Machine learning

Claire Boyer

September 30th, 2020

SORBONNE UNIVERSITÉ
CRÉATEURS DE FUTURS DEPUIS 1257

ISUP

Thanks to

- Stéphane Gaïffas
- Erwan Scornet
- Gérard Biau
- Maxime Sangnier
- Laurent Rouvière

Today

Summary

Kernel methods

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

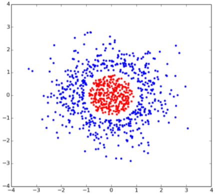
Kernel methods

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

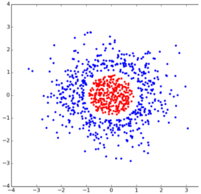
Idea

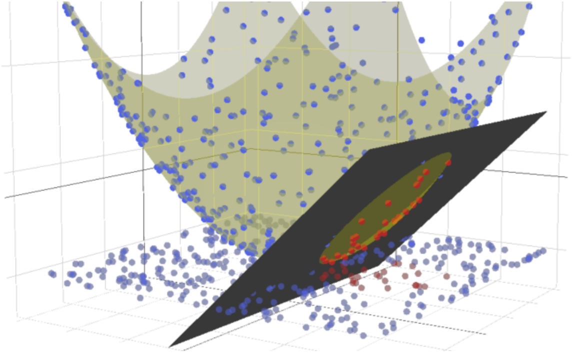
- to implicitly define an inner product in a high-dimensional space
- replacing the original inner product in the input space with positive definite kernels immediately extends algorithms such as SVMs to a linear separation in that high-dimensional space, or, equivalently, to a non-linear separation in the input space

When data are not linearly separable?



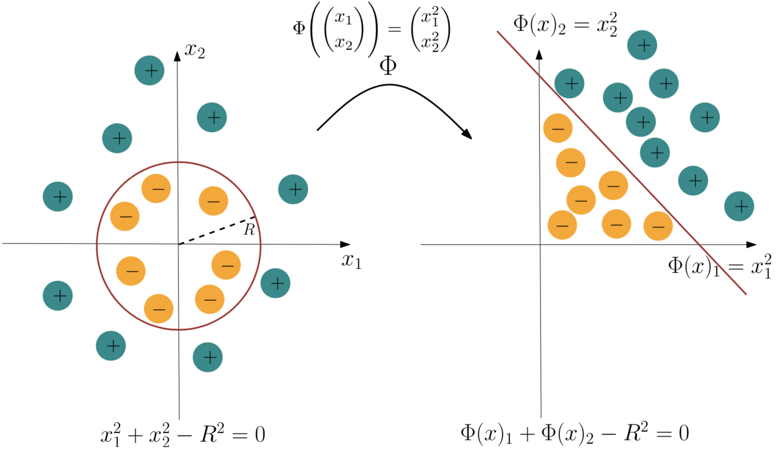
When data are not linearly separable?





Stolen from `http://efavdb.com/svm-classification/`

When data are not linearly separable



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
- One way: use a non-linear mapping φ from the input space $\mathcal{X}$ to a higher-dimensional space $\mathcal{H}$, where linear separation is possible

Ex: polynomial mapping

Polynomial mapping

The **polynomial** mapping $\varphi : \mathbb{R}^2 \to \mathbb{R}^6$ for $x = (x_1, x_2) \in \mathbb{R}^2$

$$\varphi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$

solves the XOR (Exclusive OR) classification problem.

Ex: polynomial mapping

Polynomial mapping

The **polynomial** mapping $\varphi : \mathbb{R}^2 \to \mathbb{R}^6$ for $x = (x_1, x_2) \in \mathbb{R}^2$

$$\varphi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$

solves the XOR (Exclusive OR) classification problem.

XOR : label y_i is blue iff one of the coordinates of x_i equals 1.

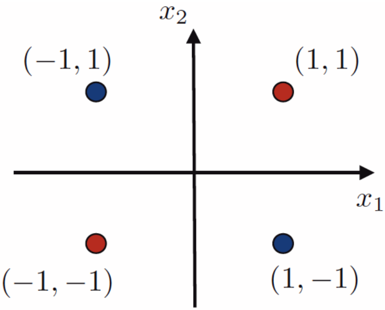


Figure: XOR problem linearly non-separable in the input space.

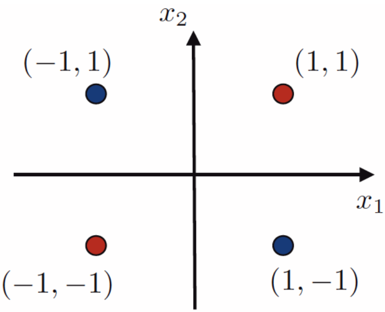


Figure: XOR problem linearly non-separable in the input space.

- Blue and red points cannot be linearly separated in $\mathbb{R}^2$

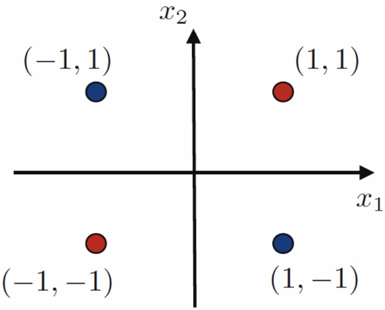
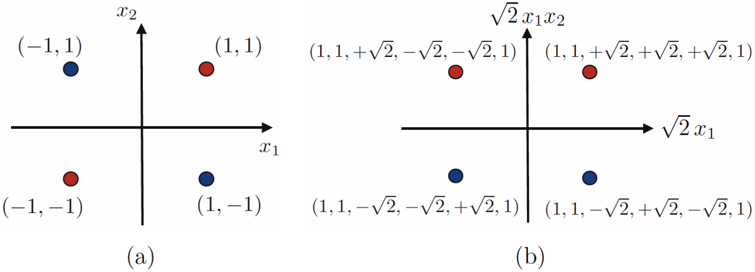


Figure: XOR problem linearly non-separable in the input space.

- Blue and red points cannot be linearly separated in $\mathbb{R}^2$
- But they can using the mapping $\varphi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$, using the hyperplane $x_1x_2 = 0$



In (b), the hyperplane $x_1x_2 = 0$ separates blue points and red points.
This mapping φ is call polynomial mapping of order 2.

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

$$\begin{aligned}
\langle \varphi(x), \varphi(x') \rangle &= \left\langle \begin{bmatrix} x_1^2 \\ x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ 1 \end{bmatrix}, \begin{bmatrix} x_1^2 \\ x_1'^2 \\ x_2'^2 \\ \sqrt{2}x_1'x_2' \\ \sqrt{2}x_1' \\ \sqrt{2}x_2' \\ 1 \end{bmatrix} \right\rangle \\
&= (x_1x_1' + x_2x_2' + 1)^2 \\
&= (\langle x, x' \rangle + 1)^2
\end{aligned}$$

Some definitions

Definition (Kernel)

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

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).

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 $x_1, \ldots, x_n$ 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

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

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

(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.

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

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

THE theorem

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 $x \in \mathcal{X}$.

THE theorem

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 reproducing kernel Hilbert space associated to the kernel k.

Some vocabulary

- Note that

  RKHS $\Rightarrow$ Hilbert space, BUT Hilbert space $\nRightarrow$ RKHS

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

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**

- 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 $\|h\|_{\mathcal{H}} = \sqrt{\langle h, h \rangle_{\mathcal{H}}}$

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 $x_1, \ldots, x_n \in \mathcal{X}$ 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 $i = 1, \ldots, n$ and write the following:

$$\begin{aligned}\sum_{1 \leqslant i,j \leqslant n} \frac{c_i c_j k(x_i, x_j)}{\sqrt{k(x_i, x_i) k(x_j, x_j)}} &= \sum_{1 \leqslant i,j \leqslant n} \frac{c_i c_j \langle \varphi(x_i), \varphi(x_j) \rangle}{\|\varphi(x_i)\| \, \|\varphi(x_j)\|} \\ &= \left\| \sum_{i=1}^{n} \frac{c_i \varphi(x_i)}{\|\varphi(x_i)\|} \right\|^2 \geqslant 0\end{aligned}$$

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 $(x, x') \mapsto \langle x, x' \rangle + b$.

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).

Exercise: 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

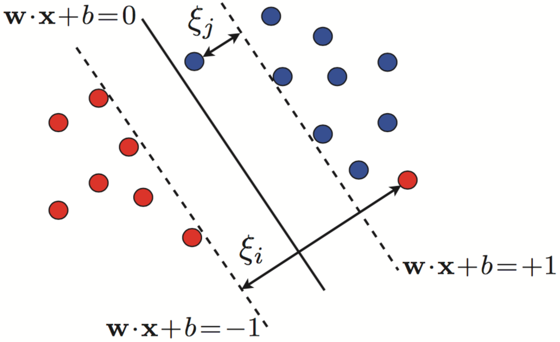


Figure: SVM: hard and soft margins

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

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

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

- 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

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

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

and the label prediction using dual variables

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

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

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

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

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

Kernel SVM

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

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

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

RBF kernel

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

the image that you will plot later :)

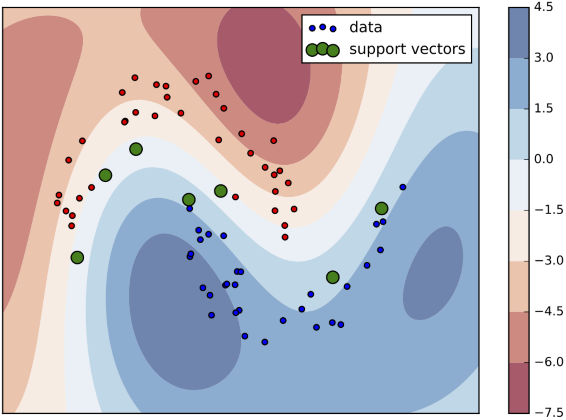


Figure: Data is separated thanks to a Gaussian mixture.

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.

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

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

This entails

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

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

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

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

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

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

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

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

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

and also

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

Then, using the Lagrangian

Then, using the Lagrangian

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

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:

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

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 !)

References on kernels & SVM

- Support Vector Machine, by Ingo Steinwart and Andreas Christmann
- Learning with kernels, by Bernhard Schlkopf and Alexander J. Smola
- Reproducing Kernel Hilbert Spaces in Probability and Statistics, by Alain Berlinet and Christine Thomas-Agnan

Summary

Non-parametric learning

Non-parametric learning algorithm (does not mean NO parameters)

- The complexity of the decision function grows with the number of data points
- Contrast with linear regression ($\simeq$ as many parameters as features)
- Usually: decision function is expressed directly in terms of the training examples
- Examples
  - k-nearest neighbors (today)
  - tree-based methods (in the next sessions)

Instance-based Learning

Learning

Store training instances

Prediction

Compute the label for a new instance based on its similarity with the stored instances.

- Also called lazy learning
- Similar to case-based reasoning
  - Doctors treating a patient based on how patients with similar symptoms were treated
  - Judges ruling court cases based on legal precedent

Reminder

Recall the problem of binary classification for $Y \in \{0, 1\}$. We show that the minimizer of the risk

$$\mathcal{R}(g) = \mathbb{E}[\mathbb{1}_{g(X) \neq Y}].$$

is the Bayes classifier

$$g^\star(x) = \begin{cases} 1 & \text{if} \quad r(x) > 1/2, \\ 0 & \text{otherwise.} \end{cases}$$

A new method for classification

Given some sample $\mathcal{D}_n = \{X_1, \ldots, X_n\}$, another strategy to construct a classifier rule is to estimate

$$r(x) = \mathbb{E}[Y|X = x],$$

and to replace $r(x)$ by its estimator $r_n(x)$.
The result is the **plug-in classifier**, given by

$$g_n(x) = \begin{cases} 1 & \text{if} \quad r_n(x) > 1/2, \\ 0 & \text{otherwise.} \end{cases}$$

Link between r_n and risk

Let us denote μ the law of X.

Theorem

Let r_n be an estimator of r and g_n be the corresponding plug-in rule. Then

$$0 \leqslant \mathcal{R}(g_n) - \mathcal{R}^\star \leqslant 2 \int_{\mathbb{R}^d} |r_n(x) - r(x)| \mu(dx).$$

Link between r_n and risk

Let us denote μ the law of X.

Theorem

Let r_n be an estimator of r and g_n be the corresponding plug-in rule. Then

$$0 \leqslant \mathcal{R}(g_n) - \mathcal{R}^\star \leqslant 2 \int_{\mathbb{R}^d} |r_n(x) - r(x)| \mu(dx).$$

- This theorem says that if we have a good estimator r_n of r in the sense
$$\int_{\mathbb{R}^d} |r_n(x) - r(x)|^2 \mu(dx) \to 0,$$
in L^1 or almost surely, then the plug-in classifier is convergent (or strongly convergent).
- Question: how to construct good estimators r_n?
- $\rightsquigarrow$ Stone's theorem

Strategy

A way to construct estimator r_n of $r(x) = \mathbb{E}[Y|X = x]$ is to choose

$$r_n(x) = \sum_{i=1}^{n} W_{ni}(x) Y_i, \quad x \in \mathbb{R}^d$$

with

- $W_{ni}(x)$ is a real Borelian function of x and $X_1, \ldots X_n$, and not of $Y_1, \ldots, Y_n$.

Strategy

A way to construct estimator r_n of $r(x) = \mathbb{E}[Y|X = x]$ is to choose

$$r_n(x) = \sum_{i=1}^{n} W_{ni}(x) Y_i, \quad x \in \mathbb{R}^d$$

with

- $W_{ni}(x)$ is a real Borelian function of x and $X_1, \ldots X_n$, and not of $Y_1, \ldots, Y_n$.
- Idea: the X_i's that are close to x should bring information on the class to assign at x
- This is a local mean estimator

Strategy

A way to construct estimator r_n of $r(x) = \mathbb{E}[Y|X = x]$ is to choose

$$r_n(x) = \sum_{i=1}^{n} W_{ni}(x) Y_i, \quad x \in \mathbb{R}^d$$

with

- $W_{ni}(x)$ is a real Borelian function of x and $X_1, \dots X_n$, and not of $Y_1, \dots, Y_n$.
- Idea: the X_i's that are close to x should bring information on the class to assign at x
- This is a local mean estimator
- Often (but not always) the $W_{ni}(x)$'s can be chosen positive and normalized to 1, so as to $(W_{n1}(x), \dots W_{nn}(x))$ is a vector of probabilities

A first choice: Kernel estimator

A first typical choice is the following

$$W_{ni}(x) = \frac{K\left(\frac{x-X_i}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{x-X_j}{h}\right)}$$

with

- K a positive measurable function of $\mathbb{R}^d$
- K is called "kernel" ($\neq$ what we have seen before)
- h is positive parameter
- h is called "window"

A first choice: Kernel estimator

$$r_n(x) = \frac{\sum_{i=1}^{n} K\left(\frac{x-X_i}{h}\right) Y_i}{\sum_{j=1}^{n} K\left(\frac{x-X_j}{h}\right)}$$

- If the denominator is zero, set $r_n(x) = (1/n)\sum_i Y_i$

A first choice: Kernel estimator

$$r_n(x) = \frac{\sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) Y_i}{\sum_{j=1}^n K\left(\frac{x - X_j}{h}\right)}$$

- If the denominator is zero, set $r_n(x) = (1/n) \sum_i Y_i$
- For instance, for the naive choice $K(z) = \mathbb{1}_{\|z\| \leqslant 1}$, we get

$$r_n(x) = \frac{\sum_{i=1}^n \mathbb{1}_{\|x - X_i\| \leqslant h} Y_i}{\sum_{j=1}^n \mathbb{1}_{\|x - X_j\| \leqslant h}}$$

showing that $r(x)$ is estimated by the mean of the (Y_i)'s such that the distance between the X_i's and x is less than h.

A first choice: Kernel estimator

$$r_n(x) = \frac{\sum_{i=1}^n K\left(\frac{x-X_i}{h}\right) Y_i}{\sum_{j=1}^n K\left(\frac{x-X_j}{h}\right)}$$

- If the denominator is zero, set $r_n(x) = (1/n)\sum_i Y_i$
- For instance, for the naive choice $K(z) = \mathbb{1}_{\|z\|\leqslant 1}$, we get

  $$r_n(x) = \frac{\sum_{i=1}^n \mathbb{1}_{\|x-X_i\|\leqslant h} Y_i}{\sum_{j=1}^n \mathbb{1}_{\|x-X_j\|\leqslant h}}$$

  showing that $r(x)$ is estimated by the mean of the (Y_i)'s such that the distance between the X_i's and x is less than h.
- In general, the weight of Y_i depends on the distance between X_i and x, depending on the choice of K

A first choice: Kernel estimator

$$r_n(x) = \frac{\sum_{i=1}^n K\left(\frac{x-X_i}{h}\right) Y_i}{\sum_{j=1}^n K\left(\frac{x-X_j}{h}\right)}$$

- If the denominator is zero, set $r_n(x) = (1/n)\sum_i Y_i$
- For instance, for the naive choice $K(z) = \mathbb{1}_{\|z\| \leqslant 1}$, we get

  $$r_n(x) = \frac{\sum_{i=1}^n \mathbb{1}_{\|x-X_i\| \leqslant h} Y_i}{\sum_{j=1}^n \mathbb{1}_{\|x-X_j\| \leqslant h}}$$

  showing that $r(x)$ is estimated by the mean of the (Y_i)'s such that the distance between the X_i's and x is less than h.
- In general, the weight of Y_i depends on the distance between X_i and x, depending on the choice of K
- Classical choices
  - Epanechnikov's kernel: $(1 - \|z\|)\mathbb{1}_{\|z\| \leqslant 1}$
  - Gaussian kernel: $e^{-\|z\|^2}$

A second choice: kNN I

A second typical choice is based on the k nearest neighbors

$$r_n(x) = \sum_{i=1}^{n} v_{ni} Y_{(i)}(x), \quad x \in \mathbb{R}^d$$

with

- $(v_{n1}, v_{n2,\ldots,v_{nn}})$ is a vector of (deterministic) weights normalized to 1
- $\big((X_{(1)}(x), Y_{(1)}(x)), \ldots, (X_{(n)}(x), Y_{(n)}(x))\big)$ is the permutation of $((X_1, Y_1), \ldots, (X_n, Y_n))$ according to increasing distances $\|X_j - x\|$, i.e.

  $$\|X_{(1)} - x\| \leqslant \ldots \leqslant \|X_{(n)} - x\|$$

- $W_{ni} = v_{n\sigma_i}$,
  with σ the permutation of $(1, \ldots, n)$ into $((1), \ldots, (n))$.

A second choice: kNN II

- A particular example is

$$v_{ni} = \begin{cases} \frac{1}{k}, & 1 \leqslant i \leqslant k \\ 0, & \text{otherwise.} \end{cases}$$

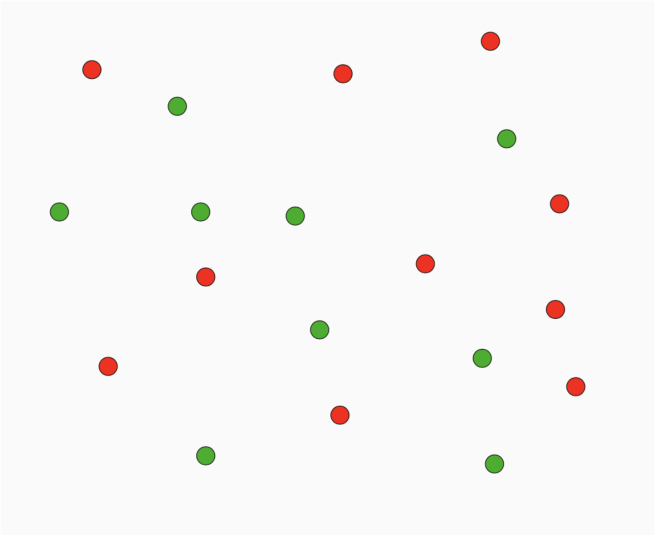
leading to

$$r_n(x) = \frac{1}{k} \sum_{i=1}^{k} Y_{(i)}(x)$$

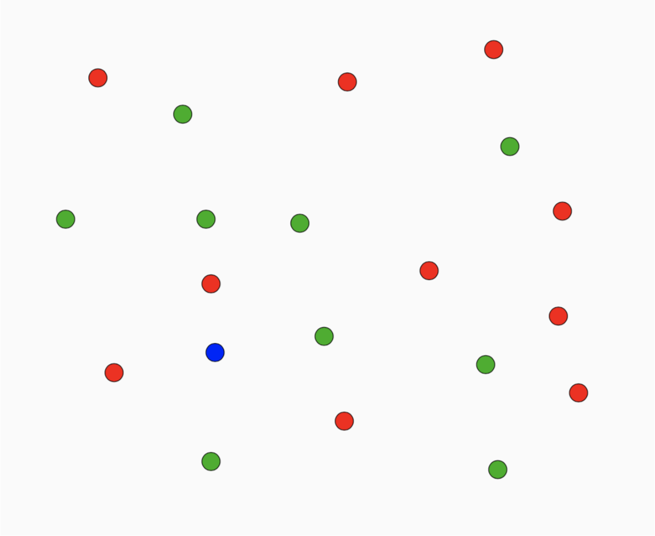
called the k-nearest neighbors estimator

- Idea: we look only at the k closest X_i of x, and we take the corresponding mean of Y_i.

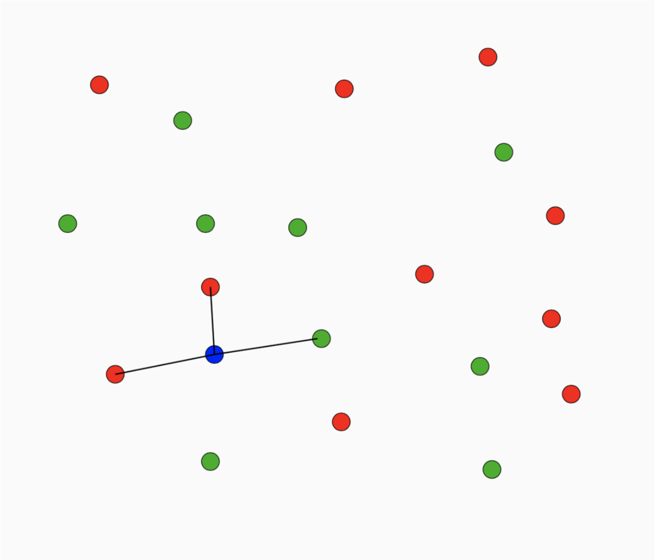
kNN with hands



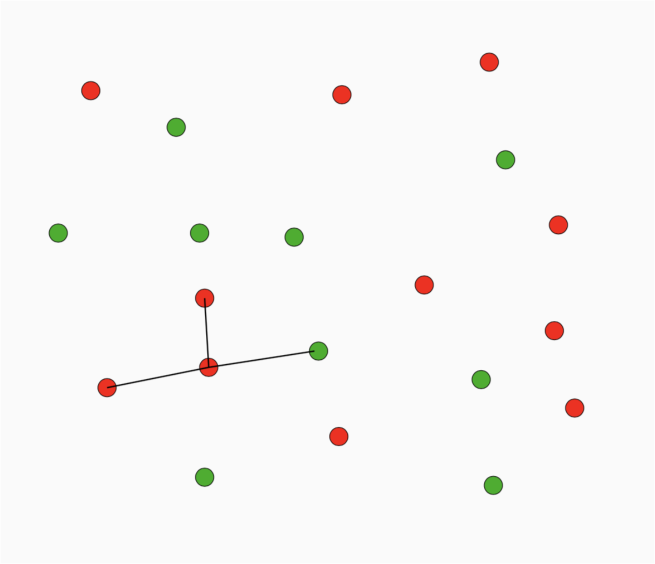
kNN with hands



kNN with hands



kNN with hands



A second choice: kNN III

Overall, the corresponding plug-in classifier can be written as follows

$$g_n(x) = \begin{cases} 1 & \text{if} \quad \sum_i W_{ni}(x) Y_i > 1/2, \\ 0 & \text{otherwise.} \end{cases}$$

If $\sum_{i=1}^{n} W_{ni}(x) = 1$,

$$g_n(x) = \begin{cases} 1 & \text{if} \quad \sum_i W_{ni}(x) \mathbb{1}_{Y_i=1} > \sum_i W_{ni}(x) \mathbb{1}_{Y_i=0}, \\ 0 & \text{otherwise.} \end{cases}$$

Stone's theorem

Theorem

Assume that for any distribution of X,

1. $\exists c$ *for all Borelian function* $f : \mathbb{R}^d \to \mathbb{R}$ *s.t.* $\mathbb{E}|f(X)| < \infty$,

$$\mathbb{E}\left(\sum_{i=1}^{n} W_{ni}(X)|f(X_i)|\right) \leqslant c\mathbb{E}|f(X)|, \quad \forall n \geqslant 1$$

2.
$$\forall a > 0, \quad \mathbb{E}\left(\sum_{i=1}^{n} W_{ni}\mathbb{1}_{\|X_i - x\| > a}\right) \to 0$$

3.
$$\mathbb{E}\left(\max_{1 \leqslant i \leqslant n} W_{ni}(X)\right) \to 0$$

Then, for any law of (X, Y), the plug-in classifier is universally convergent

$$\mathbb{E}\mathcal{R}(g_n) \to \mathcal{R}^\star.$$

Conditions of Stone's theorem

- Condition 2 means that the contribution of weights outside of any closed ball centered in X should be asymptotically negligible: only points in a local neighbourhood are needed
- Condition 3 prevents from one point to have a disproportionate influence on the estimator
- Condition 1 is called Stone's condition $\rightsquigarrow$ technical condition

Proof I

According to the first Theorem, it suffices to prove that for every distribution of (X, Y)

$$\mathbb{E}|r_n(X) - r(X)|^2 = \mathbb{E} \int_{\mathbb{R}^d} |r_n(x) - r(x)|^2 \mu(\mathrm{d}x) \to 0.$$

Introduce the notation

$$\hat{r}_n(x) = \sum_{i=1}^{n} W_{ni}(x) r(X_i).$$

Then, by the simple inequality $(a+b)^2 \leqslant 2(a^2+b^2)$, we have

$$\begin{aligned}\mathbb{E}|r_n(X) - r(X)|^2 &= \mathbb{E}|r_n(X) - \hat{r}_n(X) + \hat{r}_n(X) - r(X)|^2 \\ &\leqslant 2\big(\mathbb{E}|r_n(X) - \hat{r}_n(X)|^2 + \mathbb{E}|\hat{r}_n(X) - r(X)|^2\big).\end{aligned} \tag{1}$$

Therefore, it is enough to show that both terms on the right-hand side tend to zero as n tends to infinity. Since the W_{ni} are nonnegative and sum to one, by Jensen's inequality, the second term is

$$
\begin{aligned}
\mathbb{E}|\hat{r}_n(X) - r(X)|^2 &= \mathbb{E}\Big|\sum_{i=1}^{n} W_{ni}(X)(r(X_i) - r(X))\Big|^2 \\
&\leqslant \mathbb{E}\Big(\sum_{i=1}^{n} W_{ni}(X)|r(X_i) - r(X)|^2\Big).
\end{aligned}
$$

If the function r, which satisfies $0 \leqslant r \leqslant 1$, is continuous with compact support, then it is uniformly continuous as well: for every

Proof III

$\varepsilon > 0$, there is an $a > 0$ such that for $\|x - x'\| \leqslant a$, $|r(x) - r(x')|^2 \leqslant \varepsilon$. Thus, since $|r(x) - r(x')| \leqslant 1$,

$$\begin{aligned}
&\mathbb{E}\Big(\sum_{i=1}^{n} W_{ni}(X) |r(X_i) - r(X)|^2 \Big) \\
&\quad \leqslant \mathbb{E}\Big(\sum_{i=1}^{n} W_{ni}(X) \mathbb{1}_{[\|X_i - X\| > a]} \Big) + \mathbb{E}\Big(\sum_{i=1}^{n} W_{ni}(X) \varepsilon \Big) \\
&\quad = \mathbb{E}\Big(\sum_{i=1}^{n} W_{ni}(X) \mathbb{1}_{[\|X_i - X\| > a]} \Big) + \varepsilon.
\end{aligned}$$

Therefore, by (ii), since ε is arbitrary,

$$\mathbb{E}\Big(\sum_{i=1}^{n} W_{ni}(X) |r(X_i) - r(X)|^2 \Big) \to 0.$$

Proof IV

In the general case, since the set of continuous functions with compact support is dense in $L^2(\mu)$, for every $\varepsilon > 0$ we can choose r_ε such that

$$\mathbb{E}|r(X) - r_\varepsilon(X)|^2 \leqslant \varepsilon.$$

By this choice, using the inequality $(a+b+c)^2 \leqslant 3(a^2+b^2+c^2)$ (which follows from the Cauchy-Schwarz inequality),

$$\mathbb{E}|\hat{r}_n(X) - r(X)|^2$$

$$\leqslant \mathbb{E}\Big(\sum_{i=1}^{n} W_{ni}(X)|r(X_i) - r(X)|^2\Big)$$

$$\leqslant 3\mathbb{E}\Big(\sum_{i=1}^{n} W_{ni}(X)\big(|r(X_i) - r_\varepsilon(X_i)|^2 + |r_\varepsilon(X_i) - r_\varepsilon(X)|^2 + |r_\varepsilon(X) - r(X$$

Proof V

Thus, using (i),

$$\begin{aligned}
&\mathbb{E}|\hat{r}_n(X) - r(X)|^2 \\
&\quad \leqslant 3C\mathbb{E}|r(X) - r_\varepsilon(X)|^2 + 3\mathbb{E}\Big(\sum_{i=1}^{n} W_{ni}(X)|r_\varepsilon(X_i) - r_\varepsilon(X)|^2\Big) \\
&\qquad\quad + 3\mathbb{E}|r_\varepsilon(X) - r(X)|^2 \\
&\quad \leqslant 3C\varepsilon + 3\mathbb{E}\Big(\sum_{i=1}^{n} W_{ni}(X)|r_\varepsilon(X_i) - r_\varepsilon(X)|^2\Big) + 3\varepsilon.
\end{aligned}$$

Therefore, $\mathbb{E}|\hat{r}_n(X) - r(X)|^2 \to 0$.

Proof VI

To handle the first term of the right-hand side of (1), observe that, for all $i \neq j$,

$$
\begin{aligned}
&\mathbb{E}\big(W_{ni}(X)(Y_i - r(X_i))W_{nj}(X)(Y_j - r(X_j))\big) \\
&\quad = \mathbb{E}\Big[\mathbb{E}\Big(W_{ni}(X)(Y_i - r(X_i))W_{nj}(X)(Y_j - r(X_j)) \,|\, X, X_1, \dots, X_n, Y_i\Big)\Big] \\
&\quad = \mathbb{E}\Big[W_{ni}(X)(Y_i - r(X_i))W_{nj}(X)\mathbb{E}\big(Y_j - r(X_j) \,|\, X, X_1, \dots, X_n, Y_i\big)\Big] \\
&\quad = \mathbb{E}\Big[W_{ni}(X)(Y_i - r(X_i))W_{nj}(X)\mathbb{E}(Y_j - r(X_j) \,|\, X_j)\Big] \\
&\qquad \text{(by independence of } (X_j, Y_j) \text{ and } X, X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_n, Y_i) \\
&\quad = \mathbb{E}\Big[W_{ni}(X)(Y_i - r(X_i))W_{nj}(X)(r(X_j) - r(X_j))\Big] \\
&\quad = 0.
\end{aligned}
$$

Proof VII

Hence,

$$\begin{aligned}\mathbb{E}|r_n(X) - \hat{r}_n(X)|^2 &= \mathbb{E}\Big|\sum_{i=1}^{n} W_{ni}(X)(Y_i - r(X_i))\Big|^2 \\ &= \sum_{i,j=1}^{n} \mathbb{E}\big(W_{ni}(X)(Y_i - r(X_i))W_{nj}(X)(Y_j - r(X_j))\big) \\ &= \sum_{i=1}^{n} \mathbb{E}\big(W_{ni}^2(X)(Y_i - r(X_i))^2\big).\end{aligned}$$

We conclude that

$$\begin{aligned}\mathbb{E}|r_n(X) - \hat{r}_n(X)|^2 &\leqslant \mathbb{E}\sum_{i=1}^{n} W_{ni}^2(X) \leqslant \mathbb{E}\Big(\max_{1\leqslant i\leqslant n} W_{ni}(X)\sum_{j=1}^{n} W_{nj}(X)\Big) \\ &= \mathbb{E}\max_{1\leqslant i\leqslant n} W_{ni}(X) \to 0\end{aligned}$$

by (*iii*), and the theorem is proved.

Getting back to the kNN

Recall that the plug-in classifier reads

$$g_n(x) = \begin{cases} 1 & \text{if} \quad \sum_i W_{ni}(x)\mathbb{1}_{Y_i=1} > \sum_i W_{ni}(x)\mathbb{1}_{Y_i=0}, \\ 0 & \text{otherwise.} \end{cases}$$

Theorem

Assume that $k \to \infty$ and $k/n \to 0$. Then, the plug-in classifier in the case of the kNN is universally convergent, i.e.

$$\mathbb{E}\mathcal{R}(g_n) \to \mathcal{R}^\star,$$

for any law of (X, Y).

To prove this theorem, one has to verify the conditions of Stone's theorem.

Proof of kNN consistency I

Lemma

If $x \in supp(\mu)$ and $k/n \to 0$, then

$$\|X_{(k)}(x) - x\| \to 0 \quad \textit{almost surely.}$$

Proof of kNN consistency II

Proof.

Take $\varepsilon > 0$ and note, since x belongs to the support of μ, that $\mu(B(x,\varepsilon)) > 0$. Observe that

$$\Big[\|X_{(k)}(x) - x\| > \varepsilon\Big] = \Big[\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{[X_i \in B(x,\varepsilon)]} < \frac{k}{n}\Big].$$

By the strong law of large numbers,

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{[X_i \in B(x,\varepsilon)]} \to \mu(B(x,\varepsilon)) \quad \text{almost surely.}$$

Since $k/n \to 0$, we conclude that $\|X_{(k)}(x) - x\| \to 0$ almost surely. $\square$

Proof of kNN consistency III

Lemma

Let ν be a probability measure on $\mathbb{R}^d$. Fix $x' \in \mathbb{R}^d$ and let, for $a \geqslant 0$,

$$B_a(x') = \Big\{x \in \mathbb{R}^d : \nu\big(B(x, \|x' - x\|)\big) \leqslant a\Big\}.$$

Then

$$\nu(B_a(x')) \leqslant \gamma_d a,$$

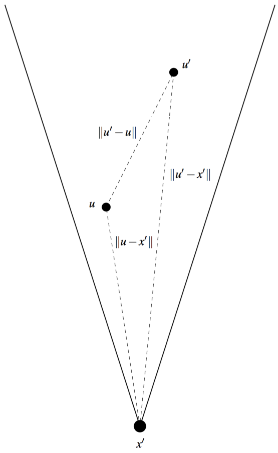
where γ_d is a positive constant depending only upon d.

Proof. Fix $x' \in \mathbb{R}^d$ and let $\mathscr{C}_1, \ldots, \mathscr{C}_{\gamma_d}$ be a collection of cones of angle $0 < \theta \leqslant \pi/6$ covering $\mathbb{R}^d$, all centered at x' but with different central directions (such a covering is always possible). In other words,

$$\bigcup_{j=1}^{\gamma_d} \mathscr{C}_j = \mathbb{R}^d.$$

Proof of kNN consistency IV

We leave it as an easy exercise to show that if $u \in \mathscr{C}_j$, $u' \in \mathscr{C}_j$, and $\|u - x'\| \leqslant \|u' - x'\|$, then $\|u - u'\| \leqslant \|u' - x'\|$ (see Figure 4).



Proof of kNN consistency V

Figure: The geometrical property of a cone of angle $0 < \theta \leqslant \pi/6$ (in dimension 2).

In addition,

$$\nu(B_a(x')) \leqslant \sum_{j=1}^{\gamma_d} \nu(\mathscr{C}_j \cap B_a(x')).$$

Let $x^\star \in \mathscr{C}_j \cap B_a(x')$. Then, by the geometrical property of cones mentioned above, we have

$$\nu\big(\mathscr{C}_j \cap B(x', \|x^\star - x'\|) \cap B_a(x')\big) \leqslant \nu\big(B(x^\star, \|x' - x^\star\|)\big) \leqslant a.$$

Since $x^\star$ was arbitrary, we conclude that

$$\nu(\mathscr{C}_j \cap B_a(x')) \leqslant a.$$

Proof of kNN consistency VI

Corollary

If distance ties occur with zero probability, then

$$\sum_{i=1}^{n} \mathbb{1}_{[X \text{ is among the } k\text{-NN of } X_i \text{ in } \{X_1, \ldots, X_{i-1}, X, X_{i+1}, \ldots, X_n\}]} \leqslant k$$

with probability one.

Proof We apply Lemma 19 with $a = k/n$ and ν the empirical measure μ_n associated with $X_1, \ldots, X_n$. With these choices,

$$B_{k/n}(X) = \Big\{x \in \mathbb{R}^d : \mu_n\big(B(x, \|X - x\|)\big) \leqslant k/n\Big\}$$

Proof of kNN consistency VII

and, with probability one,

$$
\begin{aligned}
& X_i \in B_{k/n}(X) \\
& \Leftrightarrow \mu_n\big(B(X_i, \|X - X_i\|)\big) \leqslant k/n \\
& \Leftrightarrow X \text{ is among the } k\text{-NN of } X_i \text{ in } \{X_1, \ldots, X_{i-1}, X, X_{i+1}, \ldots, X_n\}.
\end{aligned}
$$

(Note that the second equivalence uses the fact that distance ties occur with zero probability.) Thus, by Lemma 19, we conclude that, with probability one,

$$
\begin{aligned}
& \sum_{i=1}^{n} \mathbb{1}_{[X \text{ is among the } k\text{-NN of } X_i \text{ in } \{X_1, \ldots, X_{i-1}, X, X_{i+1}, \ldots, X_n\}]} \\
& \quad = \sum_{i=1}^{n} \mathbb{1}_{[X_i \in B_{k/n}(X)]} = n \times \mu_n(B_{k/n}(X)) \leqslant k\gamma_d.
\end{aligned}
$$

Proof of kNN consistency VIII

Lemma (Stone's lemma)

Assume that distance ties occur with zero probability. Then, for every Borel measurable function $f : \mathbb{R}^d \to \mathbb{R}$ such that $\mathbb{E}|f(X)| < \infty$, we have

$$\sum_{i=1}^{k} \mathbb{E}\big|f(X_{(i)}(X))\big| \leqslant k\gamma_d \mathbb{E}|f(X)|,$$

where γ_d is a positive constant depending only upon d.

Proof of kNN consistency IX

Proof. Take f as in the lemma. Then

$$\sum_{i=1}^{k} \mathbb{E}\big|f(X_{(i)}(X))\big|$$

$$= \mathbb{E}\Big(\sum_{i=1}^{n} |f(X_i)| \mathbb{1}_{[X_i \text{ is among the } k\text{-NN of } X \text{ in } \{X_1,\ldots,X_n\}]}\Big)$$

$$= \mathbb{E}\Big(|f(X)| \sum_{i=1}^{n} \mathbb{1}_{[X \text{ is among the } k\text{-NN of } X_i \text{ in } \{X_1,\ldots,X_{i-1},X,X_{i+1},$$

(by exchanging X and X_i)

$$\leqslant \mathbb{E}(|f(X)| k\gamma_d),$$

by the previous Corollary.

Now to show the universal consistency of g_n, we have to verify Conditions of Stone's theorem.

- Condition 3 is clear, since $k \to \infty$
- Condition 2: Note that

$$\mathbb{E}\left(\sum_{i=1}^{n} W_{ni}\mathbb{1}_{\|X_i - X\| > a}\right) = \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{n} \mathbb{1}_{\|X_{(i)}(X) - X\| > a}\right).$$

Then $\mathbb{E}\left(\sum_{i=1}^{n} W_{ni}\mathbb{1}_{\|X_i - X\| > a}\right) \to 0$ if for all $a > 0$

$$\mathbb{P}\left(\|X_{(k)}(X) - X\| > a\right) \to 0.$$

But,

$$\mathbb{P}\left(\|X_{(k)}(X) - X\| > a\right) = \int_{\mathbb{R}^d} \mathbb{P}\left(\|X_{(k)}(x) - x\| > a\right)\mu(dx).$$

Concluding the proof of kNN consistency II

For a fixed x in the support of μ, Lemma 18 says

$$\mathbb{P}\left(\|X_{(k)}(x) - x\| > a\right) \to 0$$

when $k/n \to 0$. Then, the conclusion follows by the Lebesgue dominated convergence theorem (the support of μ is of μ-measure 1).

- Condition 1: take f such that $\mathbb{E}|f(X)| < \infty$ we have to show that for some constant C

$$\mathbb{E}\left[\frac{1}{k}\sum_{i=1}^{n} |f(X_i)| \mathbb{1} X_i \in kNN(X)\right] \leqslant C\mathbb{E}|f(X)|.$$

Since,

$$\mathbb{E}\left[\frac{1}{k}\sum_{i=1}^{n} |f(X_i)| \mathbb{1} X_i \in kNN(X)\right] = \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k} |f(X_{(i)}(X))|\right),$$

this is precisely the statement of Stone's lemma.

Choice of k

Small k: noisy decision

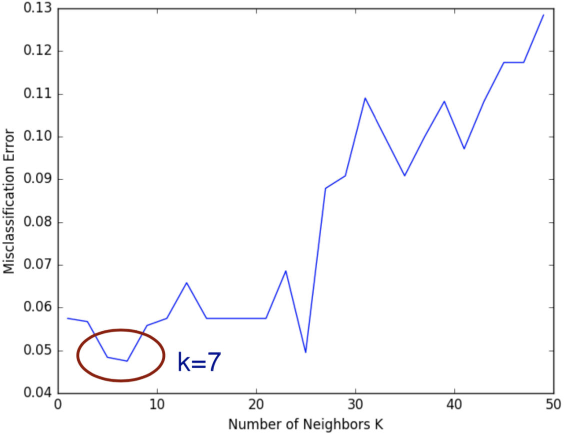
The idea behind using more than 1 neighbors is to average out the noise

Large k

- May lead to better prediction performance
- If we set k too large, we may end up looking at samples that are not neighbors (are far away from the point of interest)
- Also, computationally intensive. Why?
- Extreme case: set $k = n$ (number of points in the dataset)
  - For classification: the majority class
  - For regression: the average value

Choice of k

Set k by cross validation, by examining the misclassification error



Thumb rule

Choose $k = \sqrt{n}$

Advantages of kNN

- Training is very fast
  - Just store the training examples
  - Can use smart indexing procedures to speed-up testing
- The training data is part of the 'model'
  - Useful in case we want to do something else with it
- Quite robust to noisy data
  - Averaging k votes
- Can learn complex functions (implicitly)!

Drawbacks of kNN

- Memory requirements
  - Must store all training data
- Prediction can be slow (will figure it out by yourself in the lab)
  - Complexity of labeling 1 new data point: $O(knp)$
  - But kNN works best with lots of samples
  - Can we further improve the running time?
- Efficient data structures (e.g., k-D trees)
- Approximate solutions based on hashing!
- High dimensional data and the curse of dimensionality
  - Computation of the distance in a high dimensional space may become meaningless
  - Need more training data
  - Dimensionality reduction

This is a non-parametric method

Curse of dimensionality

- They suffer from the curse of dimensionality :

When the dimension increases
$\Rightarrow$ neighborhoods become empty
$\Rightarrow$ bad convergence rate

This is a non-parametric method

Curse of dimensionality

- They suffer from the curse of dimensionality :

  When the dimension increases
  $\Rightarrow$ neighborhoods become empty
  $\Rightarrow$ bad convergence rate

Theorem

Given n random points drawn in the hypercube $[0,1]^d$ then

$$\frac{\max_{i\neq j}\|X_i - X_j\|_p}{\min_{i\neq j}\|X_i - X_j\|_p} = 1 + O\left(\sqrt{\frac{d}{\log(n)}}\right)$$

- When d is large, all the points are almost equidistant...
- Nearest neighbors are meaningless!

Some more issues

- Normalize the scale of the attributes
- Simple option: linearly scale the range of each feature to be, e.g., in the range of [0,1]
- Linearly scale each dimension to have 0 mean and variance 1

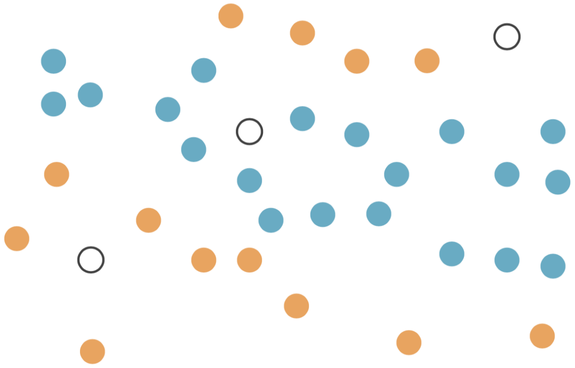
Decision boundary with kNN

Decision boundary in classification:

- Line separating the positive from negative regions

What decision boundary is the kNN building?

The nearest neighbors algorithm does not explicitly compute decision boundaries, but those can be inferred



Voronoi tesselation

Think about the 1NN.

Voronoi cell of x

- Set of all points of the space closer to x than any other point of the training set
- Shape?

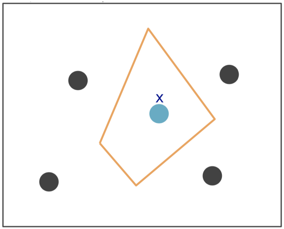
[1]Wikipedia: https://en.wikipedia.org/wiki/Voronoi_diagram

Voronoi tesselation

Think about the 1NN.

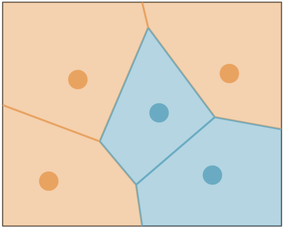
Voronoi cell of x

- Set of all points of the space closer to x than any other point of the training set
- Shape? Polyhedron



Voronoi tessellation (or diagram) of the space

Union of all Voronoi cells



1

[1]Wikipedia: https://en.wikipedia.org/wiki/Voronoi_diagram

kNN variants

Weighted kNN

- Weight the vote of each neighbor x_i according to the distance to the test point x

$$w_i = \frac{1}{d(x, x_i)^2}$$

- Other kernel functions can be used to weight the distance of neighbors