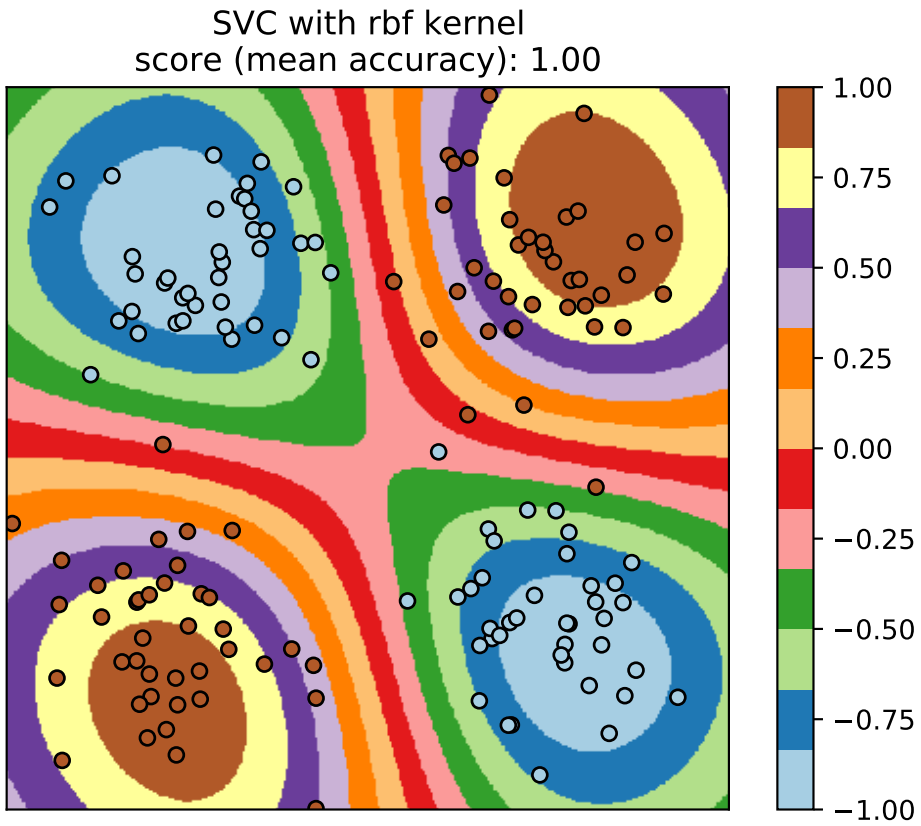
Fondamentaux théoriques du machine learning



Gradient algorithms II

- Bounds and actual convergence speeds
- Line search (GD)
- Extensions of SGD

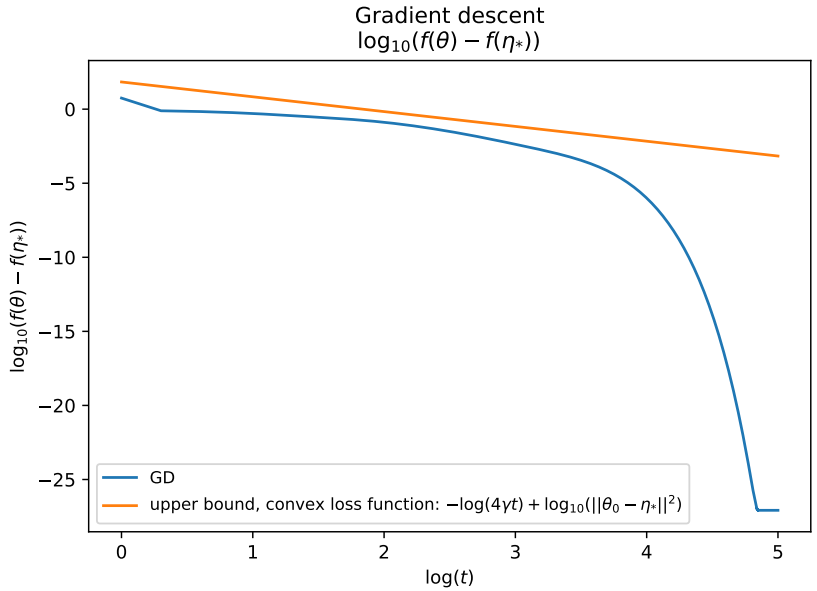
Kernels

- Introduction
- Representer theorem

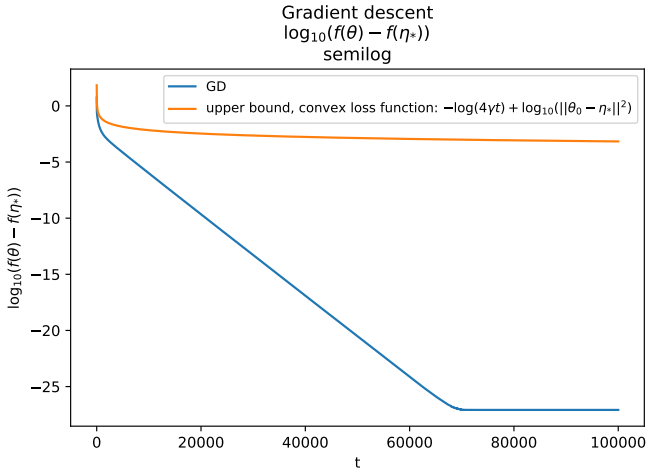
Support vector machines

- Linear separation
- Optimization problem
- Link with empirical risk minimization
- Kernels

Upper bounds are just bounds



In lecture 6, we have shown that for a smooth, convex function, with the appropriate learning rate we have a convergence rate in $O(\frac{1}{t})$ for the function values.



It seems that with this function, we observe

- a phase of convergence $\mathcal{O}(\frac{1}{t})$, since $\log_{10}\left(f(\theta) - f(\eta_*)\right)$ decreases approximately as $-\log(t)$
- a phase of exponential convergence, approximately when $\log(t) \geq 4$

Exercice 1 : **Why ?**

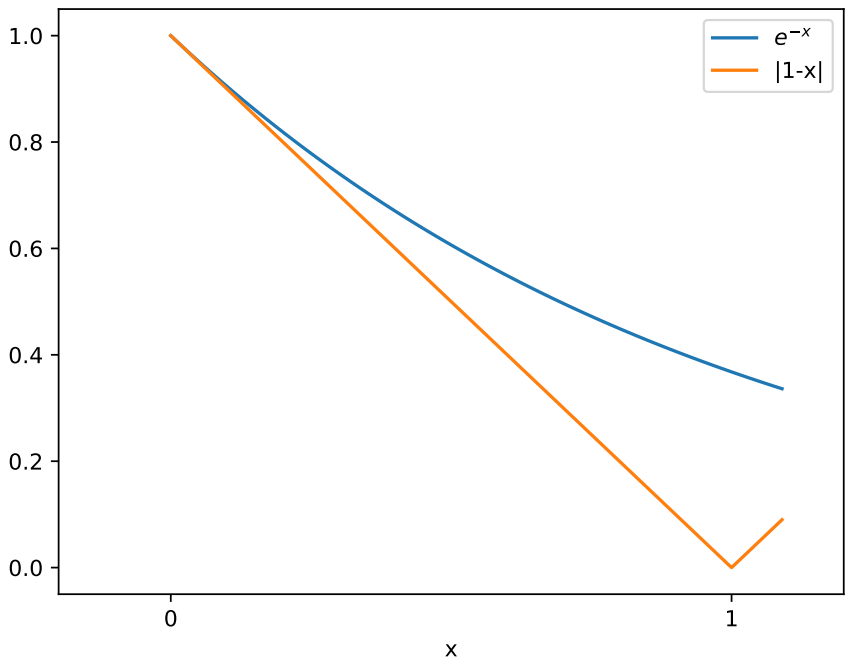
Convergence in function values

We recall that

$$f(\theta_t) - f(\eta^*) = \frac{1}{2}(\theta_0 - \eta^*)^T (I - \gamma H)^{2t} H (\theta_0 - \eta^*) \qquad (1)$$

The eigenvalues of $(I - \gamma H)^{2t} H$ are the $\lambda(1 - \gamma\lambda)^{2t}$, with λ being an eigenvalue of H.
If $\gamma \leq \frac{1}{L}$, we have $0 \leq \gamma\lambda \leq 1$.

If $\gamma \leq \frac{1}{L}$, we have $0 \leq \gamma\lambda \leq 1$ and $|1 - \gamma\lambda| \leq \exp(-\gamma\lambda)| \leq 1$.



Convergence rate

$$\sup_{\alpha \geq 0} \alpha \exp(-\alpha) = \frac{1}{e} \tag{2}$$

The maximum is attained for $\alpha = 1$.
However, when t increases, α increases, and we might observe an exponential convergence phase. This depends on the distribution on the eigenvalues of H.
In the practical session, H had several very small eigenvalues.

Local optimization of γ

Considering an fixed iteration step θ_t, we note

$$\alpha(\gamma) = \theta_t - \gamma \nabla_{\theta_t} f \tag{3}$$

The **exact line seach** method attempts to find the optimal step γ^*, at each iteration.

Local optimization of γ

Considering an fixed iteration step θ_t, we note

$$\alpha(\gamma) = \theta_t - \gamma \nabla_{\theta_t} f \tag{4}$$

Given the position θ_t, find γ^* that minimizes $\gamma \mapsto g(\gamma)$ with

$$\begin{aligned} g(\gamma) &= f(\theta_t - \gamma \nabla_{\theta_t} f) \\ &= f(\alpha(\gamma)) \end{aligned} \tag{5}$$

Line search

We note that

$$\begin{aligned}\nabla_{\alpha(\gamma)} f &= H\alpha(\gamma) - \frac{1}{n} X^T y \\ &= H(\theta_t - \gamma \nabla_{\theta_t} f) - \frac{1}{n} X^T y \\ &= \nabla_{\theta_t} f - \gamma H \nabla_{\theta_t} f\end{aligned} \tag{6}$$

Minimization of g

We can derivate g with respect to γ.

$$\begin{aligned} g'(\gamma) &= \langle \nabla_{\alpha(\gamma)} f, -\alpha'(\gamma) \rangle \\ &= -\langle \nabla_{\theta_t} f - \gamma H \nabla_{\theta_t} f, \nabla_{\theta_t} f \rangle \\ &= -\langle \nabla_{\theta_t} f, \nabla_{\theta_t} f \rangle + \langle \gamma H \nabla_{\theta_t} f, \nabla_{\theta_t} f \rangle \\ &= -||\nabla_{\theta_t} f||^2 + \gamma \langle H \nabla_{\theta_t} f, \nabla_{\theta_t} f \rangle \end{aligned} \tag{7}$$

Minimization of g

$$\begin{aligned} g'(\gamma) &= \langle \nabla_{\alpha(\gamma)} f, -\alpha'(\gamma) \rangle \\ &= -\langle \nabla_{\theta_t} f - \gamma H \nabla_{\theta_t} f, \nabla_{\theta_t} f \rangle \\ &= -\langle \nabla_{\theta_t} f, \nabla_{\theta_t} f \rangle + \langle \gamma H \nabla_{\theta_t} f, \nabla_{\theta_t} f \rangle \\ &= -||\nabla_{\theta_t} f||^2 + \gamma \langle H \nabla_{\theta_t} f, \nabla_{\theta_t} f \rangle \end{aligned} \tag{8}$$

In order to cancel the derivative, we must have that

$$\gamma^* = \frac{||\nabla_{\theta_t} f||^2}{\langle H \nabla_{\theta_t} f, \nabla_{\theta_t} f \rangle} \tag{9}$$

We note that this is correct if $\nabla_{\theta_t} f \neq 0$. If $\nabla_{\theta_t} f = 0$, this means that $\theta_t = \eta^*$, as f is convex.

Orthogonality of descent directions

If we note $\theta^*_{t+1} = \theta_t - \gamma^* \nabla_{\theta_t} f = \alpha(\gamma^*)$, then $g'(\gamma^*) = 0$ translates into

$$\langle \nabla_{\theta^*_{t+1}} f, \nabla_{\theta_t} f \rangle = 0 \tag{10}$$

Two optimal directions of the gradient updates are **orthogonal**. Importantly, this is true in the general case, not only for least-squares.

Heavy-ball

We add a **momentum term** to the gradient update.

$$\theta_{t+1} = \theta_t - \gamma \nabla_{\theta_t} f + \beta(\theta_t - \theta_{t-1}) \tag{11}$$

SGA

Output an average of the θ_t after running the algorithm (see tomorrow's practical session)

SAG

By storing the previously computed gradients, we progressively build an estimation of the batch gradient, while updating θ (see tomorrow's practical session).

Introduction to Kernel methods

Replace inputs $x \in \mathcal{X}$ by a function $\phi(x) \in \mathcal{H}$, with $\mathcal{H}$ a $\mathbb{R}$-Hilbert space. We then perform linear predictions on $\phi(x)$. This means that estimators have the form :

$$f(x) = \langle \theta, \phi(x) \rangle_{\mathcal{H}} \tag{12}$$

Introduction to Kernel methods

Replace inputs $x \in \mathcal{X}$ by a function $\phi(x) \in \mathcal{H}$, with $\mathcal{H}$ a $\mathbb{R}$-Hilbert space. We then perform linear predictions on $\phi(x)$. This means that estimators have the form :

$$f(x) = \langle \theta, \phi(x) \rangle_{\mathcal{H}} \tag{13}$$

- $\langle ., . \rangle_{\mathcal{H}}$: inner product defined on $\mathcal{H}$. When there is no ambiguity, we will note $\langle ., . \rangle = \langle ., . \rangle_{\mathcal{H}}$.
- $\theta \in \mathcal{H}$
- $\phi(x)$: **feature** associated to x, $\mathcal{H}$: **feature space** .

Interest

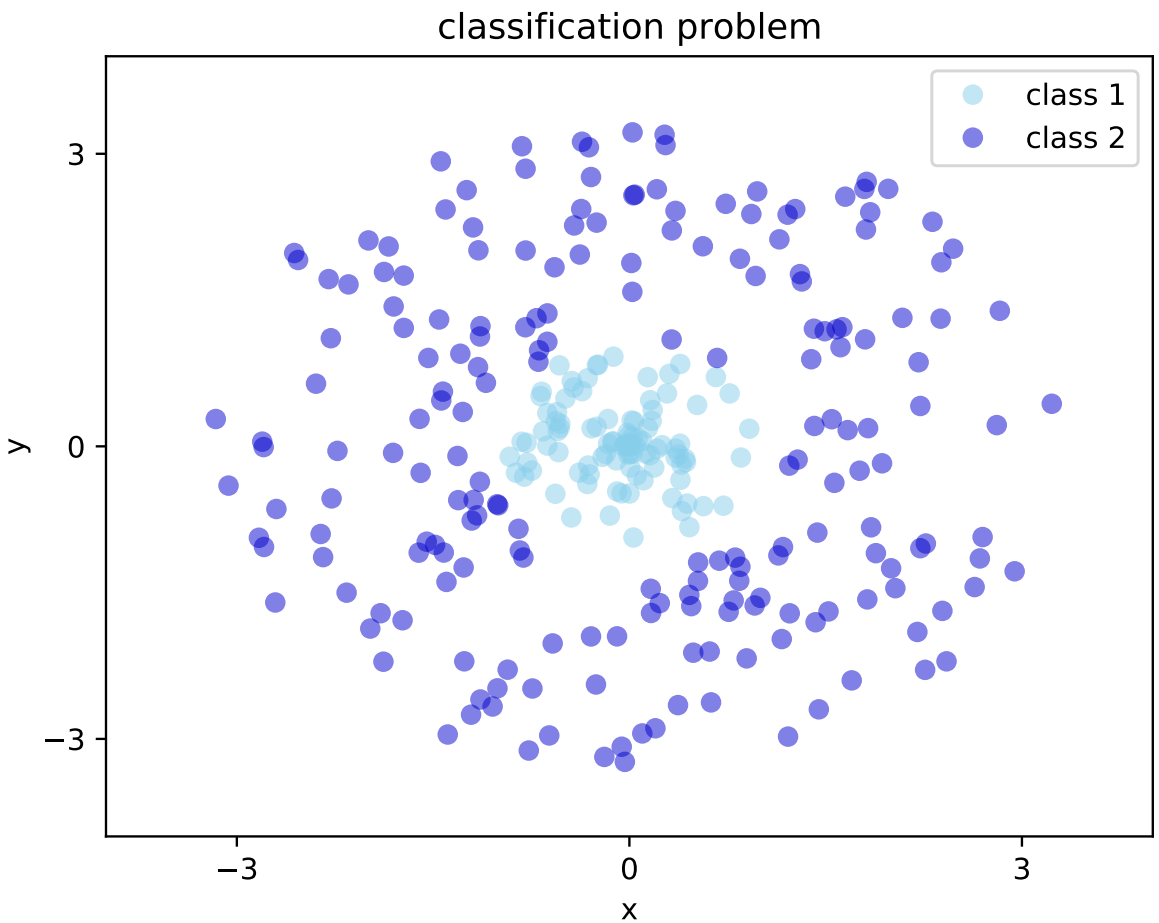
- Kernel methods provide stable algorithms, with theoretical convergence guarantees.
- They can benefit from the smoothness (regularity) of the target function, whereas local averaging methods cannot.
- They can be applied in high dimension.

In some supervised learning problems with many observations, such as computer vision and natural language processing, they are now outperformed by neural networks.

Feature maps

- $\mathcal{X}$ need not be a vector space.
- $\phi(x)$ can provide more useful **representation** of the input for the considered problem (classification, regression).
- The prediction function is then allowed to depend **non-linearly** on x.

Nonlinear data separation



Feature space

Often, $\mathcal{H} = \mathbb{R}^d$, but importantly, we will see that d can even be **infinite**, thanks to a computation trick called the **kernel trick**.

Representer theorem

We consider a framework where we look for a minimizer $\hat{\theta}$ of a loss such as

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, \langle \theta, \phi(x_i) \rangle + \lambda ||\theta||^2 \tag{14}$$

The key aspect is that the input observations $x_i \in \mathcal{X}$ are only accessed through inner products with θ.

Representer theorem

Theorem

Let $\Psi : \mathbb{R}^{n+1} \to \mathbb{R}$ *be a strictly increasing function with respect to the last variable. Then, the minimum of*

$$L(\theta) = \Psi(\langle \theta, \phi(x_i) \rangle, \ldots, \langle \theta, \phi(x_n) \rangle, ||\theta||^2)$$

is attained for $\hat{\theta} \in Vect(\{\phi(x_i)\}$. *We can write*

$$\theta = \sum_{i=1}^{n} \alpha_i \phi(x_i), \alpha \in \mathbb{R}^n$$

Representer theorem

Exercice 2 : **Proove the theorem**

Proof I

Let $\mathcal{H}_D = \{\sum_{i=1}^{n} \alpha_i \phi(x_i), \alpha \in \mathbb{R}^n\}$. For all $\theta \in \mathcal{H}$, we have a decomposition

$$\theta = \theta_D + \theta_{D^\perp} \tag{15}$$

with $\theta_D \in \mathcal{H}_D$ and $\theta_{D^\perp} \in \mathcal{H}_D^\perp$.
Then, $\forall i \in \{1, \ldots, n\}$,

$$\begin{aligned} \langle \theta, \phi(x_i) \rangle &= \langle \theta_D, \phi(x_i) \rangle + \langle \theta_{D^\perp}, \phi(x_i) \rangle \\ &= \langle \theta_D, \phi(x_i) \rangle \end{aligned} \tag{16}$$

Furthermore,

$$||\theta||^2 = ||\theta_D||^2 + ||\theta_{D^\perp}||^2 \tag{17}$$

Proof II

Hence

$$\begin{aligned} & \Psi(\langle\theta, \phi(x_i)\rangle, \ldots, \langle\theta, \phi(x_n)\rangle, ||\theta||^2) \\ = \Psi(\langle\theta_D, \phi(x_i)\rangle, \ldots, \langle\theta_D, \phi(x_n)\rangle, ||\theta_D||^2 + ||\theta_{D^\perp}||^2) & \qquad (18) \\ \geq \Psi(\langle\theta_D, \phi(x_i)\rangle, \ldots, \langle\theta_D, \phi(x_n)\rangle, ||\theta_D||^2) & \end{aligned}$$

This means that

$$\begin{aligned} & \inf_{\theta\in\mathcal{H}} \Psi(\langle\theta, \phi(x_i)\rangle, \ldots, \langle\theta, \phi(x_n)\rangle, ||\theta||^2) \\ = & \inf_{\theta\in\mathcal{H}_D} \Psi(\langle\theta, \phi(x_i)\rangle, \ldots, \langle\theta, \phi(x_n)\rangle, ||\theta||^2) \end{aligned} \qquad (19)$$

Application to supervised learning

The loss

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, \langle \theta, \phi(x_i)\rangle + \lambda||\theta||^2 \tag{20}$$

is of the form Ψ and is increasing in the last variable.
As a direct consequence, the minimum of 20 is attained at

$$\theta = \sum_{i=1}^{n} \alpha_i \phi(x_i), \alpha \in \mathbb{R}^n \tag{21}$$

We note that no convexity hypothesis on l is required.

Consequence

We note

- α the vector such that $\theta = \sum_{i=1}^{n} \alpha_i \phi(x_i)$.
- $K \in \mathbb{R}^{n,n}$ the matrix defined by

$$K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$$

Consequence

We remark that $\forall i \in [1, n]$,

$$\begin{aligned}\langle \theta, \phi(x_i) \rangle &= \sum_{j=1}^{n} \alpha_j \langle \phi(x_j), \phi(x_i) \rangle \\ &= \sum_{j=1}^{n} \alpha_j K_{ij} \\ &= (K\alpha)_i\end{aligned}$$

And we also remark that

$$
\begin{aligned}
||\theta||^2 &= \langle \theta, \theta \rangle \\
&= \langle \sum_{i=1}^{n} \alpha_i \phi(x_i), \sum_{i=j}^{n} \alpha_j \phi(x_j) \rangle \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \langle \phi(x_i), \phi(x_j) \rangle \\
&= \sum_{i=1}^{n} \alpha_i (\sum_{j=1}^{n} K_{ij} \alpha_j) \\
&= \sum_{i=1}^{n} \alpha_i (K\alpha)_i \\
&= \alpha^T K \alpha
\end{aligned}
$$

Finally

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, (K\alpha)_i) + \lambda \alpha^T K \alpha$$

$L(\theta)$ can be written **only** with K and α, instead of $\phi(x_i)$.
Natural question : But this does not make sense, as $\phi(x_i)$ and $\phi(x_j)$ are required to compute $K_{ij} = k(x_i, x_j)$?
Yes, **but**, in some situations, it is possible to compute $k(x_i, x_j)$ **without explicit knowledge** of ϕ. This is known as the **kernel trick**.

Alternate minimization problem

$$\begin{aligned} &\inf_{\theta \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} l(y_i, \langle \theta, \phi(x_i) \rangle + \lambda ||\theta||^2 \\ = &\inf_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} l(y_i, (K\alpha)_i) + \lambda \alpha^T K \alpha \end{aligned} \tag{22}$$

Alternate minimization problem

$$\begin{aligned} &\inf_{\theta \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} l(y_i, \langle \theta, \phi(x_i) \rangle + \lambda ||\theta||^2 \\ = &\inf_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} l(y_i, (K\alpha)_i) + \lambda \alpha^T K \alpha \end{aligned} \tag{23}$$

It might be easier to optimize in $\mathbb{R}^n$ than in $\mathcal{H}$, especially if $\mathcal{H}$ is infinite dimensional.

Evaluation fonction

Also, we can rewrite the evaluation function as :

$$f(x) = \theta^T \phi(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x)$$

Gram matrix

The kernel matrix is a matrix of inner products. It is often called a Gram matrix. If we note the design matrix

$$\phi = \begin{pmatrix} \phi(x_1)^T \\ \dots \\ \phi(x_i)^T \\ \dots \\ \phi(x_n)^T \end{pmatrix}$$

Then

$$K = \phi\phi^T \in \mathbb{R}^{n,n} \tag{24}$$

Gram matrix

K is symmetric positive semi-definite. $\forall \alpha \in \mathbb{R}^n$,

$$\begin{aligned} \alpha K \alpha &= \alpha \phi \phi^T \alpha \\ &= (\phi^T \alpha)^T (\phi^T \alpha) \\ &= ||\phi^T \alpha||^2 \end{aligned} \tag{25}$$

Then, if λ is an eigenvalue of K, with eigenvector α_λ,

$$\begin{aligned} \alpha_\lambda K \alpha_\lambda &= \alpha_\lambda \lambda \alpha_\lambda \\ &= \lambda ||\alpha_\lambda||^2 \end{aligned} \tag{26}$$

Approximations

- when n is large, it can become too costly to compute and store K ($\mathcal{O}(n^2)$) and to solve the optimization problem ($\mathcal{O}n^3$)).
- to avoid explicitely computing and storing K, **low-rank approximations** may be used (such as Nyström method)
- to solve the optimization problem, **low-rank decomposition** may be used.

See also

- Kernels on structured objects (graphds, texts, etc)
- Reproducing kernel Hilbert space (RKHS) (the space $\mathcal{H}$ that corresponds to k and ϕ)
- Adaptivity of kernel methods to the smoothness of the target function. If the optimization if performed in the right way, the convergence is faster for functions that are more than simply Lipshitz-continuous (in a future lecture).

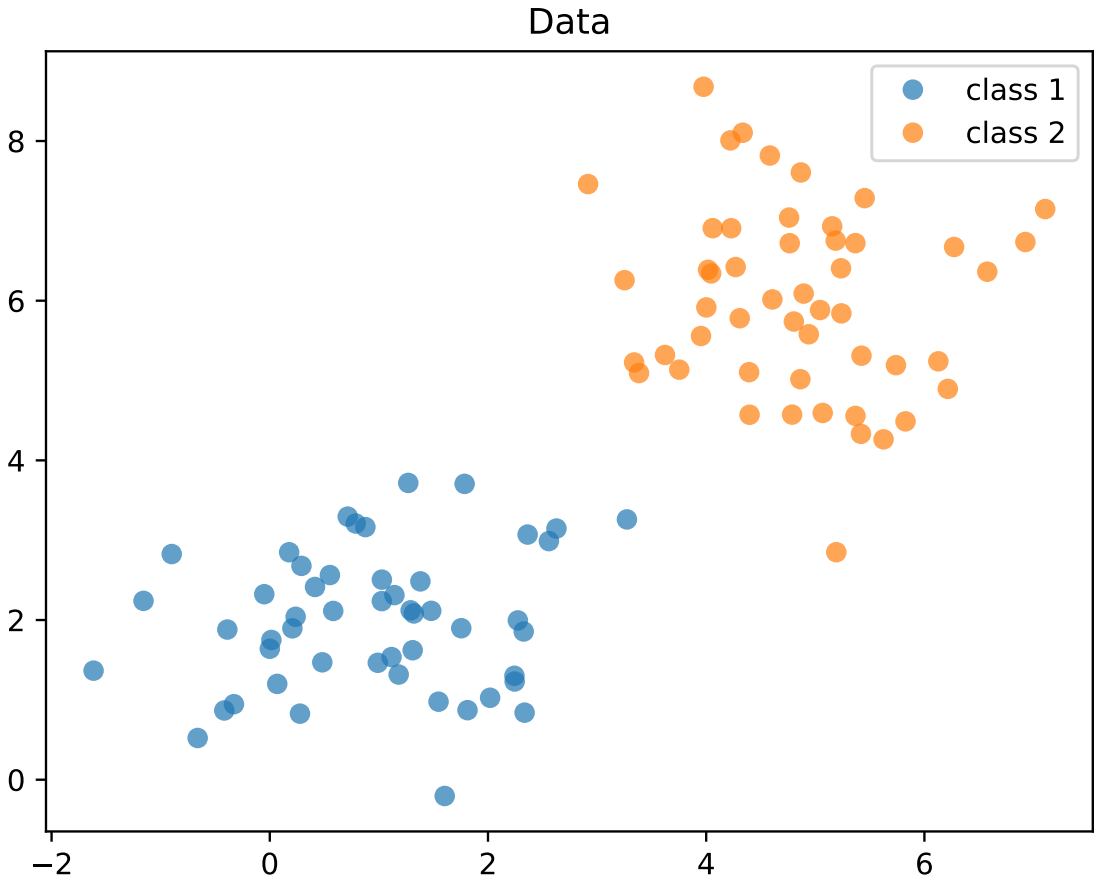


Figure – Linearly separable data

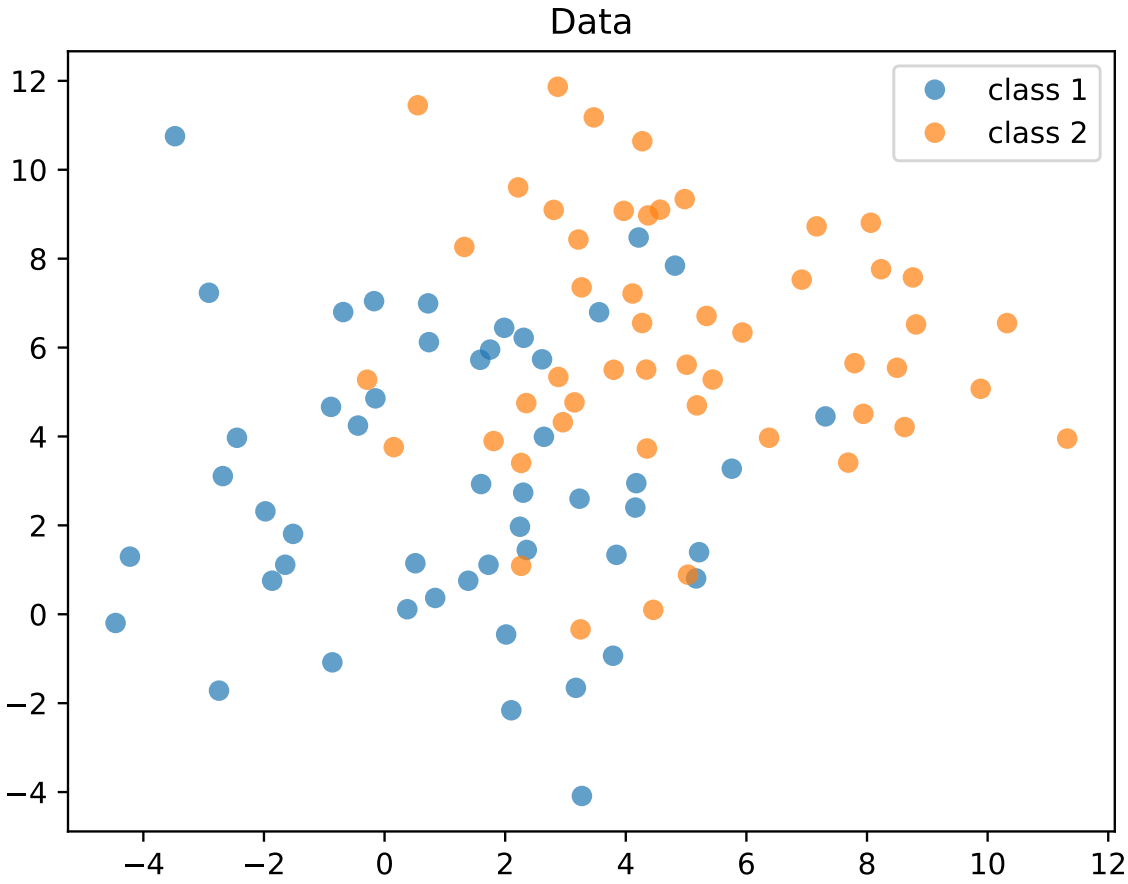


Figure – Non linearly-separable data

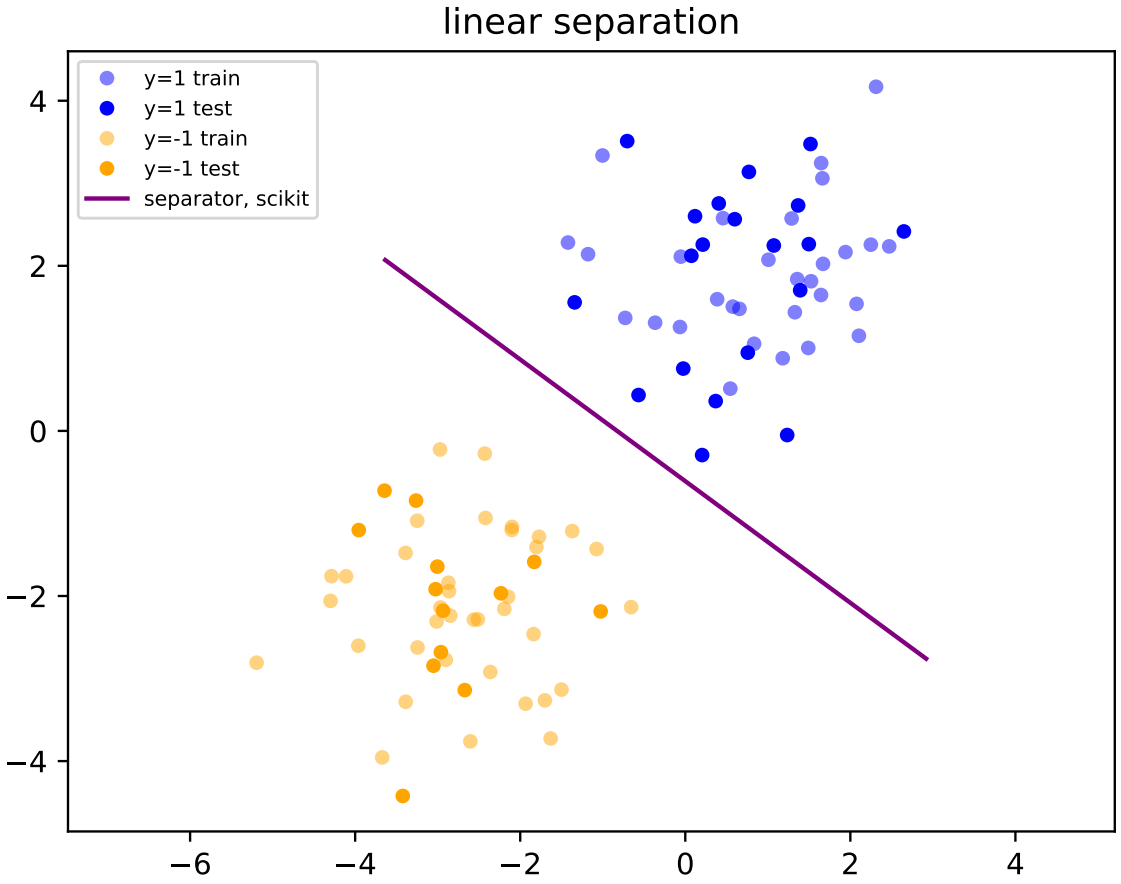


Figure – Linear separator

Linear separator

- $\mathcal{X} = \mathbb{R}^d$
- $\mathcal{Y} = \{-1, 1\}$

Equation of a linear separator

$$\langle w, x \rangle + b = 0 \tag{27}$$

- $w \in \mathbb{R}^d$
- $x \in \mathbb{R}^d$
- $b \in \mathbb{R}$

Affine subspace

$$H = \{x \in \mathbb{R}^d, \langle w, x \rangle + b = 0\} \tag{28}$$

is an affine subspace.

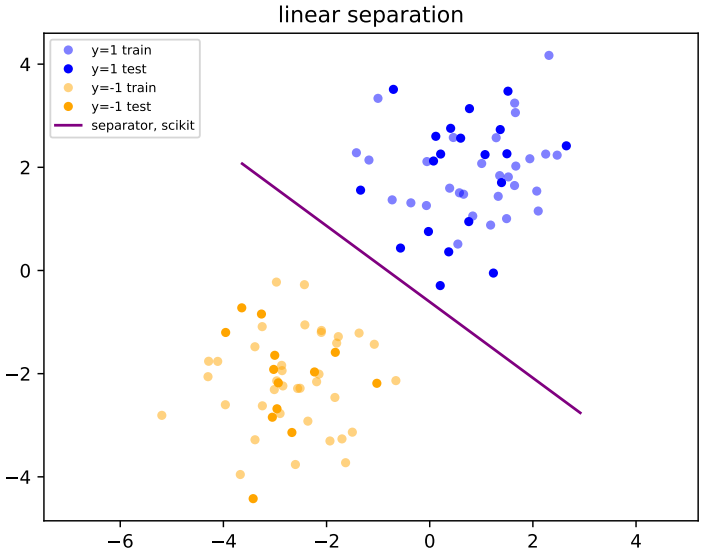
Any vector $x \in \mathbb{R}^d$ can uniquely be decomposed as

$$x = \lambda_w^x \frac{w}{||w||} + x_{w^\perp} \tag{29}$$

with $x_{x^\perp} \in \text{vect}(w)^\perp$. $x \in H$ if and only if

$$\begin{aligned} \langle w, x \rangle + b = 0 \\ \Leftrightarrow \langle w, \lambda_w^x \frac{w}{||w||} + x_{w^\perp} \rangle + b = 0 \\ \Leftrightarrow \langle w, \lambda_w^x \frac{w}{||w||} \rangle + b = 0 \\ \Leftrightarrow \lambda_w^x ||w|| + b = 0 \\ \Leftrightarrow \lambda_w^x = \frac{-b}{||w||} \end{aligned} \tag{30}$$

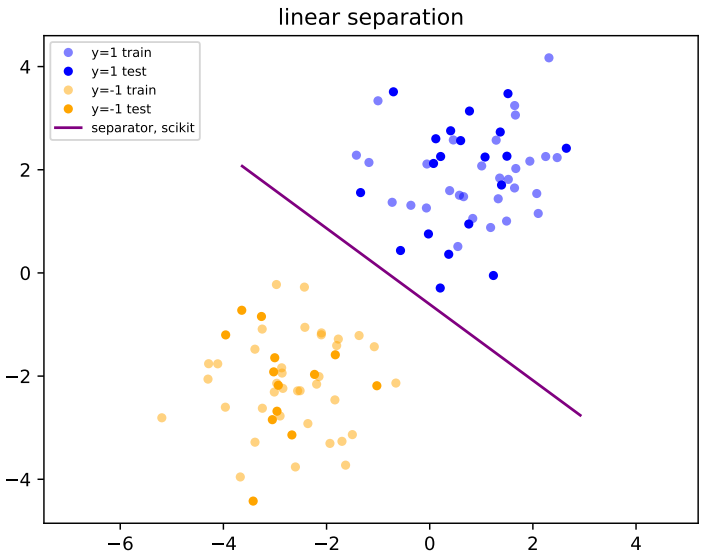
We first consider a linearly separable situation.



We note $h_{w,b}(x) = \langle w, x \rangle + b$. We look for separators that satisfies :

- $\forall x_i$ such that $y_i = 1$, $h_{w,b}(x) \geq 0$
- $\forall x_i$ such that $y_i = -1$, $h_{w,b}(x) \leq 0$

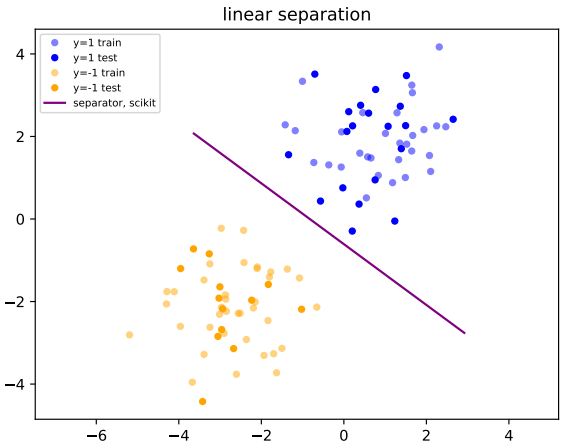
We first consider a linearly separable situation.



We note $h_{w,b}(x) = \langle w, x \rangle + b$. We look for separators that satisfies :

- $\forall x_i$ such that $y_i = 1$, $h_{w,b}(x) \geq 0$
- $\forall x_i$ such that $y_i = -1$, $h_{w,b}(x) \leq 0$

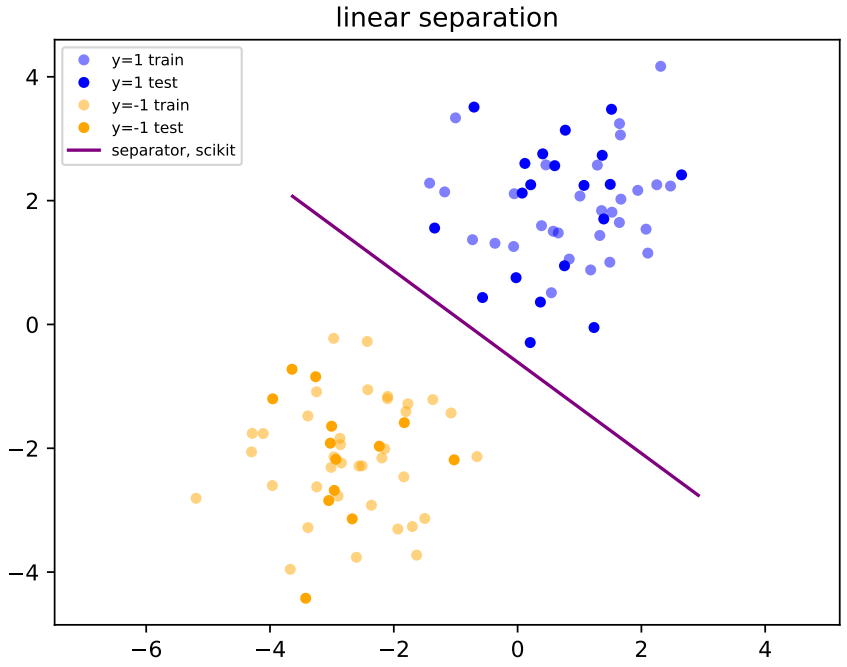
However, there exists an infinite number of such parameters. How could we choose the best one ?



- $\forall x_i$ such that $y_i = 1$, $h_{w,b}(x) \geq 0$
- $\forall x_i$ such that $y_i = -1$, $h_{w,b}(x) \leq 0$

The **margin** is the distance from H to the dataset. We look for the separator with the largest margin, leading to **Support vector classification (SVC)**.

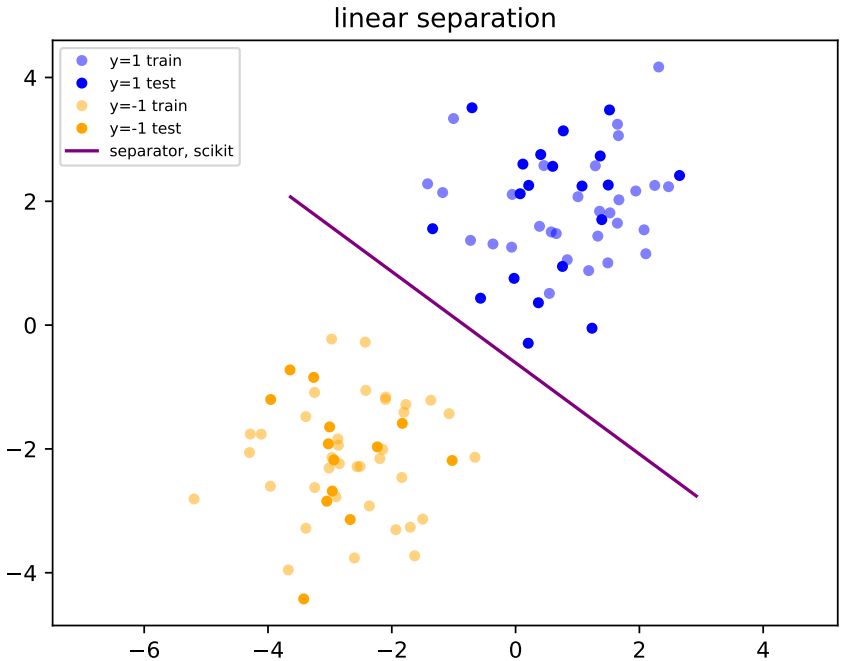
Margin



Let x be a point such that $h_{w,b}(x) = \langle w, x \rangle + b = c$, with $c \in \mathbb{R}$.

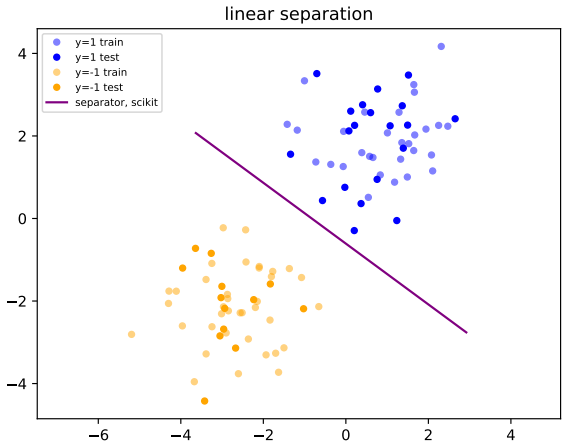
Exercice 3 : Compute the distance from x to H.

Margin



Let x be a point such that $h_{w,b}(x) = \langle w, x \rangle + b = c$, with $c \in \mathbb{R}$.
The distance is $\frac{c}{||w||}$.

Support vectors

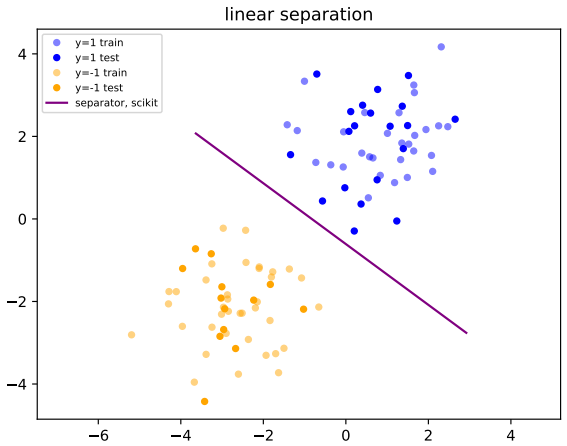


The **support vectors** are the vectors such that $|h_{w,b}(x)|$ is minimal among the dataset.

- the margin M is the distance from H to these vectors.
- if H is the optimal separator, there has to be a vector x_- and x_+ on each side, such that

$$M = d(x_-, H) = d(x_+, H) \tag{31}$$

Support vectors



Exercice 4 : Show that if H is optimal, then

$$M = d(x_-, H) = d(x_+, H) \tag{32}$$

Rescaling

Important remark : multiplying w and b by a constant $\lambda \neq 0$ does not change H, as :

$$\begin{aligned} &\langle \lambda w, x \rangle + \lambda b = 0 \\ &\Leftrightarrow \lambda(\langle w, x \rangle + b) = 0 \\ &\Leftrightarrow \langle w, x \rangle + b = 0 \end{aligned} \tag{33}$$

Rescaling

Important remark : multiplying w and b by a constant $\lambda \neq 0$ does not change H.
If the support vector x is such that $h_{w,b}(x) = c$, we have seen that the margin is

$$\frac{|c|}{||w||} \tag{34}$$

When looking for the optimal H, we can impose, without loss of generality, that $|c| = 1$.
This means that we look for w with minimal norm, such that H separates the data (since the margin is $\frac{1}{||w||}$.

Optimization problem

We can now formulate the optimization problem.

$$\arg\min_{w,b} \frac{1}{2}\langle w, w\rangle \tag{35}$$

subject to :

$$\forall i \in [1, n], y_i(\langle w, x_i\rangle + b) \geq 1 \tag{36}$$

Slack variables

When the dataset is not linearly separable, the approach is to authorize some of the samples to have a margin smaller that 1. This means relawing the constraint, from

$$y_i(\langle w, x_i \rangle + b) \geq 1 \tag{37}$$

to

$$y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i \tag{38}$$

The ξ are called the *slack variables*, they are ≥ 0. The smaller the slack variabes, the better.

Optimization problem

In the general case, the optimization problem is :

$$\arg\min_{w,b,\xi} \frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{n}\xi_i \tag{39}$$

subject to :

$$\forall i \in [1, n], y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i \tag{40}$$

and

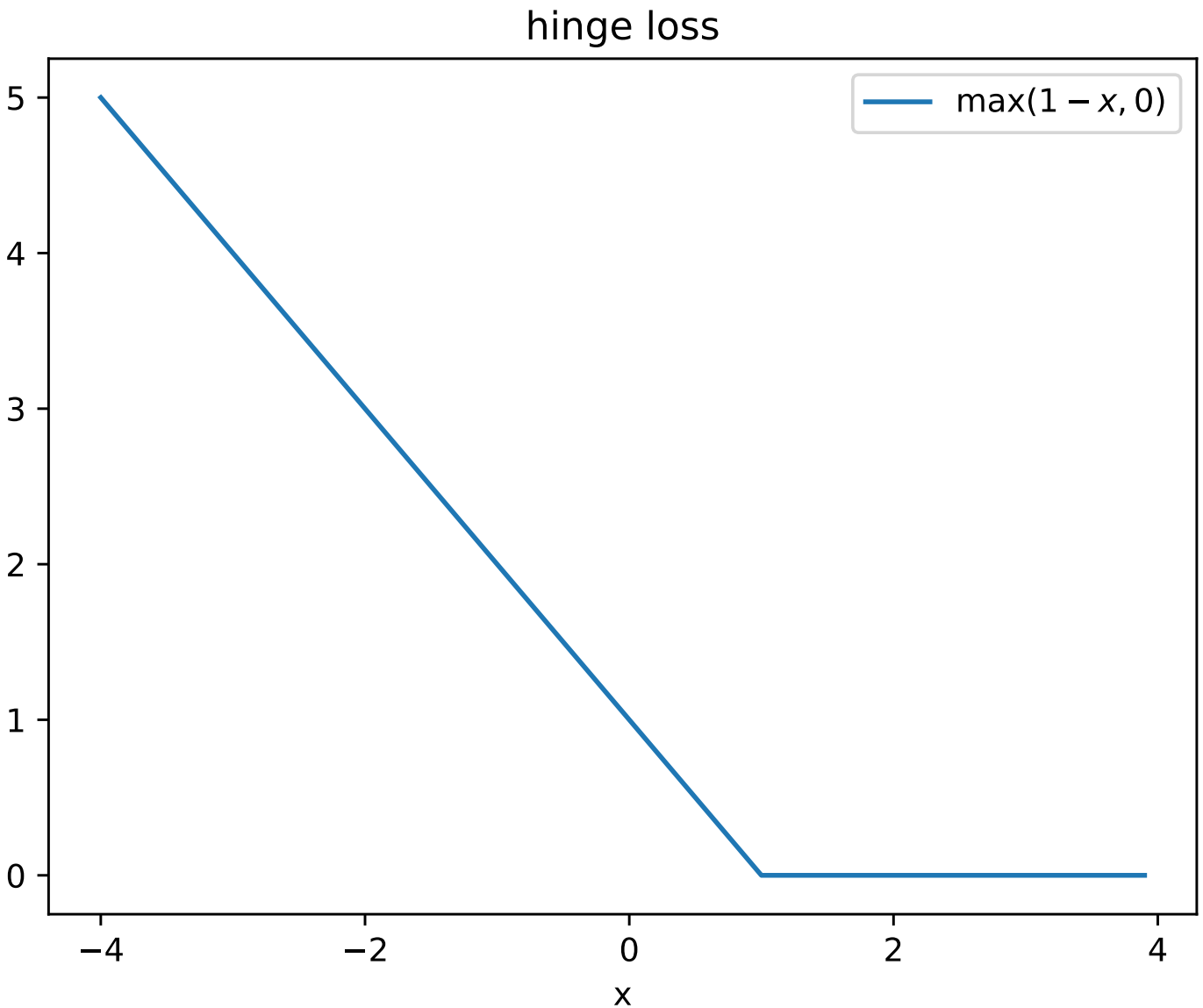
$$\forall i \in [1, n], \xi_i \geq 0 \tag{41}$$

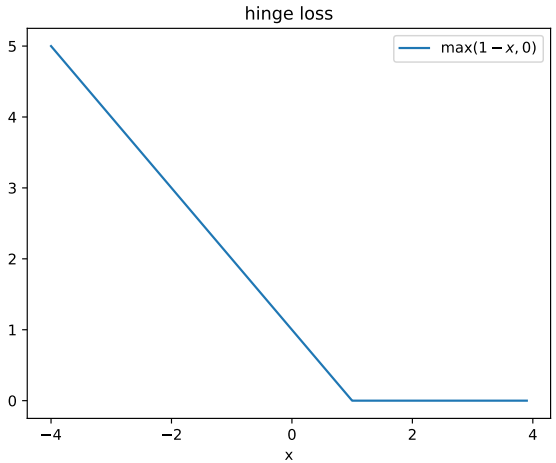
Resolution

In order to find the optimal parameters, a method is to consider the **Lagrangian** of the problem, and to solve the **dual problem.**
One aspect that appears when performing this transformation is that only the inner products $\langle x_i, x_j \rangle$ are involved in this form of the problem.
Hence, in this context it is also possible to use **feature maps** $\phi(x)$, **kernels** and the kernel trick.
This is often done for SVMs.

Margin vs ERM

The margin maximisation seems to differ from empirical risk minimization (ERM), which we have studied earlier.
However, with a specific loss function, we an show that margin maximisation is in fact an ERM.


hinge loss
max(1 − x, 0)
5
4
3
2
1
0
−4
−2
0
2
4
x



- estimation : $h(x) = \langle w, x \rangle + b$
- label : $y \in \{-1, 1\}$

Hinge loss :

$$L_{\text{hinge}}(h(x), y) = \max(0, 1 - yh(x)) \tag{42}$$

Hinge loss

- estimation : $h(x) = \langle w, x \rangle + b$
- label : $y \in \{-1, 1\}$

Hinge loss :

$$L_{\text{hinge}}(h(x), y) = \max(0, 1 - yh(x)) \tag{43}$$

The hinge loss can be seen as an approximation of the binary loss.

Problem reformulation

We recall the constraints on ξ

$$y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i \tag{44}$$

and

$$\xi_i \geq 0 \tag{45}$$

Equivalently,

$$\xi_i \geq max\big(0, 1 - y_i(\langle w, x_i \rangle + b)\big) \tag{46}$$

Problem reformulation

The slack variables should be minimal. Hence, we can write that for the optimal solution, the inequality is in fact an equality ;

$$\xi_i = max\big(0, 1 - y_i(\langle w, x_i \rangle + b)\big) \tag{47}$$

Problem reformulation

Finally, we can rewrite the problem as

$$\arg\min_{w,b} \frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{n} max\big(0, 1 - y_i(\langle w, x_i\rangle + b)\big) \tag{48}$$

or equivalently

$$\arg\min_{w,b} \frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{n} L_{\text{hinge}}(h(x_i)), y_i) \tag{49}$$

Which is an ERM problem with a $L2$ regularization.

Duality

An consequence of the lagrangian resolution of the optimization problem is that there exists α such that

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{50}$$

Hence

$$h_{w,b}(x) = \sum_{i=1}^{n} \alpha_i y_i \langle x_i, x \rangle + b \tag{51}$$

Only the inner products $\langle x_i, x_j \rangle$ are involved in the estimator. Similarly, the dual problem formulation also only uses the inner products. This motivates the use of kernels.

Separation function

$$h(x) = \sum_{i=1}^{n} \alpha_i y_i k(x_i, x) + b \tag{52}$$

Gaussian kernel (RBF)

$$k(x, x') = \exp\left(-\gamma ||x - x'||^2 \right) \tag{53}$$

With $k(x, x') = \langle \phi(x), \phi(x') \rangle$, $\phi(x) \in \mathcal{H}$

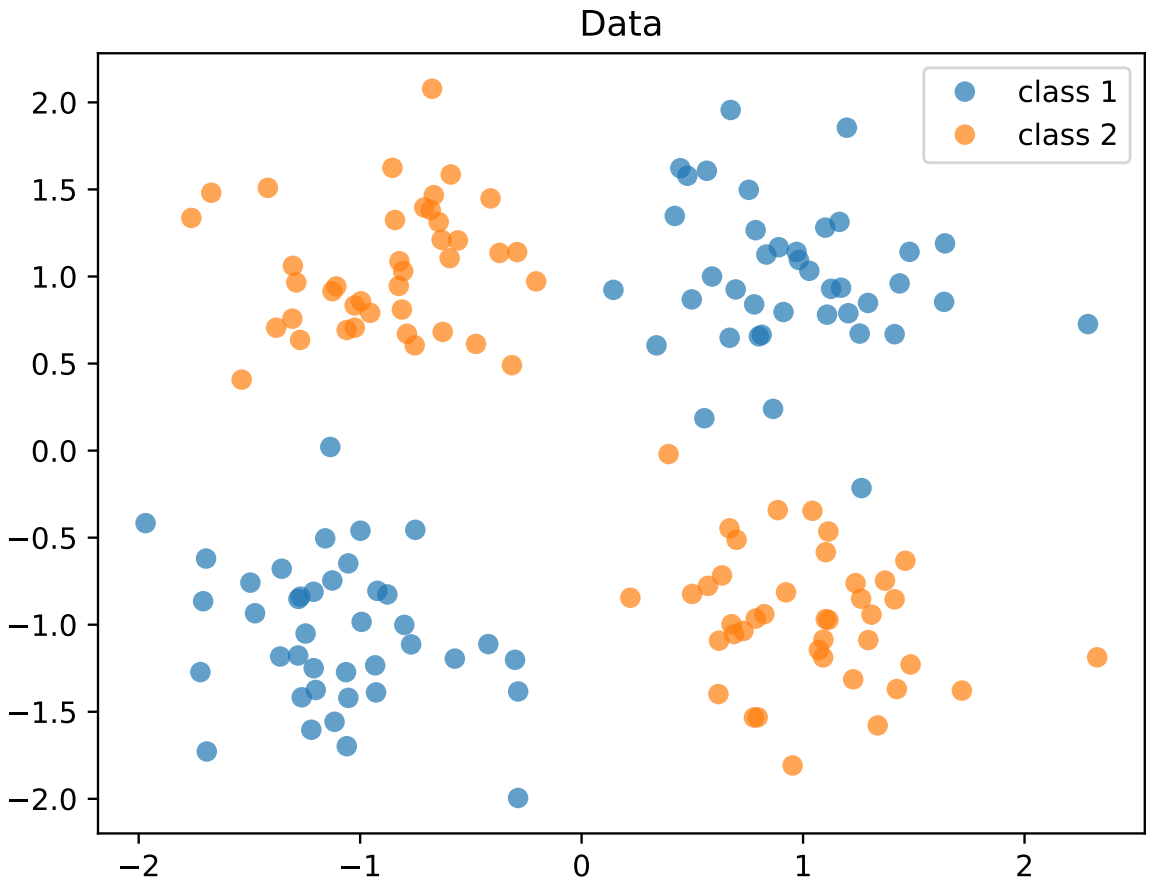
- $\mathcal{H}$ is of infinite dimension.
- γ is a parameter that should be carefully tuned.

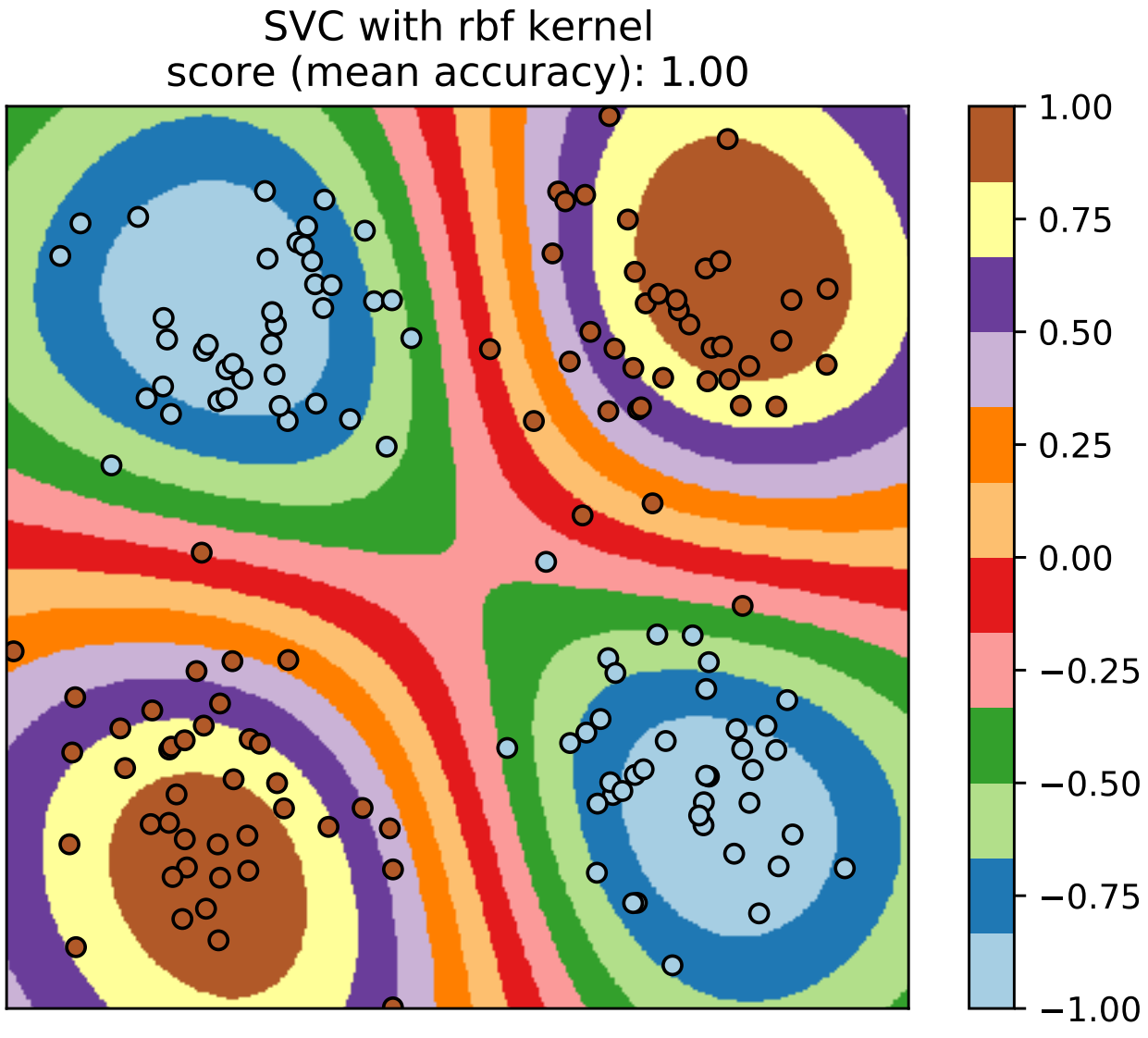
Linear kernel

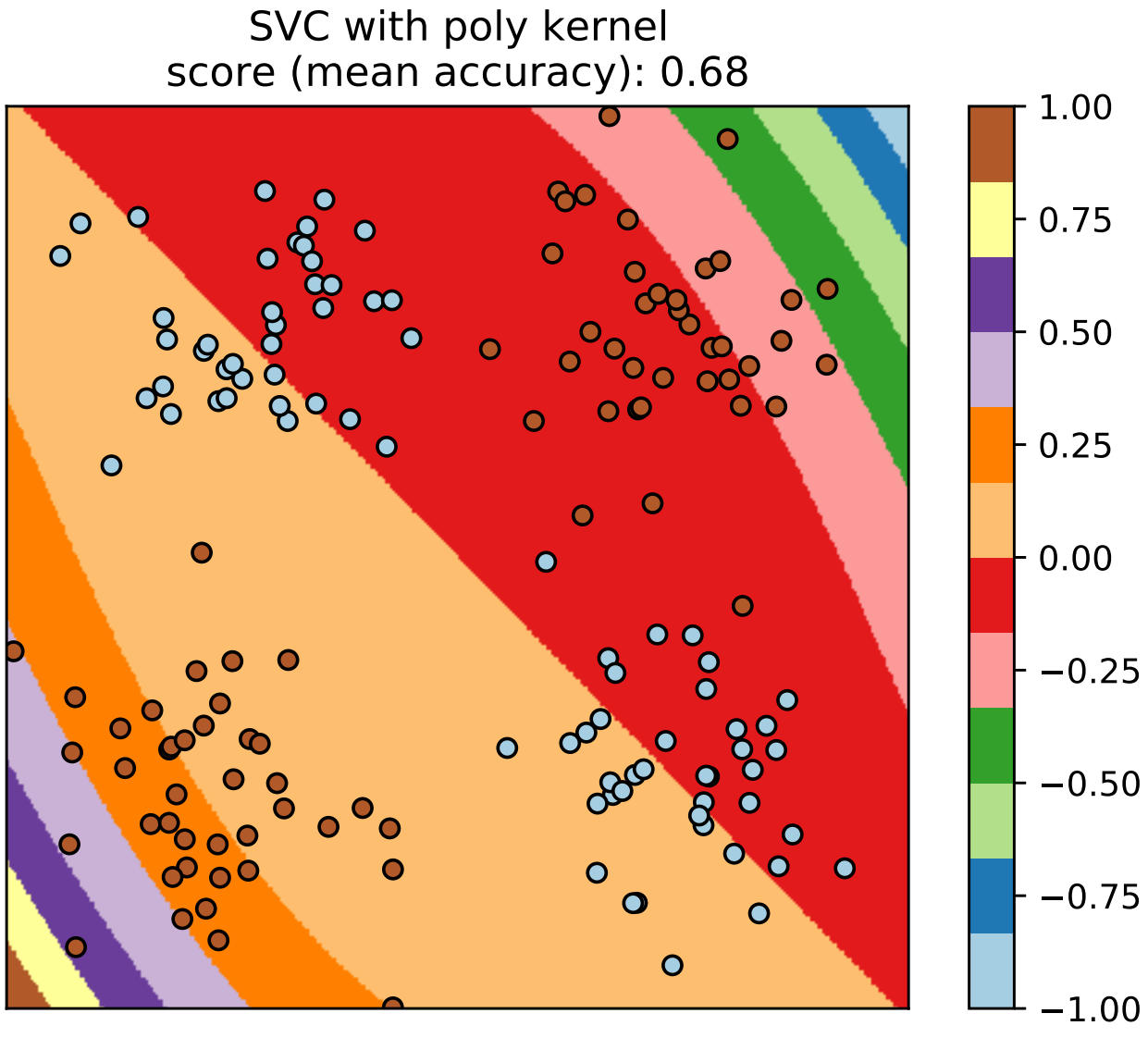
$$k(x, x') = \langle x, x' \rangle \tag{54}$$

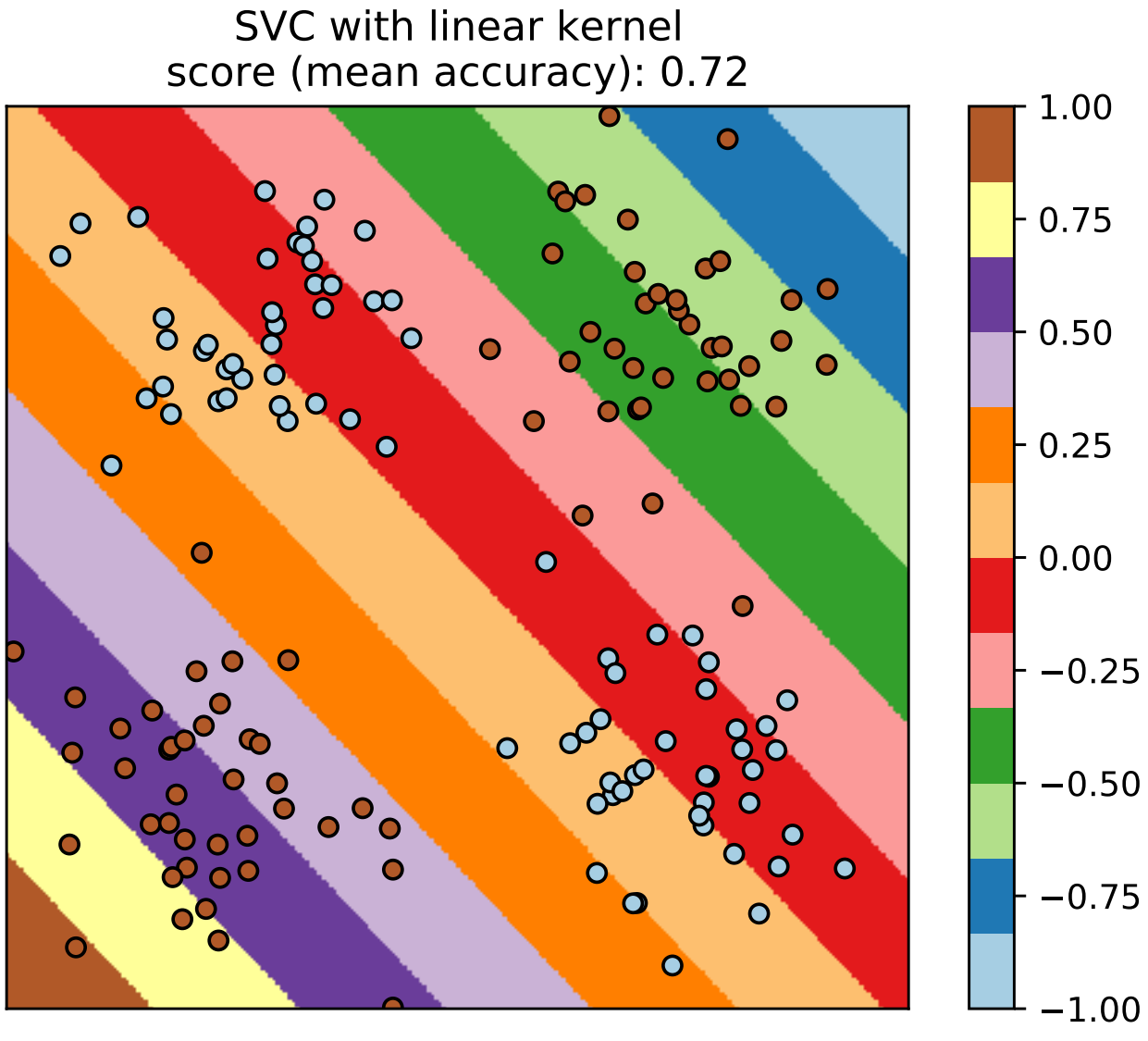
Application

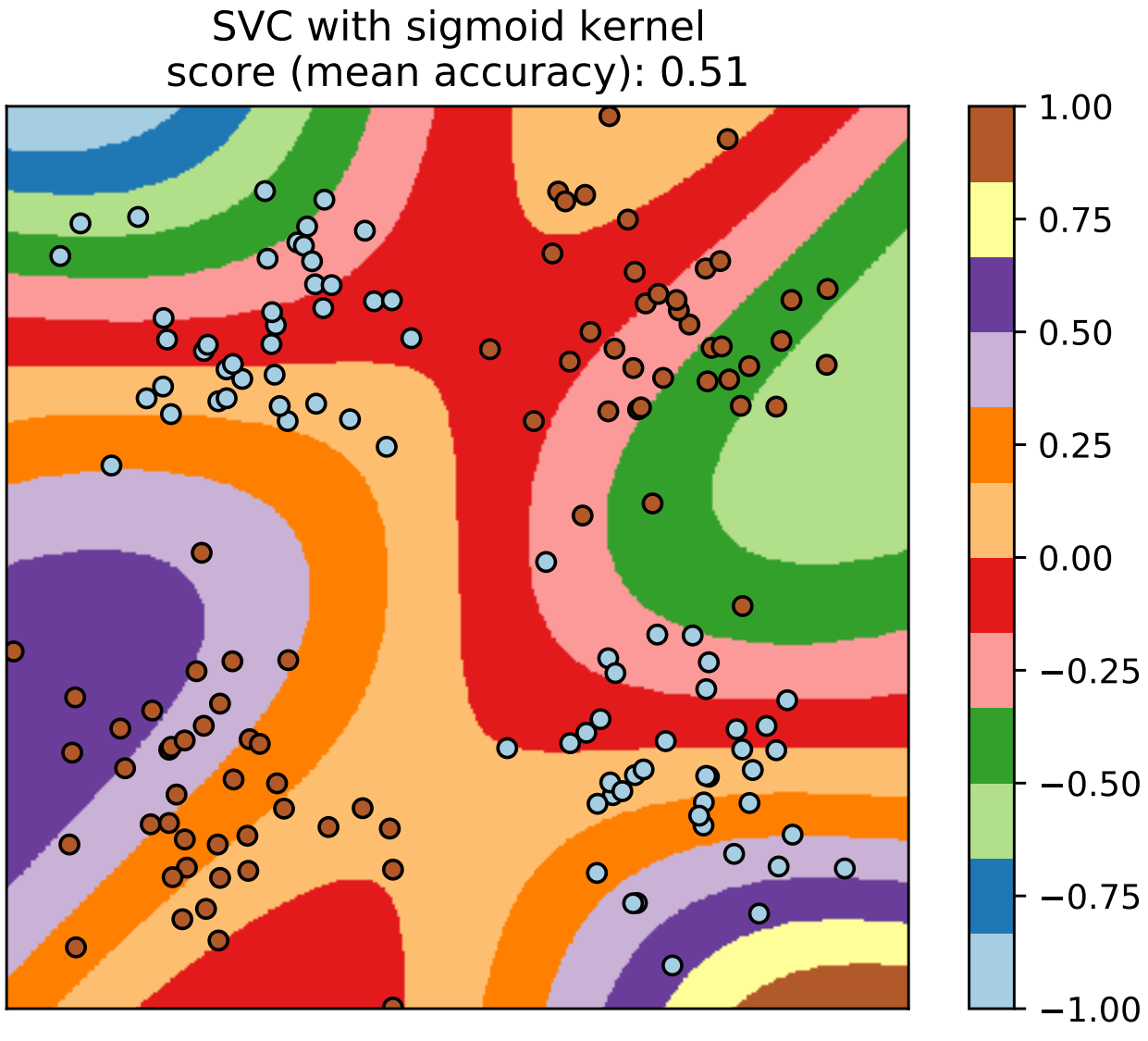
We would like to classify these data with a SVC.




SVC with rbf kernel
score (mean accuracy): 1.00
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


SVC with poly kernel
score (mean accuracy): 0.68
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


SVC with linear kernel
score (mean accuracy): 0.72
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


SVC with sigmoid kernel
score (mean accuracy): 0.51
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

See also

- SVMs for regression
- SVMs for unsupervised learning
- Sequential minimal optimization (SMO)