Support Vector Machines

Benoit Gaüzère

INSA Rouen Normandie - Laboratoire LITIS

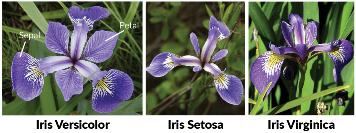
October 23, 2023

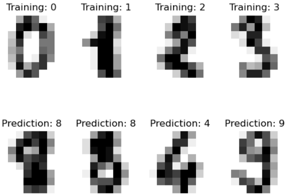
Classification I

Principle

Identify the category of an object

- Binary classification: Positive or Negative, $0, 1$ or $-1, 1$ (Breast Cancer,Spam detection, ...)
- Multi classification: More than 2 classes (object classification, iris: plant species, ...)





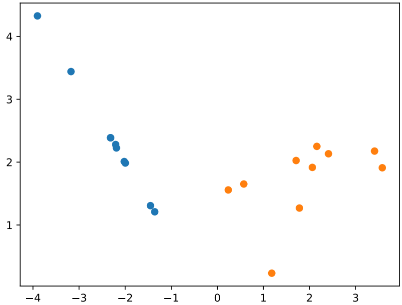
Classification II

Formally

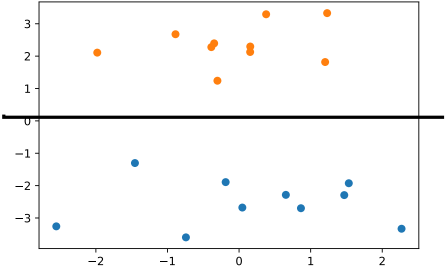
- Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1...N}$ with $\mathcal{Y} \in \{-1, 1\}$
- Prediction function f:
  - $f : \mathcal{X} \to \{-1, 1\}$
  - $f : \mathcal{X} \to \mathbb{R} : \begin{cases} f(\mathbf{x}) > 0 \to 1 \\ f(\mathbf{x}) < 0 \to -1 \end{cases}$
- Metrics: Accuracy, precision, recall, AUC, ...

Linearly separable problem

It exists at least one line which separates the data in two classes (in 2D)



Linear Classifier



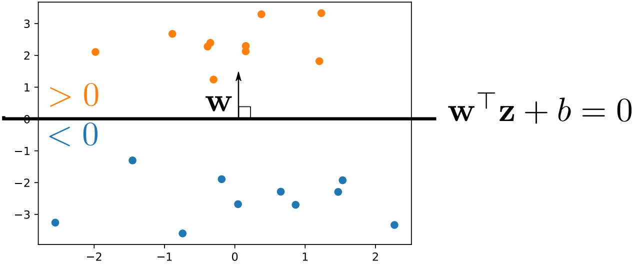
- Find a separator between classes
- Parameters of model : $\boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}$
- Decision function:

$$f(\mathbf{x}) = \boldsymbol{w}^\top \mathbf{x} + b \begin{cases} f(\mathbf{x}) > 0 \rightarrow 1 \\ f(\mathbf{x}) < 0 \rightarrow -1 \end{cases}$$

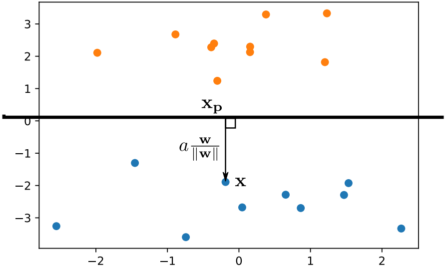
Hyperplane

Hyperplane $\mathcal{H}_{\boldsymbol{w},b}$:

$$\mathcal{H}_{\boldsymbol{w},b} = \{\boldsymbol{z} \in R^d | f(\boldsymbol{z}) = \boldsymbol{w}^\top \boldsymbol{z} + b = 0\}$$



Distance to the hyperplane



$$\begin{aligned} \mathbf{x} &= \mathbf{x}_p + a\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \\ \Rightarrow \quad a &= \frac{\boldsymbol{w}^\top \mathbf{x} + b}{\|\boldsymbol{w}\|} \end{aligned}$$

Proof

Distance $d(\mathcal{H}, \mathbf{x})$

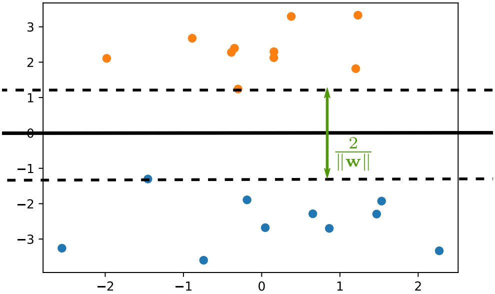
$$d(\mathcal{H}, \mathbf{x}) = |a| = \frac{|\boldsymbol{w}^\top \mathbf{x} + b|}{\|\boldsymbol{w}\|}$$

Definition of margin

Margin

- Minimum distance between a point and $\mathcal{H}$
- Canonical hyperplane :

$$\min_{\mathbf{x}_i \forall i \in 1 \dots N} \boldsymbol{w}^\top \mathbf{x}_i + b = 1$$



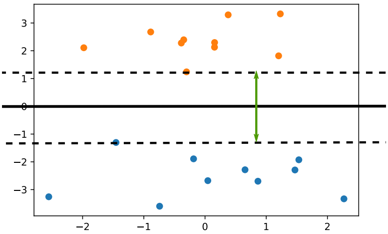
Maximization of margin

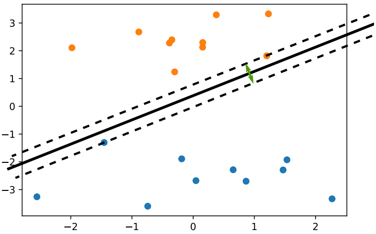
Better classifier separates the data

- Many different hyperplanes separate the data
- How to select the best ?
- $\Rightarrow$ Maximize the margin

Maximization of the margin

- Maximize the margin $\Leftrightarrow$ maximize $\frac{2}{\|\boldsymbol{w}\|}$
- $\boldsymbol{w}^\star = \operatorname{argmin}_{\boldsymbol{w}} \|\boldsymbol{w}\|$





Linear Separable SVM

Principle of SVM

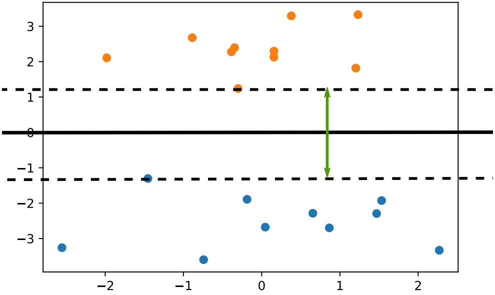
Find an hyperplane $\mathcal{H}$ which :

- separates well the data

$$y_i f(\mathbf{x}_i) > 1, \forall i \in 1 \dots N$$

- maximizes the margin

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|^2$$



Objective function

Hard-margin

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2$$

s.t.

$$y_i(\boldsymbol{w}^\top \mathbf{x}_i + b) \geq 1, \forall i \in 1 \dots N$$

Control + data term

Data term

$$y_i(\boldsymbol{w}^\top \mathbf{x}_i + b) \geq 1$$

- N Constraints
- Ensures that the train set is separated

Control term

$$\|\boldsymbol{w}\|^2$$

- Maximize the margin
- Selects the "best" model

How to resolve the SVM problem ?

Constraints

- We know how to optimize $\|\boldsymbol{w}\|^2$
- But the constraints ?

Solution

- Transform the problem
- Use Lagrangian dual

Lagrangian equivalence

Lagrangian

- Dual formulation of a constrained optimization problem
- Transform constraints to term
- Introduction of Lagrange multipliers for each constraint

Lagrangian of SVM

$$\mathcal{L}(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{N} \alpha_i (y_i(\boldsymbol{w}^\top \mathbf{x}_i + b) - 1)$$

Dual problem I

Dual SVM problem

- Annilihate the gradient wrt to primal variables
- Rewrite $\mathcal{L}(\boldsymbol{w}, b, \alpha)$ to eliminate primal variables
- Minimizing primal $\Leftrightarrow$ maximizing dual

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

s. t.

$$\alpha_i \geq 0, \forall i \in 1 \dots N$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

Dual problem II

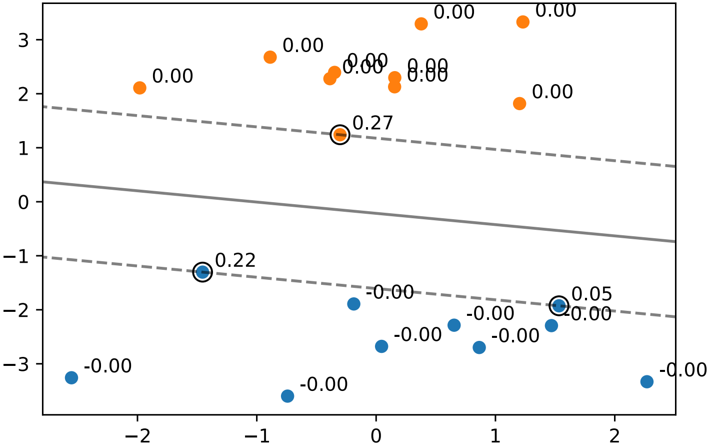
Resolving dual formulation

- Quadratic programming (use a solver)
- Compute optimal $\alpha^\star$

Dual variables α

- $\alpha^\star \in \mathbb{R}^N$ is the solution of dual SVM
- $\alpha_i^\star \neq 0$ for x_i in the margin
- $\alpha_i^\star = 0$ else.

Support vectors



Classification function

Retrieving $\boldsymbol{w}$

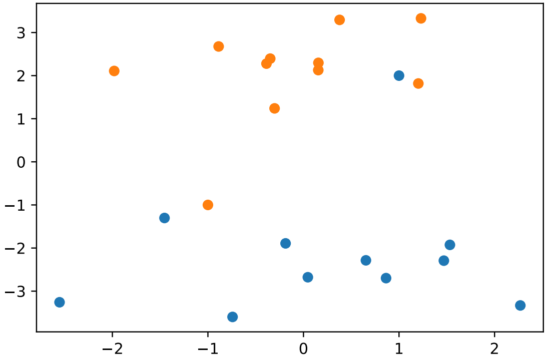
- $\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i \boldsymbol{y}_i \mathbf{x}_i$
- Decision function $f(\mathbf{x}')$

$$f(\mathbf{x}') = \boldsymbol{w}^\top \mathbf{x}' + b = \sum_{i=1}^{N} \alpha_i \boldsymbol{y}_i \mathbf{x}_i^\top \mathbf{x}' + b$$

Observations

- No need of $\boldsymbol{w}$ to predict
- Only scalar product between data
- Only few support vectors (sparsity)

How to deal with non separable case ?



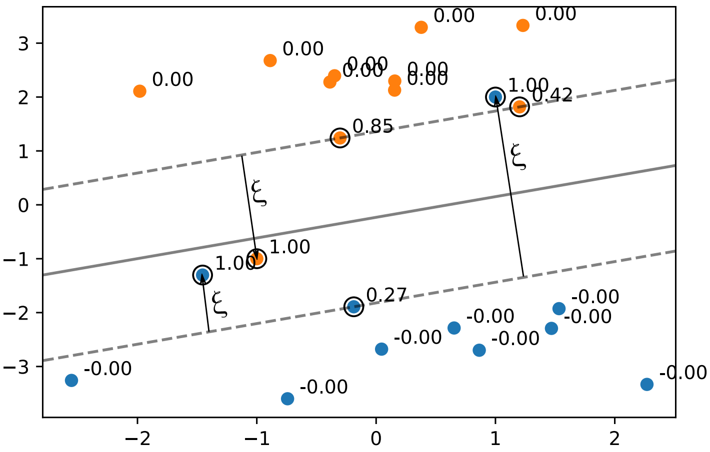
The ξ slack variables

Allow some errors

- Relax the margin by allowing errors
- Constraints:
$$\boldsymbol{y}_i(\boldsymbol{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i$$
- $\xi_i \geq 0$

Must be minimized

- Fit to data term
- We want to minimize the errors
- $\min \sum_i \xi_i$



0.00
0.00
0.00
0.00
0.00
0.00
0.00
0.00
1.00
0.42
0.85
ξ
ξ
1.00
1.00
ξ
0.27
-0.00
-0.00
-0.00
-0.00
-0.00
-0.00
-0.00
-0.00
3
2
1
0
−1
−2
−3
−2
−1
0
1
2

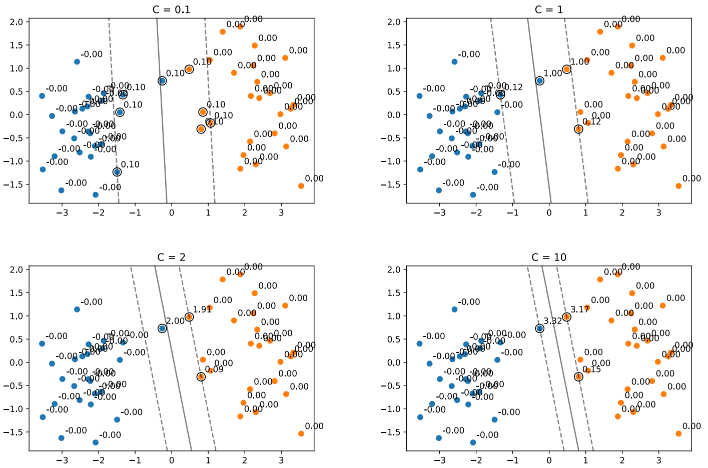
SVM-C Objective function

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

s.t.

$$\begin{aligned} & y_i(\boldsymbol{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i && ,\forall i \in 1 \ldots N \\ & \xi_i \geq 0 && ,\forall i \in 1 \ldots N \end{aligned}$$

- $C > 0$
- C balances the regularization and fit to data term
- Big C : small errors, small margin
- Low C : big errors, big margin



C = 0.1
C = 1
C = 2
C = 10

Dual formulation

Support vector values

- $0 \leq \alpha_i \leq C$

C parameter

- Controls the balance regularization/fit to data term
- Needs to be tuned

Let's try it

SVM for Regression : SVR

Regression

Regression problem

- Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1\ldots N}$ with $\mathcal{Y} \in \mathbb{R}$
- Prediction function f:

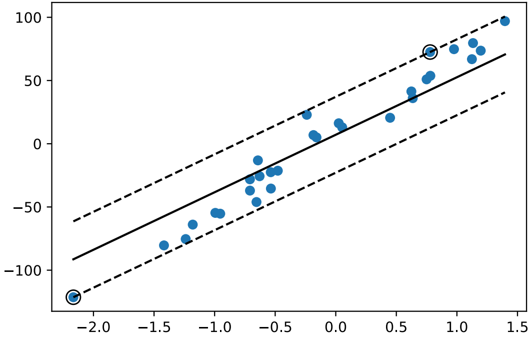
$$f(\mathbf{x}_i) = \boldsymbol{w}^\top \mathbf{x}_i + b \simeq \boldsymbol{y}_i$$

- Metrics: RMSE, MAE, R^2, ...
- Methods: Kernel Ridge Regression, ...

From classification to regression

How to adapt margin to regression ?

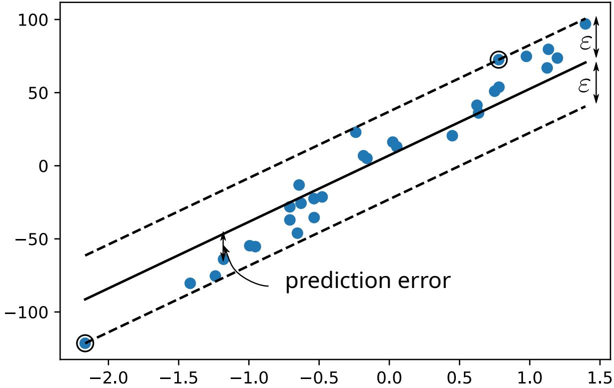
- We must gather the data
- We don't want to split
- How to ?



Adapting margin to regression

Solution

- Margin: contains the data
- $\boldsymbol{w}^\top \mathbf{x}_i + b \simeq y_i \Leftrightarrow \boldsymbol{w}^\top \mathbf{x}_i + b = y_i \pm \varepsilon$
- Adapt the size of margin ε to contain the data



SVR Objective function

SVR problem formulation

$$
\begin{aligned}
\min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 \\
\text{s t.} \quad & y_i - \boldsymbol{w}^\top x_i - b \le \varepsilon, \forall i \in 1 \dots N \\
& \boldsymbol{w}^\top x_i + b - y_i \le \varepsilon, \forall i \in 1 \dots N
\end{aligned}
$$

- $2N$ constraints
- ε insensitive cost

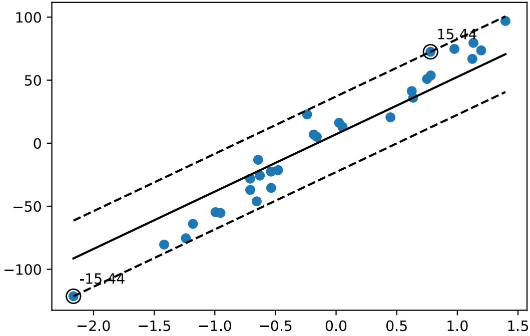
Hyperparameters

- ε : define the size of the margin
- Condition: it exists $\boldsymbol{w}, b$ which contains the data within ε.

Resolution

Dual variables

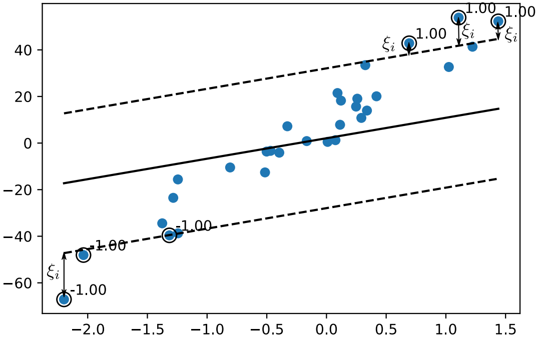
- $2N$ dual variables : $\alpha_i, \alpha^\star \geq 0$,
- One vector can be only on one margin
- $\alpha_i \neq 0 \Rightarrow \alpha_i^\star = 0$, and vice versa
- Constraint satisfied : $\alpha_i^{(\star)} = 0$



SVR with errors

Integrating errors

- Allowing to be outside the margin
- Manage outliers
- Relax the constraints with ξ_i values



SVR objective function I

Primal

$$
\begin{aligned}
\min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^\star) & \\
\text{s. t.} \quad & y_i - \boldsymbol{w}^\top x_i - b \leq \varepsilon + \xi_i & , \forall i \in 1 \dots N \\
& \boldsymbol{w}^\top x_i + b - y_i \leq \varepsilon + \xi_i^\star & , \forall i \in 1 \dots N \\
& \xi_i, \xi_i^\star \geq 0 & , \forall i \in 1 \dots N
\end{aligned}
$$

- $4N$ constraints
- New hyperparameter: $C \geq 0$

Dual Resolution

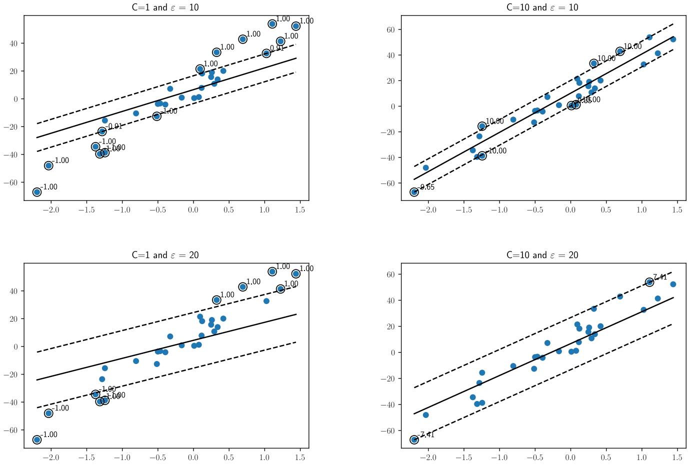
Dual variables

- $\alpha_i, \alpha_i^\star$ for errors constraints
- $\nu_i, \nu_i^\star$ for positivity on $\xi_i^{(\star)}$

Resolution

- $\nu_i^{(\star)} = C - \alpha_i^{(\star)}$
- $\boldsymbol{w} = \sum_{i=1}^N (\alpha_i - \alpha_i^\star)\mathbf{x}_i$
- $f(\mathbf{x}) = \sum_{i=1}^N (\alpha_i - \alpha_i^\star)\mathbf{x}_i^\top \mathbf{x} + b$

C and ε



Others variants of SVM

- Multiclass formulation: $\mathcal{Y} \in \mathbb{N}$
- One class SVM : unsupervised method to detect outliers
- ν-SVM : variant of C-SVM

Let's try it

Let's revisit SVM

Objective function

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

s. t.

$$\alpha_i \geq 0, \forall i \in 1 \ldots N$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

Decision function

$$f(\mathbf{x}') = \boldsymbol{w}^\top \mathbf{x}' + b = \sum_{i=1}^{N} \alpha_i \boldsymbol{y}_i \mathbf{x}_i^\top \mathbf{x}' + b$$

Observations

What does it mean ?

- Decision function is a linear combination of input data
- We don't need explicit data vectors $\mathbf{x}_i$
- We only need values of $\langle \mathbf{x}_i, \mathbf{x}_j \rangle, \forall i, j \in \{1..N\}^2$

Observations

Intuition

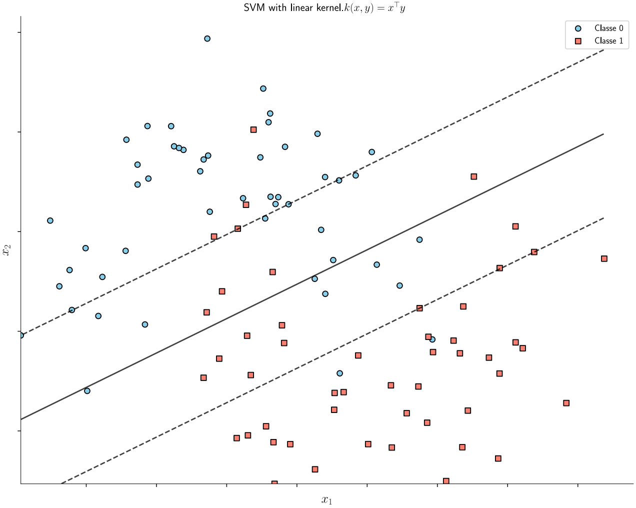
By simply modifying the dot product, the algorithm works in another space

This is the **Kernel Trick**

1. Define your algorithm as it access to data only through scalar products
2. Redefine your scalar product between data by a kernel $k(\cdot, \cdot)$
3. Replace standard scalar product by k
4. Enjoy

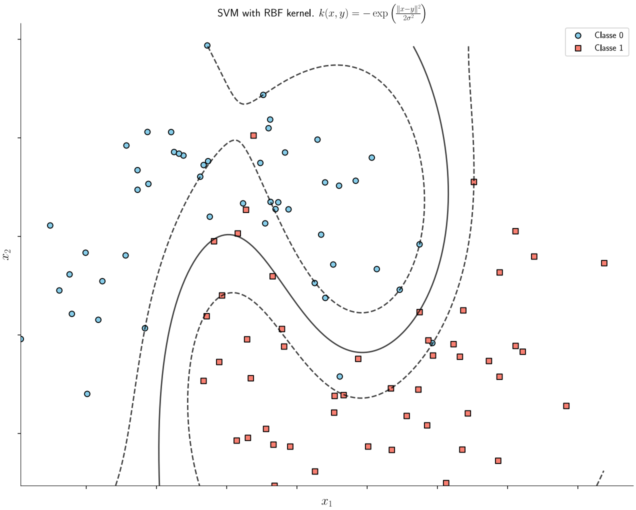
Non linear SVM

Linear SVM



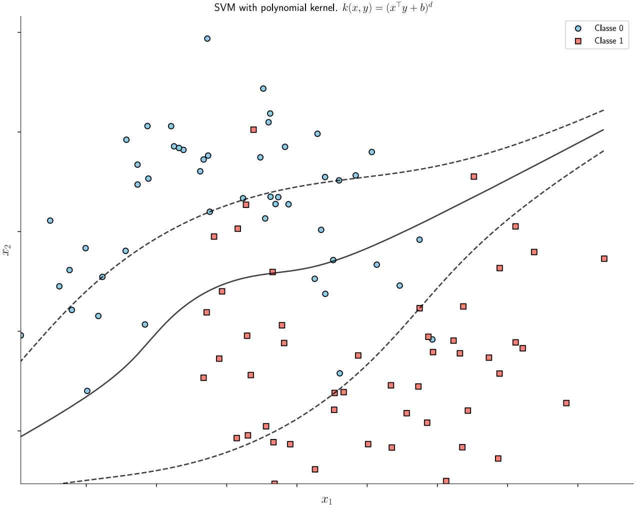
Non linear SVM

Linear SVM



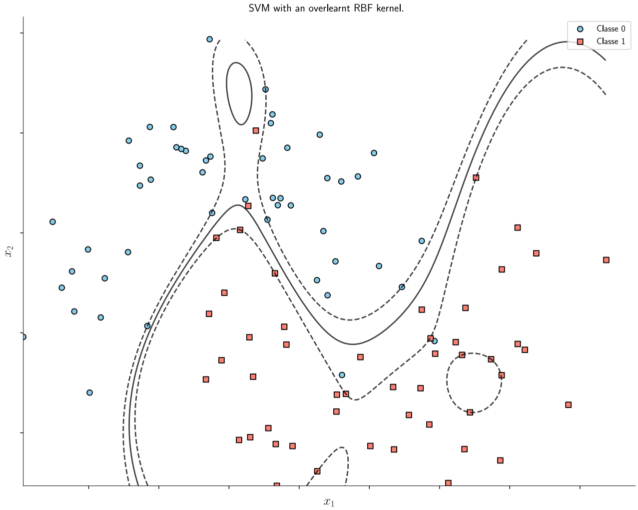
Non linear SVM

Linear SVM



Non linear SVM

Linear SVM



What is a kernel ?

What can be k ?

Prerequisites

Some definitions and notations

- $\mathcal{X}$: Non empty input space (set of $\mathbb{R}^N$, graphs, objects, ...)
- $x \in \mathcal{X}$, $\mathbf{x} \in \mathbb{R}^d$
- $\mathcal{H}$: feature space with a dot product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$
- $\Phi : \mathcal{X} \to \mathcal{H}$: embedding function from $\mathcal{X}$ to $\mathcal{H}$

Kernel

Definition

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

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

Positive Definite Kernels

Gram Matrix

Given a kernel $k : \mathcal{X}^2 \to \mathbb{R}$, and $\{x_1, \ldots, x_n\} \subseteq \mathcal{X}$, the corresponding Gram Matrix $\mathbf{K}$ is a $n \times n$ matrix whose elements :

$$\mathbf{K}_{i,j} := k(x_i, x_j)$$

Positive Semi-Definite Matrix

- if $\mathbf{K}$ is symmetric and $\mathbf{c}^T\mathbf{K}\mathbf{c} > 0, \forall \mathbf{c} \neq 0$, K is a positive definite matrix
- if $\mathbf{K}$ is symmetric and $\mathbf{c}^T\mathbf{K}\mathbf{c} \geq 0, \forall \mathbf{c} \neq 0$, K is a positive semi-definite matrix.

Equivalently: $\sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{c}_i \mathbf{c}_j \mathbf{K}_{i,j} \geq 0$

Positive Definite Kernels I

Definition

If for any subset $\mathcal{X}' \subseteq \mathcal{X}, |\mathcal{X}'| = n$, the associated Gram Matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ is positive semi-definite, then k is a positive definite kernel on $\mathcal{X}$.

- Usually, we talk about kernels. Positiveness is implicit.
- Verifying $\mathbf{K}$ positive semi-definiteness consists in computing eigenvalues $\lambda_1 > \cdots > \lambda_n$. if $\lambda_n \geq 0$, then $\mathbf{K}$ is positive semi-definite.
- Keep in mind that k corresponds to a scalar product in $\mathcal{H}$, so:
  - $k(x_i, x_j) = k(x_j, x_i)$: Then $\mathbf{K}$ is symmetric.
  - Consider $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\mathbf{K} = \mathbf{X}\mathbf{X}^\top$. Eigenvalues > 0 follows.

Reproducing Kernel Hilbert Space

RKHS

- $\mathcal{H}$ is a:
  - pre-Hilbert space of functions
  - endowed with a dot product
  - and we add a norm $\|f\| := \sqrt{\langle f, f \rangle}$
- $\mathcal{H}$ is a Hilbert space.
- Hilbert space: Generalization of euclidean space to finite or infinite dimension

> $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS) associated to kernel k

Let's summarize

From kernel to feature space

Given a valid kernel k, we can associate a RKHS $\mathcal{H}$ which corresponds to the feature space of k.

From feature space to kernel

Now consider that you have $\Phi : \mathcal{X} \to \mathcal{H}$ a mapping function.
A positive kernel k is defined by:

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

Kernels in Practice

Linear Kernel

$k(\mathbf{s}, \mathbf{t}) = \mathbf{s}^\top \mathbf{t}$

- $\mathbf{s}, \mathbf{t} \in \mathbb{R}^d$
- symmetric: $\mathbf{s}^\top \mathbf{t} = \mathbf{t}^\top \mathbf{s}$
- positive:

$$\begin{aligned}
\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) &= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \mathbf{x}_i^\top \mathbf{x}_j \\
&= \left(\sum_{i=1}^{n} \alpha_i \mathbf{x}_i \right)^\top \left(\sum_{j=1}^{n} \alpha_j \mathbf{x}_j \right) \\
&= \left\| \sum_{i=1}^{n} \alpha_i x_i \right\|^2
\end{aligned}$$

Product kernel

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

- $x, x' \in \mathcal{X}$
- for some $g : \mathcal{X} \to \mathbb{R}$
- symmetric: by construction
- positive:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j g(\mathbf{x}_i)g(\mathbf{x}_j)$$

$$= \left(\sum_{i=1}^{n}\alpha_i g(\mathbf{x}_i)\right)\left(\sum_{j=1}^{n}\alpha_j g(\mathbf{x}_j)\right)$$

$$= \left(\sum_{i=1}^{n}\alpha_i g(\mathbf{x}_i)\right)^2$$

Polynomial kernels I

A first approach

- $\mathbf{s}, \mathbf{t} \in \mathbb{R}^N$
- All ordered combinations of degree d, e.g.:

$$\Phi : \mathbb{R}^2 \to \mathcal{H} = \mathbb{R}^4$$
$$(\mathbf{s}_1, \mathbf{s}_2) \mapsto (\mathbf{s}_1^2, \mathbf{s}_2^2, \mathbf{s}_1\mathbf{s}_2, \mathbf{s}_2\mathbf{s}_1)$$

- Dimension of $\mathcal{H}$: $\frac{(d+N-1)!}{d!(N-1)!}$
- Untractable !

Polynomial kernels II

$k(\mathbf{s}, \mathbf{t}) = \langle \mathbf{s}, \mathbf{t} \rangle^d$, $\mathbf{s}, \mathbf{t} \in \mathbb{R}^N$

- Two valid feature spaces:
- All ordered combinations of degree d, e.g.:

$$\begin{aligned} \Phi : \mathbb{R}^2 &\rightarrow \mathcal{H} = \mathbb{R}^4 \\ (\mathbf{s}_1, \mathbf{s}_2) &\mapsto (\mathbf{s}_1^2, \mathbf{s}_2^2, \mathbf{s}_1\mathbf{s}_2, \mathbf{s}_2\mathbf{s}_1) \end{aligned}$$

- All unordered combinations of degree d, e.g.:

$$\begin{aligned} \Phi : \mathbb{R}^2 &\rightarrow \mathcal{H} = \mathbb{R}^3 \\ (\mathbf{s}_1, \mathbf{s}_2) &\mapsto (\mathbf{s}_1^2, \mathbf{s}_2^2, \sqrt{2}\mathbf{s}_1\mathbf{s}_2) \end{aligned}$$

- Also: $(\mathbf{s}^\mathsf{T}\mathbf{t} + c)^d$, $c \in \mathbb{R}^+$.
- High dimensional feature space but k is computed in $\mathcal{O}(n)$

Generalisation: finite kernel

Embedding

- Let Φ_j, for $j = 1, \ldots, p$ be a finite dictionary of functions $\mathcal{X} \to \mathbb{R}$ (polynomials, wavelets, ...)
- Feature map:

$$\begin{aligned} \Phi : \mathcal{X} &\to \mathbb{R}^p \\ \mathbf{s} &\mapsto \big(\Phi_1(x), \ldots, \Phi_p(x')\big) \end{aligned}$$

- Linear kernel in the feature space:

$$k(x, x') = \big(\Phi_1(x), \ldots, \Phi_p(x)\big)^\top \big(\Phi_1(x'), \ldots, \Phi_p(x')\big)$$

Gaussian kernel

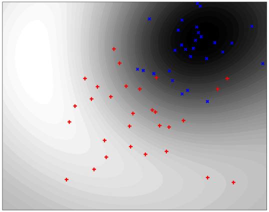
$k(\mathbf{s}, \mathbf{t}) = \exp\left(-\frac{\|\mathbf{s}-\mathbf{t}\|^2}{2\sigma^2}\right)$

- for $\sigma = 1$:

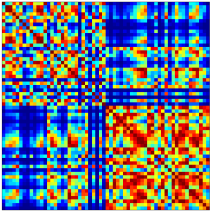
$$\Phi(\mathbf{s}) = \left(\frac{\exp\frac{\|\mathbf{s}\|^2}{2j}}{\sqrt{j!}^{1/j}}\binom{j}{n_1, \ldots, n_k}^{1/2}\mathbf{s}_1^{n_1}\ldots\mathbf{s}_k^{n_k}\right)_{j=0,\ldots,\infty,\sum_{i=1}^k n_i = j}$$

- Feature space has an infinite dimension
- Overlearning ⚠
- σ controls the influence area of the kernel
- σ is another hyperparameter

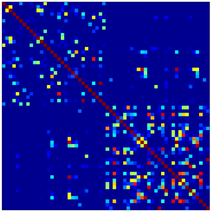
Examples of Gram matrices with different bandwidth



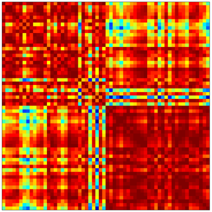
raw data



Gram matrix for $b = 2$



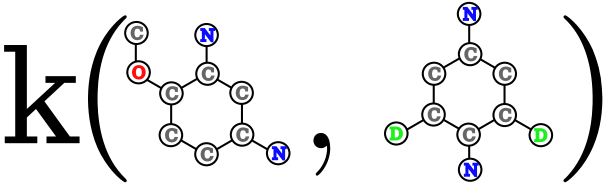
$b = .5$



$b = 10$

Kernels on structures

- $\mathcal{X}$ may not be a vector space.
- we can define kernels on any kind of data :
  - Strings
  - Time series
  - Graphs
  - Images
  - ...



Kernel from distances I

Kernel and distance

- Distance is a dissimilarity measure between vectors or objects

$$
\begin{aligned}
{d_m}^2(\mathbf{s},\mathbf{t}) =& \|\mathbf{s}-\mathbf{t}\|_2^2 \\
=& (\mathbf{s}-\mathbf{t})^\top(\mathbf{s}-\mathbf{t}) \\
=& \mathbf{s}^\top\mathbf{s}+\mathbf{t}^\top\mathbf{t}-2\mathbf{s}^\top\mathbf{t} \\
=& \langle\mathbf{s},\mathbf{s}\rangle+\langle\mathbf{t},\mathbf{t}\rangle-2\langle\mathbf{s},\mathbf{t}\rangle \\
=& k(\mathbf{s},\mathbf{s})+k(\mathbf{t},\mathbf{t})-2\,k(\mathbf{s},\mathbf{t})
\end{aligned}
$$

- For normalized kernels ($k(x,x')=1$) kernel is proportional to the opposite of squared distance
- Kernels correspond to similarity measures

Kernel from distances II

From distance to kernels

- We can define a kernel from an euclidean distance
- Usually we plug a distance in Gaussian Kernel
- Use of distance map

$$\mathcal{X} \rightarrow \mathbb{R}^n$$
$$\Phi(x) = (d_m(x, x_1), \ldots, d_m(x, x_n))$$

- Related to kernel feature map

Kernel jungle

Appendix D

List of kernels

D.1 Kernel definitions and computations

Reference	Title	Page
Definition 9.1	Polynomial kernel	292
Computation 9.6	All-subsets kernel	295
Computation 9.8	Gaussian kernel	296
Computation 9.12	ANOVA kernel	299
Computation 9.18	Alternative recursion for ANOVA kernel	302
Computation 9.24	General graph kernels	307
Definition 9.33	Exponential diffusion kernel	313
Definition 9.34	von Neumann diffusion kernel	313
Computation 9.35	Evaluating diffusion kernels	314
Computation 9.46	Evaluating randomised kernels	321
Definition 9.37	Intersection kernel	315
Definition 9.38	Union-complement kernel	316
Remark 9.40	Agreement kernel	316
Definition 9.41	Derived subsets kernel	317
Section 9.6	Kernels on real numbers	318
Remark 9.45	Spline kernels	320
Definition 10.5	Vector space kernel	331
Computation 10.8	Latent semantic kernels	338
Definition 11.7	The p-spectrum kernel	348
Computation 11.10	The p-spectrum recursion	349
Remark 11.13	Blended spectrum kernel	350
Computation 11.17	All-subsequences kernel	353
Computation 11.24	Fixed length subsequences kernel	358
Computation 11.33	Naive recursion for gap-weighted subsequences kernel	364
Computation 11.36	Gap-weighted subsequences kernel	366
Computation 11.45	Trie-based string kernels	373

448

D.2 Kernel algorithms 449

D.2 Kernel algorithms

Reference	Title	Page
Algorithm 9.14	ANOVA kernel	300
Algorithm 9.25	Simple graph kernels	308
Algorithm 11.20	All–non-contiguous subsequences kernel	356
Algorithm 11.25	Fixed length subsequences kernel	358
Algorithm 11.38	Gap-weighted subsequences kernel	367
Algorithm 11.40	Character weighting string kernel	370
Algorithm 11.41	Soft matching string kernel	371
Algorithm 11.42	Gap number weighting string kernel	372
Algorithm 11.46	Trie-based p-spectrum kernel	374
Algorithm 11.51	Trie-based mismatch kernel	377
Algorithm 11.54	Trie-based restricted gap-weighted kernel	380
Algorithm 11.62	Co-rooted subtree kernel	386
Algorithm 11.65	All-subtree kernel	389
Algorithm 12.8	Fixed length HMM kernel	409
Algorithm 12.14	Pair HMM kernel	415
Algorithm 12.17	Hidden tree model kernel	419
Algorithm 12.34	Fixed length Markov model Fisher kernel	435

[Shawe-Taylor and Cristianini, 2004]

Invalid kernels

Danger

Some similarity measures may be invalid kernels

- $k(x, y) = \max(x, y)$.
- Optimal assignment kernel: [Fröhlich et al., 2005]
- and many more ...
- The use is not forbidden, but handle with care
- $\rightarrow$ operating in Krein spaces: [Loosli et al., 2013]

Kernel algebra

Convex cone:

The set of kernels forms a convex cone, closed under pointwise convergence.

- **Linear combination**:
  - if k_1 an k_2 are kernels, a_1,$a_2 \geq 0$, then $a_1 k_1 + a_2 k_2$ is a kernel
  - if k_1, k_2, ...are kernels, and $k(x, x') := \lim_{n \to \infty} k_n(x, x')$ exists for all x, x', then k is a kernel
- **Product kernel**:
  if k_1 an k_2 are kernels, then $k_1 k_2(x, x') := k_1(x, x') k_2(x, x')$ is a kernel.

And some molecular graphs kernels

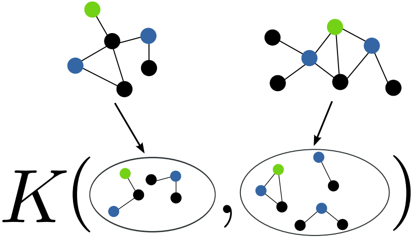
How to define the similarity between molecules ?

Graph kernel based on bags of patterns

(1) Extraction of a set of patterns,

(2) Comparison between patterns,

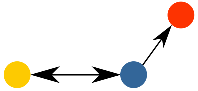
(3) Comparison between bags of patterns.

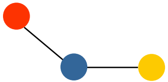


Patterns

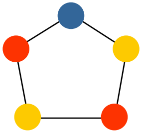
Linear Patterns

- **Random Walks** ∞ **Kashima et al. [2003]**
  - − Tottering
  - \+ Mahé et al. [2004b].





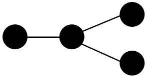
- **Paths** $\not\infty$ **Ralaivola et al. [2005]**
  - − Low branching description

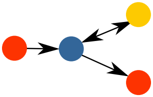


- **Cyclic patterns** $\not\infty$ **Horváth et al. [2004]**
  - \+ Cyclic information
  - \+ Relevant in chemoinformatics
  - − Only a partial cyclic information

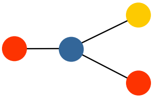
Patterns

- **Graphlets** $\not\propto$ **Shervashidze et al. [2009]**
  - ✚ Non linear structures.
  - ▬ Non labeled patterns.
  - ✚ Linear complexity.





- **Tree-patterns** $\propto$ **Mahé and Vert [2009], ?**
  - ✚ Non linear and labeled patterns.



- **Treelets** $\not\propto$ **Gaüzère et al. [2012b]**
  - ✚ Non linear and labeled patterns.

Convolution Kernels

Counting function

- $f_p(G)$: Number of occurences of pattern p in G.

Kernel definition

$$k_{\mathcal{T}}(G, G') = \sum_{p \in \mathcal{P}(G) \cap \mathcal{P}(G')} k_p(G, G')$$

- $\mathcal{P}(G)$: Set of patterns extracted from G.
- $k_p(G, G') = k(f_p(G), f_p(G'))$.
- $k_p(.,.)$: Similarity according to p.

Molecular similarity

$\Longleftrightarrow$

Similarity of their bags of patterns

Conclusion

SVM

- Nice framework
- Good mathematical foundations
- Kernel trick : extension to non linear models and any data

Limitations

- Need to define and compute a kernel
- Still need to handcraft features (or kernel)

References

References

10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009, 2009. IEEE Computer Society.

Edoardo Amaldi and Viggo Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. Theoretical Computer Science, 209(1):237–260, 1998.

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

A. Barla, F. Odone, and A. Verri. Histogram intersection kernel for image classification. In Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on, volume 3, pages III–513. IEEE, 2003. ISBN 0780377508.

Yakir Berchenko and Mina Teicher. Graph embedding through random walk for shortest paths problems. In Proceedings of the 5th international conference on Stochastic algorithms: foundations and applications, SAGA'09, pages 127–140, Berlin, Hei-