Nonlinear supervised learning with kernels

Sylvain Le Corff

Kernel methods

- Widely used in machine learning.
- Extend algorithms such as SVMs to define non-linear decision boundaries.

Idea

- Implicitly defining an inner product in a high-dimensional space.
- Replacing the original inner product in the input space with positive definite kernels immediately extends algorithms such as SVMs to a non-linear separation in the input space.

Kernel methods

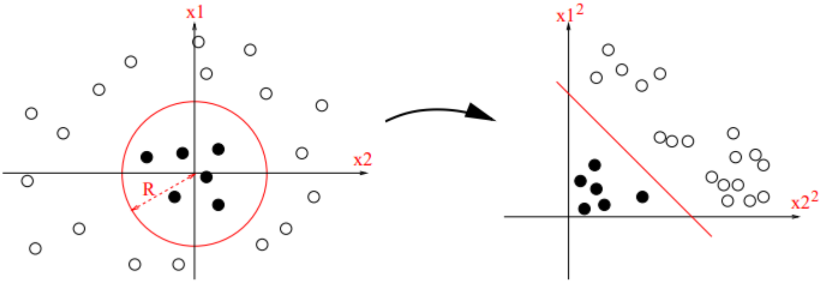
SVM

In practice, linear separation is often not possible.

Implicit lifting to a higher dimensional space

- Use more complex functions to separate the two sets
- Use a non-linear mapping φ from the input space $\mathcal{X}$ to a higher-dimensional space $\mathcal{H}$, where linear separation is possible.

Polynomial mapping

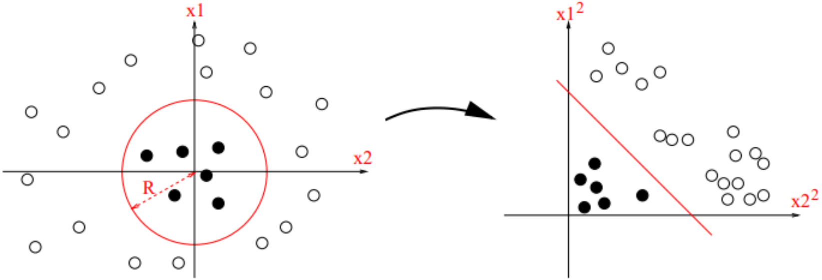


Polynomial mapping

The **polynomial** mapping $\varphi : \mathbb{R}^2 \to \mathbb{R}^3$ for $x = (x_1, x_2) \in \mathbb{R}^2$

$$\varphi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

solves the classification problem: label $Y_i = 1$ if the data point is in the circle of radius R.



Note that for $x, \tilde{x} \in \mathbb{R}^2$ we have

$$\begin{aligned}\langle \varphi(x), \varphi(x') \rangle &= x_1^2 \tilde{x}_1^2 + x_2^2 \tilde{x}_2^2 + 2 x_1 x_2 \tilde{x}_1 \tilde{x}_2 \\ &= \langle x, \tilde{x} \rangle^2 .\end{aligned}$$

Some definitions

Definition (Kernel)

A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a kernel over $\mathcal{X}$.

The idea is to define a kernel k such that

$$\forall (x, x') \in \mathcal{X} \times \mathcal{X}, \qquad k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}.$$

- for some mapping $\varphi = \mathcal{X} \to \mathcal{H}$ to a Hilbert space $\mathcal{H}$
- $\mathcal{H}$ is called a feature space

Some definitions

Definition (Kernel)

A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a kernel over $\mathcal{X}$.

The idea is to define a kernel k such that

$$\forall (x, x') \in \mathcal{X} \times \mathcal{X}, \qquad k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}.$$

- for some mapping $\varphi = \mathcal{X} \to \mathcal{H}$ to a Hilbert space $\mathcal{H}$
- $\mathcal{H}$ is called a feature space

Interpretation: k can be interpreted as a similarity measure between elements of the input space $\mathcal{X}$ (or the "raw feature" space).

Kernel trick in ML

- Many machine learning algorithms (in particular, linear SVMs) can be expressed only in terms of inner products between vectors
- Computing the explicit mappings $\varphi(x_1), \varphi(x_2)$ and their inner product $\langle \varphi(x_1), \varphi(x_2) \rangle_{\mathcal{H}}$ can be computationally expensive!
- Kernel trick: avoid the explicit mapping $\varphi(x)$ by directly computing the inner product $\langle \varphi(x_1), \varphi(x_2) \rangle_{\mathcal{H}}$ via the kernel function $k(x_1, x_2)$

Good properties of kernels

Efficiency:

- k is often significantly more efficient to compute than φ and an inner product in $\mathcal{H}$.
- in several common examples, the computation of $k(x, x')$ can be achieved in $O(\dim \mathcal{X})$ while that of $\langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$ typically requires $O(\dim(\mathcal{H}))$ work, with $\dim(\mathcal{H}) \gg N$.
- in some cases, $\dim(\mathcal{H}) = \infty$.

Flexibility:

- No need to explicitly define or compute a mapping φ
- The kernel k can be arbitrarily chosen so long as the existence of φ is guaranteed, i.e. k satisfies Mercer's condition

More definitions on kernels

Definition (Symmetry)

We say that a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is symmetric if for all $(x, x') \in \mathcal{X} \times \mathcal{X}$

$$k(x, x') = k(x', x).$$

More definitions on kernels

Definition (Symmetry)

We say that a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is symmetric if for all $(x, x') \in \mathcal{X} \times \mathcal{X}$

$$k(x, x') = k(x', x).$$

Definition (Positive Definite Symmetric (PDS) kernel)

We say that a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is Positive Definite Symmetric (PDS) if for any $\{x_1, \ldots, x_n\} \subset \mathcal{X}$ the matrix $K := (k(x_i, x_j))_{1 \leqslant i,j \leqslant n}$ is symmetric positive semidefinite (SPSD), i.e.

$$K := (k(x_i, x_j))_{1 \leqslant i,j \leqslant n} \succeq 0.$$

More definitions on kernels

Recall that K is SPSD if

- the eigenvalues of K are all non-negative,
- or, for any vector $u \in \mathbb{R}^n$

$$u^T K u = \sum_{ij} u_i u_j k(x_i, x_j) \geqslant 0$$

(with K symmetric).

For a sample $x_1, \dots, x_n$ we call $K = [K(x_i, x_j)]_{1 \leqslant i,j \leqslant n}$ the Gram matrix of this sample.

Some properties of "stability"

Definition (Hadamard product)

$A \odot B$ between two matrices A and B (or vectors) with the same dimensions is given by

$$(A \odot B)_{i,j} = A_{i,j} \odot B_{i,j}$$

Theorem

The sum, product, pointwise limit and composition with a power series $\sum_{n \geqslant 0} a_n x^n$ with $a_n \geqslant 0$ for all $n \geqslant 0$ preserves the PDS property.

Proof I

(Sum) Consider two $n \times n$ Gram matrices K, K' of PDS kernels K, K' and take $u \in \mathbb{R}^n$. Observe that

$$u^\top (K + K')u = u^\top Ku + u^\top K'u \geqslant 0$$

So PDS is preserved by the sum and finite sums by reccurence.

Proof II

(Product) Now, to prove that the product $K \odot K'$ is PDS, write $K = MM^\top$, where M is the square-root of K (which is SDP) and note that

$$\begin{aligned} u^\top (K \odot K') u &= \sum_{1 \leqslant i,j \leqslant n} u_i u_j K_{i,j} K'_{i,j} \\ &= \sum_{1 \leqslant i,j \leqslant n} \sum_{k=1}^{n} u_i u_j M_{i,k} M_{k,j} K'_{i,j} \\ &= \sum_{k=1}^{n} z_k^\top K' z_k \geqslant 0 \end{aligned}$$

with $z_k = u \odot M_{\bullet,k}$. This proves that finite products of PDS kernels is PDS.

Proof III

(Pointwise limit) Assume that $K_\ell \to K$ as $\ell \to +\infty$ pointwise, where K_ℓ is a sequence of PDS kernels.
It means that any associated sequence of Gram matrices K_ℓ and the its limit K satisfies $K_\ell \to K$ entrywise, so that for any $u \in \mathbb{R}^n$ we have

$$u^\top K_\ell u \to u^\top K u$$

so $u^\top K u \geqslant 0$ since $u^\top K_\ell u \to u$ for all ℓ. This proves stability of PDS property under pointwise limit.

(Composition w/ a power series) Now, let K be a kernel such that $|K(x, x')| < r$ for all $x, x' \in \mathcal{X}$ and $\sum_{\ell \geqslant 0} a_\ell x^\ell$ a power series with radius of convergence r.
By stability under sum and product, we have that

$$\sum_{\ell=0}^{L} a_\ell K^\ell$$

Proof IV

is PDS, and

$$\lim_{L \to +\infty} \sum_{\ell=0}^{L} a_\ell K^\ell = \sum_{\ell \geqslant 0} a_\ell K^\ell$$

remains PDS since PDS is kept under pointwise limit. This concludes the proof of the theorem.

More properties on kernels

Theorem (Cauchy-Schwarz)

The following inequality holds for k, k' two PDS kernels

$$k(x, x')^2 \leqslant k(x, x)k(x', x')$$

for any $x, x' \in \mathcal{X}$.

It is called the *Cauchy-Schwarz inequality* for PSD kernels.

Proof

Take $x, x' \in \mathcal{X}$ and consider the Gram matrix

$$G = \begin{bmatrix} k(x,x) & k(x,x') \\ k(x',x) & k(x',x') \end{bmatrix}.$$

Since k is PDS, then $G \succcurlyeq 0$, which entails that

$$0 \leqslant \det G = k(x,x)k(x',x') - k(x,x')^2.$$

Sufficient condition for the reproducing property

Theorem (Reproducing Kernel Hilbert Space (RKHS))

Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a PDS kernel. Then, there is a Hilbert space $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and a mapping $\varphi : \mathcal{X} \to \mathcal{H}$ such that

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

and such that the ***reproducing property*** *holds:*

$$h(x) = \langle h, k(x, \cdot) \rangle_{\mathcal{H}}$$

for any $h \in \mathcal{H}$ and $x \in \mathcal{X}$.

Sufficient condition for the reproducing property

Theorem (Reproducing Kernel Hilbert Space (RKHS))

Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a PDS kernel. Then, there is a Hilbert space $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and a mapping $\varphi : \mathcal{X} \to \mathcal{H}$ such that

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

*and such that the **reproducing property** holds:*

$$h(x) = \langle h, k(x, \cdot) \rangle_{\mathcal{H}}$$

for any $h \in \mathcal{H}$ and $x \in \mathcal{X}$.

We say that $\mathcal{H}$ is a reproducting kernel Hilbert space associated to the kernel k.

Some vocabulary

- Note that

$$\text{RKHS} \Rightarrow \text{Hilbert space}, \qquad \text{BUT} \qquad \text{Hilbert space} \not\Rightarrow \text{RKHS}$$

Some vocabulary

- Note that

  RKHS $\Rightarrow$ Hilbert space, BUT Hilbert space $\not\Rightarrow$ RKHS
- The Hilbert space $\mathcal{H}$ is called the **features space** associated to k

Some vocabulary

- Note that

  $$\text{RKHS} \Rightarrow \text{Hilbert space}, \qquad \text{BUT} \qquad \text{Hilbert space} \not\Rightarrow \text{RKHS}$$
- The Hilbert space $\mathcal{H}$ is called the **features space** associated to k
- The corresponding mapping $\varphi : \mathcal{X} \to \mathcal{H}$ is called the **features mapping**

Some vocabulary

- Note that

  $$\text{RKHS} \Rightarrow \text{Hilbert space}, \qquad \text{BUT} \qquad \text{Hilbert space} \not\Rightarrow \text{RKHS}$$

- The Hilbert space $\mathcal{H}$ is called the **features space** associated to k
- The corresponding mapping $\varphi : \mathcal{X} \to \mathcal{H}$ is called the **features mapping**
- $\mathcal{H}$ is endowed with an inner product $\langle h, h' \rangle_{\mathcal{H}}$ for $h, h' \in \mathcal{H}$ and a norm $\|h\|_{\mathcal{H}} = \sqrt{\langle h, h \rangle_{\mathcal{H}}}$

Some vocabulary

- Note that

  RKHS $\Rightarrow$ Hilbert space, BUT Hilbert space $\not\Rightarrow$ RKHS
- The Hilbert space $\mathcal{H}$ is called the **features space** associated to k
- The corresponding mapping $\varphi : \mathcal{X} \to \mathcal{H}$ is called the **features mapping**
- $\mathcal{H}$ is endowed with an inner product $\langle h, h' \rangle_{\mathcal{H}}$ for $h, h' \in \mathcal{H}$ and a norm $\|h\|_{\mathcal{H}} = \sqrt{\langle h, h \rangle_{\mathcal{H}}}$
- The feature space might not be unique in general

Some examples of RKHS

1. any finite-dimensional Hilbert space of functions is a RKHS, with $k(x, x') = \sum_{i=1}^{\dim(\mathcal{H})} e_i(x)e_i(x')$.

Some examples of RKHS

1. any finite-dimensional Hilbert space of functions is a RKHS, with $k(x, x') = \sum_{i=1}^{\dim(\mathcal{H})} e_i(x) e_i(x')$.
2. the space $L^2(\mathbb{R})$ is not a RKHS.

Some examples of RKHS

1. any finite-dimensional Hilbert space of functions is a RKHS, with $k(x, x') = \sum_{i=1}^{\dim(\mathcal{H})} e_i(x) e_i(x')$.
2. the space $L^2(\mathbb{R})$ is not a RKHS.
3. the space of $\mathcal{F} = \left\{ f : f(0) = 0, f \text{ absolutely continuous}, f, f' \in L^2(\mathbb{R}) \right\}$ is a RKHS with $k(x, x') = e^{-|x-x'|}$.

In summary

- Choose a kernel k you think relevant
- If it's PDS, then there is a mapping φ and a RKHS $\mathcal{H}$ for it

In summary

- Choose a kernel k you think relevant
- If it's PDS, then there is a mapping φ and a RKHS $\mathcal{H}$ for it
- Feature engineering becomes kernel engineering with kernel methods

In summary

- Choose a kernel k you think relevant
- If it's PDS, then there is a mapping φ and a RKHS $\mathcal{H}$ for it
- Feature engineering becomes kernel engineering with kernel methods
- Any linear algorithm based on computing inner products can be extended into a non-linear version by replacing the inner products by a kernel function $\rightsquigarrow$ kernel trick

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

Normalized kernel

Definition

The **normalized kernel** k' associated to a kernel k is given by

$$k'(x, x') = \frac{k(x, x')}{\sqrt{k(x, x)k(x', x')}}$$

if $k(x, x)k(x', x') > 0$ and $k(x, x') = 0$ otherwise.

Theorem

If k is a PDS kernel, its normalized kernel k' is PDS.

Proof

Let $x_1, \dots, x_n \in \mathcal{X}$ and $c \in \mathbb{R}^n$. If $k(x_i, x_i) = 0$ or $k(x_j, x_j) = 0$ then $k(x_i, x_j) = 0$ using Cauchy-Schwarz, so $k'(x_i, x_j) = 0$.
So, we can assume $k(x_i, x_i) > 0$ for all $i = 1, \dots, n$ and write the following:

$$\sum_{1 \leqslant i,j \leqslant n} \frac{c_i c_j k(x_i, x_j)}{\sqrt{k(x_i, x_i) k(x_j, x_j)}} = \sum_{1 \leqslant i,j \leqslant n} \frac{c_i c_j \langle \varphi(x_i), \varphi(x_j) \rangle}{\|\varphi(x_i)\| \, \|\varphi(x_j)\|}$$

$$= \left\| \sum_{i=1}^{n} \frac{c_i \varphi(x_i)}{\|\varphi(x_i)\|} \right\|^2 \geqslant 0$$

which proves the theorem.

A few remarks

Remark

- *We have that $k(x, x')$ is the cosine of the angle between $\varphi(x)$ and $\varphi(x')$ if k is a normalized kernel (if none is zero).*
- *Once again, $k(x, x')$ is a similarity measure between x and x'*

A few remarks

Remark

- *We have that* $k(x, x')$ *is the cosine of the angle between* $\varphi(x)$ *and* $\varphi(x')$ *if* k *is a normalized kernel (if none is zero).*
- *Once again,* $k(x, x')$ *is a similarity measure between* x *and* x'

Remark

If k *is a normalized kernel, then*

$$\|\varphi(x)\|_{\mathcal{H}} = \langle \varphi(x), \varphi(x) \rangle_{\mathcal{H}} = k(x, x) = 1$$

for any $x \in \mathcal{X}$.

Some famous kernels

The polynomial kernel.

For $c > 0$ and $q \in \mathbb{N} \setminus \{0\}$ we define the polynomial kernel

$$K(x, x') = (\langle x, x' \rangle + c)^q.$$

It is a PDS kernel,

Some famous kernels

The polynomial kernel.

For $c > 0$ and $q \in \mathbb{N} \setminus \{0\}$ we define the polynomial kernel

$$K(x, x') = (\langle x, x' \rangle + c)^q.$$

It is a PDS kernel, since it is the power of the PDS kernel $(x, x') \mapsto \langle x, x' \rangle + b$.

Some famous kernels

The polynomial kernel.

For $c > 0$ and $q \in \mathbb{N} \setminus \{0\}$ we define the polynomial kernel

$$K(x, x') = (\langle x, x' \rangle + c)^q.$$

It is a PDS kernel, since it is the power of the PDS kernel $(x, x') \mapsto \langle x, x' \rangle + b$.

We already computed its mapping $\varphi(x)$: it contains all the monomials of degree less than q of the coordinates of x.

Some famous kernels

The Gaussian or the Radial Basis Function (RBF) kernel.

For $\gamma > 0$ it is given by

$$k(x, x') = \exp(-\gamma \left\| x - x' \right\|_2^2)$$

Some famous kernels

The Gaussian or the Radial Basis Function (RBF) kernel.

For $\gamma > 0$ it is given by

$$k(x, x') = \exp(-\gamma \left\| x - x' \right\|_2^2)$$

Proposition

The RBF kernel is a PDS and normalized kernel.

Some famous kernels

The Gaussian or the Radial Basis Function (RBF) kernel.

For $\gamma > 0$ it is given by

$$k(x, x') = \exp(-\gamma \left\| x - x' \right\|_2^2)$$

Proposition

The RBF kernel is a PDS and normalized kernel.

By far, the RBF kernel is the most widely used: uses as a similarity measure the Euclidean norm

Proof

First remark that

$$\exp(-\gamma \left\|x - x'\right\|_2^2) = \frac{\exp(2\gamma\langle x, x'\rangle)}{\exp(\gamma \left\|x\right\|^2)\exp(\gamma \left\|x'\right\|^2)}$$
$$= \frac{k'(x, x')}{\sqrt{k'(x, x)k'(x', x')}}$$

with $k'(x, x') = \exp(2\gamma\langle x, x'\rangle)$ and that k' is PDS since

$$k'(x, x') = \sum_{n \geqslant 0} \frac{(2\gamma\langle x, x'\rangle)^n}{n!}$$

namely a series of the PDS kernel $(x, x') \mapsto 2\gamma\langle x, x'\rangle$.

Some famous kernels

The tanh kernel or the sigmoid kernel.

$$k'(x, x') = \tanh(a\langle x, x'\rangle + c) = \frac{e^{a\langle x,x'\rangle + c} - e^{-a\langle x,x'\rangle - c}}{e^{a\langle x,x'\rangle + c} + e^{-a\langle x,x'\rangle - c}}$$

for $a, c > 0$. It is again a PDS kernel (same argument as for the RBF kernel).

Some famous kernels

The tanh kernel or the sigmoid kernel.

$$k'(x,x') = \tanh(a\langle x,x'\rangle + c) = \frac{e^{a\langle x,x'\rangle + c} - e^{-a\langle x,x'\rangle - c}}{e^{a\langle x,x'\rangle + c} + e^{-a\langle x,x'\rangle - c}}$$

for $a, c > 0$. It is again a PDS kernel (same argument as for the RBF kernel).

<u>Exercise</u>: compute its mapping.

Kernel-based algorithms?

Question

How to use kernels for classification and regression?

Kernel-based algorithms?

Question

How to use kernels for classification and regression?

Recall the linear SVM

Figure: SVM: hard and soft margins

Recall the linear SVM

Linear SVM

- Back to the primal problem

$$\min_{w\in\mathbb{R}^d, b\in\mathbb{R}, s\in\mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n s_i$$

$$\text{s.t.} \quad y_i(\langle x_i, w\rangle + b) \geqslant 1 - s_i \ \text{ and } \ s_i \geqslant 0 \ \text{ for all } \ i = 1, \ldots, n$$

- or equivalently

$$\text{argmin}_{w\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n \ell(y_i, \langle x_i, w\rangle + b)$$

  where $\ell(y, y') = \max(0, 1 - yy') = (1 - yy')_+$ is the hinge loss.

- Label prediction given by

$$y = \text{sign}(\langle x, w\rangle + b)$$

Recall the linear SVM

Linear SVM

- Back to the primal problem

$$\min_{w\in\mathbb{R}^d, b\in\mathbb{R}, s\in\mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n s_i$$

$$\text{s.t.} \quad y_i(\langle x_i, w\rangle + b) \geqslant 1 - s_i \;\text{ and }\; s_i \geqslant 0 \;\text{ for all }\; i = 1, \ldots, n$$

- or equivalently

$$\text{argmin}_{w\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n \ell(y_i, \langle x_i, w\rangle + b)$$

where $\ell(y, y') = \max(0, 1 - yy') = (1 - yy')_+$ is the hinge loss.

- Label prediction given by

$$y = \text{sign}(\langle x, w\rangle + b)$$

Kernel SVM

Principle

- Replace x_i by $\varphi(x_i)$. In the primal this leads to

$$\text{argmin}_{w\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\ell(y_i, \langle\varphi(x_i), w\rangle + b)$$

- Label prediction is given by

$$y = \text{sign}(\langle\varphi(x), w\rangle + b)$$

Problem

In the primal, you need to compute $\varphi(x)$!

Linear SVM

Dual problem

$$\max_{\alpha \in \mathbb{R}^n} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

subject to $\quad 0 \leqslant \alpha_i \leqslant C \;$ and $\; \sum_{i=1}^{n} \alpha_i y_i = 0 \;$ for all $\; i = 1, \ldots, n$

and the label prediction using dual variables

$$x \mapsto \operatorname{sign}(\langle w, x \rangle + b) = \operatorname{sign}\left(\sum_{i=1}^{n} \alpha_i y_i \langle x, x_i \rangle + b \right)$$

depends only on the features x_i via their inner products $\langle x_i, x_j \rangle$

Linear SVM

Dual problem

$$\max_{\alpha \in \mathbb{R}^n} \quad \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

$$\text{subject to} \quad 0 \leqslant \alpha_i \leqslant C \;\; \text{and} \;\; \sum_{i=1}^n \alpha_i y_i = 0 \;\; \text{for all} \;\; i = 1, \ldots, n$$

and the label prediction using dual variables

$$x \mapsto \operatorname{sign}(\langle w, x \rangle + b) = \operatorname{sign}\left(\sum_{i=1}^n \alpha_i y_i \langle x, x_i \rangle + b \right)$$

Depends only on the features x_i via their inner products $\langle x_i, x_j \rangle$

Kernel SVM

Remark (Fundamental remark)

The dual problem depends only on the features via their inner products.

Kernel SVM

Remark (Fundamental remark)

The dual problem depends only on the features via their inner products.

Given some kernel k, let's replace the "raw" inner products $\langle x_i, x_j \rangle$ by the "new" inner products $k(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$

Kernel SVM

Remark (Fundamental remark)

The dual problem depends only on the features via their inner products.

Given some kernel k, let's replace the "raw" inner products $\langle x_i, x_j \rangle$ by the "new" inner products $k(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$

The kernel trick

To train the SVM with a kernel, you don't need to know or compute the $\varphi(x_i)$!

Kernel SVM

Remark (Fundamental remark)

The dual problem depends only on the features via their inner products.

Given some kernel k, let's replace the "raw" inner products $\langle x_i, x_j \rangle$ by the "new" inner products $k(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$

The kernel trick

To train the SVM with a kernel, you don't need to know or compute the $\varphi(x_i)$!

Take-home message: kernel trick

- Kernel + SVM = ♡
- But do it in the dual problem only!

Kernel SVM

Dual problem

$$\max_{\alpha \in \mathbb{R}^n} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$\text{subject to} \quad 0 \leqslant \alpha_i \leqslant C \quad \text{and} \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \quad \text{for all} \quad i = 1, \ldots, n$$

Label prediction

The label prediction using dual variables

$$x \mapsto \operatorname{sign}\left(\sum_{i=1}^{n} \alpha_i y_i k(x, x_i) + b\right)$$

with the intercept given by

$$b = y_i - \sum_{j=1}^{n} \alpha_j y_j k(x_j, x_i)$$

for any i such that $0 < \alpha_i < C$ (support vector) (cf previous lecture)

Kernel & support vectors

This proves that the hypothesis solution writes

$$h(x) = \text{sign}\Big(\sum_{i:\alpha_i \neq 0} \alpha_i y_i k(x, x_i) + b \Big),$$

namely a combination of functions $k(x_i, \cdot)$ where x_i are the support vectors.

For the RBF kernel

The decision function is

$$x \mapsto \sum_{i:\alpha_i \neq 0} \alpha_i y_i \exp\big(-\gamma \, \|x - x_i\|_2^2 \big) + b$$

It is a mixture of Gaussian "densities". Let's recall that the x_i with $\alpha_i \neq 0$ are the support vectors

Kernel and regression

The kernel trick is not only for the SVM!

Theorem ((Kimeldorf & Wahba 1971, Schölkopf et al. 2001))

If k is a PDS kernel and $\mathcal{H}$ its corresponding RKHS, for any increasing function g and any function $L : \mathbb{R}^n \to \mathbb{R}$, the optimization problem

$$\min_{h \in \mathcal{H}} g(\|h\|_{\mathcal{H}}) + L(h(x_1), \ldots, h(x_n))$$

admits only solutions of the form

$$h^\star = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot).$$

This theorem is called the representer theorem.

The representer theorem

It means that in the case of a penalization increasing with $\| \cdot \|_{\mathcal{H}}$, any optimal solution $h^\star$ lives in a finite dimensional vector space of $\mathcal{H}$, even if $\mathcal{H}$ is infinite-dimensional!

Kernel Ridge regression

- Consider this time a continuous label $y_i \in \mathbb{R}$, features $x_i \in \mathcal{X}$ for $i = 1, \ldots, n$ and a features mapping $\varphi : \mathcal{X} \to \mathcal{H}$ with PDS kernel k

Kernel Ridge regression

- Consider this time a continuous label $y_i \in \mathbb{R}$, features $x_i \in \mathcal{X}$ for $i = 1, \ldots, n$ and a features mapping $\varphi : \mathcal{X} \to \mathcal{H}$ with PDS kernel k
- Kernel Ridge regression considers the problem

$$\min_w \left\{ \sum_{i=1}^n \ell(y_i, \langle w, \varphi(x_i) \rangle) + \frac{\lambda}{2} \|w\|_2^2 \right\}$$

where λ is a penalization parameter, and $\ell(y, y') = \frac{1}{2}(y - y')^2$ is the least-squares loss

Kernel Ridge regression

- Consider this time a continuous label $y_i \in \mathbb{R}$, features $x_i \in \mathcal{X}$ for $i = 1, \ldots, n$ and a features mapping $\varphi : \mathcal{X} \to \mathcal{H}$ with PDS kernel k
- Kernel Ridge regression considers the problem

$$\min_w \left\{ \sum_{i=1}^{n} \ell(y_i, \langle w, \varphi(x_i) \rangle) + \frac{\lambda}{2} \|w\|_2^2 \right\}$$

where λ is a penalization parameter, and $\ell(y, y') = \frac{1}{2}(y - y')^2$ is the least-squares loss

- Can be written as

$$\min_w F(x) \quad \text{with} \quad F(w) = \|y - Xw\|_2^2 + \lambda \|w\|_2^2$$

with X the matrix with rows containing the $\varphi(x_i)$ and $y = [y_1 \cdots y_n] \in \mathbb{R}^n$

Kernel Ridge regression

$$\min_{w} \|y - Xw\|_2^2 + \lambda \|w\|_2^2$$

Kernel Ridge regression

$$\min_{w} \|y - Xw\|_2^2 + \lambda \|w\|_2^2$$

- This problem is strongly convex, and admits a global minimum iff

Kernel Ridge regression

$$\min_w \|y - Xw\|_2^2 + \lambda \|w\|_2^2$$

- This problem is strongly convex, and admits a global minimum iff

$$\nabla F(w) = 0 \quad \text{namely} \quad (X^\top X + \lambda \mathrm{Id})w = X^\top y$$

Kernel Ridge regression

$$\min_{w} \|y - Xw\|_2^2 + \lambda \|w\|_2^2$$

- This problem is strongly convex, and admits a global minimum iff
$$\nabla F(w) = 0 \quad \text{namely} \quad (X^\top X + \lambda \mathrm{Id})w = X^\top y$$

- Note that $X^\top X + \lambda \mathrm{Id}$ is always invertible. Thus kernel ridge admits a closed-form solution.

Kernel Ridge regression

$$\min_w \|y - Xw\|_2^2 + \lambda \|w\|_2^2$$

- This problem is strongly convex, and admits a global minimum iff
$$\nabla F(w) = 0 \quad \text{namely} \quad (X^\top X + \lambda \mathrm{Id})w = X^\top y$$
- Note that $X^\top X + \lambda \mathrm{Id}$ is always invertible. Thus kernel ridge admits a closed-form solution.
- Requires to solve a $D \times D$ linear system, where D is the dimension of $\mathcal{H}$
- What if D is large ?

Kernel trick & regression

Let's use the kernel trick, as we did for SVM

- Representer theorem says that we can find α such that

$$h(x) = \langle w, \varphi(x) \rangle = \sum_{i=1}^{n} \alpha_i k(x_i, x) = \sum_{i=1}^{n} \alpha_i \langle \varphi(x_i), \varphi(x) \rangle$$

  for any $x \in \mathcal{X}$
- This means that

$$w = X^\top \alpha$$

New trick

Now use this trick

For any matrix X, we have

$$(X^\top X + \lambda \mathrm{Id})^{-1} X^\top = X^\top (X X^\top + \lambda \mathrm{Id})^{-1}$$

This entails

$$w = (X^\top X + \lambda \mathrm{Id})^{-1} X^\top y = X^\top (X X^\top + \lambda \mathrm{Id})^{-1} y$$

which gives (note that $(XX^\top)_{i,j} = \langle \varphi(x_i), \varphi(x_j) \rangle = k(x_i, x_j)$)

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

Proof of the trick

Note that

$$(X^\top X + \lambda \mathrm{Id}) X^\top = X^\top (X X^\top + \lambda \mathrm{Id}).$$

Multiplying on the left by $(X^\top X + \lambda \mathrm{Id})^{-1}$ leads to

$$X^\top = (X^\top X + \lambda \mathrm{Id})^{-1} X^\top (X X^\top + \lambda \mathrm{Id}).$$

and then on the right by $(X X^\top + \lambda \mathrm{Id})^{-1}$ concludes with

$$(X X^\top + \lambda \mathrm{Id})^{-1} X^\top = (X^\top X + \lambda \mathrm{Id})^{-1} X^\top$$

A cute trick. But let's do it like we did for the SVMs (just to be sure...)

Another way for Kernel Ridge regression

An alternative formulation of

$$\min_{w} \sum_{i=1}^{n} (y_i - \langle w, \varphi(x_i) \rangle)^2 + \lambda \|w\|_2^2$$

is the **constrained version**, given by

$$\min_{w} \sum_{i=1}^{n} (y_i - \langle w, \varphi(x_i) \rangle)^2 \ \text{ subject to } \ \|w\|_2^2 \leqslant r^2$$

and also

$$\min_{w} \sum_{i=1}^{n} s_i^2 \ \text{ subject to } \ \|w\|_2^2 \leqslant r^2 \ \text{ and } \ s_i = y_i - \langle w, \varphi(x_i) \rangle$$

Then, using the Lagrangian

$$L(w, s, \alpha, \lambda) = \min_w \sum_{i=1}^{n} s_i^2 + \min_w \sum_{i=1}^{n} \alpha_i (y_i - s_i - \langle w, \varphi(x_i) \rangle) \\ + \lambda(\|w\|_2^2 - r^2)$$

Then, using the Lagrangian

$$L(w, s, \alpha, \lambda) = \min_w \sum_{i=1}^{n} s_i^2 + \min_w \sum_{i=1}^{n} \alpha_i (y_i - s_i - \langle w, \varphi(x_i) \rangle) + \lambda(\|w\|_2^2 - r^2)$$

KKT conditions

$$\nabla_w L = -\sum_{i=1}^{n} \alpha_i \varphi(x_i) + 2\lambda w \Rightarrow w = \frac{1}{2\lambda} \sum_{i=1}^{n} \alpha_i \varphi(x_i)$$

$$\nabla_{s_i} L = 2s_i - \alpha_i \Rightarrow s_i = \alpha_i / 2$$

and the slackness complementary conditions:

$$\alpha_i (y_i - s_i - \langle w, \varphi(x_i) \rangle) = 0 \quad \text{and} \quad \lambda(\|w\|_2^2 - r^2) = 0$$

Plugging the expressions of w and s_i in functions of α in L gives after some algebra the dual objective

$$D(\alpha) = -\lambda \sum_{i=1}^{n} \alpha_i^2 + 2 \sum_{i=1}^{n} \alpha_i y_i \\ - \sum_{1 \leqslant i,j \leqslant n} \alpha_i \alpha_j \langle \varphi(x_i), \varphi(x_j) \rangle - \lambda r^2$$

(where we replaced $2\lambda\alpha_i$ by α_i)

Plugging the expressions of w and s_i in functions of α in L gives after some algebra the dual objective

$$\begin{aligned} D(\alpha) = &-\lambda \sum_{i=1}^{n} \alpha_i^2 + 2 \sum_{i=1}^{n} \alpha_i y_i \\ &- \sum_{1 \leqslant i,j \leqslant n} \alpha_i \alpha_j \langle \varphi(x_i), \varphi(x_j) \rangle - \lambda r^2 \end{aligned}$$

(where we replaced $2\lambda\alpha_i$ by α_i) which can be written matricially as

$$\begin{aligned} D(\alpha) &= -\lambda \|\alpha\|_2^2 + 2\langle \alpha, y \rangle - \alpha^\top X X^\top \alpha \\ &= 2\langle \alpha, y \rangle - \alpha^\top (K + \lambda \mathrm{Id}) \alpha \end{aligned}$$

Plugging the expressions of w and s_i in functions of α in L gives after some algebra the dual objective

$$
\begin{aligned}
D(\alpha) = -\lambda \sum_{i=1}^{n} \alpha_i^2 + 2 \sum_{i=1}^{n} \alpha_i y_i \\
- \sum_{1 \leqslant i,j \leqslant n} \alpha_i \alpha_j \langle \varphi(x_i), \varphi(x_j) \rangle - \lambda r^2
\end{aligned}
$$

(where we replaced $2\lambda\alpha_i$ by α_i) which can be written matricially as

$$
\begin{aligned}
D(\alpha) &= -\lambda \|\alpha\|_2^2 + 2\langle \alpha, y \rangle - \alpha^\top X X^\top \alpha \\
&= 2\langle \alpha, y \rangle - \alpha^\top (K + \lambda \mathrm{Id}) \alpha
\end{aligned}
$$

with optimum achieved for

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

what we already got.

Conclusion on kernels

- Solving a problem in the dual benefits from the kernel trick

Conclusion on kernels

- Solving a problem in the dual benefits from the kernel trick
- Allows to construct complex non-linear decision functions

Conclusion on kernels

- Solving a problem in the dual benefits from the kernel trick
- Allows to construct complex non-linear decision functions
- OK if n is not too large... (if the $n \times n$ Gram matrix K fits in memory)
- Otherwise, stick to the primal! (and forget about kernels...)
- But don't forget about feature engineering (yes, again !)