Kernel methods

Edouard Pauwels

M2-MAT SID

Have you already encountered kernels ?

$$k(x, y)$$

Learning setting

Suppervised learning
Prediction of a label in $\mathcal{Y}$. $\mathcal{X}$ is the input feature space.
$\mathcal{D}_n = \big\{(x_1, y_1), \dots, (x_n, y_n)\big\}$, is the learning sample.
Construct $f_n \colon \mathcal{X} \mapsto \mathcal{Y}$

Unsupervised learning :
Learning sample from the feature space $\mathcal{D}_n = \big\{x_1, \dots, x_n\big\} \subset \mathcal{X}$, infer properties of $\mathcal{X}$ (clustering, PCA), construct an oulier detector ...

Learning setting

Suppervised learning
Prediction of a label in $\mathcal{Y}$. $\mathcal{X}$ is the input feature space.
$\mathcal{D}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, is the learning sample.
Construct $f_n \colon \mathcal{X} \mapsto \mathcal{Y}$

Unsupervised learning :
Learning sample from the feature space $\mathcal{D}_n = \{x_1, \ldots, x_n\} \subset \mathcal{X}$, infer properties of $\mathcal{X}$ (clustering, PCA), construct an oulier detector ...

Kernels :
Induce a new representation of the feature space $\mathcal{X}$:

- Handle specific characteristics of $\mathcal{X}$ (e.g. non numeric data).
- A general framework for non linear modeling.

Learning setting

Suppervised learning
Prediction of a label in $\mathcal{Y}$. $\mathcal{X}$ is the input feature space.
$\mathcal{D}_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$, is the learning sample.
Construct $f_n \colon \mathcal{X} \mapsto \mathcal{Y}$

Unsupervised learning :
Learning sample from the feature space $\mathcal{D}_n = \{x_1, \dots, x_n\} \subset \mathcal{X}$, infer properties of $\mathcal{X}$ (clustering, PCA), construct an oulier detector . . .

Kernels :
Induce a new representation of the feature space $\mathcal{X}$:

- Handle specific characteristics of $\mathcal{X}$ (e.g. non numeric data).
- A general framework for non linear modeling.

Usage :
kernelized supervised learning, kernel smoothing, kernel density estimation, kernel PCA, spectral clustering . . .

What is a kernel ?

Denote by $\mathcal{X}$ a the space where your intput data lives.

What is a kernel ?

Denote by $\mathcal{X}$ a the space where your intput data lives.

- Most often it is $\mathbb{R}^p$.
- More complicated examples :
  - Sequences in an alphabet (DNA)
  - Graphs (molecules, social networks)
  - Large feature space (time series)
  - etc ...

What is a kernel ?

Denote by $\mathcal{X}$ a the space where your intput data lives.

- Most often it is $\mathbb{R}^p$.
- More complicated examples :
  - Sequences in an alphabet (DNA)
  - Graphs (molecules, social networks)
  - Large feature space (time series)
  - etc ...

What ? A kernel is a symmetric function

$$k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$$

$k(x, z) = k(z, x)$ is a measure of similarity of two inputs x, z (the larger the more similar).

Denote by $\mathcal{X}$ a the space where your intput data lives.

- Most often it is $\mathbb{R}^p$.
- More complicated examples :
  - Sequences in an alphabet (DNA)
  - Graphs (molecules, social networks)
  - Large feature space (time series)
  - etc ...

What ? A kernel is a symmetric function

$$k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$$

$k(x, z) = k(z, x)$ is a measure of similarity of two inputs x, z (the larger the more similar).

Why ? Generalize scalar product and Euclidean distances.

What is a kernel ?

Denote by $\mathcal{X}$ a the space where your intput data lives.

- Most often it is $\mathbb{R}^p$.
- More complicated examples :
  - Sequences in an alphabet (DNA)
  - Graphs (molecules, social networks)
  - Large feature space (time series)
  - etc ...

What ? A kernel is a symmetric function

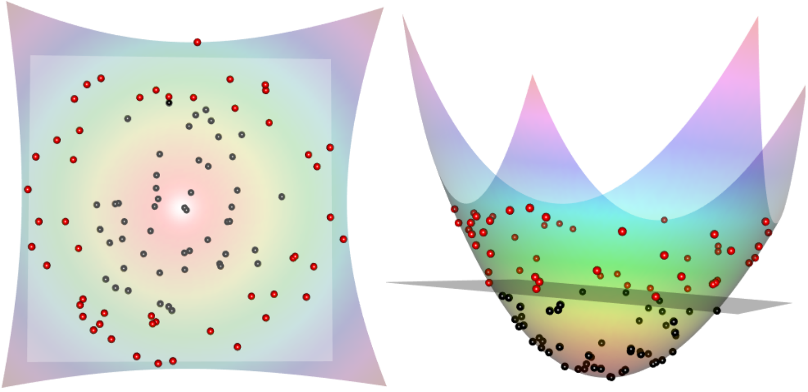
$$k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$$

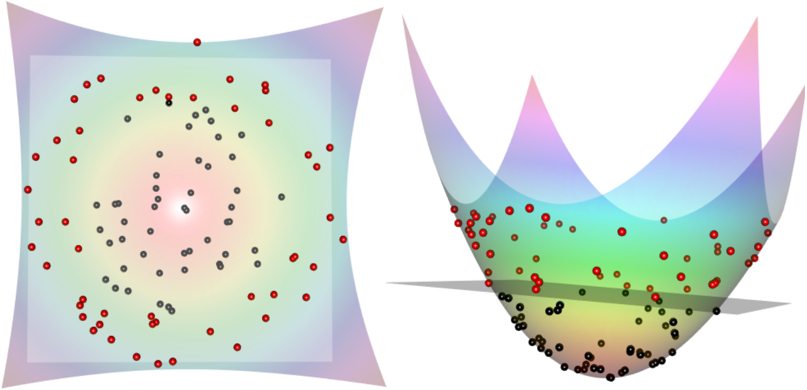
$k(x, z) = k(z, x)$ is a measure of similarity of two inputs x, z (the larger the more similar).

Why ? Generalize scalar product and Euclidean distances.

Kernels are used to

- Extend linear methods (supervised/ unsupervised) to nonlinear methods.
- Handle data which cannot be encoded by vectors (non numeric data, graphs).





Main idea : Build a non linear model by constructing a linear model in higher dimension.

Importance of feature space : non numeric data

```
X
['fndsuninsdunisdissidfundiudsuiffddussusniuifndnfsu',
 'idnsudfndidusuiuusidifisfnsdunsiuuuifudnsssfunsidu',
 'nddnnfdfnndfudfnfffsnfnfsnsdisnfuisuifsidfundinssn',
 'ffsndnunndsdnusidfunisdfiufinnundfdsunnunsudssfffs',
 'unudidiifsnndsndsinnuuisnnsnsdsusfuiufdnusdidfdunf',
 'suufffiiddiundiiuuudfddsdnsdnnnunddnffnindiuindisd',
 'fuisdussudduissufnsnnunsdnufudusfsusiufusiinsnuiid',
 'dssisffdnniifidniuffdfdiiisuffduffisfinuusidfundiu',
 'isdsuufsuusufnisdsdfsdunnuiididnddiuinsnndduiffuun',
 'ifuidfndinufunssunuifunsidffnifdffdsdnsuiffsffffnn',
 'uudfsuduufniinnsuiufnsdfdsufnfunsiddsuufifffnfsfnn',
 'dundffundfifiiuiuifnuuunuifnisfsuundsffiffsdfufdff',
 'fuufdnsinnuddfsnusdfnssfsiiuidfnninfunsidnsfnufusu',
 'susufsfinffnddudddsifunidiffnnndddniiunffsidfunnin',
```

Importance of feature space : non numeric data

```
X

['fndsuninsdunisdissidfundiudsuiffddussusniuifndnfsu',
 'idnsudfndidusuiuusidifisfnsdunsiuuuifudnsssfunsidu',
 'nddnnfdfnndfudfnfffsnfnfsnsdisnfuisuifsidfundinssn',
 'ffsndnunndsdnusidfunisdfiufinnundfdsunnunsudssfffs',
 'unudidiifsnndsndsinnuuisnnsnsdsusfuiufdnusdidfdunf',
 'suufffiiddiundiiuuudfddsdnsdnnnunddnffnindiuindisd',
 'fuisdussudduissufnsnnunsdnufudusfsusiufusiinsnuiid',
 'dssisffdnniifidniuffdfdiiisuffduffisfinuusidfundiu',
 'isdsuufsuusufnisdsdfsdunnuiididnddiuinsnndduiffuun',
 'ifuidfndinufunssunuifunsidffnifdffdsdnsuiffsffffnn',
 'uudfsuduufniinnsuiufnsdfdsufnfunsiddsuufifffnfsfnn',
 'dundffundfifiiuiuifnuuunuifnisfsuundsffiffsdfufdff',
 'fuufdnsinnuddfsnusdfnssfsiiuidfnninfunsidnsfnufusu',
 'susufsfinffnddudddsifunidiffnnndddniiunffsidfunnin',
```

Main idea : Handle features implicitely only through computation of similarities.

Recap on scalar product

x, z vectors in $\mathbb{R}^p$.

$$\langle x, z \rangle =$$

Recap on scalar product

x, z vectors in $\mathbb{R}^p$.

$$\langle x, z \rangle = \sum_{i=1}^{p} x[i]z[i] = x^T z$$

Recap on scalar product

x, z vectors in $\mathbb{R}^p$.

$$\langle x, z \rangle = \sum_{i=1}^{p} x[i]z[i] = x^T z$$

Symmetry :

Bilinearity :

Recap on scalar product

x, z vectors in $\mathbb{R}^p$.

$$\langle x, z \rangle = \sum_{i=1}^{p} x[i]z[i] = x^T z$$

Symmetry : $\langle x, z \rangle = \langle z, x \rangle$.
Bilinearity :

x, z vectors in $\mathbb{R}^p$.

$$\langle x, z \rangle = \sum_{i=1}^{p} x[i]z[i] = x^T z$$

Symmetry : $\langle x, z \rangle = \langle z, x \rangle$.
Bilinearity : $\langle x_1 + x_2, z \rangle = \langle x_1, z \rangle + \langle x_2, z \rangle$, $\quad \langle \alpha x, z \rangle = \alpha \langle x, z \rangle$, $\alpha \in \mathbb{R}$.

Recap on scalar product

x, z vectors in $\mathbb{R}^p$.

$$\langle x, z \rangle = \sum_{i=1}^{p} x[i]z[i] = x^T z$$

Symmetry : $\langle x, z \rangle = \langle z, x \rangle$.
Bilinearity : $\langle x_1 + x_2, z \rangle = \langle x_1, z \rangle + \langle x_2, z \rangle, \quad \langle \alpha x, z \rangle = \alpha \langle x, z \rangle,\ \alpha \in \mathbb{R}$.
Design : given a training sample $\mathcal{D}_n = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$ the design matrix represents samples by row :

$$X = \begin{pmatrix} - & x_1^T & - \\ - & x_2^T & - \\ & \vdots & \\ - & x_n^T & - \end{pmatrix} \in \mathbb{R}^{n \times p}$$

Recap on scalar product

x, z vectors in $\mathbb{R}^p$.

$$\langle x, z\rangle = \sum_{i=1}^{p} x[i]z[i] = x^T z$$

Symmetry : $\langle x, z\rangle = \langle z, x\rangle$.
Bilinearity : $\langle x_1 + x_2, z\rangle = \langle x_1, z\rangle + \langle x_2, z\rangle$, $\quad \langle \alpha x, z\rangle = \alpha \langle x, z\rangle$, $\alpha \in \mathbb{R}$.
Design : given a training sample $\mathcal{D}_n = \{x_1, \dots, x_n\} \subset \mathbb{R}^p$ the design matrix represents samples by row :

$$X = \begin{pmatrix} - & x_1^T & - \\ - & x_2^T & - \\ & \vdots & \\ - & x_n^T & - \end{pmatrix} \in \mathbb{R}^{n \times p}$$

We have for example for $z \in \mathbb{R}^p$.

Xz (size ?) $=$

Recap on scalar product

x, z vectors in $\mathbb{R}^p$.

$$\langle x, z \rangle = \sum_{i=1}^{p} x[i]z[i] = x^T z$$

Symmetry : $\langle x, z \rangle = \langle z, x \rangle$.
Bilinearity : $\langle x_1 + x_2, z \rangle = \langle x_1, z \rangle + \langle x_2, z \rangle, \quad \langle \alpha x, z \rangle = \alpha \langle x, z \rangle, \ \alpha \in \mathbb{R}$.
Design : given a training sample $\mathcal{D}_n = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$ the design matrix represents samples by row :

$$X = \begin{pmatrix} - & x_1^T & - \\ - & x_2^T & - \\ & \vdots & \\ - & x_n^T & - \end{pmatrix} \in \mathbb{R}^{n \times p}$$

We have for example for $z \in \mathbb{R}^p$.

$$Xz \text{ (size ?)} = \begin{pmatrix} \langle x_1, z \rangle \\ \langle x_2, z \rangle \\ \vdots \\ \langle x_n, z \rangle \end{pmatrix}$$

Recap on scalar product

x, z vectors in $\mathbb{R}^p$.

$$\langle x, z \rangle = \sum_{i=1}^{p} x[i]z[i] = x^T z$$

Symmetry : $\langle x, z \rangle = \langle z, x \rangle$.
Bilinearity : $\langle x_1 + x_2, z \rangle = \langle x_1, z \rangle + \langle x_2, z \rangle, \quad \langle \alpha x, z \rangle = \alpha \langle x, z \rangle, \ \alpha \in \mathbb{R}$.
Design : given a training sample $\mathcal{D}_n = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$ the design matrix represents samples by row :

$$X = \begin{pmatrix} - & x_1^T & - \\ - & x_2^T & - \\ & \vdots & \\ - & x_n^T & - \end{pmatrix} \in \mathbb{R}^{n \times p}$$

We have for example for $z \in \mathbb{R}^p$.

$$Xz \text{ (size ?)} = \begin{pmatrix} \langle x_1, z \rangle \\ \langle x_2, z \rangle \\ \vdots \\ \langle x_n, z \rangle \end{pmatrix} \qquad XX^T \text{ (size ?)} =$$

Recap on scalar product

x, z vectors in $\mathbb{R}^p$.

$$\langle x, z\rangle = \sum_{i=1}^{p} x[i]z[i] = x^T z$$

Symmetry : $\langle x, z\rangle = \langle z, x\rangle$.
Bilinearity : $\langle x_1 + x_2, z\rangle = \langle x_1, z\rangle + \langle x_2, z\rangle, \quad \langle \alpha x, z\rangle = \alpha \langle x, z\rangle,\ \alpha \in \mathbb{R}$.
Design : given a training sample $\mathcal{D}_n = \{x_1, \dots, x_n\} \subset \mathbb{R}^p$ the design matrix represents samples by row :

$$X = \begin{pmatrix} - & x_1^T & - \\ - & x_2^T & - \\ & \vdots & \\ - & x_n^T & - \end{pmatrix} \in \mathbb{R}^{n\times p}$$

We have for example for $z \in \mathbb{R}^p$.

$$Xz \text{ (size ?)} = \begin{pmatrix} \langle x_1, z\rangle \\ \langle x_2, z\rangle \\ \vdots \\ \langle x_n, z\rangle \end{pmatrix} \qquad XX^T \text{ (size ?)} = \begin{pmatrix} \langle x_1, x_1\rangle & \langle x_1, x_2\rangle & \dots & \langle x_1, x_n\rangle \\ \langle x_2, x_1\rangle & \langle x_2, x_2\rangle & \dots & \langle x_2, x_n\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle x_n, x_1\rangle & \langle x_n, x_2\rangle & \dots & \langle x_n, x_n\rangle \end{pmatrix}$$

Outline

Gram Matrix

Kernel : throughout $k\colon \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a symmetric function.

Input sample : $\mathcal{D}_n = \{x_1, \dots, x_n\} \subset \mathcal{X}$,

Gram Matrix

Kernel : throughout $k\colon \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a symmetric function.

Input sample : $\mathcal{D}_n = \{x_1, \ldots, x_n\} \subset \mathcal{X}$,

Exercice (fil rouge) : try to explicit all the notion with the linear kernel $(x, z) \mapsto x^T z$ and $\mathcal{D}_n$ given by the design matrix $X \in \mathbb{R}^{n \times p}$.

Gram Matrix

Kernel : throughout $k \colon \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a symmetric function.

Input sample : $\mathcal{D}_n = \{x_1, \ldots, x_n\} \subset \mathcal{X}$,

Exercice (fil rouge) : try to explicit all the notion with the linear kernel $(x, z) \mapsto x^T z$ and $\mathcal{D}_n$ given by the design matrix $X \in \mathbb{R}^{n \times p}$.

Gram matrix : representation by pairwise comparison (symmetric ?)

$$K_n = (k(x_i, x_j))_{i,j=1}^n = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \ldots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \ldots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \ldots & k(x_n, x_n) \end{pmatrix} \in \mathbb{R}^{n \times n}$$

Kernel : throughout $k\colon \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a symmetric function.

Input sample : $\mathcal{D}_n = \{x_1, \dots, x_n\} \subset \mathcal{X}$,

Exercice (fil rouge) : try to explicit all the notion with the linear kernel $(x, z) \mapsto x^T z$ and $\mathcal{D}_n$ given by the design matrix $X \in \mathbb{R}^{n\times p}$.

Gram matrix : representation by pairwise comparison (symmetric ?)

$$K_n = (k(x_i, x_j))_{i,j=1}^n = \begin{pmatrix} k(x_1,x_1) & k(x_1,x_2) & \dots & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & \dots & k(x_2,x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n,x_1) & k(x_n,x_2) & \dots & k(x_n,x_n) \end{pmatrix} \in \mathbb{R}^{n\times n}$$

Fil rouge : what is the Gram matrix for the linear kernel (design matrix $X \in \mathbb{R}^{n\times p}$) ?

Gram Matrix

Kernel : throughout $k\colon \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a symmetric function.

Input sample : $\mathcal{D}_n = \{x_1, \dots, x_n\} \subset \mathcal{X}$,

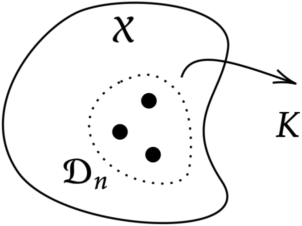
Exercice (fil rouge) : try to explicit all the notion with the linear kernel $(x, z) \mapsto x^T z$ and $\mathcal{D}_n$ given by the design matrix $X \in \mathbb{R}^{n \times p}$.

Gram matrix : representation by pairwise comparison (symmetric ?)

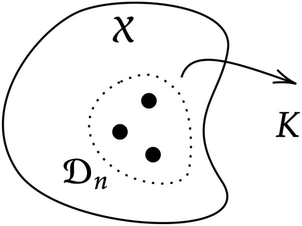
$$K_n = (k(x_i, x_j))_{i,j=1}^n = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{pmatrix} \in \mathbb{R}^{n \times n}$$

Fil rouge : what is the Gram matrix for the linear kernel (design matrix $X \in \mathbb{R}^{n \times p}$) ?

$$\begin{pmatrix} \langle x_1, x_1 \rangle & \langle x_1, x_2 \rangle & \dots & \langle x_1, x_n \rangle \\ \langle x_2, x_1 \rangle & \langle x_2, x_2 \rangle & \dots & \langle x_2, x_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle x_n, x_1 \rangle & \langle x_n, x_2 \rangle & \dots & \langle x_n, x_n \rangle \end{pmatrix} = XX^T \in \mathbb{R}^{n \times n}$$



$$K = \begin{pmatrix} 1 & 0.1 & 0.5 \\ 0.1 & 1 & 0.3 \\ 0.5 & 0.3 & 1 \end{pmatrix}$$



$$K = \begin{pmatrix} 1 & 0.1 & 0.5 \\ 0.1 & 1 & 0.3 \\ 0.5 & 0.3 & 1 \end{pmatrix}$$

Example : There is no easy scalar product on the space of strings. But we can measure similarity (number of common substrings).

```
X
['fndsuninsdunisdissidfundiudsuiffddussusniuifndnfsu',
 'idnsudfndidusuiuusidifisfnsdunsiuuuifudnsssfunsidu',
 'nddnnfdfnndfudfnfffsnfnfsnsdisnfuisuifsidfundinssn',
```

Symmetric matrix : $S \in \mathbb{R}^{n \times n}$ is symmetric if . . .

Symmetric matrix : $S \in \mathbb{R}^{n \times n}$ is symmetric if $S^T = S$.

Symmetric matrix : $S \in \mathbb{R}^{n \times n}$ is symmetric if $S^T = S$.

Eigenvalues : If $S \in \mathbb{R}^{n \times n}$ is symmetric then it is ...

Symmetric matrix : $S \in \mathbb{R}^{n \times n}$ is symmetric if $S^T = S$.

Eigenvalues : If $S \in \mathbb{R}^{n \times n}$ is symmetric then it is diagonalizable with real eigenvalues

Symmetric matrix : $S \in \mathbb{R}^{n \times n}$ is symmetric if $S^T = S$.

Eigenvalues : If $S \in \mathbb{R}^{n \times n}$ is symmetric then it is diagonalizable with real eigenvalues

Positivity : A symmetric matrix $S \in \mathbb{R}^{n \times n}$ is positive semidefinite (psd) if one of the following equivalent condition holds :

- ...
- ...

Recap on positive semidefinite matrices

Symmetric matrix : $S \in \mathbb{R}^{n \times n}$ is symmetric if $S^T = S$.

Eigenvalues : If $S \in \mathbb{R}^{n \times n}$ is symmetric then it is diagonalizable with real eigenvalues

Positivity : A symmetric matrix $S \in \mathbb{R}^{n \times n}$ is positive semidefinite (psd) if one of the following equivalent condition holds :

- For any $w \in \mathbb{R}^n$, $w^T S w \geq 0$.
- All the eigenvalues of S are non negative.

Symmetric matrix : $S \in \mathbb{R}^{n \times n}$ is symmetric if $S^T = S$.

Eigenvalues : If $S \in \mathbb{R}^{n \times n}$ is symmetric then it is diagonalizable with real eigenvalues

Positivity : A symmetric matrix $S \in \mathbb{R}^{n \times n}$ is positive semidefinite (psd) if one of the following equivalent condition holds :

- For any $w \in \mathbb{R}^n$, $w^T S w \geq 0$.
- All the eigenvalues of S are non negative.

Fil rouge : is the gram matrix of the linear kernel $K_n = XX^T$ psd ?

Recap on positive semidefinite matrices

Symmetric matrix : $S \in \mathbb{R}^{n \times n}$ is symmetric if $S^T = S$.

Eigenvalues : If $S \in \mathbb{R}^{n \times n}$ is symmetric then it is diagonalizable with real eigenvalues

Positivity : A symmetric matrix $S \in \mathbb{R}^{n \times n}$ is positive semidefinite (psd) if one of the following equivalent condition holds :

- For any $w \in \mathbb{R}^n$, $w^T S w \geq 0$.
- All the eigenvalues of S are non negative.

Fil rouge : is the gram matrix of the linear kernel $K_n = XX^T$ psd ?

$$w^T XX^T w = (X^T w)^T X^T w = \left\langle X^T w, X^T w \right\rangle = \|X^T w\|^2 \geq 0$$

Positive definite kernel

k is called positive definite if the gram matrix is positive semi-definite

- for any n
- for any dataset $\mathcal{D}_n = \{x_1, \ldots, x_n\} \subset \mathcal{X}$.

Positive definite kernel

k is called positive definite if the gram matrix is positive semi-definite

- for any n
- for any dataset $\mathcal{D}_n = \{x_1, \ldots, x_n\} \subset \mathcal{X}$.

Why ? Because we want k to behave similarly as a scalar product.

Positive definite kernel

k is called positive definite if the gram matrix is positive semi-definite

- for any n
- for any dataset $\mathcal{D}_n = \{x_1, \ldots, x_n\} \subset \mathcal{X}$.

Why ? Because we want k to behave similarly as a scalar product.

Fil rouge : is the linear kernel positive definite ?

Examples

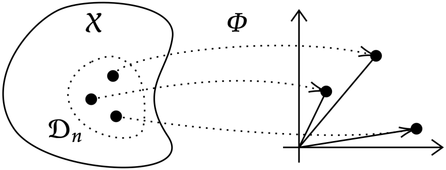
Linear kernel : $k\colon (x, z) \mapsto x^T z$ is positive definite.

Linear kernel : $k\colon (x,z) \mapsto x^T z$ is positive definite.

Feature map : Let $\mathcal{X}$ is any set and $\Phi\colon \mathcal{X} \mapsto \mathbb{R}^p$, then

$$k\colon (x,z) \mapsto \langle \Phi(x), \Phi(z) \rangle = \Phi(x)^T \Phi(z).$$

is positive definite.

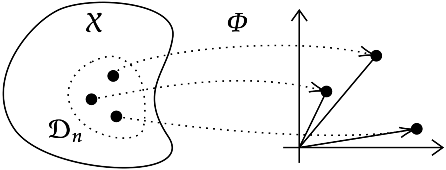


Linear kernel : $k\colon (x,z) \mapsto x^T z$ is positive definite.

Feature map : Let $\mathcal{X}$ is any set and $\Phi\colon \mathcal{X} \mapsto \mathbb{R}^p$, then

$$k\colon (x,z) \mapsto \langle \Phi(x), \Phi(z) \rangle = \Phi(x)^T \Phi(z).$$

is positive definite.



All positive definite kernels are of this form.

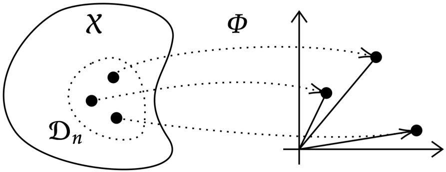
Feature space

Theorem (Aronszajn, 1950) : k is positive definite on $\mathcal{X}$ if and only if there exists a Hilbert space $\mathcal{H}$ and a mapping

$$\Phi : \mathcal{X} \mapsto \mathcal{H}$$

such that for all $x, z \in \mathcal{X}$,

$$k(x,z) = \langle \Phi(x), \Phi(z) \rangle_{\mathcal{H}} \,.$$

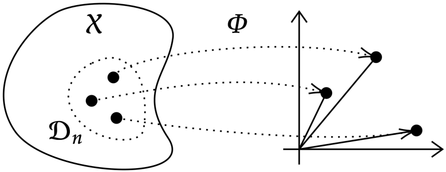


Theorem (Aronszajn, 1950) : k is positive definite on $\mathcal{X}$ if and only if there exists a Hilbert space $\mathcal{H}$ and a mapping

$$\Phi : \mathcal{X} \mapsto \mathcal{H}$$

such that for all $x, z \in \mathcal{X}$,

$$k(x,z) = \langle \Phi(x), \Phi(z) \rangle_{\mathcal{H}} .$$



Warning : $\mathcal{H}$ could have infinite dimension. Φ is only manipulated implicitely through k.

Kernel trick

Start with : a "linear" algorithm formulated only in terms or pairwise inner products $\langle \cdot, \cdot \rangle$.
Kernelized version : replace $\langle \cdot, \cdot \rangle$ by a positive definite kernel $k(\cdot, \cdot)$.

Kernel trick

Start with : a "linear" algorithm formulated only in terms or pairwise inner products $\langle \cdot, \cdot \rangle$.
Kernelized version : replace $\langle \cdot, \cdot \rangle$ by a positive definite kernel $k(\cdot, \cdot)$.

Example : An algorithm based only onth Gram matrix $XX^T \in \mathbb{R}^{n \times n}$ can be obtained by remplacing it by $K_n \in \mathbb{R}^{n \times n}$.

Start with : a "linear" algorithm formulated only in terms or pairwise inner products $\langle \cdot, \cdot \rangle$.
Kernelized version : replace $\langle \cdot, \cdot \rangle$ by a positive definite kernel $k(\cdot, \cdot)$.

Example : An algorithm based only onth Gram matrix $XX^T \in \mathbb{R}^{n \times n}$ can be obtained by remplacing it by $K_n \in \mathbb{R}^{n \times n}$.

Feature space interpretion : This amounts to manipulate a different training set $\mathcal{D}_n = \{\Phi(x_1), \ldots, \Phi(x_n)\}$, which is possibly infinite dimensional.

Kernel trick

Start with : a "linear" algorithm formulated only in terms or pairwise inner products $\langle \cdot, \cdot \rangle$.
Kernelized version : replace $\langle \cdot, \cdot \rangle$ by a positive definite kernel $k(\cdot, \cdot)$.

Example : An algorithm based only onth Gram matrix $XX^T \in \mathbb{R}^{n \times n}$ can be obtained by remplacing it by $K_n \in \mathbb{R}^{n \times n}$.

Feature space interpretion : This amounts to manipulate a different training set $\mathcal{D}_n = \{\Phi(x_1), \ldots, \Phi(x_n)\}$, which is possibly infinite dimensional.

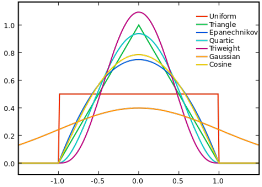
Remark : No need to compute Φ explicitely if the original algorithm only use values of scalar products.

Examples of positive definite kernels

- Gaussian kernel : $(x, z) \mapsto e^{\frac{-\|x-z\|^2}{\sigma^2}}$, $\sigma > 0$.
- Polynomial kernel : $(x, z) \mapsto (c + x^t z)^d$, $d \in \mathbb{N}$, $c \geq 0$.
- Laplacian kernel : $(x, z) \mapsto e^{\frac{-\|x-z\|}{\sigma}}$, $\sigma > 0$.

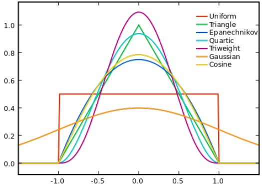
Examples of positive definite kernels

- Gaussian kernel : $(x, z) \mapsto e^{\frac{-\|x-z\|^2}{\sigma^2}}$, $\sigma > 0$.
- Polynomial kernel : $(x, z) \mapsto (c + x^t z)^d$, $d \in \mathbb{N}$, $c \geq 0$.
- Laplacian kernel : $(x, z) \mapsto e^{\frac{-\|x-z\|}{\sigma}}$, $\sigma > 0$.
- Many functions of the form $k(x, z) = \rho(x - z)$.



Examples of positive definite kernels

- Gaussian kernel : $(x, z) \mapsto e^{\frac{-\|x-z\|^2}{\sigma^2}}$, $\sigma > 0$.
- Polynomial kernel : $(x, z) \mapsto (c + x^t z)^d$, $d \in \mathbb{N}$, $c \geq 0$.
- Laplacian kernel : $(x, z) \mapsto e^{\frac{-\|x-z\|}{\sigma}}$, $\sigma > 0$.
- Many functions of the form $k(x, z) = \rho(x - z)$.

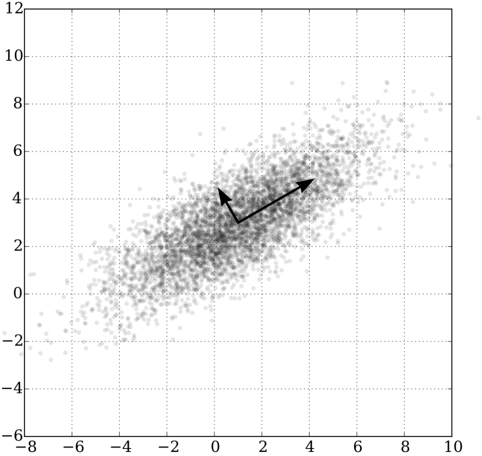


Further examples include

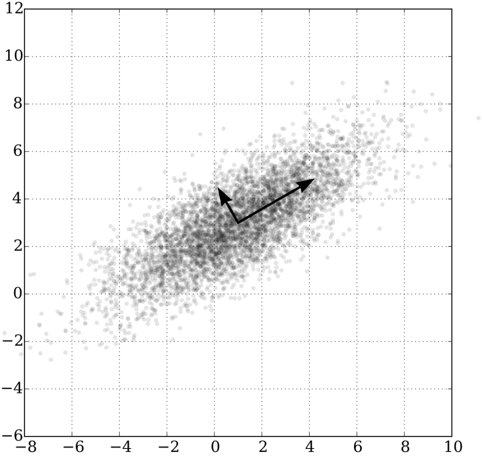
- Kernels for strings
- Kernels for graphs
- Kernels on graphs
- ...

Outline

Principal component analysis

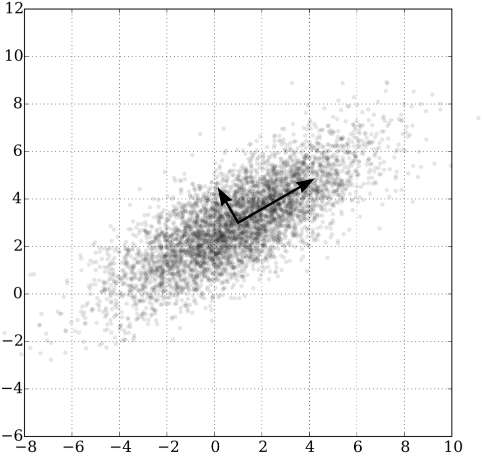


Principal component analysis



How is it done ?

Principal component analysis



How is it done ? Simultaneous diagonalization of covariance $X^T X$ and Gram XX^T matrices.

Kernel PCA

PCA : $X \in \mathbb{R}^{n \times p}$, design matrix. $XX^T \in \mathbb{R}^{n \times n}$ the gram matrix.
First step :

Kernel PCA

PCA : $X \in \mathbb{R}^{n \times p}$, design matrix. $XX^T \in \mathbb{R}^{n \times n}$ the gram matrix.
First step : centering the design.

Kernel PCA

PCA : $X \in \mathbb{R}^{n\times p}$, design matrix. $XX^T \in \mathbb{R}^{n\times n}$ the gram matrix.
First step : centering the design.

Mean Vector :

$$m = X^T \mathbf{1}/n$$

where $\mathbf{1}$ is the vector of all 1 in dimension n.

PCA : $X \in \mathbb{R}^{n \times p}$, design matrix. $XX^T \in \mathbb{R}^{n \times n}$ the gram matrix.
First step : centering the design.

Mean Vector :

$$m = X^T \mathbf{1}/n$$

where $\mathbf{1}$ is the vector of all 1 in dimension n.

Centered design :

$$\tilde{X} = X - \mathbf{1}m^T = X - \mathbf{1}\mathbf{1}^T/nX = X - UX$$

where $U \in \mathbb{R}^{n \times n}$ has constant entries $1/n$.

Kernel PCA

PCA : $X \in \mathbb{R}^{n \times p}$, design matrix. $XX^T \in \mathbb{R}^{n \times n}$ the gram matrix.
First step : centering the design.

Mean Vector :

$$m = X^T \mathbf{1}/n$$

where $\mathbf{1}$ is the vector of all 1 in dimension n.

Centered design :

$$\tilde{X} = X - \mathbf{1}m^T = X - \mathbf{1}\mathbf{1}^T/nX = X - UX$$

where $U \in \mathbb{R}^{n \times n}$ has constant entries $1/n$.

Centered gram matrix :

$$\tilde{X}\tilde{X}^T = (X - UX)(X - UX)^T = XX^T - UXX^T - XX^T U + UXX^T U$$

PCA : $X \in \mathbb{R}^{n \times p}$, design matrix. $XX^T \in \mathbb{R}^{n \times n}$ the gram matrix.
First step : centering the design.

Mean Vector :

$$m = X^T \mathbf{1}/n$$

where $\mathbf{1}$ is the vector of all 1 in dimension n.

Centered design :

$$\tilde{X} = X - \mathbf{1}m^T = X - \mathbf{1}\mathbf{1}^T/nX = X - UX$$

where $U \in \mathbb{R}^{n \times n}$ has constant entries $1/n$.

Centered gram matrix :

$$\tilde{X}\tilde{X}^T = (X - UX)(X - UX)^T = XX^T - UXX^T - XX^T U + UXX^T U$$

Kernel trick : Centering in feature space using kernel k and Gram matrix K_n

$$\tilde{K}_n = K_n - UK_n - K_n U + UK_n U$$

Kernel PCA

PCA : $X^T X$ centered gram matrix.

PCA : $X^T X$ centered gram matrix.

Eigendecomposition :

- $v_1 \in \mathbb{R}^n$ eigenvector associated to $\lambda_1 \geq 0$, the largest eigenvalue of XX^T with $\|v_1\| = 1$.
- $v_2 \in \mathbb{R}^n$ eigenvector associated to $\lambda_2 \geq 0$, the second largest eigenvalue of XX^T with $\|v_2\| = 1$.

Kernel PCA

PCA : X^TX centered gram matrix.

Eigendecomposition :

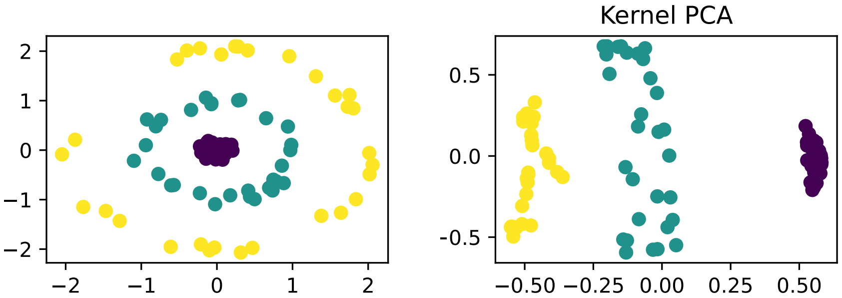
- $v_1 \in \mathbb{R}^n$ eigenvector associated to $\lambda_1 \geq 0$, the largest eigenvalue of XX^T with $\|v_1\| = 1$.
- $v_2 \in \mathbb{R}^n$ eigenvector associated to $\lambda_2 \geq 0$, the second largest eigenvalue of XX^T with $\|v_2\| = 1$.

Observations in principal plan : Coordinates of the projection given by $\sqrt{\lambda_1}v_1$ and $\sqrt{\lambda_2}v_2$ vectors in $\mathbb{R}^n$.

Kernel PCA : given K_n,

- Center : $K_n \leftarrow K_n - UK_n - K_nU + UK_nU$.
- Eigendecomposition of K_n : $\lambda_1, \lambda_2 \in \mathbb{R}$, $v_1, v_2 \in \mathbb{R}^n$.
- Principal plan representation : $\sqrt{\lambda_1}v_1$ and $\sqrt{\lambda_2}v_2$

Nonlinear PCA



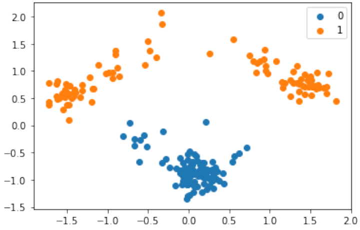
Kernel PCA : example (practical session)

How to get a graphical representation of a dataset of strings ?

Kernel PCA : example (practical session)

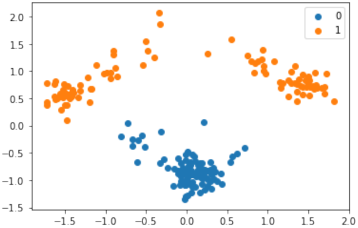
How to get a graphical representation of a dataset of strings ?

```
X
['fndsuninsdunisdissidfundiudsuiffddussusniuifndnfsu',
 'idnsudfndidusuiuusidifisfnsdunsiuuuifudnsssfunsidu',
 'nddnnfdfnndfudfnfffsnfnfsnsdisnfuisuifsidfundinssn',
 'ffsndnunndsdnusidfunisdfiufinnundfdsunnunsudssfffs',
 'unudidiifsnndsndsinnuuisnnsnsdsusfuiufdnusdidfdunf',
 'suufffiiddiundiiuuudfddsdnsdnnnunddnffnindiuindisd',
 'fuisdussudduissufnsnnunsdnufudusfsusiufusiinsnuiid',
 'dssisffdnniifidniuffdfdiiisuffduffisfinuusidfundiu',
 'isdsuufsuusufnisdsdfsdunnuiididnddiuinsnndduiffuun',
 'ifuidfndinufunssunuifunsidffnifdffdsdnsuiffsffffnn',
 'uudfsuduufniinnsuiufnsdfdsufnfunsiddsuufifffnfsfnn',
 'dundffundfifiiuiuifnuuunuifnisfsuundsffiffsdfufdff',
 'fuufdnsinnuddfsnusdfnssfsiiuidfnninfunsidnsfnufusu',
 'susufsfinffnddudddsifunidiffnnndddniiunffsidfunnin',
```



How to get a graphical representation of a dataset of strings ?

```
X
['fndsuninsdunisdissidfundiudsuiffddussusniuifndnfsu',
 'idnsudfndidusuiuusidifisfnsdunsiuuuifudnsssfunsidu',
 'nddnnfdfnndfudfnfffsnfnfsnsdisnfuisuifsidfundinssn',
 'ffsndnunndsdnusidfunisdfiufinnundfdsunnunsudssfffs',
 'unudidiifsnndsndsinnuuisnnsnsdsusfuiufdnusdidfdunf',
 'suufffiiddiundiiuuudfddsdnsdnnnunddnffnindiuindisd',
 'fuisdussudduissufnsnnunsdnufudusfsusiufusiinsnuiid',
 'dssisffdnniifidniuffdfdiiisuffduffisfinuusidfundiu',
 'isdsuufsuusufnisdsdfsdunnuiididnddiuinsnndduiffuun',
 'ifuidfndinufunssunuifunsidffnifdffdsdnsuiffsffffnn',
 'uudfsuduufniinnsuiufnsdfdsufnfunsiddsuufifffnfsfnn',
 'dundffundfifiiuiuifnuuunuifnisfsuundsffiffsdfufdff',
 'fuufdnsinnuddfsnusdfnssfsiiuidfnninfunsidnsfnufusu',
 'susufsfinffnddudddsifunidiffnnndddniiunffsidfunnin',
```



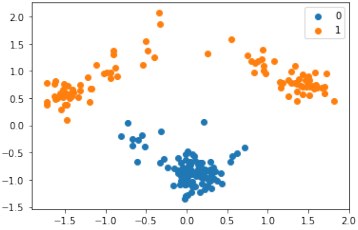
0 class : random strings of length 30 with letters `s`,`i`,`d`,`f`,`u`,`n`.

1 class : same but contain `sidfun` or `funsid`.

Kernel PCA : example (practical session)

How to get a graphical representation of a dataset of strings ?

```
X
['fndsuninsdunisdissidfundiudsuiffddussusniuifndnfsu',
 'idnsudfndidusuiuusidifisfnsdunsiuuuifudnsssfunsidu',
 'nddnnfdfnndfudfnfffsnfnfsnsdisnfuisuifsidfundinssn',
 'ffsndnunndsdnusidfunisdfiufinnundfdsunnunsudssfffs',
 'unudidiifsnndsndsinnuuisnnsnsdsusfuiufdnusdidfdunf',
 'suufffiiddiundiiuuudfddsdnsdnnnunddnffnindiuindisd',
 'fuisdussudduissufnsnnunsdnufudusfsusiufusiinsnuiid',
 'dssisffdnniifidniuffdfdiiisuffduffisfinuusidfundiu',
 'isdsuufsuusufnisdsdfsdunnuiididnddiuinsnndduiffuun',
 'ifuidfndinufunssunuifunsidffnifdffdsdnsuiffsffffnn',
 'uudfsuduufniinnsuiufnsdfdsufnfunsiddsuufifffnfsfnn',
 'dundffundfifiiuiuifnuuunuifnisfsuundsffiffsdfufdff',
 'fuufdnsinnuddfsnusdfnssfsiiuidfnninfunsidnsfnufusu',
 'susufsfinffnddudddsifunidiffnnndddniiunffsidfunnin',
```



0 class : random strings of length 30 with letters `s`,`i`,`d`,`f`,`u`,`n`.
1 class : same but contain `sidfun` or `funsid`.

k number of common substrings of a given size.

Outline

Kernel trick : alternative view

Construct a nonlinear algorithm by replacing $\langle \cdot, \cdot \rangle$ by a positive definite kernel $k(\cdot, \cdot)$.

Example : An algorithm based only onth Gram matrix $XX^T \in \mathbb{R}^{n \times n}$ can be obtained by remplacing it by $K_n \in \mathbb{R}^{n \times n}$.

Feature space interpretion : different training set $\mathcal{D}_n = \left\{ \Phi(x_1), \ldots, \Phi(x_n) \right\}$, possibly infinite dimensional. No need to comput Φ explicitely, just $k(\cdot, \cdot)$.

Construct a nonlinear algorithm by replacing $\langle \cdot, \cdot \rangle$ by a positive definite kernel $k(\cdot, \cdot)$.

Example : An algorithm based only onth Gram matrix $XX^T \in \mathbb{R}^{n \times n}$ can be obtained by remplacing it by $K_n \in \mathbb{R}^{n \times n}$.

Feature space interpretion : different training set $\mathcal{D}_n = \{\Phi(x_1), \dots, \Phi(x_n)\}$, possibly infinite dimensional. No need to comput Φ explicitely, just $k(\cdot, \cdot)$.

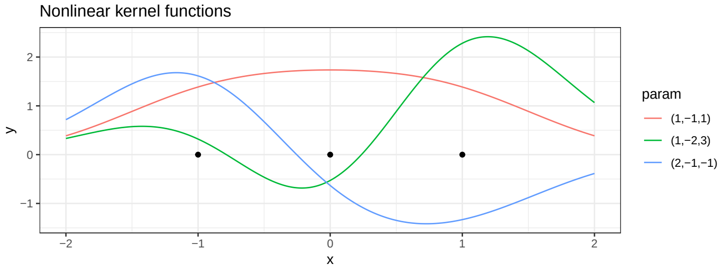
Alternative view : Replace a linear function $f_w : x \mapsto \langle w, x \rangle$ with parameter w by a nonlinear function which depends on the dataset :

$$f_\alpha : x \mapsto \sum_{i=1}^{n} \alpha_i k(x_i, x)$$

Illustration

Gaussian kernel : $k\colon (x,z) \mapsto e^{\frac{-\|x-z\|^2}{\sigma^2}}$, $\sigma = 1$,
Inputs dataset : $x_1 = -1,\ x_2 = 0,\ x_3 = 1.$

$$f_\alpha : x \mapsto \sum_{i=1}^{3} \alpha_i k(x_i, x)$$



Inputs dataset : $\mathcal{D}_n = (x_1, \ldots, x_n)$.
Kernel function : $k\colon (x, z) \mapsto k(x, z)$, symmetric, positive definite
Parameterized functions : $f_\alpha : x \mapsto \sum_{i=1}^n \alpha_i k(x_i, x)$, $\alpha \in \quad$.

Linear algebra

Inputs dataset : $\mathcal{D}_n = (x_1, \ldots, x_n)$.
Kernel function : $k\colon (x, z) \mapsto k(x, z)$, symmetric, positive definite
Parameterized functions : $f_\alpha : x \mapsto \sum_{i=1}^n \alpha_i k(x_i, x)$, $\alpha \in \mathbb{R}^n$.

Inputs dataset : $\mathcal{D}_n = (x_1, \dots, x_n)$.
Kernel function : $k\colon (x,z) \mapsto k(x,z)$, symmetric, positive definite
Parameterized functions : $f_\alpha : x \mapsto \sum_{i=1}^n \alpha_i k(x_i, x)$, $\alpha \in \mathbb{R}^n$.

Gram matrix : representation by pairwise comparison (symmetric ?)

$$K_n = (k(x_i, x_j))_{i,j=1}^n = \begin{pmatrix} k(x_1,x_1) & k(x_1,x_2) & \dots & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & \dots & k(x_2,x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n,x_1) & k(x_n,x_2) & \dots & k(x_n,x_n) \end{pmatrix} \in \mathbb{R}^{n \times n}$$

Inputs dataset : $\mathcal{D}_n = (x_1, \dots, x_n)$.
Kernel function : $k\colon (x,z) \mapsto k(x,z)$, symmetric, positive definite
Parameterized functions : $f_\alpha : x \mapsto \sum_{i=1}^n \alpha_i k(x_i, x)$, $\alpha \in \mathbb{R}^n$.

Gram matrix : representation by pairwise comparison (symmetric ?)

$$K_n = (k(x_i, x_j))_{i,j=1}^n = \begin{pmatrix} k(x_1,x_1) & k(x_1,x_2) & \dots & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & \dots & k(x_2,x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n,x_1) & k(x_n,x_2) & \dots & k(x_n,x_n) \end{pmatrix} \in \mathbb{R}^{n\times n}$$

For $\alpha \in \mathbb{R}^n$,

$$K_n\alpha = \begin{pmatrix} \sum_{i=1}^n \alpha_i K(x_i, x_1) \\ \vdots \\ \sum_{i=1}^n \alpha_i K(x_i, x_n) \end{pmatrix} =$$

Inputs dataset : $\mathcal{D}_n = (x_1, \dots, x_n)$.
Kernel function : $k \colon (x, z) \mapsto k(x, z)$, symmetric, positive definite
Parameterized functions : $f_\alpha : x \mapsto \sum_{i=1}^n \alpha_i k(x_i, x)$, $\alpha \in \mathbb{R}^n$.

Gram matrix : representation by pairwise comparison (symmetric ?)

$$K_n = (k(x_i, x_j))_{i,j=1}^n = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{pmatrix} \in \mathbb{R}^{n \times n}$$

For $\alpha \in \mathbb{R}^n$,

$$K_n \alpha = \begin{pmatrix} \sum_{i=1}^n \alpha_i K(x_i, x_1) \\ \vdots \\ \sum_{i=1}^n \alpha_i K(x_i, x_n) \end{pmatrix} = \begin{pmatrix} f_\alpha(x_1) \\ \vdots \\ f_\alpha(x_n) \end{pmatrix}$$

Inputs dataset : $\mathcal{D}_n = (x_1, \dots, x_n)$.
Kernel function : $k \colon (x, z) \mapsto k(x, z)$, symmetric, positive definite
Parameterized functions : $f_\alpha : x \mapsto \sum_{i=1}^n \alpha_i k(x_i, x)$, $\alpha \in \mathbb{R}^n$.

Gram matrix : representation by pairwise comparison (symmetric ?)

$$K_n = (k(x_i, x_j))_{i,j=1}^n = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{pmatrix} \in \mathbb{R}^{n \times n}$$

For $\alpha \in \mathbb{R}^n$,

$$K_n \alpha = \begin{pmatrix} \sum_{i=1}^n \alpha_i K(x_i, x_1) \\ \vdots \\ \sum_{i=1}^n \alpha_i K(x_i, x_n) \end{pmatrix} = \begin{pmatrix} f_\alpha(x_1) \\ \vdots \\ f_\alpha(x_n) \end{pmatrix}$$

Setting $\kappa_n \colon \mathbb{R}^p \to \mathbb{R}^n$, such that $\kappa_n(x) = (k(x_i, x))_{i=1}^n$, we have

$$\langle \alpha, \kappa_n(x) \rangle = \alpha^T \kappa(x) = \sum_{i=1}^n \alpha_i k(x_i, x) = f_\alpha(x)$$

Gaussian kernel : $k : (x, z) \mapsto e^{\frac{-\|x-z\|^2}{\sigma^2}}$, $\sigma = 1$,
Parameterized function : $f_\alpha : x \mapsto \sum_{i=1}^{n} \alpha_i k(x_i, x)$.

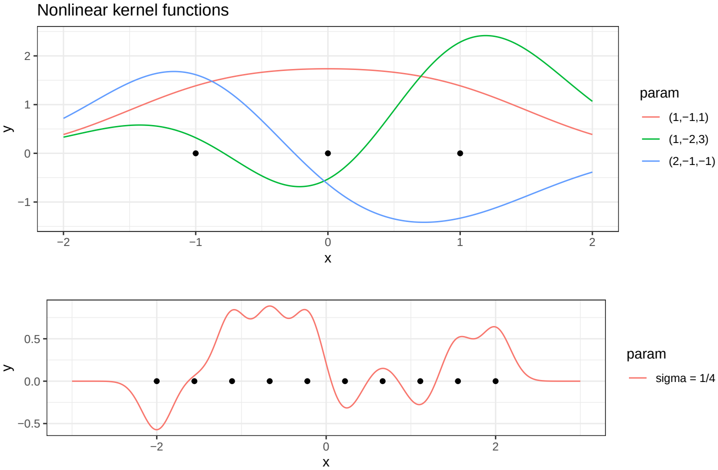
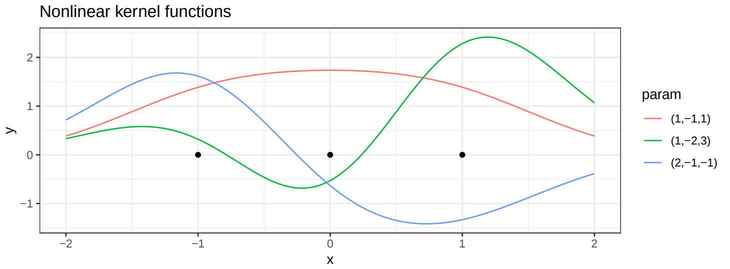
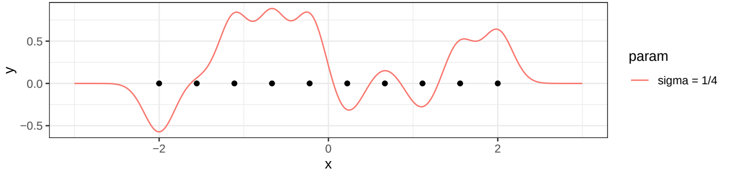


Illustration 2

Gaussian kernel : $k : (x, z) \mapsto e^{\frac{-\|x-z\|^2}{\sigma^2}}$, $\sigma = 1$,
Parameterized function : $f_\alpha : x \mapsto \sum_{i=1}^{n} \alpha_i k(x_i, x)$.





What determines the complexity of the model ? Does it remind anything ?

Empirical risk minimization

$\mathcal{X} \subset \mathbb{R}^p$ input space, $\mathcal{Y} \subset \mathbb{R}$ output space. $\ell \colon \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}^+$ a loss function.

Empirical risk minimization over RKHS : $S = (x_i, y_i)_{i=1}^n$, iid copies of X and Y.

$$\min_{f \in \mathcal{F}} R_n(f) := \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

where $\mathcal{F}$ is a class of functions from $\mathcal{X}$ to $\mathbb{R}$. f_n is the argmin.

Empirical risk minimization

$\mathcal{X} \subset \mathbb{R}^p$ input space, $\mathcal{Y} \subset \mathbb{R}$ output space. $\ell \colon \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}^+$ a loss function.

Empirical risk minimization over RKHS : $S = (x_i, y_i)_{i=1}^n$, iid copies of X and Y.

$$\min_{f \in \mathcal{F}} R_n(f) := \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

where $\mathcal{F}$ is a class of functions from $\mathcal{X}$ to $\mathbb{R}$. f_n is the argmin.

Examples :

Empirical risk minimization

$\mathcal{X} \subset \mathbb{R}^p$ input space, $\mathcal{Y} \subset \mathbb{R}$ output space. $\ell \colon \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}^+$ a loss function.

Empirical risk minimization over RKHS : $S = (x_i, y_i)_{i=1}^n$, iid copies of X and Y.

$$\min_{f \in \mathcal{F}} R_n(f) := \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i)$$

where $\mathcal{F}$ is a class of functions from $\mathcal{X}$ to $\mathbb{R}$. f_n is the argmin.

Examples :

- Linear regression. $y_i \in \mathbb{R}$, $\mathcal{F}$ are linear functions $f_w \colon x \mapsto \langle w, x \rangle$, ℓ is the square loss.

$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n (\langle w, x_i \rangle - y_i)^2$$

- Logistic regression. $y_i \in \{-1, 1\}$, $\mathcal{F}$ are linear functions, ℓ bernouilli log likelihood combined with logit function : $\ell(s, y) = \log(1 + \exp(sy))$.

$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n \log(1 + \exp(y_i \langle w, x_i \rangle))$$

- SVM, same with hinge loss.

Empirical risk minimization : "kernel trick"

$\mathcal{X} \subset \mathbb{R}^p$ input space, $\mathcal{Y} \subset \mathbb{R}$ output space.
$\ell \colon \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}^+$ a loss function.

Empirical risk minimization over RKHS : $S = (x_i, y_i)_{i=1}^n$, iid copies of X and Y.

$$\min_{f \in \mathcal{F}} R_n(f) := \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i)$$

where $\mathcal{F}$ is a class of functions from $\mathcal{X}$ to $\mathbb{R}$. f_n is the argmin.

Idea : Take any linear method, and replace linear functions, of the form

$$f_w \colon x \mapsto \langle w, x \rangle = \sum_{i=1}^p w[i]x[i]$$

by a nonlinear one

$$f_\alpha \colon x \mapsto \sum_{i=1}^n \alpha_i k(x_i, x) = \langle \kappa_n(x), \alpha \rangle \, .$$

Kernel linear regression

$\mathcal{X} \subset \mathbb{R}^p$ input space, $\mathcal{Y} \subset \mathbb{R}$ output space. Square loss.

$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 \qquad \to \qquad \min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} (f_\alpha(x_i) - y_i)^2$$

where $f_\alpha : x \mapsto \sum_{i=1}^n \alpha_i k(x_i, x)$.

Kernel linear regression

$\mathcal{X} \subset \mathbb{R}^p$ input space, $\mathcal{Y} \subset \mathbb{R}$ output space. Square loss.

$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 \qquad \rightarrow \qquad \min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} (f_\alpha(x_i) - y_i)^2$$

where $f_\alpha : x \mapsto \sum_{i=1}^{n} \alpha_i k(x_i, x)$.

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \alpha_j k(x_i, x_j) - y_i \right)^2 = \frac{1}{n} \|K_n \alpha - y\|^2.$$

Solution : If K_n is invertible,

Kernel linear regression

$\mathcal{X} \subset \mathbb{R}^p$ input space, $\mathcal{Y} \subset \mathbb{R}$ output space. Square loss.

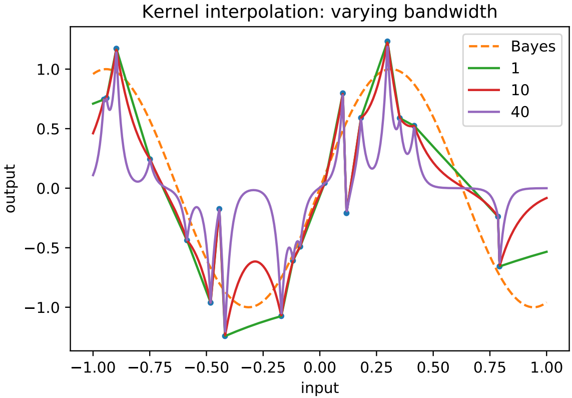
$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 \qquad \rightarrow \qquad \min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} (f_\alpha(x_i) - y_i)^2$$

where $f_\alpha : x \mapsto \sum_{i=1}^{n} \alpha_i k(x_i, x)$.

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \alpha_j k(x_i, x_j) - y_i \right)^2 = \frac{1}{n} \|K_n \alpha - y\|^2.$$

Solution : If K_n is invertible, then $\alpha = K_n^{-1} y$ and the empirical risk is null.

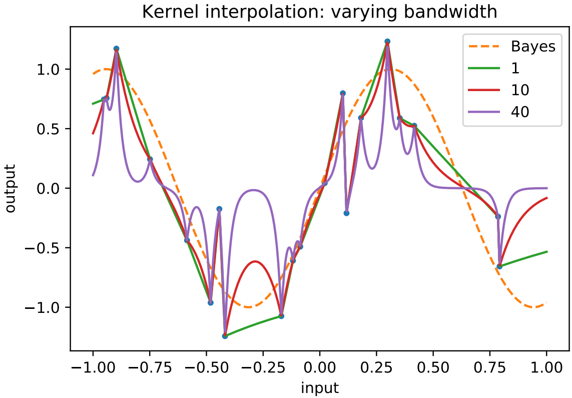
Laplacian kernel : $k\colon (x, z) \mapsto e^{-\gamma \|x - z\|}$.



What is going to happen ? How to avoid it ?

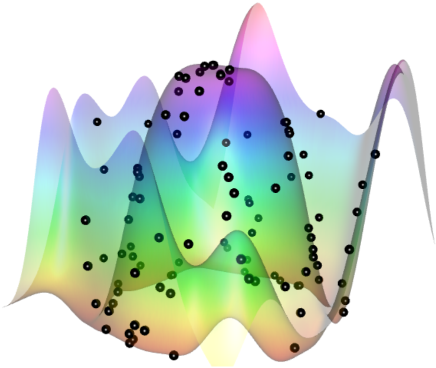
Vanilla linear regression $\simeq$ interpolation

Laplacian kernel : $k\colon (x, z) \mapsto e^{-\gamma \|x-z\|}$.



What is going to happen ? How to avoid it ?
What determines the complexity of the model ? Does it remind anything ?

Gaussian kernel : $k\colon (x,z) \mapsto e^{-\|x-z\|^2/\sigma^2}$.



Which other method can interpolate ? What is the advantage of this one ?

Ridge regression

$X \in \mathbb{R}^{n\times p}$, design matrix, $y \in \mathbb{R}^n$ observations.

$$\min_{w\in\mathbb{R}^p} \sum_{i=1}^{n} (\langle w, x_i\rangle - y_i)^2 + \lambda\|w\|^2 \qquad \rightarrow \qquad \min_{\alpha\in\mathbb{R}^n} \sum_{i=1}^{n} (f_\alpha(x_i) - y_i)^2 + ?$$

Replace $\langle w, x_i\rangle$ by $\sum_{i=1}^n \alpha_j k(x_j, x_i)$, but replace $\|w\|^2$ by what ?

Ridge regression

$X \in \mathbb{R}^{n \times p}$, design matrix, $y \in \mathbb{R}^n$ observations.

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 + \lambda \|w\|^2 \qquad \to \qquad \min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} (f_\alpha(x_i) - y_i)^2 + ?$$

Replace $\langle w, x_i \rangle$ by $\sum_{i=1}^{n} \alpha_j k(x_j, x_i)$, but replace $\|w\|^2$ by what ?

Exercise : For any $w \in \mathbb{R}^p$, there is $\alpha \in \mathbb{R}^n$ and $z \in \mathbb{R}^p$, with $w = X^T \alpha + z$ and $Xz = 0$.

Ridge regression

$X \in \mathbb{R}^{n \times p}$, design matrix, $y \in \mathbb{R}^n$ observations.

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 + \lambda \|w\|^2 \qquad \rightarrow \qquad \min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} (f_\alpha(x_i) - y_i)^2 + ?$$

Replace $\langle w, x_i \rangle$ by $\sum_{i=1}^{n} \alpha_j k(x_j, x_i)$, but replace $\|w\|^2$ by what ?

Exercise : For any $w \in \mathbb{R}^p$, there is $\alpha \in \mathbb{R}^n$ and $z \in \mathbb{R}^p$, with $w = X^T \alpha + z$ and $Xz = 0$.

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

Ridge regression

$X \in \mathbb{R}^{n \times p}$, design matrix, $y \in \mathbb{R}^n$ observations.

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 + \lambda \|w\|^2 \qquad \to \qquad \min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} (f_\alpha(x_i) - y_i)^2 + ?$$

Replace $\langle w, x_i \rangle$ by $\sum_{i=1}^{n} \alpha_j k(x_j, x_i)$, but replace $\|w\|^2$ by what ?

Exercise : For any $w \in \mathbb{R}^p$, there is $\alpha \in \mathbb{R}^n$ and $z \in \mathbb{R}^p$, with $w = X^T \alpha + z$ and $Xz = 0$.

$$\begin{aligned} & \min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 + \lambda \|w\|^2 \\ = & \min_{w \in \mathbb{R}^p} \|Xw - y\|^2 + \lambda \|w\|^2 \end{aligned}$$

Ridge regression

$X \in \mathbb{R}^{n \times p}$, design matrix, $y \in \mathbb{R}^n$ observations.

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 + \lambda \|w\|^2 \qquad \rightarrow \qquad \min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} (f_\alpha(x_i) - y_i)^2 + ?$$

Replace $\langle w, x_i \rangle$ by $\sum_{i=1}^{n} \alpha_j k(x_j, x_i)$, but replace $\|w\|^2$ by what ?

Exercise : For any $w \in \mathbb{R}^p$, there is $\alpha \in \mathbb{R}^n$ and $z \in \mathbb{R}^p$, with $w = X^T \alpha + z$ and $Xz = 0$.

$$\begin{aligned}
& \min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 + \lambda \|w\|^2 \\
= & \min_{w \in \mathbb{R}^p} \|Xw - y\|^2 + \lambda \|w\|^2 \\
= & \min_{\alpha \in \mathbb{R}^n, Xz = 0} \|X(X^T \alpha + z) - y\|^2 + \lambda \|X^T \alpha + z\|^2
\end{aligned}$$

Ridge regression

$X \in \mathbb{R}^{n \times p}$, design matrix, $y \in \mathbb{R}^n$ observations.

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 + \lambda \|w\|^2 \qquad \to \qquad \min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} (f_\alpha(x_i) - y_i)^2 + ?$$

Replace $\langle w, x_i \rangle$ by $\sum_{i=1}^{n} \alpha_j k(x_j, x_i)$, but replace $\|w\|^2$ by what ?

Exercise : For any $w \in \mathbb{R}^p$, there is $\alpha \in \mathbb{R}^n$ and $z \in \mathbb{R}^p$, with $w = X^T \alpha + z$ and $Xz = 0$.

$$\begin{aligned}
& \min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 + \lambda \|w\|^2 \\
= & \min_{w \in \mathbb{R}^p} \|Xw - y\|^2 + \lambda \|w\|^2 \\
= & \min_{\alpha \in \mathbb{R}^n, Xz = 0} \|X(X^T \alpha + z) - y\|^2 + \lambda \|X^T \alpha + z\|^2 \\
= & \min_{\alpha \in \mathbb{R}^n, Xz = 0} \|X(X^T \alpha + z) - y\|^2 + \lambda \|X^T \alpha\|^2 + 2\alpha^T X z + \|z\|^2
\end{aligned}$$

Ridge regression

$X \in \mathbb{R}^{n \times p}$, design matrix, $y \in \mathbb{R}^n$ observations.

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 + \lambda \|w\|^2 \qquad \rightarrow \qquad \min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} (f_\alpha(x_i) - y_i)^2 + ?$$

Replace $\langle w, x_i \rangle$ by $\sum_{j=1}^{n} \alpha_j k(x_j, x_i)$, but replace $\|w\|^2$ by what ?

Exercise : For any $w \in \mathbb{R}^p$, there is $\alpha \in \mathbb{R}^n$ and $z \in \mathbb{R}^p$, with $w = X^T \alpha + z$ and $Xz = 0$.

$$\begin{aligned}
& \min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 + \lambda \|w\|^2 \\
= & \min_{w \in \mathbb{R}^p} \|Xw - y\|^2 + \lambda \|w\|^2 \\
= & \min_{\alpha \in \mathbb{R}^n, Xz = 0} \|X(X^T \alpha + z) - y\|^2 + \lambda \|X^T \alpha + z\|^2 \\
= & \min_{\alpha \in \mathbb{R}^n, Xz = 0} \|X(X^T \alpha + z) - y\|^2 + \lambda \|X^T \alpha\|^2 + 2\alpha^T X z + \|z\|^2 \\
= & \min_{\alpha \in \mathbb{R}^n, Xz = 0} \|XX^T \alpha - y\|^2 + \lambda \|X^T \alpha\|^2 + \|z\|^2
\end{aligned}$$

Ridge regression

$X \in \mathbb{R}^{n \times p}$, design matrix, $y \in \mathbb{R}^n$ observations.

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 + \lambda \|w\|^2 \qquad \rightarrow \qquad \min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} (f_\alpha(x_i) - y_i)^2 + ?$$

Replace $\langle w, x_i \rangle$ by $\sum_{j=1}^{n} \alpha_j k(x_j, x_i)$, but replace $\|w\|^2$ by what ?

Exercise : For any $w \in \mathbb{R}^p$, there is $\alpha \in \mathbb{R}^n$ and $z \in \mathbb{R}^p$, with $w = X^T\alpha + z$ and $Xz = 0$.

$$\begin{aligned}
& \min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2 + \lambda \|w\|^2 \\
= & \min_{w \in \mathbb{R}^p} \|Xw - y\|^2 + \lambda \|w\|^2 \\
= & \min_{\alpha \in \mathbb{R}^n, Xz = 0} \|X(X^T\alpha + z) - y\|^2 + \lambda \|X^T\alpha + z\|^2 \\
= & \min_{\alpha \in \mathbb{R}^n, Xz = 0} \|X(X^T\alpha + z) - y\|^2 + \lambda \|X^T\alpha\|^2 + 2\alpha^T Xz + \|z\|^2 \\
= & \min_{\alpha \in \mathbb{R}^n, Xz = 0} \|XX^T\alpha - y\|^2 + \lambda \|X^T\alpha\|^2 + \|z\|^2 \\
= & \min_{\alpha \in \mathbb{R}^n} \|XX^T\alpha - y\|^2 + \lambda \|X^T\alpha\|^2 = \min_{\alpha \in \mathbb{R}^n} \|XX^T\alpha - y\|^2 + \lambda \alpha^T XX^T \alpha
\end{aligned}$$

Kernel ridge regression

$X \in \mathbb{R}^{n\times p}$, design matrix, $y \in \mathbb{R}^n$ observations.

Replace w by $X^T\alpha$

$$\min_{w\in\mathbb{R}^p} \|Xw - y\|^2 + \lambda\|w\|^2$$

$$\min_{\alpha\in\mathbb{R}^n} \|XX^T\alpha - y\|^2 + \lambda\alpha^T XX^T\alpha \qquad \to \qquad \min_{\alpha\in\mathbb{R}^n} \|K_n\alpha - y\|^2 + \lambda\alpha^T K_n\alpha$$

Kernel ridge regression

$X \in \mathbb{R}^{n \times p}$, design matrix, $y \in \mathbb{R}^n$ observations.

Replace w by $X^T\alpha$

$$\min_{w \in \mathbb{R}^p} \|Xw - y\|^2 + \lambda \|w\|^2$$

$$\min_{\alpha \in \mathbb{R}^n} \|XX^T\alpha - y\|^2 + \lambda \alpha^T XX^T \alpha \quad \rightarrow \quad \min_{\alpha \in \mathbb{R}^n} \|K_n\alpha - y\|^2 + \lambda \alpha^T K_n \alpha$$

Solution : $K_n(K_n\alpha - y) + \lambda K_n \alpha = K_n(K_n + \lambda I)\alpha - K_n y = 0.$

Kernel ridge regression

$X \in \mathbb{R}^{n \times p}$, design matrix, $y \in \mathbb{R}^n$ observations.

Replace w by $X^T\alpha$

$$\min_{w \in \mathbb{R}^p} \|Xw - y\|^2 + \lambda \|w\|^2$$

$$\min_{\alpha \in \mathbb{R}^n} \|XX^T\alpha - y\|^2 + \lambda \alpha^T XX^T \alpha \qquad \rightarrow \qquad \min_{\alpha \in \mathbb{R}^n} \|K_n\alpha - y\|^2 + \lambda \alpha^T K_n \alpha$$

Solution : $K_n(K_n\alpha - y) + \lambda K_n \alpha = K_n(K_n + \lambda I)\alpha - K_n y = 0.$

$$\alpha = (K_n + \lambda I)^{-1} y$$

Interpretation ?

Kernel ridge regression

$X \in \mathbb{R}^{n \times p}$, design matrix, $y \in \mathbb{R}^n$ observations.

Replace w by $X^T\alpha$

$$\min_{w \in \mathbb{R}^p} \|Xw - y\|^2 + \lambda \|w\|^2$$

$$\min_{\alpha \in \mathbb{R}^n} \|XX^T\alpha - y\|^2 + \lambda \alpha^T XX^T \alpha \qquad \rightarrow \qquad \min_{\alpha \in \mathbb{R}^n} \|K_n\alpha - y\|^2 + \lambda \alpha^T K_n \alpha$$

Solution : $K_n(K_n\alpha - y) + \lambda K_n \alpha = K_n(K_n + \lambda I)\alpha - K_n y = 0.$

$$\alpha = (K_n + \lambda I)^{-1} y$$

Interpretation ?

Prediction : $w = X^T\alpha$

$$x \mapsto \langle x, w \rangle = \left\langle x, X^T\alpha \right\rangle = \langle Xx, \alpha \rangle = \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle$$

$$x \mapsto \sum_{i=1}^{n} \alpha_i k(x, x_i) = \alpha^T \kappa_n(x) = y^T (K_n + \lambda I)^{-1} \kappa_n(x).$$

Remark on regularization using $\alpha^T K_n \alpha$

$\mathcal{X} \subset \mathbb{R}^p$ input space, $\mathcal{Y} \subset \mathbb{R}$ output space. $\ell \colon \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}^+$ a loss function.

Empirical risk minimization over RKHS : $S = (x_i, y_i)_{i=1}^n$, iid copies of X and Y.

$$\min_{f \in \mathcal{F}} R_n(f) := \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

where $\mathcal{F}$ is a class of functions from $\mathcal{X}$ to $\mathbb{R}$. f_n is the argmin.

Remark on regularization using $\alpha^T K_n \alpha$

$\mathcal{X} \subset \mathbb{R}^p$ input space, $\mathcal{Y} \subset \mathbb{R}$ output space. $\ell \colon \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}^+$ a loss function.

Empirical risk minimization over RKHS : $S = (x_i, y_i)_{i=1}^n$, iid copies of X and Y.

$$\min_{f \in \mathcal{F}} R_n(f) := \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i)$$

where $\mathcal{F}$ is a class of functions from $\mathcal{X}$ to $\mathbb{R}$. f_n is the argmin.

Statistical learning assumption : S is an i.i.d sample from $P_{X,Y}$. Expected risk :

$$R(f) := \mathbb{E}_{XY}[\ell(f(X), Y)]$$

Remark on regularization using $\alpha^T K_n \alpha$

$\mathcal{X} \subset \mathbb{R}^p$ input space, $\mathcal{Y} \subset \mathbb{R}$ output space. $\ell \colon \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}^+$ a loss function.

Empirical risk minimization over RKHS : $S = (x_i, y_i)_{i=1}^n$, iid copies of X and Y.

$$\min_{f \in \mathcal{F}} R_n(f) := \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i)$$

where $\mathcal{F}$ is a class of functions from $\mathcal{X}$ to $\mathbb{R}$. f_n is the argmin.

Statistical learning assumption : S is an i.i.d sample from $P_{X,Y}$. Expected risk :

$$R(f) := \mathbb{E}_{XY}[\ell(f(X), Y)]$$

Generalization bound : Under assumptions, with high probability, if $\mathcal{F} = \{f_\alpha,\ \alpha^T K_n \alpha \leq R\}$

$$\min_{f \in \mathcal{F}} R(f) \leq \min_{f \in \mathcal{F}} R_n(f) + cst \times \frac{R}{\sqrt{n}}$$

Remark on regularization using $\alpha^T K_n \alpha$

$\mathcal{X} \subset \mathbb{R}^p$ input space, $\mathcal{Y} \subset \mathbb{R}$ output space. $\ell \colon \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}^+$ a loss function.

Empirical risk minimization over RKHS : $S = (x_i, y_i)_{i=1}^n$, iid copies of X and Y.

$$\min_{f \in \mathcal{F}} R_n(f) := \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

where $\mathcal{F}$ is a class of functions from $\mathcal{X}$ to $\mathbb{R}$. f_n is the argmin.

Statistical learning assumption : S is an i.i.d sample from $P_{X,Y}$. Expected risk :

$$R(f) := \mathbb{E}_{XY}[\ell(f(X), Y)]$$

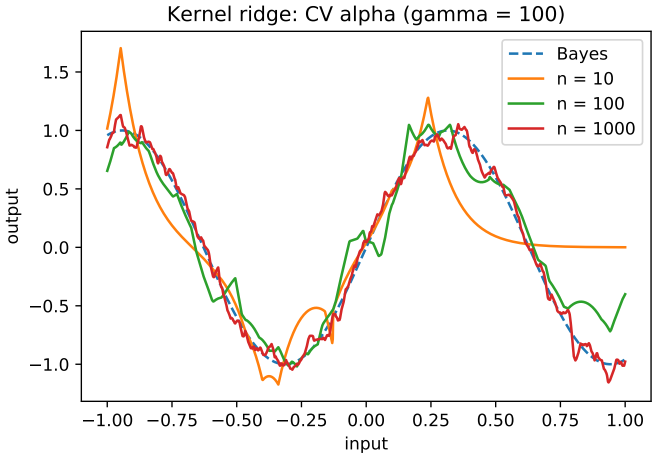
Generalization bound : Under assumptions, with high probability, if $\mathcal{F} = \{f_\alpha, \, \alpha^T K_n \alpha \leq R\}$

$$\min_{f \in \mathcal{F}} R(f) \leq \min_{f \in \mathcal{F}} R_n(f) + cst \times \frac{R}{\sqrt{n}}$$

Take-away : penalyzing $\lambda \alpha^T K_n \alpha$ allows to effectively control estimation error.

What do you expect as the sample size grows ?

What do you expect as the sample size grows ?
Which other methods have this property ?



Outline

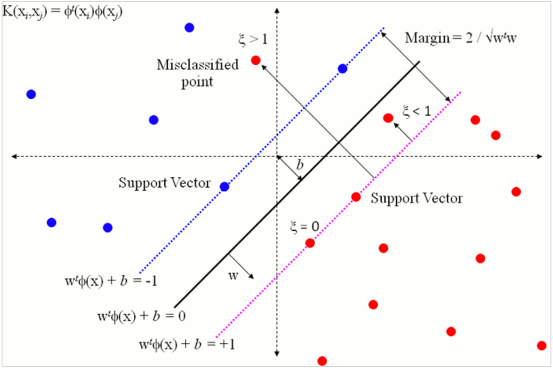
Linear SVM : $\mathcal{D}_n = \{x_1, \dots, x_n\} \subset \mathbb{R}^p$, $(y_i)_{i=1}^n$ in $-1, 1$
Find $w \in \mathbb{R}^p$, $b \in \mathbb{R}^p$ such that $\mathrm{sign}(w^T x_i + b) \simeq y_i$, $i = 1 \dots n$.

Linear SVM : $\mathcal{D}_n = \left\{x_1, \ldots, x_n\right\} \subset \mathbb{R}^p$, $(y_i)_{i=1}^n$ in $-1, 1$
Find $w \in \mathbb{R}^p$, $b \in \mathbb{R}^p$ such that $\mathrm{sign}(w^T x_i + b) \simeq y_i$, $i = 1 \ldots n$. Fix $C > 0$

$$\min_{w \in \mathbb{R}^p, b \in \mathbb{R}} \quad \|w\|^2 + C \sum_{i=1}^n \max\left(1 - y_i(w^T x_i + b), 0\right)$$

Linear SVM : $\mathcal{D}_n = \left\{x_1, \ldots, x_n\right\} \subset \mathbb{R}^p$, $(y_i)_{i=1}^n$ in $-1, 1$
Find $w \in \mathbb{R}^p$, $b \in \mathbb{R}^p$ such that $\mathrm{sign}(w^T x_i + b) \simeq y_i$, $i = 1 \ldots n$. Fix $C > 0$

$$\min_{w \in \mathbb{R}^p, b \in \mathbb{R}} \quad \|w\|^2 + C \sum_{i=1}^{n} \max\left(1 - y_i(w^T x_i + b), 0\right)$$



Linear SVM : $\mathcal{D}_n = \{x_1, \dots, x_n\} \subset \mathbb{R}^p$, $(y_i)_{i=1}^n$ in $-1, 1$
Find $w \in \mathbb{R}^p$, $b \in \mathbb{R}^p$ such that $\operatorname{sign}(w^T x_i + b) \simeq y_i$, $i = 1 \dots n$. Fix $C > 0$

$$\min_{w \in \mathbb{R}^p, b \in \mathbb{R}} \quad \|w\|^2 + C \sum_{i=1}^n \max\left(1 - y_i(w^T x_i + b), 0\right)$$

(Exercise : we can consider $w = X^T \alpha$, $\alpha \in \mathbb{R}^n$).

Linear SVM : $\mathcal{D}_n = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$, $(y_i)_{i=1}^n$ in $-1, 1$
Find $w \in \mathbb{R}^p$, $b \in \mathbb{R}^p$ such that $\mathrm{sign}(w^T x_i + b) \simeq y_i$, $i = 1 \ldots n$. Fix $C > 0$

$$\min_{w \in \mathbb{R}^p, b \in \mathbb{R}} \quad \|w\|^2 + C \sum_{i=1}^{n} \max\left(1 - y_i(w^T x_i + b), 0\right)$$

(Exercise : we can consider $w = X^T \alpha$, $\alpha \in \mathbb{R}^n$).

We may take $w = X^T \alpha$ for some $\alpha \in \mathbb{R}^n$,

Linear SVM : $\mathcal{D}_n = \left\{x_1, \ldots, x_n\right\} \subset \mathbb{R}^p$, $(y_i)_{i=1}^n$ in $-1, 1$
Find $w \in \mathbb{R}^p$, $b \in \mathbb{R}^p$ such that $\mathrm{sign}(w^T x_i + b) \simeq y_i$, $i = 1 \ldots n$. Fix $C > 0$

$$\min_{w \in \mathbb{R}^p, b \in \mathbb{R}} \quad \|w\|^2 + C \sum_{i=1}^{n} \max\left(1 - y_i(w^T x_i + b), 0\right)$$

(Exercise : we can consider $w = X^T \alpha$, $\alpha \in \mathbb{R}^n$).

We may take $w = X^T \alpha$ for some $\alpha \in \mathbb{R}^n$,
Then $\|w\|^2 = w^T w = \alpha^T X X^T \alpha$ and $w^T x_i = \alpha^T X x_i = \sum_{j=1}^n \alpha_j \langle x_j, x_i \rangle$.

Linear SVM : $\mathcal{D}_n = \{x_1, \dots, x_n\} \subset \mathbb{R}^p$, $(y_i)_{i=1}^n$ in $-1, 1$
Find $w \in \mathbb{R}^p$, $b \in \mathbb{R}^p$ such that $\text{sign}(w^T x_i + b) \simeq y_i$, $i = 1 \dots n$. Fix $C > 0$

$$\min_{w \in \mathbb{R}^p, b \in \mathbb{R}} \quad \|w\|^2 + C \sum_{i=1}^n \max\left(1 - y_i(w^T x_i + b), 0\right)$$

(Exercise : we can consider $w = X^T \alpha$, $\alpha \in \mathbb{R}^n$).

We may take $w = X^T \alpha$ for some $\alpha \in \mathbb{R}^n$,
Then $\|w\|^2 = w^T w = \alpha^T X X^T \alpha$ and $w^T x_i = \alpha^T X x_i = \sum_{j=1}^n \alpha_j \langle x_j, x_i \rangle$.

$$\min_{\alpha \in \mathbb{R}^n, b \in \mathbb{R}} \quad \alpha^T X X^T \alpha + C \sum_{i=1}^n \max\left(1 - y_i \left(\sum_{k=1}^n \alpha_k \langle x_k, x_i \rangle + b\right), 0\right)$$

Linear SVM : $\mathcal{D}_n = \left\{x_1, \ldots, x_n\right\} \subset \mathbb{R}^p$, $(y_i)_{i=1}^n$ in $-1, 1$
Find $w \in \mathbb{R}^p$, $b \in \mathbb{R}^p$ such that $\mathrm{sign}(w^T x_i + b) \simeq y_i$, $i = 1 \ldots n$. Fix $C > 0$

$$\min_{w \in \mathbb{R}^p, b \in \mathbb{R}} \quad \|w\|^2 + C \sum_{i=1}^n \max\left(1 - y_i(w^T x_i + b), 0\right)$$

(Exercise : we can consider $w = X^T \alpha$, $\alpha \in \mathbb{R}^n$).

We may take $w = X^T \alpha$ for some $\alpha \in \mathbb{R}^n$,
Then $\|w\|^2 = w^T w = \alpha^T X X^T \alpha$ and $w^T x_i = \alpha^T X x_i = \sum_{j=1}^n \alpha_j \langle x_j, x_i \rangle$.

$$\min_{\alpha \in \mathbb{R}^n, b \in \mathbb{R}} \quad \alpha^T X X^T \alpha + C \sum_{i=1}^n \max\left(1 - y_i \left(\sum_{k=1}^n \alpha_k \langle x_k, x_i \rangle + b\right), 0\right)$$

Warning : Prediction at x

$$\mathrm{sign}(w^T x + b) = \mathrm{sign}(\alpha^T X x + b) = \mathrm{sign}\left(\sum_{j=1}^n \alpha_j \langle x_j, x \rangle + b\right)$$

$C > 0$

$$\min_{\alpha \in \mathbb{R}^n, b \in \mathbb{R}} \quad \alpha^T X X^T \alpha + C \sum_{i=1}^{n} \max \left(1 - y_i \left(\sum_{j=1}^{n} \alpha_j \langle x_j, x_i \rangle + b \right), 0 \right)$$

$C > 0$

$$\min_{\alpha \in \mathbb{R}^n, b \in \mathbb{R}} \quad \alpha^T X X^T \alpha + C \sum_{i=1}^{n} \max \left(1 - y_i \left(\sum_{j=1}^{n} \alpha_j \langle x_j, x_i \rangle + b \right), 0 \right)$$

Replace XX^T by K_n and $\langle \cdot, \cdot \rangle$ by $k(\cdot, \cdot)$.

$$\min_{\alpha \in \mathbb{R}^n, b \in \mathbb{R}} \quad \alpha^T K_n \alpha + C \sum_{i=1}^{n} \max \left(1 - y_i \left(\sum_{j=1}^{n} \alpha_j k(x_j, x_i) + b \right), 0 \right)$$

$C > 0$

$$\min_{\alpha \in \mathbb{R}^n, b \in \mathbb{R}} \quad \alpha^T X X^T \alpha + C \sum_{i=1}^{n} \max\left(1 - y_i \left(\sum_{j=1}^{n} \alpha_j \langle x_j, x_i \rangle + b\right), 0\right)$$

Replace XX^T by K_n and $\langle \cdot, \cdot \rangle$ by $k(\cdot, \cdot)$.

$$\min_{\alpha \in \mathbb{R}^n, b \in \mathbb{R}} \quad \alpha^T K_n \alpha + C \sum_{i=1}^{n} \max\left(1 - y_i \left(\sum_{j=1}^{n} \alpha_j k(x_j, x_i) + b\right), 0\right)$$

Prediction : at x

$$\mathrm{sign}\left(\sum_{j=1}^{n} \alpha_j k(x_j, x) + b\right) = \mathrm{sign}(\alpha^T \kappa_n(x) + b)$$

with $\kappa_n(x) = (k(x_i, x))_{i=1}^{n}$.

$C > 0$

$$\min_{\alpha\in\mathbb{R}^n, b\in\mathbb{R}} \quad \alpha^T XX^T\alpha + C\sum_{i=1}^{n} \max\left(1 - y_i\left(\sum_{j=1}^{n}\alpha_j\left\langle x_j, x_i\right\rangle + b\right), 0\right)$$

Replace XX^T by K_n and $\langle\cdot,\cdot\rangle$ by $k(\cdot,\cdot)$.

$$\min_{\alpha\in\mathbb{R}^n, b\in\mathbb{R}} \quad \alpha^T K_n\alpha + C\sum_{i=1}^{n} \max\left(1 - y_i\left(\sum_{j=1}^{n}\alpha_j k(x_j, x_i) + b\right), 0\right)$$

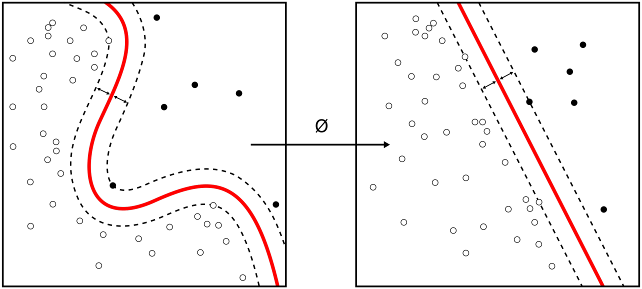
Prediction : at x

$$\mathrm{sign}\left(\sum_{j=1}^{n}\alpha_j k(x_j, x) + b\right) = \mathrm{sign}(\alpha^T\kappa_n(x) + b)$$

with $\kappa_n(x) = (k(x_i, x))_{i=1}^{n}$.

Warning : C is a tuning parameter controling regularization / data-fitting tradeoff in order to avoid overfitting.

Intuition : nonlinear decision in $\mathcal{X}$ from linear separation in higher space implicitely through the kernel trick.



Kernel Logistic regression

$$\min_{\alpha \in \mathbb{R}^n, b \in \mathbb{R}} \quad \alpha^T K_n \alpha + C \sum_{i=1}^{n} \log \left(1 + \exp \left(y_i \sum_{j=1}^{n} \alpha_j k(x_j, x_i) + b \right) \right)$$

What is the advantage ?

Outline

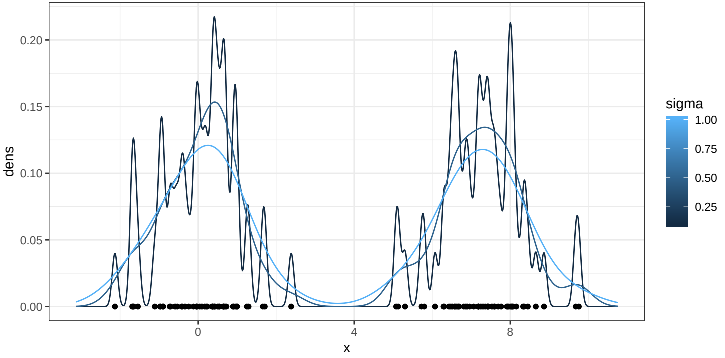
Density based

Gaussian kernel with bandwidth σ

$$k(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{\|y-x\|^2}{\sigma^2}}$$

Kernel density estimator :

$$p_\sigma : x \mapsto \frac{1}{n} \sum_{i=1}^{n} k(x, x_i)$$



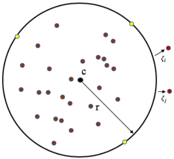
A variant of one class SVM

Main idea, find a ball of minimal radius which encloses all the points :

$$\min_{r \in \mathbb{R}, c \in \mathbb{R}^p} \quad r^2$$
$$\text{s.t.} \quad \|x_i - c\|^2 \leq r^2,\ i = 1 \dots, n.$$

Too restrictive, add slack, $\nu > 0$

$$\min_{r \in \mathbb{R}, c \in \mathbb{R}^p} \quad r^2 + \frac{1}{n\nu} \sum_{i=1}^{n} \xi_i$$
$$\text{s.t.} \quad \|x_i - c\|^2 \leq r^2 + \xi_i,\ i = 1 \dots, n.$$



Kernel trick : ϕ: $x \mapsto X \in \mathbb{R}^P$ sends x to a high (infinite) dimensional feature space.
Implicitely : $x_i \to \phi(x_i)$, $i = 1, \dots, n$.
Positive definite kernel (*ex :* Gaussian) implicitely encodes ϕ.

Kernel trick :

$$\min_{r\in\mathbb{R}, c\in\mathbb{R}^p} \quad r^2 + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i$$
$$\text{s.t.} \quad \|x_i - c\|^2 \leq r^2 + \xi_i,\ i = 1\ldots, n.$$

Kernel trick :

$$\min_{r \in \mathbb{R}, c \in \mathbb{R}^p} \quad r^2 + \frac{1}{n\nu} \sum_{i=1}^{n} \xi_i$$
$$\text{s.t.} \quad \|x_i - c\|^2 \leq r^2 + \xi_i,\ i = 1 \ldots, n.$$

$$\|x_i - c\|^2 = x_i^T x_i - 2c^T x_i + c^T c = x_i^T x_i - 2\alpha^T X x + \alpha^T X X^T \alpha.$$

A variant of one class SVM

Kernel trick :

$$\min_{r\in\mathbb{R},c\in\mathbb{R}^p} \quad r^2 + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i$$
$$\text{s.t.} \quad \|x_i - c\|^2 \leq r^2 + \xi_i,\ i = 1\ldots,n.$$

$\|x_i - c\|^2 = x_i^T x_i - 2c^T x_i + c^T c = x_i^T x_i - 2\alpha^T X x + \alpha^T X X^T \alpha.$
We may take $c = X^T\alpha$ with $\alpha \in \mathbb{R}^n$ and use the kernel trick :

Kernel trick :

$$\min_{r\in\mathbb{R},c\in\mathbb{R}^p} \quad r^2 + \frac{1}{n\nu}\sum_{i=1}^n \xi_i$$
$$\text{s.t.} \quad \|x_i - c\|^2 \le r^2 + \xi_i,\ i = 1\ldots,n.$$

$\|x_i - c\|^2 = x_i^T x_i - 2c^T x_i + c^T c = x_i^T x_i - 2\alpha^T X x + \alpha^T X X^T \alpha.$
We may take $c = X^T\alpha$ with $\alpha \in \mathbb{R}^n$ and use the kernel trick :

$$\min_{r\in\mathbb{R},c\in\mathbb{R}^p} \quad r^2 + \frac{1}{n\nu}\sum_{i=1}^n \xi_i$$
$$\text{s.t.} \quad k(x_i, x_i) - 2\sum_{j=1}^n \alpha_j k(x_j, x_i) + \alpha^T K_n \alpha \le r^2 + \xi_i,\ i = 1\ldots,n.$$

A variant of one class SVM

Kernel trick :

$$\min_{r\in\mathbb{R},c\in\mathbb{R}^p} \quad r^2 + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i$$
$$\text{s.t.} \quad \|x_i - c\|^2 \leq r^2 + \xi_i,\ i = 1\ldots,n.$$

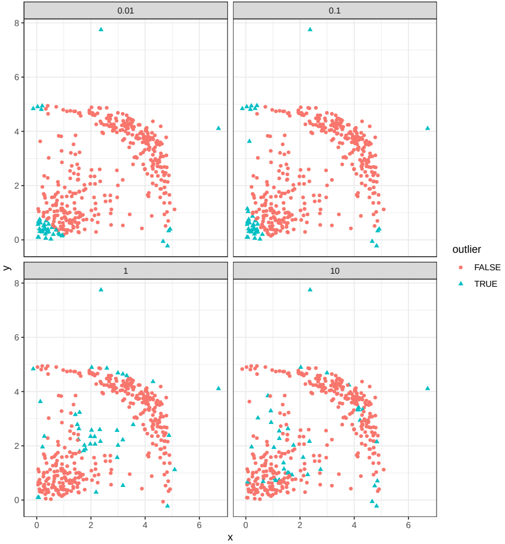
$\|x_i - c\|^2 = x_i^T x_i - 2c^T x_i + c^T c = x_i^T x_i - 2\alpha^T Xx + \alpha^T XX^T\alpha.$
We may take $c = X^T\alpha$ with $\alpha \in \mathbb{R}^n$ and use the kernel trick :

$$\min_{r\in\mathbb{R},c\in\mathbb{R}^p} \quad r^2 + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i$$
$$\text{s.t.} \quad k(x_i, x_i) - 2\sum_{j=1}^{n}\alpha_j k(x_j, x_i) + \alpha^T K_n\alpha \leq r^2 + \xi_i,\ i = 1\ldots,n.$$

Score : $s(x) = r^2 - k(x, x) + 2\sum_{j=1}^{n}\alpha_j k(x_j, x) + \alpha^T K_n\alpha.$

A variant of one class SVM

Gaussian kernel with varying bandwidth



Outline

Conclusion

- A generic framework to build nonlinear models.
- "Decouple", learning algorithms and data representation.
- More parameters to tune.
- Only need pairwise similarity : can handle non numeric data.
- Perform well on many problems.

Conclusion

- A generic framework to build nonlinear models.
- "Decouple", learning algorithms and data representation.
- More parameters to tune.
- Only need pairwise similarity : can handle non numeric data.
- Perform well on many problems.

Take away : It all depends on the kernel which you choose.

Practical