Leads to a similar solution: $\quad \boldsymbol{\beta} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i$

$$\lambda_1,\ldots,\lambda_n \text{ obtained by maximizing } L_D = \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle$$

$$\text{s.t.} \quad 0 \leq \lambda_i \leq C, \quad \textstyle\sum_{i=1}^{n}\lambda_i y_i = 0, \quad \lambda_i\big\{y_i(\boldsymbol{\beta}^\top \mathbf{x}_i + \beta_0) - (1-\zeta_i)\big\} = 0$$

— Support Vector Machine. Cortes & Vapnik (1995)

The Kernel Trick:

1. Transform the feature: $\mathbf{x} \to \phi(\mathbf{x})$
2. Identify a linear method that depends on the feature only via inner-products $\mathbf{x}^\top\widetilde{\mathbf{x}}$.
3. In the transformed feature $\phi(\mathbf{x})$, the method depends on inner-products $\langle\phi(\mathbf{x}),\phi(\widetilde{\mathbf{x}})\rangle$.
   Use kernel to compute this inner-product $\langle\phi(\mathbf{x}),\phi(\widetilde{\mathbf{x}})\rangle = K(\mathbf{x},\widetilde{\mathbf{x}})$.

Linear method in $\phi(\mathbf{x})$-space $\quad\Rightarrow\quad$ Non-linear structure in the $\mathbf{x}$-space

- The feature map can be arbitrary/non-unique. Even infinite dimensional!! But, we don't care!

  As long as we have a valid (and appropriate!) kernel, we are good.

  E.g. SVM with Gaussian RBF: $\exp\left(-\dfrac{\|\mathbf{x}-\widetilde{\mathbf{x}}\|^2}{\sigma^2}\right)$ has been very successful.

Kernel PCA

$\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$. Assumed centered, $\sum_{i=1}^n \mathbf{x}_i = 0$ (W.l.o.g.!).

Kernel PCA

$\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$. Assumed centered, $\sum_{i=1}^{n} \mathbf{x}_i = 0$ (W.l.o.g.!).

Usual PCA: Find directions with largest variation.

Kernel PCA

$\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$. Assumed centered, $\sum_{i=1}^n \mathbf{x}_i = 0$ (W.l.o.g.!).

Usual PCA: Find directions with largest variation.

$$\boldsymbol{\beta}_1 = \underset{\|\boldsymbol{\beta}\|=1}{\arg\max} \frac{1}{n} \sum_{i=1}^{n} \big(\langle \boldsymbol{\beta}, \mathbf{x}_i \rangle\big)^2 = \underset{\|\boldsymbol{\beta}\|=1}{\arg\max} \quad \widehat{\mathrm{var}}\big(\langle \boldsymbol{\beta}, \mathbf{X} \rangle\big)$$

Kernel PCA

$\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$. Assumed centered, $\sum_{i=1}^n \mathbf{x}_i = 0$ (W.l.o.g.!).

Usual PCA: Find directions with largest variation.

$$\boldsymbol{\beta}_1 = \arg\max_{\|\boldsymbol{\beta}\|=1} \frac{1}{n} \sum_{i=1}^{n} \big(\langle \boldsymbol{\beta}, \mathbf{x}_i \rangle\big)^2 = \arg\max_{\|\boldsymbol{\beta}\|=1} \quad \widehat{\mathrm{var}}\big(\langle \boldsymbol{\beta}, \mathbf{X} \rangle\big)$$

$$\boldsymbol{\beta}_2 = \arg\max_{\|\boldsymbol{\beta}\|=1, \boldsymbol{\beta}^\top \boldsymbol{\beta}_1 = 0} \frac{1}{n} \sum_{i=1}^{n} \big(\langle \boldsymbol{\beta}, \mathbf{x}_i \rangle\big)^2 = \arg\max_{\|\boldsymbol{\beta}\|=1, \boldsymbol{\beta}^\top \boldsymbol{\beta}_1 = 0} \quad \widehat{\mathrm{var}}\big(\langle \boldsymbol{\beta}, \mathbf{X} \rangle\big)$$

Kernel PCA

$\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$. Assumed centered, $\sum_{i=1}^n \mathbf{x}_i = 0$ (W.l.o.g.!).

Usual PCA: Find directions with largest variation.

$$\boldsymbol{\beta}_1 = \underset{\|\boldsymbol{\beta}\|=1}{\arg\max} \frac{1}{n} \sum_{i=1}^{n} \big(\langle \boldsymbol{\beta}, \mathbf{x}_i \rangle\big)^2 = \underset{\|\boldsymbol{\beta}\|=1}{\arg\max} \quad \widehat{\text{var}}\big(\langle \boldsymbol{\beta}, \mathbf{X} \rangle\big)$$

$$\boldsymbol{\beta}_2 = \underset{\|\boldsymbol{\beta}\|=1, \boldsymbol{\beta}^\top \boldsymbol{\beta}_1 = 0}{\arg\max} \frac{1}{n} \sum_{i=1}^{n} \big(\langle \boldsymbol{\beta}, \mathbf{x}_i \rangle\big)^2 = \underset{\|\boldsymbol{\beta}\|=1, \boldsymbol{\beta}^\top \boldsymbol{\beta}_1 = 0}{\arg\max} \quad \widehat{\text{var}}\big(\langle \boldsymbol{\beta}, \mathbf{X} \rangle\big)$$

$$\vdots$$

$$\boldsymbol{\beta}_l = \underset{\|\boldsymbol{\beta}\|=1, \boldsymbol{\beta}^\top \boldsymbol{\beta}_1 = \cdots = \boldsymbol{\beta}^\top \boldsymbol{\beta}_{l-1} = 0}{\arg\max} \frac{1}{n} \sum_{i=1}^{n} \big(\langle \boldsymbol{\beta}, \mathbf{x}_i \rangle\big)^2 = \underset{\|\boldsymbol{\beta}\|=1, \boldsymbol{\beta}^\top \boldsymbol{\beta}_1 = \cdots = \boldsymbol{\beta}^\top \boldsymbol{\beta}_{l-1} = 0}{\arg\max} \quad \widehat{\text{var}}\big(\langle \boldsymbol{\beta}, \mathbf{X} \rangle\big)$$

Kernel PCA

$\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$. Assumed centered, $\sum_{i=1}^n \mathbf{x}_i = 0$ (W.l.o.g.!).

Usual PCA: Find directions with largest variation.

$$\boldsymbol{\beta}_1 = \underset{\|\boldsymbol{\beta}\|=1}{\arg\max} \frac{1}{n} \sum_{i=1}^n \left(\langle \boldsymbol{\beta}, \mathbf{x}_i \rangle\right)^2 = \underset{\|\boldsymbol{\beta}\|=1}{\arg\max} \quad \widehat{\operatorname{var}}\left(\langle \boldsymbol{\beta}, \mathbf{X} \rangle\right)$$

$$\boldsymbol{\beta}_2 = \underset{\|\boldsymbol{\beta}\|=1, \boldsymbol{\beta}^\top \boldsymbol{\beta}_1 = 0}{\arg\max} \frac{1}{n} \sum_{i=1}^n \left(\langle \boldsymbol{\beta}, \mathbf{x}_i \rangle\right)^2 = \underset{\|\boldsymbol{\beta}\|=1, \boldsymbol{\beta}^\top \boldsymbol{\beta}_1 = 0}{\arg\max} \quad \widehat{\operatorname{var}}\left(\langle \boldsymbol{\beta}, \mathbf{X} \rangle\right)$$

$$\vdots$$

$$\boldsymbol{\beta}_l = \underset{\|\boldsymbol{\beta}\|=1, \boldsymbol{\beta}^\top \boldsymbol{\beta}_1 = \cdots = \boldsymbol{\beta}^\top \boldsymbol{\beta}_{l-1} = 0}{\arg\max} \frac{1}{n} \sum_{i=1}^n \left(\langle \boldsymbol{\beta}, \mathbf{x}_i \rangle\right)^2 = \underset{\|\boldsymbol{\beta}\|=1, \boldsymbol{\beta}^\top \boldsymbol{\beta}_1 = \cdots = \boldsymbol{\beta}^\top \boldsymbol{\beta}_{l-1} = 0}{\arg\max} \quad \widehat{\operatorname{var}}\left(\langle \boldsymbol{\beta}, \mathbf{X} \rangle\right)$$

Turns out: $\displaystyle \frac{1}{n} \sum_{i=1}^n \left(\langle \boldsymbol{\beta}, \mathbf{x}_i \rangle\right)^2 = \boldsymbol{\beta}^\top \mathbf{S} \boldsymbol{\beta}, \qquad \mathbf{S} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top$ — sample covariance matrix.

Kernel PCA

$\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$. Assumed centered, $\sum_{i=1}^n \mathbf{x}_i = 0$ (W.l.o.g.!).

Usual PCA: Find directions with largest variation.

$$\boldsymbol{\beta}_1 = \operatorname*{arg\,max}_{\|\boldsymbol{\beta}\|=1} \frac{1}{n}\sum_{i=1}^n \big(\langle \boldsymbol{\beta}, \mathbf{x}_i\rangle\big)^2 = \operatorname*{arg\,max}_{\|\boldsymbol{\beta}\|=1} \widehat{\operatorname{var}}\big(\langle \boldsymbol{\beta}, \mathbf{X}\rangle\big)$$

$$\boldsymbol{\beta}_2 = \operatorname*{arg\,max}_{\|\boldsymbol{\beta}\|=1, \boldsymbol{\beta}^\top\boldsymbol{\beta}_1=0} \frac{1}{n}\sum_{i=1}^n \big(\langle \boldsymbol{\beta}, \mathbf{x}_i\rangle\big)^2 = \operatorname*{arg\,max}_{\|\boldsymbol{\beta}\|=1, \boldsymbol{\beta}^\top\boldsymbol{\beta}_1=0} \widehat{\operatorname{var}}\big(\langle \boldsymbol{\beta}, \mathbf{X}\rangle\big)$$

$$\vdots$$

$$\boldsymbol{\beta}_l = \operatorname*{arg\,max}_{\|\boldsymbol{\beta}\|=1, \boldsymbol{\beta}^\top\boldsymbol{\beta}_1=\cdots=\boldsymbol{\beta}^\top\boldsymbol{\beta}_{l-1}=0} \frac{1}{n}\sum_{i=1}^n \big(\langle \boldsymbol{\beta}, \mathbf{x}_i\rangle\big)^2 = \operatorname*{arg\,max}_{\|\boldsymbol{\beta}\|=1, \boldsymbol{\beta}^\top\boldsymbol{\beta}_1=\cdots=\boldsymbol{\beta}^\top\boldsymbol{\beta}_{l-1}=0} \widehat{\operatorname{var}}\big(\langle \boldsymbol{\beta}, \mathbf{X}\rangle\big)$$

Turns out: $\displaystyle \frac{1}{n}\sum_{i=1}^n \big(\langle \boldsymbol{\beta}, \mathbf{x}_i\rangle\big)^2 = \boldsymbol{\beta}^\top \mathbf{S}\boldsymbol{\beta}$, $\displaystyle \mathbf{S} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\top$ — sample covariance matrix.

The PC directions are given by the eigenvectors of $\mathbf{S}$: $\mathbf{S}\boldsymbol{\beta}_l = \lambda_l \boldsymbol{\beta}_l$.

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top. \quad \mathbf{S}\boldsymbol{\beta} = \lambda\boldsymbol{\beta} \quad \Rightarrow \quad \boldsymbol{\beta} \in \mathrm{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \quad \Rightarrow \quad \boldsymbol{\beta} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i.$$

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^\top. \quad \mathbf{S}\boldsymbol{\beta} = \lambda\boldsymbol{\beta} \quad \Rightarrow \quad \boldsymbol{\beta} \in \operatorname{span}\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \quad \Rightarrow \quad \boldsymbol{\beta} = \sum_{i=1}^{n}\alpha_i\mathbf{x}_i.$$

$$\boldsymbol{\beta}^\top\boldsymbol{\beta} = \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\langle\mathbf{x}_i, \mathbf{x}_j\rangle$$

$$\boldsymbol{\beta}^\top\widetilde{\boldsymbol{\beta}} = \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\widetilde{\alpha}_j\langle\mathbf{x}_i, \mathbf{x}_j\rangle$$

$$\boldsymbol{\beta}^\top\mathbf{S}\boldsymbol{\beta} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{j'=1}^{n}\alpha_j\alpha_{j'}\langle\mathbf{x}_i, \mathbf{x}_j\rangle\langle\mathbf{x}_i, \mathbf{x}_{j'}\rangle$$

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top. \quad \mathbf{S}\boldsymbol{\beta} = \lambda\boldsymbol{\beta} \quad \Rightarrow \quad \boldsymbol{\beta} \in \operatorname{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \quad \Rightarrow \quad \boldsymbol{\beta} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i.$$

$$\boldsymbol{\beta}^\top \boldsymbol{\beta} = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$$

$$\boldsymbol{\beta}^\top \widetilde{\boldsymbol{\beta}} = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \widetilde{\alpha}_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle = \boldsymbol{\alpha}^\top \mathbf{K} \widetilde{\boldsymbol{\alpha}}$$

$$\boldsymbol{\beta}^\top \mathbf{S} \boldsymbol{\beta} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{j'=1}^{n} \alpha_j \alpha_{j'} \langle \mathbf{x}_i, \mathbf{x}_j \rangle \langle \mathbf{x}_i, \mathbf{x}_{j'} \rangle = \boldsymbol{\alpha}^\top \mathbf{K}^2 \boldsymbol{\alpha}$$

$$\mathbf{K} = \Big(\big(\langle \mathbf{x}_i, \mathbf{x}_j \rangle \big) \Big)_{1 \le i,j \le n}, \quad \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top.$$

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^\top. \quad \mathbf{S}\boldsymbol{\beta} = \lambda\boldsymbol{\beta} \quad \Rightarrow \quad \boldsymbol{\beta} \in \mathrm{span}\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \quad \Rightarrow \quad \boldsymbol{\beta} = \sum_{i=1}^{n}\alpha_i\mathbf{x}_i.$$

$$\boldsymbol{\beta}^\top\boldsymbol{\beta} = \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\langle\mathbf{x}_i, \mathbf{x}_j\rangle = \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha}$$

$$\boldsymbol{\beta}^\top\widetilde{\boldsymbol{\beta}} = \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\widetilde{\alpha}_j\langle\mathbf{x}_i, \mathbf{x}_j\rangle = \boldsymbol{\alpha}^\top\mathbf{K}\widetilde{\boldsymbol{\alpha}}$$

$$\boldsymbol{\beta}^\top\mathbf{S}\boldsymbol{\beta} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{j'=1}^{n}\alpha_j\alpha_{j'}\langle\mathbf{x}_i, \mathbf{x}_j\rangle\langle\mathbf{x}_i, \mathbf{x}_{j'}\rangle = \boldsymbol{\alpha}^\top\mathbf{K}^2\boldsymbol{\alpha}$$

$$\mathbf{K} = \big(\big(\langle\mathbf{x}_i, \mathbf{x}_j\rangle\big)\big)_{1\le i,j\le n}, \quad \boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^\top.$$

$$\boldsymbol{\beta}_1 \equiv \arg\max_{\boldsymbol{\alpha}} \ \boldsymbol{\alpha}^\top\mathbf{K}^2\boldsymbol{\alpha} \quad \text{s.t.} \quad \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} = 1$$

$$\boldsymbol{\beta}_2 \equiv \arg\max_{\boldsymbol{\alpha}} \ \boldsymbol{\alpha}^\top\mathbf{K}^2\boldsymbol{\alpha} \quad \text{s.t.} \quad \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} = 1, \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha}_1 = 0$$

$$\dots$$

$$\boldsymbol{\beta}_l \equiv \arg\max_{\boldsymbol{\alpha}} \ \boldsymbol{\alpha}^\top\mathbf{K}^2\boldsymbol{\alpha} \quad \text{s.t.} \quad \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} = 1, \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha}_1 = \cdots = \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha}_{l-1} = 0$$

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^\top. \quad \mathbf{S}\boldsymbol{\beta} = \lambda\boldsymbol{\beta} \quad \Rightarrow \quad \boldsymbol{\beta} \in \operatorname{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \quad \Rightarrow \quad \boldsymbol{\beta} = \sum_{i=1}^{n}\alpha_i\mathbf{x}_i.$$

$$\boldsymbol{\beta}^\top\boldsymbol{\beta} = \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\langle\mathbf{x}_i, \mathbf{x}_j\rangle = \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha}$$

$$\boldsymbol{\beta}^\top\widetilde{\boldsymbol{\beta}} = \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\widetilde{\alpha}_j\langle\mathbf{x}_i, \mathbf{x}_j\rangle = \boldsymbol{\alpha}^\top\mathbf{K}\widetilde{\boldsymbol{\alpha}}$$

$$\boldsymbol{\beta}^\top\mathbf{S}\boldsymbol{\beta} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{j'=1}^{n}\alpha_j\alpha_{j'}\langle\mathbf{x}_i, \mathbf{x}_j\rangle\langle\mathbf{x}_i, \mathbf{x}_{j'}\rangle = \boldsymbol{\alpha}^\top\mathbf{K}^2\boldsymbol{\alpha}$$

$$\mathbf{K} = \big(\left(\langle\mathbf{x}_i, \mathbf{x}_j\rangle\right)\big)_{1\le i,j\le n}, \quad \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top.$$

$$\boldsymbol{\beta}_1 \equiv \arg\max_{\boldsymbol{\alpha}} \quad \boldsymbol{\alpha}^\top\mathbf{K}^2\boldsymbol{\alpha} \quad \text{s.t.} \quad \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} = 1$$

$$\boldsymbol{\beta}_2 \equiv \arg\max_{\boldsymbol{\alpha}} \quad \boldsymbol{\alpha}^\top\mathbf{K}^2\boldsymbol{\alpha} \quad \text{s.t.} \quad \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} = 1, \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha}_1 = 0$$

$$\ldots$$

$$\boldsymbol{\beta}_l \equiv \arg\max_{\boldsymbol{\alpha}} \quad \boldsymbol{\alpha}^\top\mathbf{K}^2\boldsymbol{\alpha} \quad \text{s.t.} \quad \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} = 1, \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha}_1 = \cdots = \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha}_{l-1} = 0$$

For a new observation $\mathbf{x}$, the l-th PC score: $\langle\boldsymbol{\beta}_l, \mathbf{x}\rangle = \sum_{i=1}^{n}\alpha_{l,i}\langle\mathbf{x}_i, \mathbf{x}\rangle$

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^\top. \quad \mathbf{S}\boldsymbol{\beta} = \lambda\boldsymbol{\beta} \quad \Rightarrow \quad \boldsymbol{\beta} \in \operatorname{span}\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \quad \Rightarrow \quad \boldsymbol{\beta} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i.$$

$$\boldsymbol{\beta}^\top\boldsymbol{\beta} = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j\langle \mathbf{x}_i, \mathbf{x}_j\rangle = \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha}$$

$$\boldsymbol{\beta}^\top\widetilde{\boldsymbol{\beta}} = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\widetilde{\alpha}_j\langle \mathbf{x}_i, \mathbf{x}_j\rangle = \boldsymbol{\alpha}^\top\mathbf{K}\widetilde{\boldsymbol{\alpha}}$$

$$\boldsymbol{\beta}^\top\mathbf{S}\boldsymbol{\beta} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{j'=1}^{n} \alpha_j\alpha_{j'}\langle \mathbf{x}_i, \mathbf{x}_j\rangle\langle \mathbf{x}_i, \mathbf{x}_{j'}\rangle = \boldsymbol{\alpha}^\top\mathbf{K}^2\boldsymbol{\alpha}$$

$$\mathbf{K} = \big(\big(\langle \mathbf{x}_i, \mathbf{x}_j\rangle\big)\big)_{1\le i,j\le n}, \quad \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top.$$

$$\boldsymbol{\beta}_1 \equiv \arg\max_{\boldsymbol{\alpha}} \ \boldsymbol{\alpha}^\top\mathbf{K}^2\boldsymbol{\alpha} \quad \text{s.t.} \quad \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} = 1$$

$$\boldsymbol{\beta}_2 \equiv \arg\max_{\boldsymbol{\alpha}} \ \boldsymbol{\alpha}^\top\mathbf{K}^2\boldsymbol{\alpha} \quad \text{s.t.} \quad \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} = 1, \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha}_1 = 0$$

$$\ldots$$

$$\boldsymbol{\beta}_l \equiv \arg\max_{\boldsymbol{\alpha}} \ \boldsymbol{\alpha}^\top\mathbf{K}^2\boldsymbol{\alpha} \quad \text{s.t.} \quad \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha} = 1, \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha}_1 = \cdots = \boldsymbol{\alpha}^\top\mathbf{K}\boldsymbol{\alpha}_{l-1} = 0$$

For a new observation $\mathbf{x}$, the l-th PC score: $\langle \boldsymbol{\beta}_l, \mathbf{x}\rangle = \sum_{i=1}^{n} \alpha_{l,i}\langle \mathbf{x}_i, \mathbf{x}\rangle$

- Obtaining the PC's depend on the inner-product matrix $\mathbf{K}$.
- Obtaining PC scores for a new observation depends on inner-product.

Kernel PCA algorithm: Schölkopf, Smola & Müller (1998)

- Take a kernel K. Define the kernel matrix $\mathbf{K} = \big(\big(K(\mathbf{x}_i, \mathbf{x}_j)\big)\big)_{1 \le i,j \le n}$.
- For $l = 1, 2, \ldots$, find:

$$\boldsymbol{\alpha}_l = \arg\max_{\boldsymbol{\alpha}} \;\; \boldsymbol{\alpha}^\top \mathbf{K}^2 \boldsymbol{\alpha} \quad \text{s.t.} \quad \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} = 1, \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}_1 = \cdots = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}_{l-1} = 0.$$

The l-th kernel PC is: $\displaystyle\sum_{i=1}^{n} \alpha_{l,i} K(\cdot, \mathbf{x}_i)$.

For an observation $\mathbf{x}$, the l-th PC score is: $\displaystyle\sum_{i=1}^{n} \alpha_{l,i} K(\mathbf{x}, \mathbf{x}_i)$.

Kernel PCA algorithm: Schölkopf, Smola & Müller (1998)

- Take a kernel K. Define the kernel matrix $\mathbf{K} = \big(\big(K(\mathbf{x}_i, \mathbf{x}_j)\big)\big)_{1 \le i,j \le n}$.
- For $l = 1, 2, \ldots$, find:

$$\boldsymbol{\alpha}_l = \arg\max_{\boldsymbol{\alpha}} \ \boldsymbol{\alpha}^\top \mathbf{K}^2 \boldsymbol{\alpha} \quad \text{s.t.} \quad \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} = 1, \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}_1 = \cdots = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}_{l-1} = 0.$$

  The l-th kernel PC is: $\displaystyle\sum_{i=1}^{n} \alpha_{l,i} K(\cdot, \mathbf{x}_i)$.

  For an observation $\mathbf{x}$, the l-th PC score is: $\displaystyle\sum_{i=1}^{n} \alpha_{l,i} K(\mathbf{x}, \mathbf{x}_i)$.

At a conceptual level:

- Transform $\mathbf{x} \to \phi(\mathbf{x}) = K(\cdot, \mathbf{x})$.
- Perform PCA in the transformed space.
- Linear in the transformed space $\Rightarrow$ Non-linear in the actual space.

Kernel PCA algorithm: Schölkopf, Smola & Müller (1998)

- Take a kernel K. Define the kernel matrix $\mathbf{K} = \big(\big(K(\mathbf{x}_i, \mathbf{x}_j)\big)\big)_{1 \le i,j \le n}$.
- For $l = 1, 2, \ldots,$ find:

$$\boldsymbol{\alpha}_l = \arg\max_{\boldsymbol{\alpha}} \ \boldsymbol{\alpha}^\top \mathbf{K}^2 \boldsymbol{\alpha} \quad \text{s.t.} \quad \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} = 1, \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}_1 = \cdots = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}_{l-1} = 0.$$

  The l-th kernel PC is: $\displaystyle\sum_{i=1}^n \alpha_{l,i} K(\cdot, \mathbf{x}_i).$

  For an observation $\mathbf{x}$, the l-th PC score is: $\displaystyle\sum_{i=1}^n \alpha_{l,i} K(\mathbf{x}, \mathbf{x}_i).$

At a conceptual level:

- Transform $\mathbf{x} \to \phi(\mathbf{x}) = K(\cdot, \mathbf{x})$.
- Perform PCA in the transformed space.
- Linear in the transformed space $\Rightarrow$ Non-linear in the actual space.

No need to form the transformation ever. All calculations involve kernel evaluations only.

Kernel PCA algorithm: Schölkopf, Smola & Müller (1998)

- Take a kernel K. Define the kernel matrix $\mathbf{K} = \big((K(\mathbf{x}_i, \mathbf{x}_j))\big)_{1 \le i,j \le n}$.
- For $l = 1, 2, \ldots$, find:

$$\boldsymbol{\alpha}_l = \arg\max_{\boldsymbol{\alpha}} \; \boldsymbol{\alpha}^\top \mathbf{K}^2 \boldsymbol{\alpha} \quad \text{s.t.} \quad \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} = 1, \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}_1 = \cdots = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}_{l-1} = 0.$$

  The l-th kernel PC is: $\displaystyle\sum_{i=1}^{n} \alpha_{l,i} K(\cdot, \mathbf{x}_i)$.

  For an observation $\mathbf{x}$, the l-th PC score is: $\displaystyle\sum_{i=1}^{n} \alpha_{l,i} K(\mathbf{x}, \mathbf{x}_i)$.

At a conceptual level:

- Transform $\mathbf{x} \to \phi(\mathbf{x}) = K(\cdot, \mathbf{x})$.
- Perform PCA in the transformed space.
- Linear in the transformed space $\Rightarrow$ Non-linear in the actual space.

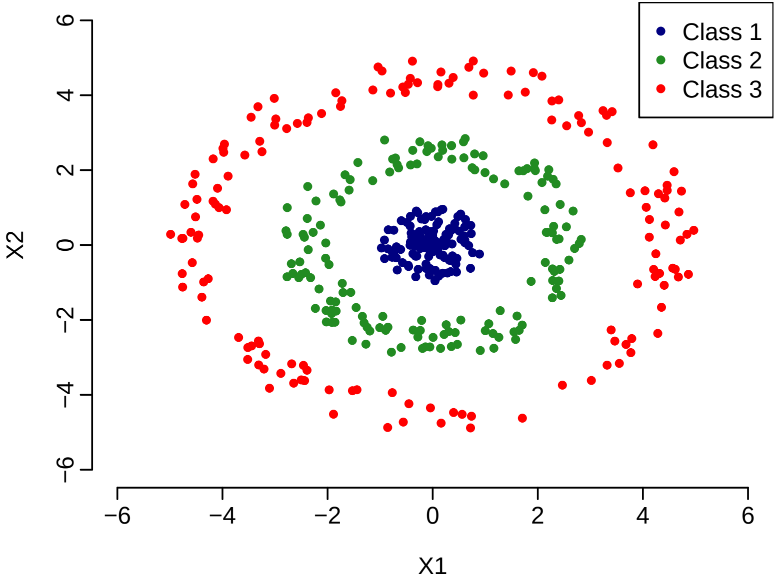
No need to form the transformation ever. All calculations involve kernel evaluations only.

⋆ We assumed centered observations.

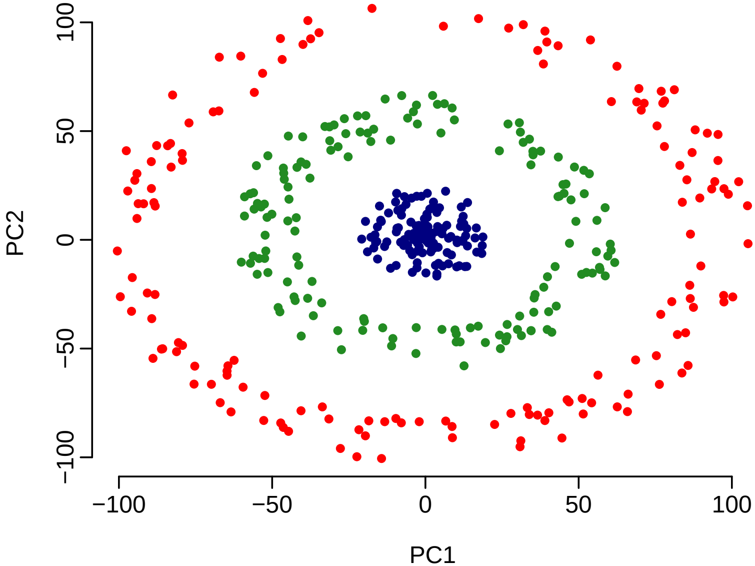
In general, need to use the double centered kernel matrix:

$$\widetilde{\mathbf{K}} = (\mathbf{I} - \mathbf{H})\mathbf{K}(\mathbf{I} - \mathbf{H}), \quad \mathbf{H} = \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top$$

An Example

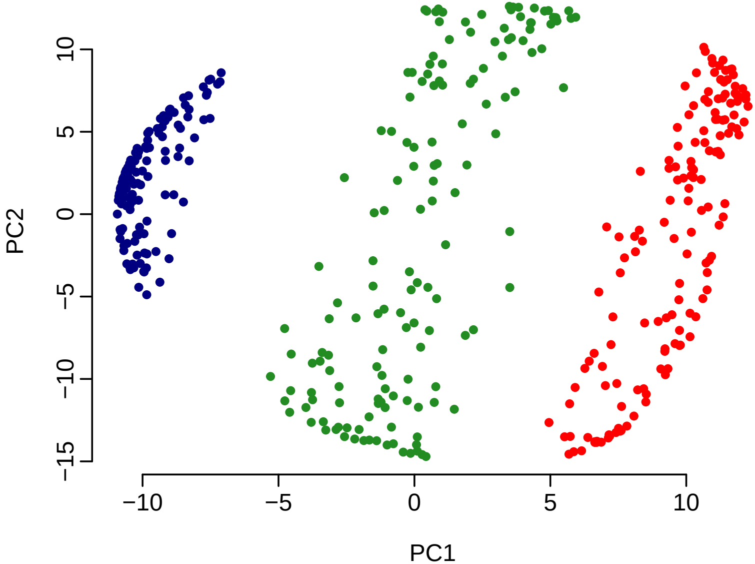


An Example



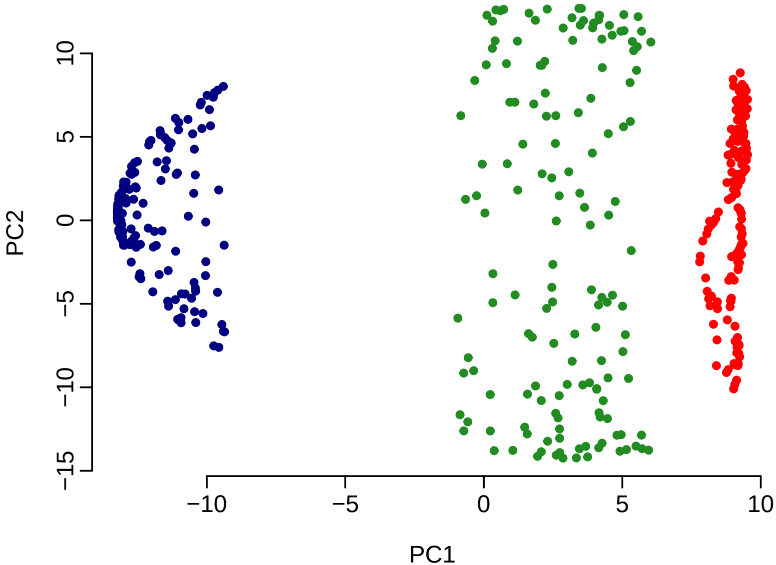
Usual PCA

An Example



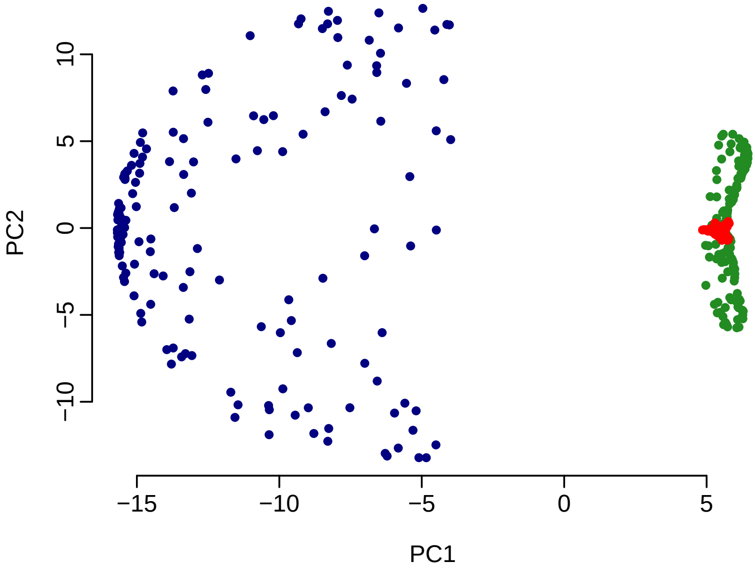
Kernel PCA with RBF kernel ($\sigma = 10$).

An Example



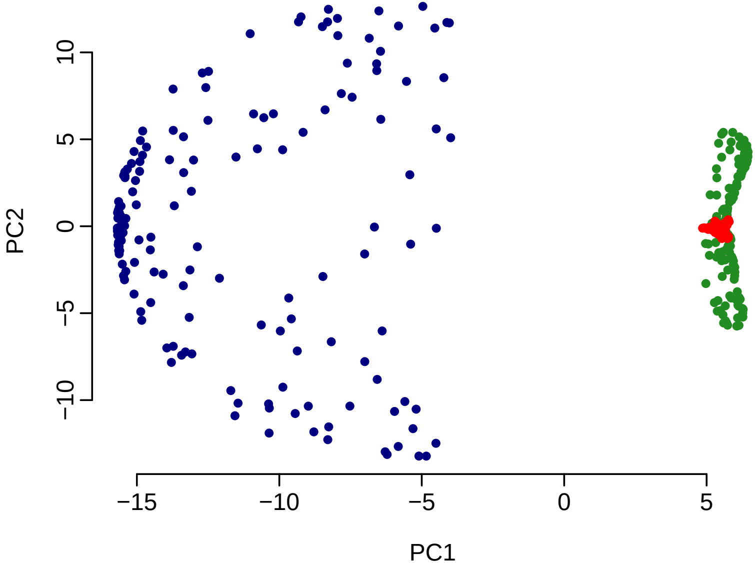
Kernel PCA with RBF kernel ($\sigma = 5$).

An Example



Kernel PCA with RBF kernel ($\sigma = 1$).

An Example



Kernel PCA with RBF kernel ($\sigma = 1$).

Choice of hyper-parameter is crucial

Kernel Two-sample Test

Two sets of observations: $\mathbf{x}_1, \ldots, \mathbf{x}_m \sim \mathbf{X}$, $\mathbf{y}_1, \ldots, \mathbf{y}_n \sim \mathbf{Y}$.

Want to check if they have the same distribution — $\mathcal{H}_0 : \mathbf{X} \overset{\mathcal{D}}{=} \mathbf{Y}$ against $\mathcal{H}_a : \mathbf{X} \overset{\mathcal{D}}{\neq} \mathbf{Y}$.

Kernel Two-sample Test

Two sets of observations: $\mathbf{x}_1, \ldots, \mathbf{x}_m \sim \mathbf{X}$, $\mathbf{y}_1, \ldots, \mathbf{y}_n \sim \mathbf{Y}$.

Want to check if they have the same distribution — $\mathcal{H}_0 : \mathbf{X} \stackrel{\mathcal{D}}{=} \mathbf{Y}$ against $\mathcal{H}_a : \mathbf{X} \stackrel{\mathcal{D}}{\neq} \mathbf{Y}$.

- Can use: $T = \left\|\bar{\mathbf{x}} - \bar{\mathbf{y}}\right\|^2$ for the test.

  Corresponds to checking: $\mathbb{E}(\mathbf{X}) = \mathbb{E}(\mathbf{Y})$ against $\mathbb{E}(\mathbf{X}) \neq \mathbb{E}(\mathbf{Y})$
  - A location alternative.
  - $\mathbf{X}$ and $\mathbf{Y}$ are well separated.

Kernel Two-sample Test

Two sets of observations: $\mathbf{x}_1, \ldots, \mathbf{x}_m \sim \mathbf{X}$, $\mathbf{y}_1, \ldots, \mathbf{y}_n \sim \mathbf{Y}$.

Want to check if they have the same distribution — $\mathcal{H}_0 : \mathbf{X} \overset{\mathcal{D}}{=} \mathbf{Y}$ against $\mathcal{H}_a : \mathbf{X} \overset{\mathcal{D}}{\neq} \mathbf{Y}$.

- Can use: $T = \left\|\bar{\mathbf{x}} - \bar{\mathbf{y}}\right\|^2$ for the test.

  Corresponds to checking: $\mathbb{E}(\mathbf{X}) = \mathbb{E}(\mathbf{Y})$ against $\mathbb{E}(\mathbf{X}) \neq \mathbb{E}(\mathbf{Y})$
  - A location alternative.
  - $\mathbf{X}$ and $\mathbf{Y}$ are well separated.
- Transform the data: $\mathbf{x} \to \phi(\mathbf{x})$.

  Hope that the sets are well separated in the transformed space.

Kernel Two-sample Test

Two sets of observations: $\mathbf{x}_1, \ldots, \mathbf{x}_m \sim \mathbf{X}$, $\mathbf{y}_1, \ldots, \mathbf{y}_n \sim \mathbf{Y}$.

Want to check if they have the same distribution — $\mathcal{H}_0 : \mathbf{X} \overset{\mathcal{D}}{=} \mathbf{Y}$ against $\mathcal{H}_a : \mathbf{X} \overset{\mathcal{D}}{\neq} \mathbf{Y}$.

- Can use: $T = \left\|\bar{\mathbf{x}} - \bar{\mathbf{y}}\right\|^2$ for the test.

  Corresponds to checking: $\mathbb{E}(\mathbf{X}) = \mathbb{E}(\mathbf{Y})$ against $\mathbb{E}(\mathbf{X}) \neq \mathbb{E}(\mathbf{Y})$
  - A location alternative.
  - $\mathbf{X}$ and $\mathbf{Y}$ are well separated.
- Transform the data: $\mathbf{x} \to \phi(\mathbf{x})$.

  Hope that the sets are well separated in the transformed space.

$$T = \left\| \frac{1}{m} \sum_{i=1}^{m} \phi(\mathbf{x}_i) - \frac{1}{n} \sum_{j=1}^{n} \phi(\mathbf{y}_j) \right\|^2.$$

Kernel Two-sample Test

Two sets of observations: $\mathbf{x}_1, \ldots, \mathbf{x}_m \sim \mathbf{X}$, $\mathbf{y}_1, \ldots, \mathbf{y}_n \sim \mathbf{Y}$.

Want to check if they have the same distribution — $\mathcal{H}_0 : \mathbf{X} \overset{\mathcal{D}}{=} \mathbf{Y}$ against $\mathcal{H}_a : \mathbf{X} \overset{\mathcal{D}}{\neq} \mathbf{Y}$.

- Can use: $T = \left\|\bar{\mathbf{x}} - \bar{\mathbf{y}}\right\|^2$ for the test.

  Corresponds to checking: $\mathbb{E}(\mathbf{X}) = \mathbb{E}(\mathbf{Y})$ against $\mathbb{E}(\mathbf{X}) \neq \mathbb{E}(\mathbf{Y})$
  - A location alternative.
  - $\mathbf{X}$ and $\mathbf{Y}$ are well separated.
- Transform the data: $\mathbf{x} \to \phi(\mathbf{x})$.

  Hope that the sets are well separated in the transformed space.

$$T = \left\| \frac{1}{m} \sum_{i=1}^{m} \phi(\mathbf{x}_i) - \frac{1}{n} \sum_{j=1}^{n} \phi(\mathbf{y}_j) \right\|^2.$$

$$= \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \langle \phi(\mathbf{y}_i), \phi(\mathbf{y}_j) \rangle - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \langle \phi(\mathbf{x}_i), \phi(\mathbf{y}_j) \rangle$$

Kernel Two-sample Test

Two sets of observations: $\mathbf{x}_1, \ldots, \mathbf{x}_m \sim \mathbf{X}$, $\mathbf{y}_1, \ldots, \mathbf{y}_n \sim \mathbf{Y}$.

Want to check if they have the same distribution — $\mathcal{H}_0 : \mathbf{X} \stackrel{\mathcal{D}}{=} \mathbf{Y}$ against $\mathcal{H}_a : \mathbf{X} \stackrel{\mathcal{D}}{\neq} \mathbf{Y}$.

- Can use: $T = \left\|\bar{\mathbf{x}} - \bar{\mathbf{y}}\right\|^2$ for the test.

  Corresponds to checking: $\mathbb{E}(\mathbf{X}) = \mathbb{E}(\mathbf{Y})$ against $\mathbb{E}(\mathbf{X}) \neq \mathbb{E}(\mathbf{Y})$
  - A location alternative.
  - $\mathbf{X}$ and $\mathbf{Y}$ are well separated.
- Transform the data: $\mathbf{x} \to \phi(\mathbf{x})$.

  Hope that the sets are well separated in the transformed space.

$$
\begin{aligned}
T &= \left\| \frac{1}{m}\sum_{i=1}^{m}\phi(\mathbf{x}_i) - \frac{1}{n}\sum_{j=1}^{n}\phi(\mathbf{y}_j) \right\|^2. \\
&= \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\langle\phi(\mathbf{y}_i), \phi(\mathbf{y}_j)\rangle - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\langle\phi(\mathbf{x}_i), \phi(\mathbf{y}_j)\rangle \\
&= \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}K(\mathbf{y}_i, \mathbf{y}_j) - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}K(\mathbf{x}_i, \mathbf{y}_j).
\end{aligned}
$$

Kernel Two-sample Test

Two sets of observations: $\mathbf{x}_1, \ldots, \mathbf{x}_m \sim \mathbf{X}$, $\mathbf{y}_1, \ldots, \mathbf{y}_n \sim \mathbf{Y}$.

Want to check if they have the same distribution — $\mathcal{H}_0 : \mathbf{X} \overset{\mathcal{D}}{=} \mathbf{Y}$ against $\mathcal{H}_a : \mathbf{X} \overset{\mathcal{D}}{\neq} \mathbf{Y}$.

- Can use: $T = \left\|\bar{\mathbf{x}} - \bar{\mathbf{y}}\right\|^2$ for the test.

  Corresponds to checking: $\mathbb{E}(\mathbf{X}) = \mathbb{E}(\mathbf{Y})$ against $\mathbb{E}(\mathbf{X}) \neq \mathbb{E}(\mathbf{Y})$
  - A location alternative.
  - $\mathbf{X}$ and $\mathbf{Y}$ are well separated.
- Transform the data: $\mathbf{x} \to \phi(\mathbf{x})$.

  Hope that the sets are well separated in the transformed space.

$$
\begin{aligned}
T &= \left\| \frac{1}{m} \sum_{i=1}^{m} \phi(\mathbf{x}_i) - \frac{1}{n} \sum_{j=1}^{n} \phi(\mathbf{y}_j) \right\|^2. \\
&= \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \langle \phi(\mathbf{y}_i), \phi(\mathbf{y}_j) \rangle - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \langle \phi(\mathbf{x}_i), \phi(\mathbf{y}_j) \rangle \\
&= \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} K(\mathbf{y}_i, \mathbf{y}_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} K(\mathbf{x}_i, \mathbf{y}_j).
\end{aligned}
$$

— Gretton, Borgwardt, Rasch, Schölkopf & Smola (2006).

Kernel Two-sample Test

Two sets of observations: $\mathbf{x}_1, \ldots, \mathbf{x}_m \sim \mathbf{X}$, $\mathbf{y}_1, \ldots, \mathbf{y}_n \sim \mathbf{Y}$.

Want to check if they have the same distribution — $\mathcal{H}_0 : \mathbf{X} \overset{\mathcal{D}}{=} \mathbf{Y}$ against $\mathcal{H}_a : \mathbf{X} \overset{\mathcal{D}}{\neq} \mathbf{Y}$.

- Can use: $T = \left\|\bar{\mathbf{x}} - \bar{\mathbf{y}}\right\|^2$ for the test.

  Corresponds to checking: $\mathbb{E}(\mathbf{X}) = \mathbb{E}(\mathbf{Y})$ against $\mathbb{E}(\mathbf{X}) \neq \mathbb{E}(\mathbf{Y})$
  - A location alternative.
  - $\mathbf{X}$ and $\mathbf{Y}$ are well separated.
- Transform the data: $\mathbf{x} \to \phi(\mathbf{x})$.

  Hope that the sets are well separated in the transformed space.

$$
\begin{aligned}
T &= \left\| \frac{1}{m} \sum_{i=1}^{m} \phi(\mathbf{x}_i) - \frac{1}{n} \sum_{j=1}^{n} \phi(\mathbf{y}_j) \right\|^2 . \\
&= \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \langle \phi(\mathbf{y}_i), \phi(\mathbf{y}_j) \rangle - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \langle \phi(\mathbf{x}_i), \phi(\mathbf{y}_j) \rangle \\
&= \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} K(\mathbf{y}_i, \mathbf{y}_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} K(\mathbf{x}_i, \mathbf{y}_j).
\end{aligned}
$$

— Gretton, Borgwardt, Rasch, Schölkopf & Smola (2006).

- For suitable choices of K, this test is exact.

  — viable against the general alternative: $\mathbf{X} \overset{\mathcal{D}}{\neq} \mathbf{Y}$.

  More on this from other speakers!

- Kernel methods can be used with arbitrary features $\mathbf{x}$

    – vectors, matrices, images, texts etc.

  All we need is to be able to evaluate the kernel.

- Kernel methods can be used with arbitrary features $\mathbf{x}$

  – vectors, matrices, images, texts etc.

  All we need is to be able to evaluate the kernel.

- Some other methods:
  - Kernel ridge regression.
  - Kernel logistic regression.
  - Kernel nearest neighbor.
  - Kernel canonical correlation analysis (CCA).
  - Kernel K-means clustering.

A look into the future

A look into the future

- Large scale problems.
  - For n observations, need at least n^2 computations for the kernel matrix.
  - Too much when n is large.
  - Reduce the computation in n.

A look into the future

- Large scale problems.
  - For n observations, need at least n^2 computations for the kernel matrix.
  - Too much when n is large.
  - Reduce the computation in n.

- Learning the kernel.
  - The choice of K is crucial for kernel methods.
  - How to select a good kernel from the data?

References

- Berlinet, A. and Thomas-Agnan, C. (2011) Reproducing Kernel Hilbert Spaces in Probability and Statistics. Springer Science & Business Media.
- Nadaraya, E. A. (1964) "On estimating regression." *Theory of Probability and Its Applications*, **9**, 141–142.
- Watson, G. S. (1964) "Smooth regression analysis." *Sankhyā (Series A)*, **26**, 359–372.
- Rosenblatt, M. (1956) "Remarks on some nonparametric estimates of a density func- tion." *The Annals of Mathematical Statistics*, **27**, 832–837.
- Parzen, E. (1962) "On estimation of a probability density function and mode." *The Annals of Mathematical Statistics*, **33**, 1065–1076.
- Kimeldorf, G. and Wahba, G. (1971) "Some results on Tchebycheffian spline functions." *Journal of Mathematical Analysis and Applications*, **33**, 82–95.
- Vapnik, V. and Chervonenkis, A. (1974). Theory of Pattern Recognition. Nauka, Moscow.
- Boser, B., Guyon, I. and Vapnik, V. (1992) "A training algorithm for an optimal margin classifier." In *Fifth annual Workshop on Computational Learning Theory*, pp. 144–152. Pittsburgh AMC.
- Cortes, C. and Vapnik, V. (1995) "Support vector networks." *Machine Learning*, **20**, 1–25.
- Schölkopf, B., Smola, A. and Müller, K.-R. (1998) "Nonlinear component analysis as a kernel eigenvalue problem." *Neural Computation*, **10**, 1299–1319.
- Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B. and Smola, A. (2006) "A kernel method for the two-sample-problem." *Advances in Neural Information Processing Systems*, **19**, 513–520.