Nonlinear SVM and kernel methods

Guillaume Obozinski

Ecole des Ponts - ParisTech

Cours MALAP 2014

Changing the dot product

Let $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\phi(\mathbf{x}) = (x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2)^\top$.

Changing the dot product

Let $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\phi(\mathbf{x}) = (x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^\top$.

$$\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$$

Changing the dot product

Let $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\phi(\mathbf{x}) = (x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^\top$.

$$\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = x_1y_1 + x_2y_2 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$$

Changing the dot product

Let $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\phi(\mathbf{x}) = (x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^\top$.

$$\begin{aligned}\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle &= x_1y_1 + x_2y_2 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 \\ &= x_1y_1 + x_2y_2 + (x_1y_1)^2 + (x_2y_2)^2 + 2(x_1y_1)(x_2y_2)\end{aligned}$$

Changing the dot product

Let $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\phi(\mathbf{x}) = (x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^\top$.

$$\begin{aligned}
\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle &= x_1y_1 + x_2y_2 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 \\
&= x_1y_1 + x_2y_2 + (x_1y_1)^2 + (x_2y_2)^2 + 2(x_1y_1)(x_2y_2) \\
&= \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle^2
\end{aligned}$$

Changing the dot product

Let $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\phi(\mathbf{x}) = (x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2)^\top$.

$$\begin{aligned}
\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle &= x_1 y_1 + x_2 y_2 + x_1^2 y_1^2 + x_2^2 y_2^2 + 2 x_1 x_2 y_1 y_2 \\
&= x_1 y_1 + x_2 y_2 + (x_1 y_1)^2 + (x_2 y_2)^2 + 2(x_1 y_1)(x_2 y_2) \\
&= \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle^2
\end{aligned}$$

For $\mathbf{w} = (0, 0, 1, 1, 0)^\top$, $\quad \mathbf{w}^\top \phi(\mathbf{x}) - 1 \leq 0 \quad \Leftrightarrow \quad \|\mathbf{x}\|^2 \leq 1.$

Changing the dot product

Let $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\phi(\mathbf{x}) = (x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^\top$.

$$\begin{aligned}
\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle &= x_1y_1 + x_2y_2 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 \\
&= x_1y_1 + x_2y_2 + (x_1y_1)^2 + (x_2y_2)^2 + 2(x_1y_1)(x_2y_2) \\
&= \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle^2
\end{aligned}$$

For $\mathbf{w} = (0, 0, 1, 1, 0)^\top, \quad \mathbf{w}^\top \phi(\mathbf{x}) - 1 \leq 0 \quad \Leftrightarrow \quad \|\mathbf{x}\|^2 \leq 1.$

Linear separators in $\mathbb{R}^5$ correspond to conic separators in $\mathbb{R}^2$.

Changing the dot product

Let $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\phi(\mathbf{x}) = (x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^\top$.

$$\begin{aligned} \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle &= x_1y_1 + x_2y_2 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 \\ &= x_1y_1 + x_2y_2 + (x_1y_1)^2 + (x_2y_2)^2 + 2(x_1y_1)(x_2y_2) \\ &= \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle^2 \end{aligned}$$

For $\mathbf{w} = (0, 0, 1, 1, 0)^\top, \quad \mathbf{w}^\top \phi(\mathbf{x}) - 1 \leq 0 \quad \Leftrightarrow \quad \|\mathbf{x}\|^2 \leq 1.$

Linear separators in $\mathbb{R}^5$ correspond to conic separators in $\mathbb{R}^2$.
http://www.youtube.com/watch?v=3liCbRZPrZA

Changing the dot product

Let $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\phi(\mathbf{x}) = (x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^\top$.

$$\begin{aligned}
\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle &= x_1y_1 + x_2y_2 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 \\
&= x_1y_1 + x_2y_2 + (x_1y_1)^2 + (x_2y_2)^2 + 2(x_1y_1)(x_2y_2) \\
&= \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle^2
\end{aligned}$$

For $\mathbf{w} = (0, 0, 1, 1, 0)^\top, \quad \mathbf{w}^\top\phi(\mathbf{x}) - 1 \leq 0 \quad \Leftrightarrow \quad \|\mathbf{x}\|^2 \leq 1.$

Linear separators in $\mathbb{R}^5$ correspond to conic separators in $\mathbb{R}^2$.
http://www.youtube.com/watch?v=3liCbRZPrZA

Let $\mathbf{x} = (x_1, \ldots, x_p) \in \mathbb{R}^p$ and

$$\phi(\mathbf{x}) = (x_1, \ldots, x_p, x_1^2, \ldots, x_p^2, \sqrt{2}x_1x_2, \ldots, \sqrt{2}x_ix_j, \ldots \sqrt{2}x_{p-1}x_p)^\top.$$

Changing the dot product

Let $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\phi(\mathbf{x}) = (x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^\top$.

$$\begin{aligned}\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle &= x_1y_1 + x_2y_2 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 \\ &= x_1y_1 + x_2y_2 + (x_1y_1)^2 + (x_2y_2)^2 + 2(x_1y_1)(x_2y_2) \\ &= \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle^2\end{aligned}$$

For $\mathbf{w} = (0, 0, 1, 1, 0)^\top$, $\quad \mathbf{w}^\top \phi(\mathbf{x}) - 1 \leq 0 \quad \Leftrightarrow \quad \|\mathbf{x}\|^2 \leq 1$.

Linear separators in $\mathbb{R}^5$ correspond to conic separators in $\mathbb{R}^2$.
http://www.youtube.com/watch?v=3liCbRZPrZA

Let $\mathbf{x} = (x_1, \ldots, x_p) \in \mathbb{R}^p$ and

$$\phi(\mathbf{x}) = (x_1, \ldots, x_p, x_1^2, \ldots, x_p^2, \sqrt{2}x_1x_2, \ldots, \sqrt{2}x_ix_j, \ldots \sqrt{2}x_{p-1}x_p)^\top.$$

Still have

$$\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle^2$$

Changing the dot product

Let $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\phi(\mathbf{x}) = (x_1, x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^\top$.

$$\begin{aligned} \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle &= x_1y_1 + x_2y_2 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 \\ &= x_1y_1 + x_2y_2 + (x_1y_1)^2 + (x_2y_2)^2 + 2(x_1y_1)(x_2y_2) \\ &= \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle^2 \end{aligned}$$

For $\mathbf{w} = (0, 0, 1, 1, 0)^\top$, $\quad \mathbf{w}^\top \phi(\mathbf{x}) - 1 \leq 0 \quad \Leftrightarrow \quad \|\mathbf{x}\|^2 \leq 1$.

Linear separators in $\mathbb{R}^5$ correspond to conic separators in $\mathbb{R}^2$.
http://www.youtube.com/watch?v=3liCbRZPrZA

Let $\mathbf{x} = (x_1, \ldots, x_p) \in \mathbb{R}^p$ and

$$\phi(\mathbf{x}) = (x_1, \ldots, x_p, x_1^2, \ldots, x_p^2, \sqrt{2}x_1x_2, \ldots, \sqrt{2}x_ix_j, \ldots \sqrt{2}x_{p-1}x_p)^\top.$$

Still have

$$\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle^2$$

But explicit mapping too expensive to compute: $\phi(\mathbf{x}) \in \mathbb{R}^{p+p(p+1)/2}$.

Reproducing kernel Hilbert space

Nice space of functions for non-parametric statistics and machine learning?

Reproducing kernel Hilbert space

Nice space of functions for non-parametric statistics and machine learning? Require that

- the *evaluation functionals* $f \mapsto f(x)$ be $\mathcal{C}^0$ for all $x \in \mathcal{X}$.

Reproducing kernel Hilbert space

Nice space of functions for non-parametric statistics and machine learning? Require that

- the *evaluation functionals* $f \mapsto f(x)$ be $\mathcal{C}^0$ for all $x \in \mathcal{X}$.
- the space should be a Hilbert space $\mathcal{H}$

Reproducing kernel Hilbert space

Nice space of functions for non-parametric statistics and machine learning? Require that

- the *evaluation functionals* $f \mapsto f(x)$ be $\mathcal{C}^0$ for all $x \in \mathcal{X}$.
- the space should be a Hilbert space $\mathcal{H}$

Reproducing kernel Hilbert space

Nice space of functions for non-parametric statistics and machine learning? Require that

- the *evaluation functionals* $f \mapsto f(x)$ be $\mathcal{C}^0$ for all $x \in \mathcal{X}$.
- the space should be a Hilbert space $\mathcal{H}$

Reproducing kernel Hilbert space

Nice space of functions for non-parametric statistics and machine learning? Require that

- the *evaluation functionals* $f \mapsto f(x)$ be $\mathcal{C}^0$ for all $x \in \mathcal{X}$.
- the space should be a Hilbert space $\mathcal{H}$

Then by the Riesz representation theorem, there must exist an element $h_x \in \mathcal{H}$ such that

$$\forall f \in \mathcal{H}, \quad f(x) = \langle h_x, f \rangle_{\mathcal{H}} .$$

Reproducing kernel Hilbert space

Nice space of functions for non-parametric statistics and machine learning? Require that

- the *evaluation functionals* $f \mapsto f(x)$ be $\mathcal{C}^0$ for all $x \in \mathcal{X}$.
- the space should be a Hilbert space $\mathcal{H}$

Then by the Riesz representation theorem, there must exist an element $h_x \in \mathcal{H}$ such that

$$\forall f \in \mathcal{H}, \quad f(x) = \langle h_x, f \rangle_{\mathcal{H}}.$$

But then by definition $h_y(x) = \langle h_x, h_y \rangle_{\mathcal{H}} = h_x(y)$.

Reproducing kernel Hilbert space

Nice space of functions for non-parametric statistics and machine learning? Require that

- the *evaluation functionals* $f \mapsto f(x)$ be $\mathcal{C}^0$ for all $x \in \mathcal{X}$.
- the space should be a Hilbert space $\mathcal{H}$

Then by the Riesz representation theorem, there must exist an element $h_x \in \mathcal{H}$ such that

$$\forall f \in \mathcal{H}, \quad f(x) = \langle h_x, f \rangle_{\mathcal{H}} .$$

But then by definition $h_y(x) = \langle h_x, h_y \rangle_{\mathcal{H}} = h_x(y)$.
Define the *reproducing kernel* as the function

$$\begin{aligned} K : \mathcal{X} \times \mathcal{X} &\to \mathbb{R} \\ (x, y) &\mapsto \langle h_x, h_y \rangle_{\mathcal{H}} . \end{aligned}$$

Reproducing kernel Hilbert space

Nice space of functions for non-parametric statistics and machine learning? Require that

- the *evaluation functionals* $f \mapsto f(x)$ be $\mathcal{C}^0$ for all $x \in \mathcal{X}$.
- the space should be a Hilbert space $\mathcal{H}$

Then by the Riesz representation theorem, there must exist an element $h_x \in \mathcal{H}$ such that

$$\forall f \in \mathcal{H}, \quad f(x) = \langle h_x, f \rangle_{\mathcal{H}} .$$

But then by definition $h_y(x) = \langle h_x, h_y \rangle_{\mathcal{H}} = h_x(y)$.
Define the *reproducing kernel* as the function

$$\begin{aligned} K : \mathcal{X} \times \mathcal{X} &\to \mathbb{R} \\ (x, y) &\mapsto \langle h_x, h_y \rangle_{\mathcal{H}} . \end{aligned}$$

By definition $h_x(\cdot) = K(x, \cdot)$

Reproducing kernel Hilbert space

Nice space of functions for non-parametric statistics and machine learning? Require that

- the *evaluation functionals* $f \mapsto f(x)$ be $\mathcal{C}^0$ for all $x \in \mathcal{X}$.
- the space should be a Hilbert space $\mathcal{H}$

Then by the Riesz representation theorem, there must exist an element $h_x \in \mathcal{H}$ such that

$$\forall f \in \mathcal{H}, \quad f(x) = \langle h_x, f \rangle_{\mathcal{H}}.$$

But then by definition $h_y(x) = \langle h_x, h_y \rangle_{\mathcal{H}} = h_x(y)$.
Define the *reproducing kernel* as the function

$$\begin{aligned} K : \mathcal{X} \times \mathcal{X} &\to \mathbb{R} \\ (x, y) &\mapsto \langle h_x, h_y \rangle_{\mathcal{H}}. \end{aligned}$$

By definition $h_x(\cdot) = K(x, \cdot)$ so that

$$f(x) = \langle K(x,\cdot), f \rangle_{\mathcal{H}} \quad \text{and} \quad \langle K(x,\cdot), K(y,\cdot) \rangle_{\mathcal{H}} = K(x, y).$$

Reproducing kernel Hilbert space

Nice space of functions for non-parametric statistics and machine learning? Require that

- the *evaluation functionals* $f \mapsto f(x)$ be $\mathcal{C}^0$ for all $x \in \mathcal{X}$.
- the space should be a Hilbert space $\mathcal{H}$

Then by the Riesz representation theorem, there must exist an element $h_x \in \mathcal{H}$ such that

$$\forall f \in \mathcal{H}, \quad f(x) = \langle h_x, f \rangle_{\mathcal{H}} .$$

But then by definition $h_y(x) = \langle h_x, h_y \rangle_{\mathcal{H}} = h_x(y)$.
Define the *reproducing kernel* as the function

$$\begin{aligned} K : \mathcal{X} \times \mathcal{X} &\to \mathbb{R} \\ (x, y) &\mapsto \langle h_x, h_y \rangle_{\mathcal{H}} . \end{aligned}$$

By definition $h_x(\cdot) = K(x, \cdot)$ so that

$$f(x) = \langle K(x, \cdot), f \rangle_{\mathcal{H}} \quad \text{and} \quad \langle K(x, \cdot), K(y, \cdot) \rangle_{\mathcal{H}} = K(x, y).$$

A space with these properties is called a *reproducing kernel Hilbert space* (RKHS).

Positive definite function

Definition (Positive definite function)

A symmetric positive definite function is a function $K : (x, y) \mapsto K(x, y)$ such that for all $x_1, \dots, x_n \in \mathcal{X}$ and $\alpha_1, \dots, \alpha_n \in \mathbb{R}$,

$$\sum_{1 \le i,j \le n} \alpha_i \alpha_j K(x_i, x_j) \ge 0.$$

A reproducing kernel is a positive definite function

Proposition

A reproducing kernel is a positive definite function.

A reproducing kernel is a positive definite function

Proposition

A reproducing kernel is a positive definite function.

Proof of the claim The reproducing kernel is necessarily a *symmetric positive definite function* since for all $x_1, \ldots, x_n \in \mathcal{X}$, and all $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$.

$$\sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) = \left\langle \sum_i \alpha_i K(x_i, \cdot), \sum_j \alpha_j K(x_j, \cdot) \right\rangle_{\mathcal{H}} \geq 0,$$

with equality if and only if $\alpha_i = 0$ for all i.

A reproducing kernel is a positive definite function

Proposition

A reproducing kernel is a positive definite function.

Proof of the claim The reproducing kernel is necessarily a *symmetric positive definite function* since for all $x_1, \ldots, x_n \in \mathcal{X}$, and all $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$.

$$\sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) = \left\langle \sum_i \alpha_i K(x_i, \cdot), \sum_j \alpha_j K(x_j, \cdot) \right\rangle_{\mathcal{H}} \geq 0,$$

with equality if and only if $\alpha_i = 0$ for all i.

A reproducing kernel is a positive definite function

Proposition

A reproducing kernel is a positive definite function.

Proof of the claim The reproducing kernel is necessarily a *symmetric positive definite function* since for all $x_1, \ldots, x_n \in \mathcal{X}$, and all $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$.

$$\sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) = \left\langle \sum_i \alpha_i K(x_i, \cdot), \sum_j \alpha_j K(x_j, \cdot) \right\rangle_{\mathcal{H}} \geq 0,$$

with equality if and only if $\alpha_i = 0$ for all i.

Converse?

A reproducing kernel is a positive definite function

Proposition

A reproducing kernel is a positive definite function.

Proof of the claim The reproducing kernel is necessarily a *symmetric positive definite function* since for all $x_1, \ldots, x_n \in \mathcal{X}$, and all $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$.

$$\sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) = \left\langle \sum_i \alpha_i K(x_i, \cdot), \sum_j \alpha_j K(x_j, \cdot) \right\rangle_{\mathcal{H}} \geq 0,$$

with equality if and only if $\alpha_i = 0$ for all i.

Converse?

Yes, any symmetric positive definite function is the reproducing kernel of a RKHS (Aronszajn, 1950).

Moore-Aronszajn theorem

Theorem

A symmetric function K on $\mathcal{X}$ is positive definite if and only if there exists a Hilbert space $\mathcal{H}$ and a mapping

$$\begin{aligned} \phi : \mathcal{X} &\to \mathcal{H} \\ x &\mapsto \phi(x) \end{aligned}$$

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

Moore-Aronszajn theorem

Theorem

A symmetric function K on $\mathcal{X}$ is positive definite if and only if there exists a Hilbert space $\mathcal{H}$ and a mapping

$$\begin{aligned} \phi : \mathcal{X} &\to \mathcal{H} \\ x &\mapsto \phi(x) \end{aligned}$$

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

When we work with kernels, we therefore always use a **feature map** but very often *implicitly*. We will not show this theorem in this course.

Common RKHSes for $\mathcal{X} = \mathbb{R}^p$

Linear kernel

- $K(x, y) = x^\top y$
- $\mathcal{H} = \{f_w : x \mapsto w^\top x \mid w \in \mathbb{R}^p\}$
- $\|f_w\|_\mathcal{H} = \|w\|_2$

Common RKHSes for $\mathcal{X} = \mathbb{R}^p$

Linear kernel

- $K(x, y) = x^\top y$
- $\mathcal{H} = \{f_w : x \mapsto w^\top x \mid w \in \mathbb{R}^p\}$
- $\|f_w\|_\mathcal{H} = \|w\|_2$

Polynomial kernel

- $K_h(x, y) = (\gamma + x^\top y)^d$
- $\mathcal{H}$

Common RKHSes for $\mathcal{X} = \mathbb{R}^p$

Linear kernel

- $K(x, y) = x^\top y$
- $\mathcal{H} = \{f_w : x \mapsto w^\top x \mid w \in \mathbb{R}^p\}$
- $\|f_w\|_{\mathcal{H}} = \|w\|_2$

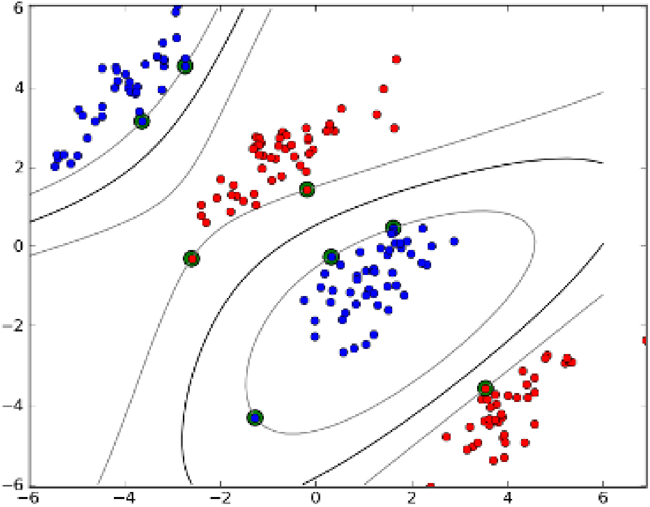
Polynomial kernel

- $K_h(x, y) = (\gamma + x^\top y)^d$
- $\mathcal{H}$

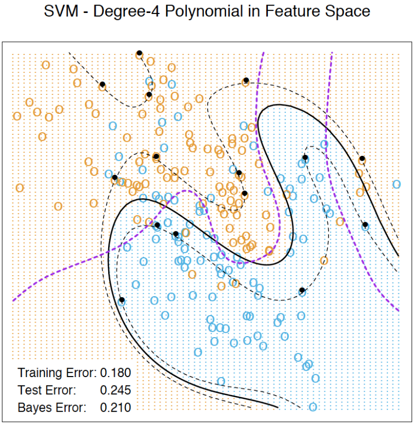
Radial Basis Function kernel (RBF)

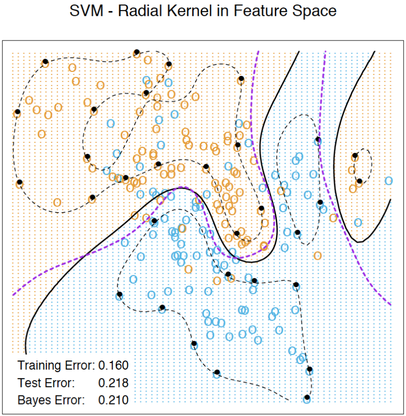
- $K_h(x, y) = \exp\left(-\frac{\|x-y\|_2^2}{2h}\right)$
- $\mathcal{H} =$ Gaussian RKHS

Nonlinear SVM : Hard margin



Nonlinear SVM: Soft margin





$\|f\|_{\mathcal{H}}$ measures the smoothness of the function f

Indeed:

$$|f(x) - f(x')| = |\left\langle f, K(x,\cdot) - K(x',\cdot)\right\rangle_{\mathcal{H}}| \leq \|f\|_{\mathcal{H}} \|K(x,\cdot) - K(x',\cdot)\|_{\mathcal{H}}$$

$\|f\|_{\mathcal{H}}$ measures the smoothness of the function f

Indeed:

$$|f(x) - f(x')| = |\left\langle f, K(x,\cdot) - K(x',\cdot)\right\rangle_{\mathcal{H}}| \leq \|f\|_{\mathcal{H}}\|K(x,\cdot) - K(x',\cdot)\|_{\mathcal{H}}$$

- f is Lipschitz with respect to the ℓ_2 distance induced by the RKHS

  $$d(x,x') = \|K(x,\cdot) - K(x',\cdot)\|_{\mathcal{H}} = \sqrt{K(x,x) + K(x',x') - 2K(x,x')}$$

- $\|f\|_{\mathcal{H}}$ is the Lipschitz constant

Some data do not live in a vector space...

- Sequence of human hemoglobin subunit gamma-1 (HGB1)

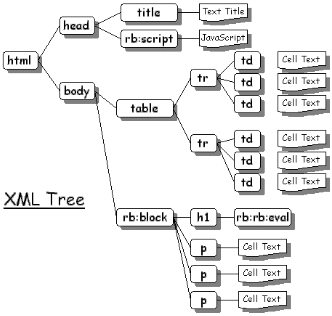
`MGHFTEEDKATITSLWGKVNVEDAGGETLGRLLVVYPWTQRFFDSFGNLSSAS...`

Some data do not live in a vector space...

- Sequence of human hemoglobin subunit gamma-1 (HGB1)

MGHFTEEDKATITSLWGKVNVEDAGGETLGRLLVVYPWTQRFFDSFGNLSSAS. . .

Xml tree of a webpage



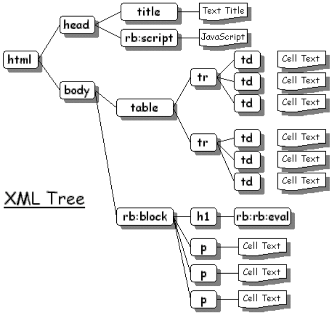
Graph structure of a molecule

Some data do not live in a vector space...

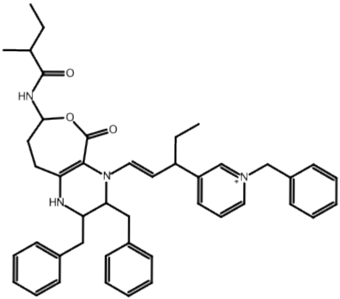
- Sequence of human hemoglobin subunit gamma-1 (HGB1)

MGHFTEEDKATITSLWGKVNVEDAGGETLGRLLVVYPWTQRFFDSFGNLSSAS...

Xml tree of a webpage



Graph structure of a molecule

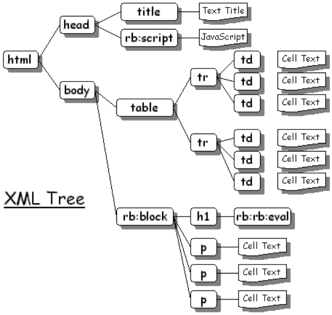


Some data do not live in a vector space...

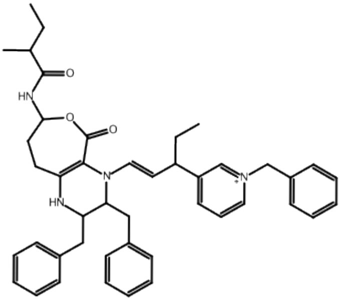
- Sequence of human hemoglobin subunit gamma-1 (HGB1)

`MGHFTEEDKATITSLWGKVNVEDAGGETLGRLLVVYPWTQRFFDSFGNLSSAS...`

Xml tree of a webpage



Graph structure of a molecule



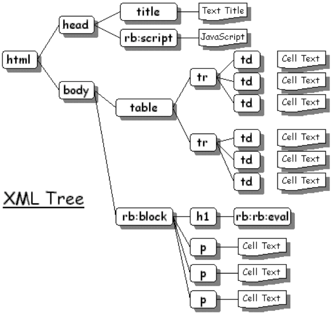
Can we learn functions of these?

Some data do not live in a vector space...

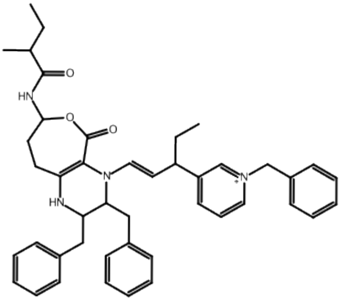
- Sequence of human hemoglobin subunit gamma-1 (HGB1)

`MGHFTEEDKATITSLWGKVNVEDAGGETLGRLLVVYPWTQRFFDSFGNLSSAS...`

Xml tree of a webpage



Graph structure of a molecule



Can we learn functions of these? $\rightarrow$ Kernels for combinatorial objects

Working with strings

- Let Σ be an alphabet of symbols or letters (e.g. $\{\texttt{A},\texttt{C},\texttt{G},\texttt{T}\}$)

Working with strings

- Let Σ be an alphabet of symbols or letters (e.g. $\{\texttt{A},\texttt{C},\texttt{G},\texttt{T}\}$)
- A sequence of letters is called a *word* or a *string*

Working with strings

- Let Σ be an alphabet of symbols or letters (e.g. $\{\texttt{A},\texttt{C},\texttt{G},\texttt{T}\}$)
- A sequence of letters is called a *word* or a *string*
- $u = u_1 \dots u_n$ a string (e.g. u=$\texttt{ATA}$)

Working with strings

- Let Σ be an alphabet of symbols or letters (e.g. $\{\texttt{A},\texttt{C},\texttt{G},\texttt{T}\}$)
- A sequence of letters is called a *word* or a *string*
- $u = u_1 \ldots u_n$ a string (e.g. u=ATA)
- $|u| = n$ is the length of the string

Working with strings

- Let Σ be an alphabet of symbols or letters (e.g. $\{\texttt{A},\texttt{C},\texttt{G},\texttt{T}\}$)
- A sequence of letters is called a *word* or a *string*
- $u = u_1 \dots u_n$ a string (e.g. u=ATA)
- $|u| = n$ is the length of the string
- $u_{1:k} = u_1 \dots u_k$ is prefix of u

Working with strings

- Let Σ be an alphabet of symbols or letters (e.g. $\{\texttt{A},\texttt{C},\texttt{G},\texttt{T}\}$)
- A sequence of letters is called a *word* or a *string*
- $u = u_1 \ldots u_n$ a string (e.g. u=ATA)
- $|u| = n$ is the length of the string
- $u_{1:k} = u_1 \ldots u_k$ is prefix of u
- $u_{k:n} = u_k \ldots u_n$ is a suffix of u

Working with strings

- Let Σ be an alphabet of symbols or letters (e.g. $\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}$)
- A sequence of letters is called a *word* or a *string*
- $u = u_1 \ldots u_n$ a string (e.g. u=$\mathtt{ATA}$)
- $|u| = n$ is the length of the string
- $u_{1:k} = u_1 \ldots u_k$ is prefix of u
- $u_{k:n} = u_k \ldots u_n$ is a suffix of u
- $uv = u_1 \ldots u_n v_1 \ldots v_m$ is the concatenation of u and v.

Working with strings

- Let Σ be an alphabet of symbols or letters (e.g. $\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}$)
- A sequence of letters is called a *word* or a *string*
- $u = u_1 \dots u_n$ a string (e.g. u=ATA)
- $|u| = n$ is the length of the string
- $u_{1:k} = u_1 \dots u_k$ is prefix of u
- $u_{k:n} = u_k \dots u_n$ is a suffix of u
- $uv = u_1 \dots u_n v_1 \dots v_m$ is the concatenation of u and v.
- v is a substring of u if there exist words u' and u'' such that $u = u'vu''$. We will then note $v \sqsubset u$.

Working with strings

- Let Σ be an alphabet of symbols or letters (e.g. $\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}$)
- A sequence of letters is called a *word* or a *string*
- $u = u_1 \dots u_n$ a string (e.g. u=$\mathtt{ATA}$)
- $|u| = n$ is the length of the string
- $u_{1:k} = u_1 \dots u_k$ is prefix of u
- $u_{k:n} = u_k \dots u_n$ is a suffix of u
- $uv = u_1 \dots u_n v_1 \dots v_m$ is the concatenation of u and v.
- v is a substring of u if there exist words u' and u'' such that $u = u'vu''$. We will then note $v \sqsubset u$.
- v is a subsequence of u, if there exist a sorted index set I such that $v = u_I$. For example $v = u_1 u_3 u_4 u_7$ is a subsequence of u since for $I = \{1, 3, 4, 7\}$ this subsequence can be written $v = u_I$.

Working with strings

- Let Σ be an alphabet of symbols or letters (e.g. $\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}$)
- A sequence of letters is called a *word* or a *string*
- $u = u_1 \ldots u_n$ a string (e.g. u=ATA)
- $|u| = n$ is the length of the string
- $u_{1:k} = u_1 \ldots u_k$ is prefix of u
- $u_{k:n} = u_k \ldots u_n$ is a suffix of u
- $uv = u_1 \ldots u_n v_1 \ldots v_m$ is the concatenation of u and v.
- v is a substring of u if there exist words u' and u'' such that $u = u'vu''$. We will then note $v \sqsubset u$.
- v is a subsequence of u, if there exist a sorted index set I such that $v = u_I$. For example $v = u_1 u_3 u_4 u_7$ is a subsequence of u since for $I = \{1, 3, 4, 7\}$ this subsequence can be written $v = u_I$.
- ε is the empty string and so $u = \varepsilon u = u\varepsilon$

Kernel for strings: p-spectrum kernel

Idea: a word is represented by the list of substrings of length p. For example the representation of `GAGA` for the 2-spectrum kernel on $\{$`A`,`C`,`G`$\}$ is

AA	AC	AG	CA	CC	CG	GA	GC	GG
(0,	0,	1,	0,	0,	0,	2,	0,	0)

Kernel for strings: p-spectrum kernel

Idea: a word is represented by the list of substrings of length p. For example the representation of `GAGA` for the 2-spectrum kernel on {`A`,`C`,`G`} is

AA	AC	AG	CA	CC	CG	GA	GC	GG
(0,	0,	1,	0,	0,	0,	2,	0,	0)

The **feature map** for a string s is

$$\phi(s) = \left(\phi_u(s)\right)_{u \in \Sigma^p} \qquad \text{with} \qquad \phi_u(s) = \#\{i \mid s_{i:(i+p-1)} = u\}$$

Kernel for strings: p-spectrum kernel

Idea: a word is represented by the list of substrings of length p. For example the representation of `GAGA` for the 2-spectrum kernel on $\{$`A,C,G`$\}$ is

AA	AC	AG	CA	CC	CG	GA	GC	GG
(0,	0,	1,	0,	0,	0,	2,	0,	0)

The **feature map** for a string s is

$$\phi(s) = \left(\phi_u(s)\right)_{u \in \Sigma^p} \qquad \text{with} \qquad \phi_u(s) = \#\{i \mid s_{i:(i+p-1)} = u\}$$

The **kernel** is

$$K(s,t) = \sum_{u \in \Sigma^p} \phi_u(s)\, \phi_u(t).$$

String kernels: other spectrum kernels

Blended spectrum kernel

$\tilde{K}_p(s,t) = \sum_{j=1}^{p} a_j K_i(s,t)$ with K_j the usual j-spectrum kernel.

String kernels: other spectrum kernels

Blended spectrum kernel

$\tilde{K}_p(s,t) = \sum_{j=1}^{p} a_j K_i(s,t)$ with K_j the usual j-spectrum kernel.

Mismatch kernel

Like the spectrum kernel but allowing mistakes...

$$\phi_u^{p,m}(s) = \#\{v \mid v \sqsubset s,\ |v| = |u|,\ d_H(u,v) \leq m\}.$$

with $d_H(u,v) = \sum_{k=1}^{n} 1_{\{u_i \neq v_i\}}$ the Hamming distance between u and v

String kernels: other spectrum kernels

Blended spectrum kernel

$\tilde{K}_p(s,t) = \sum_{j=1}^{p} a_j K_i(s,t)$ with K_j the usual j-spectrum kernel.

Mismatch kernel

Like the spectrum kernel but allowing mistakes...

$$\phi_u^{p,m}(s) = \#\{v \mid v \sqsubset s,\ |v| = |u|,\ d_H(u,v) \leq m\}.$$

with $d_H(u,v) = \sum_{k=1}^{n} 1_{\{u_i \neq v_i\}}$ the Hamming distance between u and v

String kernels: subsequence kernels

Denote $\mathcal{I}_n = \{1, \ldots, n\}$

Feature map:

$$\phi_u(s) = \#\left\{ I \subset \mathcal{I}_{|s|} \mid u = s_I \right\}$$

String kernels: subsequence kernels

Denote $\mathcal{I}_n = \{1, \ldots, n\}$

Feature map:

$$\phi_u(s) = \#\big\{ I \subset \mathcal{I}_{|s|} \mid u = s_I \big\}$$

Kernel:

$$\begin{aligned} K(s,t) &= \sum_{u \in \Sigma^*} \phi_u(s)\, \phi_u(t) \\ &= \sum_{(I,J)} 1_{\{s_I = t_J\}} \\ &= \#\big\{ (I,J) \mid s_I = t_J \big\} \end{aligned}$$

- The empty substring ε is counted only once in each string.

Subsequence kernels: dynamic programming

$$K(sa, t) = K(s, t) + \sum_{k:t_k=a} K(s, t_{1:k-1})$$

Subsequence kernels: dynamic programming

$$K(sa, t) = K(s, t) + \sum_{k : t_k = a} K(s, t_{1:k-1})$$

So that, if we denote $\kappa_{ij} := K(s_{1:i}, t_{1:j})$, the recursion becomes

$$\kappa_{i,j} = \kappa_{i-1,j} + \sum_{k=1}^{j} 1_{\{t_k = s_i\}} \, \kappa_{i-1,k-1}$$

Subsequence kernels: dynamic programming

$$K(sa, t) = K(s, t) + \sum_{k:t_k=a} K(s, t_{1:k-1})$$

So that, if we denote $\kappa_{ij} := K(s_{1:i}, t_{1:j})$, the recursion becomes

$$\kappa_{i,j} = \kappa_{i-1,j} + \sum_{k=1}^{j} 1_{\{t_k=s_i\}} \kappa_{i-1,k-1}$$

	ε	t_1	$\dots$	t_j	$\dots$
ε	1	1	$\dots$	1	$\dots$
s_1	1	$\kappa_{1,1}$	$\dots$	$\kappa_{1,j}$	$\dots$
s_2					
$\vdots$					
s_{i-1}	1	$\kappa_{i-1,1}$	$\dots$	$\kappa_{i-1,j}$	
s_i	1	$\kappa_{i,1}$	$\dots$	$\kappa_{i,j}$ $\vdots$	
$\vdots$					

Other types of kernels

- Fisher kernels
- Tree kernels
- Graph kernels
- Dedicated kernels for genomics/proteomics
- Set kernels

and more

Kernel combinations

Assume K, K_1 and K_2 are positive definite functions, then the following are still p.d. kernel functions:

Kernel combinations

Assume K, K_1 and K_2 are positive definite functions, then the following are still p.d. kernel functions:

Sum of kernels: For $\alpha_1, \alpha_2 > 0$, $\tilde{K}(x, y) = \alpha_1 K_1(x, y) + \alpha_2 K(x, y)$

Kernel combinations

Assume K, K_1 and K_2 are positive definite functions,
then the following are still p.d. kernel functions:

Sum of kernels: For $\alpha_1, \alpha_2 > 0$, $\tilde{K}(x, y) = \alpha_1 K_1(x, y) + \alpha_2 K(x, y)$

Limits of kernels: $K(x, y) = \lim_{n\to\infty} K_n(x, y)$

Kernel combinations

Assume K, K_1 and K_2 are positive definite functions, then the following are still p.d. kernel functions:

Sum of kernels:	For $\alpha_1, \alpha_2 > 0$, $\tilde{K}(x, y) = \alpha_1 K_1(x, y) + \alpha_2 K(x, y)$
Limits of kernels:	$K(x, y) = \lim_{n \to \infty} K_n(x, y)$
Pointwise product:	$\tilde{K}(x, y) = K_1(x, y)\, K_2(x, y)$

Kernel combinations

Assume K, K_1 and K_2 are positive definite functions, then the following are still p.d. kernel functions:

Sum of kernels:	For $\alpha_1, \alpha_2 > 0$, $\tilde{K}(x, y) = \alpha_1 K_1(x, y) + \alpha_2 K(x, y)$
Limits of kernels:	$K(x, y) = \lim_{n \to \infty} K_n(x, y)$
Pointwise product:	$\tilde{K}(x, y) = K_1(x, y) \, K_2(x, y)$
Pairwise kernel:	$\tilde{K}(x, y) = \sum_{z \in \mathcal{Z}} K(x, z) K(z, y)$

Kernel combinations

Assume K, K_1 and K_2 are positive definite functions, then the following are still p.d. kernel functions:

Sum of kernels: For $\alpha_1, \alpha_2 > 0$, $\tilde{K}(x,y) = \alpha_1 K_1(x,y) + \alpha_2 K(x,y)$

Limits of kernels: $K(x,y) = \lim_{n\to\infty} K_n(x,y)$

Pointwise product: $\tilde{K}(x,y) = K_1(x,y)\,K_2(x,y)$

Pairwise kernel: $\tilde{K}(x,y) = \sum_{z\in\mathcal{Z}} K(x,z)K(z,y)$

Normalized kernel: $\tilde{K}(x,y) = \frac{K(x,y)}{\sqrt{K(x,x)K(y,y)}} = \cos\angle(\phi(x),\phi(y))$

Kernel combinations

Assume K, K_1 and K_2 are positive definite functions, then the following are still p.d. kernel functions:

Sum of kernels: For $\alpha_1, \alpha_2 > 0$, $\tilde{K}(x,y) = \alpha_1 K_1(x,y) + \alpha_2 K(x,y)$

Limits of kernels: $K(x,y) = \lim_{n\to\infty} K_n(x,y)$

Pointwise product: $\tilde{K}(x,y) = K_1(x,y)\, K_2(x,y)$

Pairwise kernel: $\tilde{K}(x,y) = \sum_{z\in\mathcal{Z}} K(x,z)K(z,y)$

Normalized kernel: $\tilde{K}(x,y) = \dfrac{K(x,y)}{\sqrt{K(x,x)K(y,y)}} = \cos\angle(\phi(x), \phi(y))$

In terms of kernel matrices

Pointwise product: $\tilde{\mathbf{K}} = \mathbf{K}_1 \odot \mathbf{K}_2$ (Hadamard product)

Kernel combinations

Assume K, K_1 and K_2 are positive definite functions, then the following are still p.d. kernel functions:

Sum of kernels: For $\alpha_1, \alpha_2 > 0$, $\tilde{K}(x,y) = \alpha_1 K_1(x,y) + \alpha_2 K(x,y)$

Limits of kernels: $K(x,y) = \lim_{n\to\infty} K_n(x,y)$

Pointwise product: $\tilde{K}(x,y) = K_1(x,y)\, K_2(x,y)$

Pairwise kernel: $\tilde{K}(x,y) = \sum_{z\in\mathcal{Z}} K(x,z)K(z,y)$

Normalized kernel: $\tilde{K}(x,y) = \dfrac{K(x,y)}{\sqrt{K(x,x)K(y,y)}} = \cos\angle(\phi(x),\phi(y))$

In terms of kernel matrices

Pointwise product: $\tilde{\mathbf{K}} = \mathbf{K}_1 \odot \mathbf{K}_2$ (Hadamard product)

Pairwise kernel: $\tilde{\mathbf{K}} = \mathbf{K}^2$ (Matrix product)

Representer theorem

Theorem (Kimmeldorf and Wahba, 1971)

Consider the optimization problem

$$\min_{f \in \mathcal{H}} L(f(x_1), \ldots, f(x_n)) + \lambda \|f\|_{\mathcal{H}}^2$$

Then any local minimum is of the form

$$f = \sum_{i=1}^{n} \alpha_i K(x_i, \cdot),$$

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

Representer theorem

Theorem (Kimmeldorf and Wahba, 1971)

Consider the optimization problem

$$\min_{f \in \mathcal{H}} L(f(x_1), \dots, f(x_n)) + \lambda \|f\|_{\mathcal{H}}^2$$

Then any local minimum is of the form

$$f = \sum_{i=1}^{n} \alpha_i K(x_i, \cdot),$$

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

Proof Indeed, let f be a local optimum and consider the subspace

$$\mathcal{S} = \{g \mid g = \sum_{i=1}^{n} \alpha_i K(x_i, \cdot), \quad \boldsymbol{\alpha} \in \mathbb{R}^n\}.$$

Representer theorem

We can decompose $f = f_{//} + f_{\perp}$ with $f_{//} = \mathrm{Proj}_{\mathcal{S}}(f)$.

Representer theorem

We can decompose $f = f_{//} + f_{\perp}$ with $f_{//} = \text{Proj}_{\mathcal{S}}(f)$. We then have

$$f_{\perp}(x_i) = \langle f_{\perp}, K(x_i, \cdot)\rangle_{\mathcal{H}} = 0 \quad \text{and} \quad \langle f_{\perp}, f_{//}\rangle_{\mathcal{H}} = 0.$$

Representer theorem

We can decompose $f = f_{//} + f_{\perp}$ with $f_{//} = \text{Proj}_{\mathcal{S}}(f)$. We then have

$$f_{\perp}(x_i) = \langle f_{\perp}, K(x_i, \cdot)\rangle_{\mathcal{H}} = 0 \quad \text{and} \quad \langle f_{\perp}, f_{//}\rangle_{\mathcal{H}} = 0.$$

Thus

$$L(f(x_1), \ldots, f(x_n)) + \lambda \|f\|^2_{\mathcal{H}}$$

Representer theorem

We can decompose $f = f_{//} + f_{\perp}$ with $f_{//} = \text{Proj}_{\mathcal{S}}(f)$. We then have

$$f_{\perp}(x_i) = \langle f_{\perp}, K(x_i, \cdot) \rangle_{\mathcal{H}} = 0 \quad \text{and} \quad \langle f_{\perp}, f_{//} \rangle_{\mathcal{H}} = 0.$$

Thus

$$\begin{aligned} & L(f(x_1), \ldots, f(x_n)) + \lambda \|f\|_{\mathcal{H}}^2 \\ = \quad & L(f_{//}(x_1), \ldots, f_{//}(x_n)) + \lambda \left(\|f_{//}\|_{\mathcal{H}}^2 + 2\langle f_{\perp}, f_{//} \rangle_{\mathcal{H}} + \|f_{\perp}\|_{\mathcal{H}}^2 \right) \end{aligned}$$

Representer theorem

We can decompose $f = f_{//} + f_{\perp}$ with $f_{//} = \mathrm{Proj}_{\mathcal{S}}(f)$. We then have

$$f_{\perp}(x_i) = \langle f_{\perp}, K(x_i, \cdot) \rangle_{\mathcal{H}} = 0 \quad \text{and} \quad \langle f_{\perp}, f_{//} \rangle_{\mathcal{H}} = 0.$$

Thus

$$\begin{aligned}
& L(f(x_1), \ldots, f(x_n)) + \lambda \|f\|_{\mathcal{H}}^2 \\
= \quad & L(f_{//}(x_1), \ldots, f_{//}(x_n)) + \lambda \left(\|f_{//}\|_{\mathcal{H}}^2 + 2\langle f_{\perp}, f_{//} \rangle_{\mathcal{H}} + \|f_{\perp}\|_{\mathcal{H}}^2 \right) \\
= \quad & L(f_{//}(x_1), \ldots, f_{//}(x_n)) + \lambda \, \|f_{//}\|_{\mathcal{H}}^2 + \lambda \, \|f_{\perp}\|_{\mathcal{H}}^2
\end{aligned}$$

Representer theorem

We can decompose $f = f_{//} + f_{\perp}$ with $f_{//} = \text{Proj}_{\mathcal{S}}(f)$. We then have

$$f_{\perp}(x_i) = \langle f_{\perp}, K(x_i, \cdot) \rangle_{\mathcal{H}} = 0 \quad \text{and} \quad \langle f_{\perp}, f_{//} \rangle_{\mathcal{H}} = 0.$$

Thus

$$\begin{aligned} & L(f(x_1), \ldots, f(x_n)) + \lambda \|f\|_{\mathcal{H}}^2 \\ = \quad & L(f_{//}(x_1), \ldots, f_{//}(x_n)) + \lambda \left(\|f_{//}\|_{\mathcal{H}}^2 + 2\langle f_{\perp}, f_{//} \rangle_{\mathcal{H}} + \|f_{\perp}\|_{\mathcal{H}}^2 \right) \\ = \quad & L(f_{//}(x_1), \ldots, f_{//}(x_n)) + \lambda \, \|f_{//}\|_{\mathcal{H}}^2 + \lambda \, \|f_{\perp}\|_{\mathcal{H}}^2 \end{aligned}$$

So that we must have $f_{\perp} = 0$.

Learning with functions from a RKHS

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) \; + \; \lambda \, \|f\|_{\mathcal{H}}^2 \qquad \text{(P)}$$

Learning with functions from a RKHS

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) \; + \; \lambda \, \|f\|_{\mathcal{H}}^2 \qquad \text{(P)}$$

By the representer theorem, the solution of the regularized empirical risk minimization problem lies in the subspace of $\mathcal{H}$ generated by the point x_i, i.e.,

Learning with functions from a RKHS

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) \; + \; \lambda \, \|f\|_{\mathcal{H}}^2 \qquad \text{(P)}$$

By the representer theorem, the solution of the regularized empirical risk minimization problem lies in the subspace of $\mathcal{H}$ generated by the point x_i, i.e.,

Learning with functions from a RKHS

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \lambda \|f\|_{\mathcal{H}}^2 \tag{P}$$

By the representer theorem, the solution of the regularized empirical risk minimization problem lies in the subspace of $\mathcal{H}$ generated by the point x_i, i.e.,

$$f^* = \sum_{i=1}^{n} \alpha_i K(x_i, \cdot) \quad \text{for some } \alpha_i \in \mathbb{R}. \tag{R}$$

Learning with functions from a RKHS

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \lambda \|f\|_{\mathcal{H}}^2 \qquad \text{(P)}$$

By the representer theorem, the solution of the regularized empirical risk minimization problem lies in the subspace of $\mathcal{H}$ generated by the point x_i, i.e.,

$$f^* = \sum_{i=1}^{n} \alpha_i K(x_i, \cdot) \quad \text{for some } \alpha_i \in \mathbb{R}. \qquad \text{(R)}$$

The solution of (P) is therefore of the form (R) with $\alpha \in \mathbb{R}^n$ the solution of

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \ell\Big(\sum_{j=1}^{n} \alpha_j K(x_j, x_i), y_i\Big) + \lambda \sum_{1 \le i,j \le n} \alpha_i \alpha_j K(x_i, x_j).$$

Kernel ridge regression

$$\min \frac{1}{2}\sum_{i=1}^{n}(f(x_i) - y_i)_2^2 + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2$$

Kernel ridge regression

$$\min \frac{1}{2}\sum_{i=1}^{n}(f(x_i) - y_i)_2^2 + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2$$

- We could use the representer theorem and solve the optimization problem w.r.t. $\boldsymbol{\alpha}$
- We will show directly that the predictor can be expresses solely with the Gram matrix.

We know that the solution to ridge regression is

$$\widehat{\mathbf{w}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$$

A matrix identity and the matrix inversion lemma

Let $\mathbf{X} \in \mathbb{R}^{n \times p}$,

$$\mathbf{X}^\top + \mathbf{X}^\top \mathbf{X}\mathbf{X}^\top = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})\mathbf{X}^\top = \mathbf{X}^\top(\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)$$

A matrix identity and the matrix inversion lemma

Let $\mathbf{X} \in \mathbb{R}^{n \times p}$,

$$\mathbf{X}^\top + \mathbf{X}^\top \mathbf{X}\mathbf{X}^\top = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})\mathbf{X}^\top = \mathbf{X}^\top (\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)$$

$$\boxed{\mathbf{X}^\top (\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1} = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top}$$

A matrix identity and the matrix inversion lemma

Let $\mathbf{X} \in \mathbb{R}^{n \times p}$,

$$\mathbf{X}^\top + \mathbf{X}^\top \mathbf{X}\mathbf{X}^\top = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})\mathbf{X}^\top = \mathbf{X}^\top(\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)$$

$$\boxed{\mathbf{X}^\top(\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1} = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top}$$

$$\mathbf{I}_p - \mathbf{X}^\top(\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X} \quad =$$

A matrix identity and the matrix inversion lemma

Let $\mathbf{X} \in \mathbb{R}^{n \times p}$,

$$\mathbf{X}^\top + \mathbf{X}^\top \mathbf{X}\mathbf{X}^\top = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})\mathbf{X}^\top = \mathbf{X}^\top(\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)$$

$$\boxed{\mathbf{X}^\top(\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1} = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top}$$

$$\mathbf{I}_p - \mathbf{X}^\top(\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X} \quad = \quad \mathbf{I}_p - (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{X}$$

A matrix identity and the matrix inversion lemma

Let $\mathbf{X} \in \mathbb{R}^{n \times p}$,

$$\mathbf{X}^\top + \mathbf{X}^\top \mathbf{X} \mathbf{X}^\top = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})\mathbf{X}^\top = \mathbf{X}^\top (\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)$$

$$\boxed{\mathbf{X}^\top (\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1} = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top}$$

$$\begin{aligned} \mathbf{I}_p - \mathbf{X}^\top (\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1} \mathbf{X} &= \mathbf{I}_p - (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{X} \\ &= (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1} \left[(\mathbf{I}_p + \mathbf{X}^\top \mathbf{X}) - \mathbf{X}^\top \mathbf{X} \right] \end{aligned}$$

A matrix identity and the matrix inversion lemma

Let $\mathbf{X} \in \mathbb{R}^{n \times p}$,

$$\mathbf{X}^\top + \mathbf{X}^\top \mathbf{X}\mathbf{X}^\top = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})\mathbf{X}^\top = \mathbf{X}^\top(\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)$$

$$\boxed{\mathbf{X}^\top(\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1} = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top}$$

$$\begin{aligned}
\mathbf{I}_p - \mathbf{X}^\top(\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X} &= \mathbf{I}_p - (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{X} \\
&= (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1}\left[(\mathbf{I}_p + \mathbf{X}^\top \mathbf{X}) - \mathbf{X}^\top \mathbf{X}\right] \\
&= (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1}
\end{aligned}$$

A matrix identity and the matrix inversion lemma

Let $\mathbf{X} \in \mathbb{R}^{n \times p}$,

$$\mathbf{X}^\top + \mathbf{X}^\top \mathbf{X}\mathbf{X}^\top = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})\mathbf{X}^\top = \mathbf{X}^\top(\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)$$

$$\boxed{\mathbf{X}^\top(\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1} = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top}$$

$$\begin{aligned} \mathbf{I}_p - \mathbf{X}^\top(\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X} &= \mathbf{I}_p - (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{X} \\ &= (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1}\left[(\mathbf{I}_p + \mathbf{X}^\top \mathbf{X}) - \mathbf{X}^\top \mathbf{X}\right] \\ &= (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1} \end{aligned}$$

Matrix inversion lemma

$$\boxed{(\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1} = \mathbf{I}_p - \mathbf{X}^\top(\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}}$$

A matrix identity and the matrix inversion lemma

Let $\mathbf{X} \in \mathbb{R}^{n \times p}$,

$$\mathbf{X}^\top + \mathbf{X}^\top \mathbf{X}\mathbf{X}^\top = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})\mathbf{X}^\top = \mathbf{X}^\top (\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)$$

$$\boxed{\mathbf{X}^\top (\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1} = (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top}$$

$$\begin{aligned} \mathbf{I}_p - \mathbf{X}^\top (\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X} &= \mathbf{I}_p - (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{X} \\ &= (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1} \left[(\mathbf{I}_p + \mathbf{X}^\top \mathbf{X}) - \mathbf{X}^\top \mathbf{X} \right] \\ &= (\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1} \end{aligned}$$

Matrix inversion lemma

$$\boxed{(\mathbf{I}_p + \mathbf{X}^\top \mathbf{X})^{-1} = \mathbf{I}_p - \mathbf{X}^\top (\mathbf{I}_n + \mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}}$$

Computational cost reduced from $\mathcal{O}(p^3)$ to $\mathcal{O}(n^2 p)$.

Kernel ridge regression

Denoting $\mathbf{k}(\mathbf{z})$ the vector with entries $[\mathbf{k}(\mathbf{z})]_i = K(\mathbf{x}_i, \mathbf{z})$, we have

$$\mathbf{z}^\top \widehat{\mathbf{w}} = \mathbf{z}^\top (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^\top \mathbf{y}$$

Kernel ridge regression

Denoting $\mathbf{k}(\mathbf{z})$ the vector with entries $[\mathbf{k}(\mathbf{z})]_i = K(\mathbf{x}_i, \mathbf{z})$, we have

$$\begin{aligned} \mathbf{z}^\top \widehat{\mathbf{w}} &= \mathbf{z}^\top (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^\top \mathbf{y} \\ &= \mathbf{z}^\top \mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I}_n)^{-1} \mathbf{y} \end{aligned}$$

Kernel ridge regression

Denoting $\mathbf{k}(\mathbf{z})$ the vector with entries $[\mathbf{k}(\mathbf{z})]_i = K(\mathbf{x}_i, \mathbf{z})$, we have

$$\begin{aligned}
\mathbf{z}^\top \widehat{\mathbf{w}} &= \mathbf{z}^\top (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^\top \mathbf{y} \\
&= \mathbf{z}^\top \mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I}_n)^{-1} \mathbf{y} \\
&= \mathbf{k}(\mathbf{z})^\top (\lambda \mathbf{I}_n + \mathbf{K})^{-1} \mathbf{y}
\end{aligned}$$

Kernel ridge regression

Denoting $\mathbf{k}(\mathbf{z})$ the vector with entries $[\mathbf{k}(\mathbf{z})]_i = K(\mathbf{x}_i, \mathbf{z})$, we have

$$\begin{aligned}
\mathbf{z}^\top \widehat{\mathbf{w}} &= \mathbf{z}^\top (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^\top \mathbf{y} \\
&= \mathbf{z}^\top \mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I}_n)^{-1} \mathbf{y} \\
&= \mathbf{k}(\mathbf{z})^\top (\lambda \mathbf{I}_n + \mathbf{K})^{-1} \mathbf{y}
\end{aligned}$$

So we have $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i K(\mathbf{x}_i, \mathbf{x})$ with

$$\boxed{\boldsymbol{\alpha} = (\lambda \mathbf{I}_n + \mathbf{K})^{-1} y}.$$

Ressources

http://www.kernel-machines.org/

References I

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404.