Lecture 5
Security and Ethical Aspects of Data

Amaya Nogales Gómez
amaya.nogales-gomez@univ-cotedazur.fr

MSc Data Science & Artificial Intelligence
Université Côte d'Azur

November 22, 2021

Tentative Content

1. Introduction
   - Preliminaries and Machine Learning basics
   - Motivation for ethics in Artificial Intelligence
2. Sources of unfairness
   - Bias in data
   - Algorithmic unfairness: Examples
3. Fairness criteria
   - Types of discrimination
   - Definitions of fairness
   - Fairness analysis of datasets
4. Algorithms and methods for fair ML
   - Pre-processing methods
   - In-processing methods
   - Post-processing methods
   - Legal and policy perspectives

Outline

- Basic concepts
- Hard-margin approach
- Soft-margin approach
- Loss functions
- Kernel methods
- Parameter selection and practical issues
- Multi-class classification
- Categorical data

Supervised Classification: Support Vector Machines

- Ω: the population.
- Population is partitioned into two classes, $\{-1, +1\}$.

Supervised Classification: Support Vector Machines

- Ω: the population.
- Population is partitioned into two classes, $\{-1, +1\}$.
- For each object in Ω, we have
  - $x = (x^1, \ldots, x^d) \in X \subset \mathbb{R}^d$: predictor variables.
  - $y \in \{-1, +1\}$: class membership.

Supervised Classification: Support Vector Machines

- Ω: the population.
- Population is partitioned into two classes, $\{-1, +1\}$.
- For each object in Ω, we have
  - $x = (x^1, \ldots, x^d) \in X \subset \mathbb{R}^d$: predictor variables.
  - $y \in \{-1, +1\}$: class membership.

- The goal is to find a hyperplane $\omega^\top x + b = 0$ that aims at separating, if possible, the two classes.

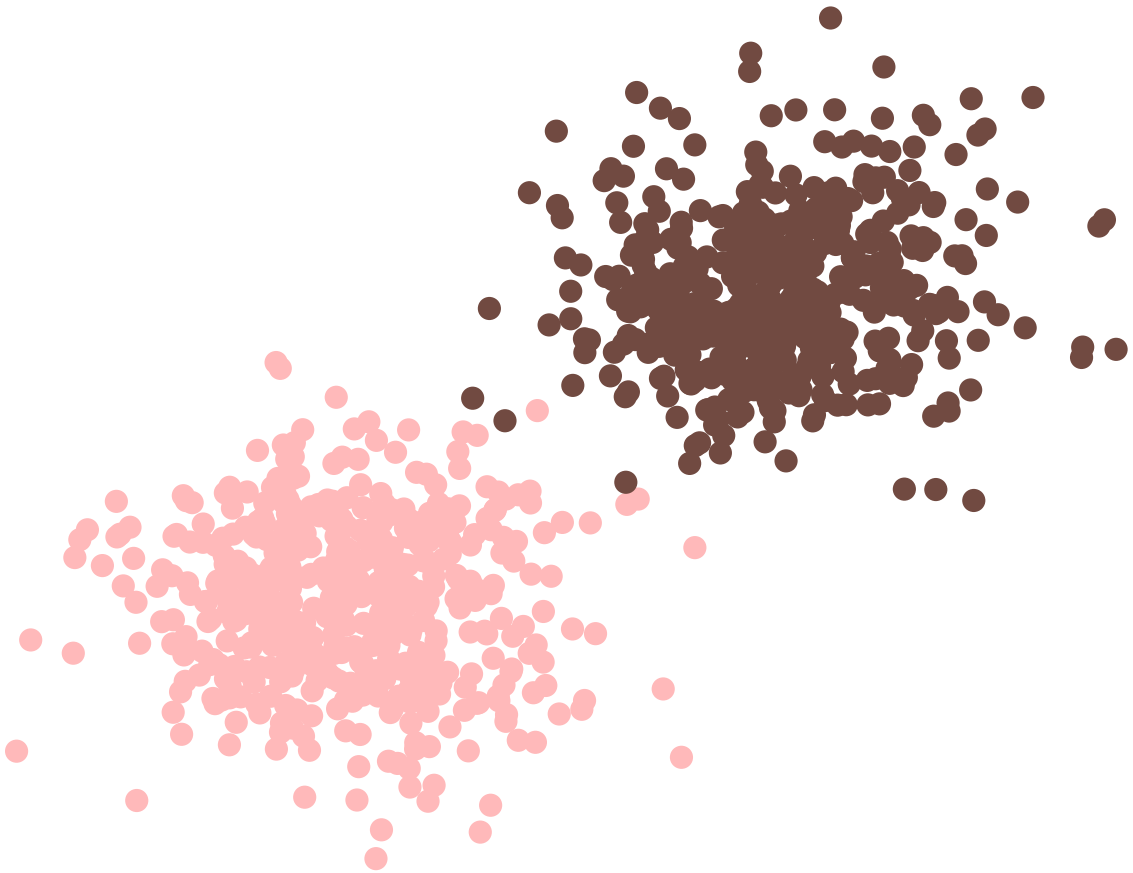
Supervised Classification: Support Vector Machines

- Ω: the population.
- Population is partitioned into two classes, $\{-1, +1\}$.
- For each object in Ω, we have
  - $x = (x^1, \ldots, x^d) \in X \subset \mathbb{R}^d$: predictor variables.
  - $y \in \{-1, +1\}$: class membership.

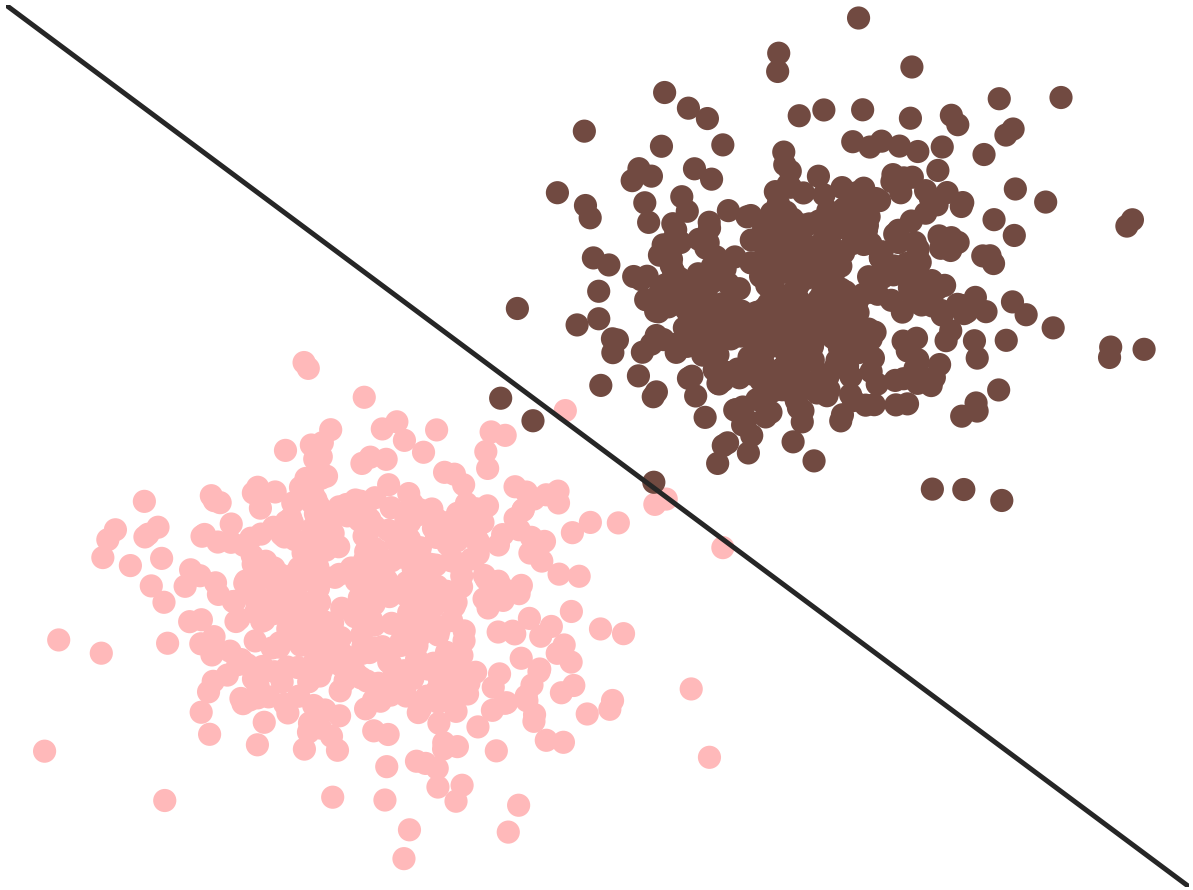
- The goal is to find a hyperplane $\omega^\top x + b = 0$ that aims at separating, if possible, the two classes.
- Future objects will be classified as

$$\begin{aligned} y &= +1 \quad \text{if} \quad \omega^\top x + b > 0 \\ y &= -1 \quad \text{if} \quad \omega^\top x + b < 0 \end{aligned} \tag{1}$$

Supervised Classification



Supervised Classification



Support Vector Machines (SVM)

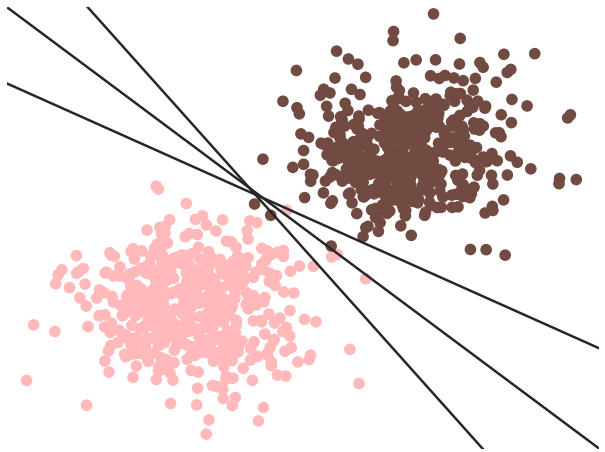
- State-of-the-art in supervised classification.
- Very good classification accuracy.
- Computationally cheap: Quadratic Programming formulation.
- SVM: In many cases competitive with existing classification methods.
- Relatively easy to use.
- Kernel techniques: many extensions.
- Regression, density estimation, kernel PCA, etc.

Separating hyperplanes

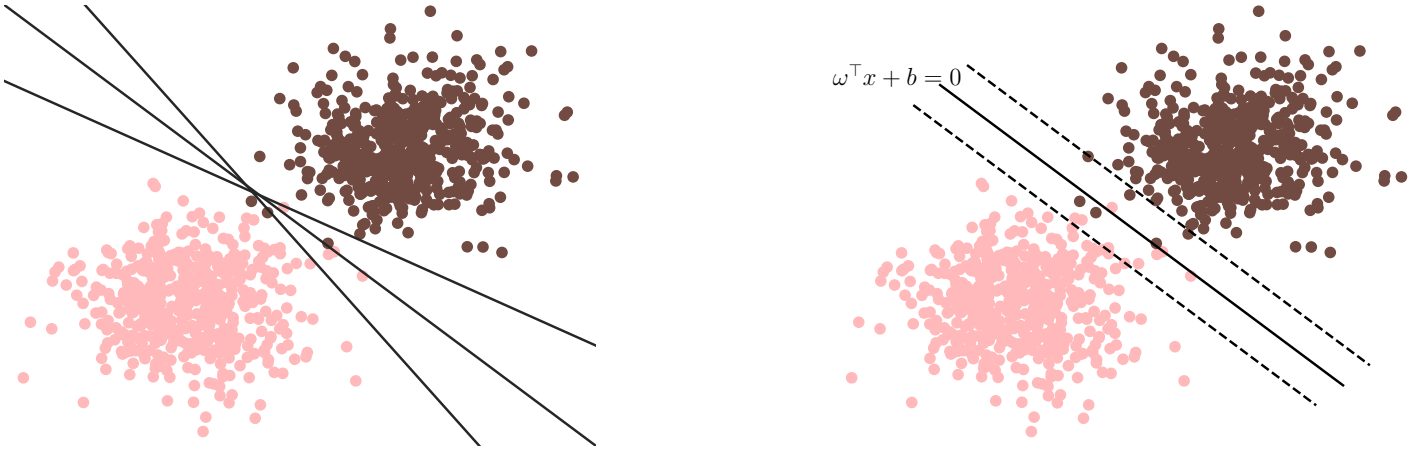
- Infinite possible separating hyperplanes.
- Each one with different properties (metrics).
- Expressed as:

$$\omega^\top x + b = 0$$

- In geometry, a hyperplane is a subspace whose dimension is one less than that of its ambient space.



Maximum margin classification



The SVM aims to find the boundary that maximizes the margin between the classes.

Linear algebra of a hyperplane

- H: hyperplane defined by $\omega^\top x + b = 0$
- Key properties:
  1. For any $x_1, x_2 \in H$
     - $\omega^\top (x_1 - x_2) = 0$ and
     - $\bar{\omega} = \omega / \|\omega\|$ is the vector normal to H.
  2. For any $x_0 \in H$, $\omega^\top x_0 = -b$

Distance of any point to the hyperplane

- The signed distance of any point x to the H is the projection of vector v ($x - x_0$, with x_0 being intersection point of H and the normal vector) into the normal vector.
- We obtain this projection via the dot product:

$$\begin{aligned} \bar{\omega} \cdot v &= \frac{\omega^\top}{\|\omega\|} \cdot (x - x_0) = \\ &= \frac{1}{\|\omega\|}(\omega^\top x - \omega^\top x_0) = \\ &= \frac{1}{\|\omega\|}(\omega^\top x + b) \end{aligned}$$

- Distance of object i to the hyperplane H:

$$d(x_i, H) = \frac{y_i}{\|\omega\|}(\omega^\top x_i + b)$$

Optimization problem

Let us recall that the margin width is the distance from the decision boundary to the closest point.

We want to find the margin as large as possible (maximization)

SVM seeks to maximize, as a function of ω, b, the quantity:

$$\arg\max_{\omega,b} \left\{ \min_i d(x_i, H) = \frac{1}{\|\omega\|} \min_i y_i(\omega^\top x_i + b) \right\}$$

Hard-Margin approach

- Training sample assumed to be linearly separable, i.e., the convex hull of the two groups are not empty and they do not overlap.
- All objects in the training sample must be correctly classified!
- The separating hyperplane is the one maximizing the smallest distance to misclassification.

Hard-margin SVM: maximal margin

- Distance between $\omega^\top x + b = +1$ and $\omega^\top x + b = -1$:

$$2/\|\omega\| = 2/\sqrt{\omega^\top \omega}$$

- A quadratic programming problem with linear constraints.

$$\min_{\omega, b} \frac{1}{2} \sum_{j=1}^{d} \omega_j^2$$

s.t.

$$y_i(\omega^\top x_i + b) \geq 1 \qquad \forall i = 1, \ldots, n$$
$$\omega \in \mathbb{R}^d$$
$$b \in \mathbb{R}.$$

But...

$$\min_{\omega, b} \frac{1}{2} \sum_{j=1}^{d} \omega_j^2$$

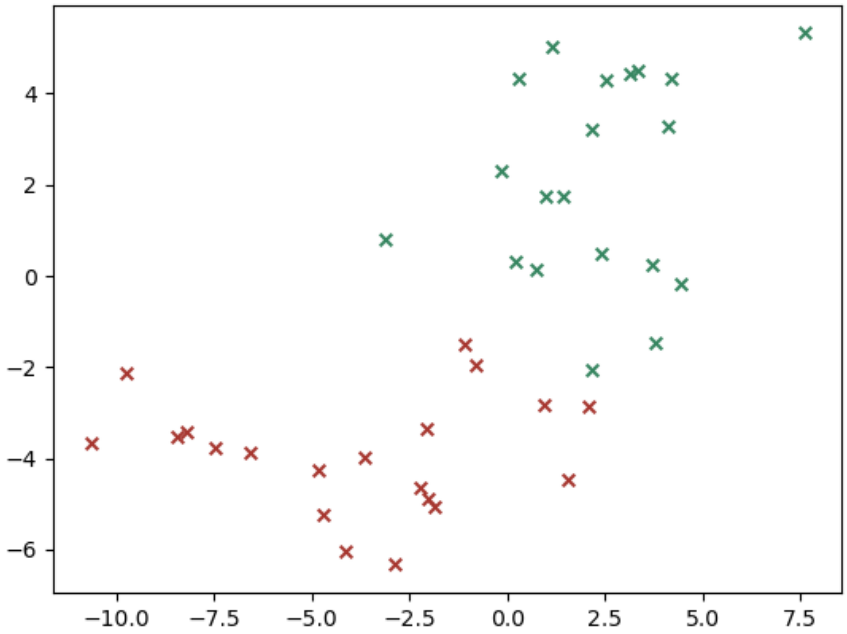
s.t.

$$y_i(\omega^\top x_i + b) \geq 1 \qquad \forall i = 1, \ldots, n$$

$$\omega \in \mathbb{R}^d$$

$$b \in \mathbb{R}.$$

Linearly separable data



But...

$$\min_{\omega, b} \frac{1}{2} \sum_{j=1}^{d} \omega_j^2$$

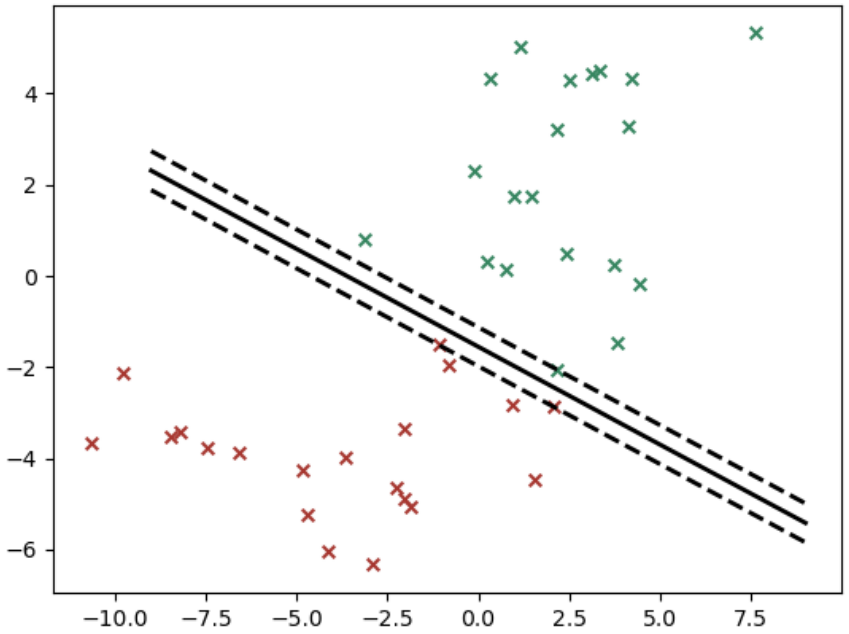
s.t.

$$y_i(\omega^\top x_i + b) \geq 1 \qquad \forall i = 1, \ldots, n$$

$$\omega \in \mathbb{R}^d$$

$$b \in \mathbb{R}.$$

Linearly separable data



But...

$$\min_{\omega, b} \frac{1}{2} \sum_{j=1}^{d} \omega_j^2$$

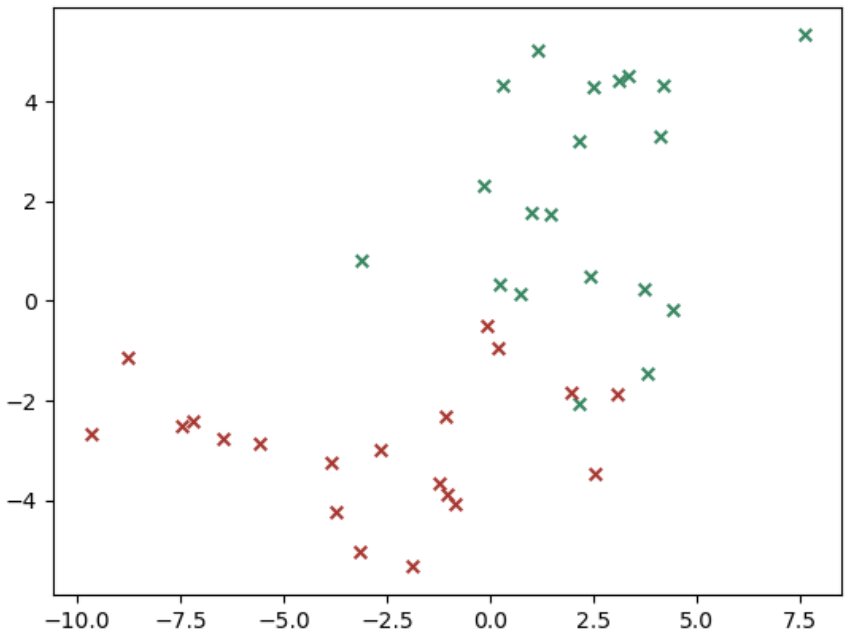
s.t.

$$y_i(\omega^\top x_i + b) \geq 1 \qquad \forall i = 1, \ldots, n$$

$$\omega \in \mathbb{R}^d$$

$$b \in \mathbb{R}.$$

Non-linearly separable data



But...

Non-linearly separable data

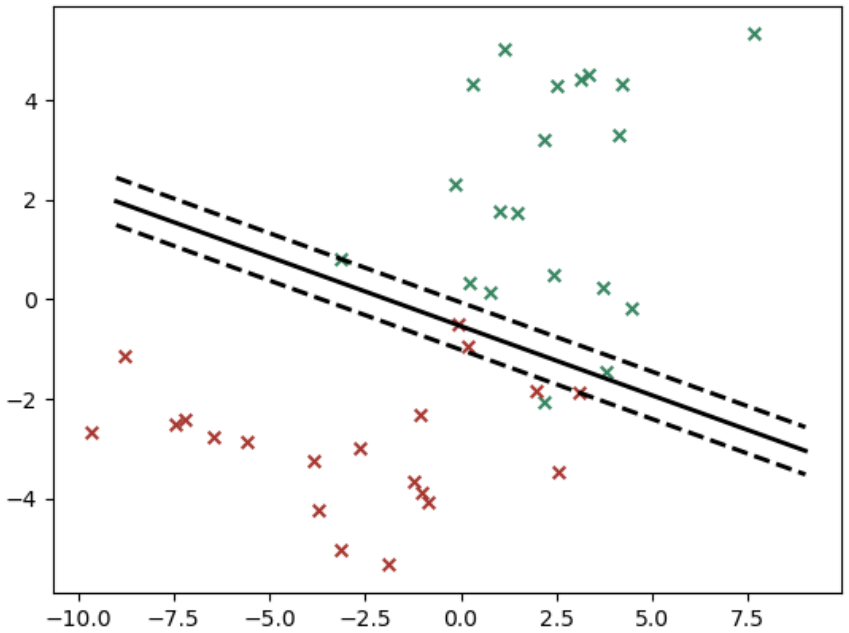
$$\min_{\omega, b} \frac{1}{2} \sum_{j=1}^{d} \omega_j^2$$

s.t.

$$y_i(\omega^\top x_i + b) \geq 1 \qquad \forall i = 1, \ldots, n$$

$$\omega \in \mathbb{R}^d$$

$$b \in \mathbb{R}.$$



INFEASIBLE!!

Hard margin SVM: limitations

- Real data, most likely, will not meet the of linear separable assumption
- Hard margin loss is too limiting when there is class overlapping
- Hard margin SVM will not be able to deal with it
- Possible solutions:
  - Keep hard margin SVM but transform the data: mapping into higher dimensional (maybe infinite) feature space

$$\phi(x) = (\phi_1(x), \phi_2(x), \cdots)$$

  - Relax the constraints (allow training errors)
  - Combination of both

A solution for non-linearly separable data

- When data are not linearly separable the hard-margin SVM problem is infeasible.
- In the **soft-margin approach**, constraints

$$y_i(\omega^\top x_i + b) \geq 1 \quad \forall i = 1, \ldots, n$$

  are perturbed.
- How? By introducing auxiliary variables ξ_i, making the new problem always feasible.

Building the soft-margin SVM

$$\min_{\omega, b, \xi} \frac{1}{2} \sum_{j=1}^{d} \omega_j^2 + \frac{C}{n} \sum_{i=1}^{n} g_i(\xi_i)$$

s.t.

$$\begin{aligned} y_i(\omega^\top x_i + b) &\geq 1 - \xi_i \qquad \forall i = 1, \ldots, n \\ \omega &\in \mathbb{R}^d \\ b &\in \mathbb{R} \\ \xi_i &\geq 0 \qquad \forall i = 1, \ldots, n, \end{aligned}$$

- $\xi = (\xi_i) \in \mathbb{R}^n$ is the vector of deviation variables.
- g_i is the loss function (convex and increasing).
- Most popular choices: *hinge* loss, $g_i(t) = C_i t$ or *squared hinge* loss, $g_i(t) = C_i t^2$.
- C is a tuning parameter.

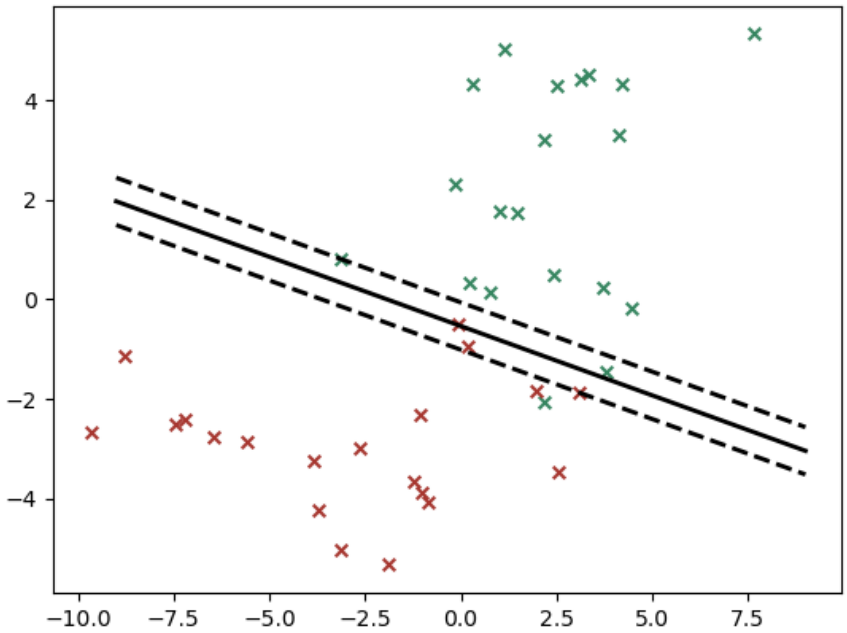
The SVM formulation

$$\min_{\omega,\, b,\, \xi} \; \frac{1}{2}\sum_{j=1}^{d} \omega_j^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i$$

s.t.

$$\begin{aligned} y_i(\omega^\top x_i + b) &\geq 1 - \xi_i \qquad \forall i = 1, \ldots, n \\ \omega &\in \mathbb{R}^d \\ b &\in \mathbb{R} \\ \xi_i &\geq 0 \qquad \forall i = 1, \ldots, n. \end{aligned}$$

Non-linearly separable data

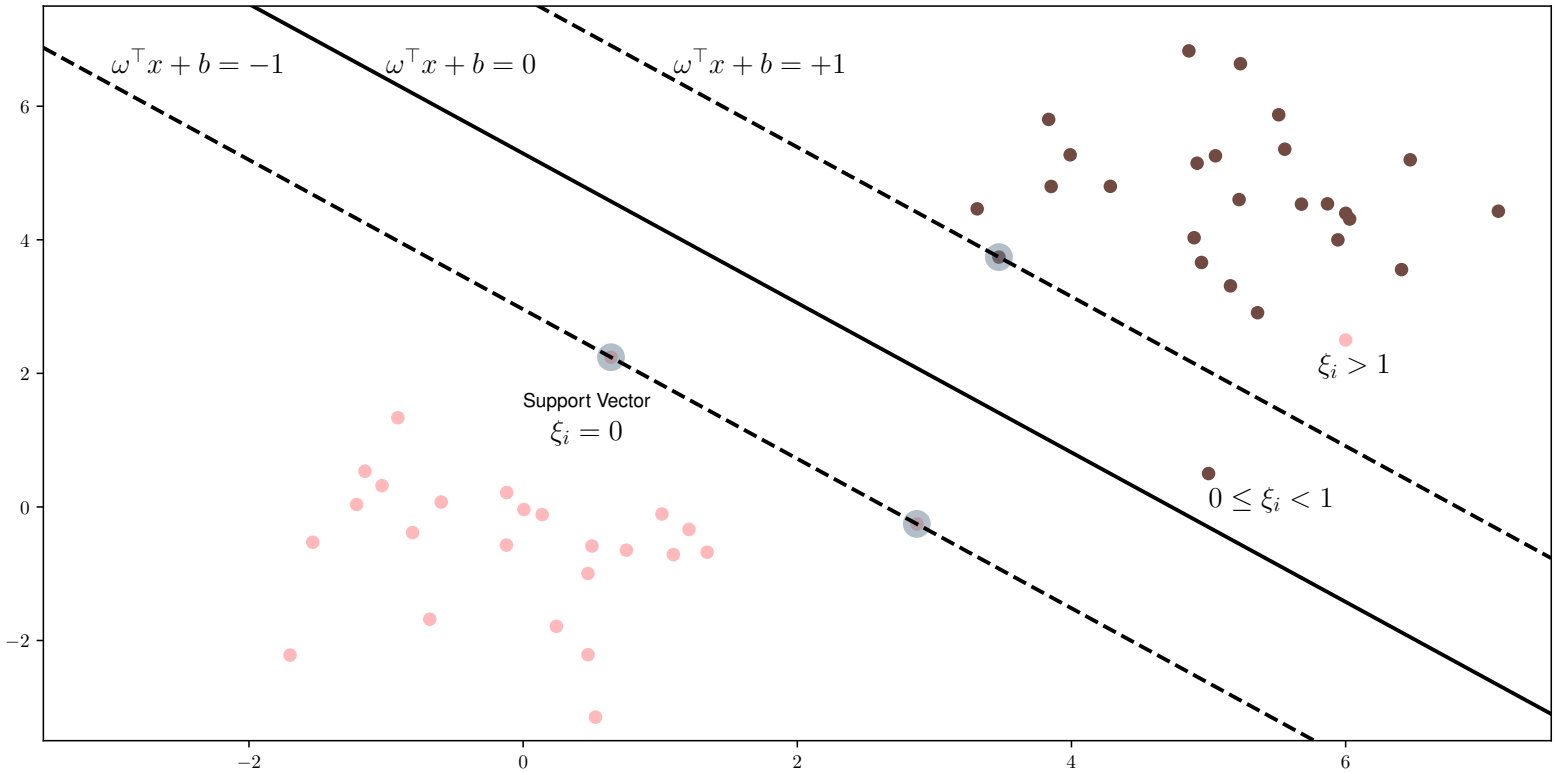


- An object i will be correctly classified if $0 \leq \xi_i < 1$
- Misclassified if $\xi_i > 1$.
- In the case $\xi_i = 1$, we get a tie (objects coincide with the hyperplane).
- $\sum_{i=1}^{n} \xi_i$ is an upper bound of the number of misclassified objects.

Soft Margin SVM

- Soft margin SVM relaxes the constraint to allow points to be inside the margin or even on the wrong side of the boundary
- The boundaries are penalized by a quantity that reflects the extent of the violation
- Slack variables $\xi_i \geq 0$ for each sample to measure the extent of the violation.

Slack variables



Slack variables

- For points on or inside the correct margin:

$$\xi_i = 0$$

- For other points:

$$\xi_i = 1 - y_i(\omega^\top x_i + b)$$

- If a point is in the decision boundary:

$$\xi_i = 1$$

- The hard margin constraint:

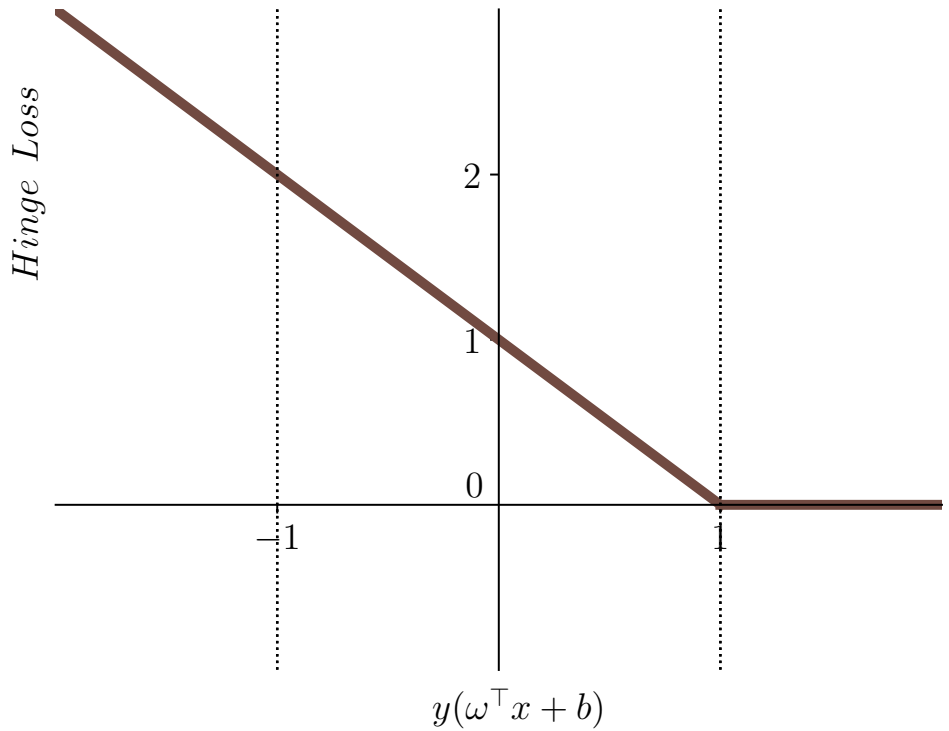
$$y_i(\omega^\top x_i + b) \geq 1 \quad \forall i = 1, \ldots, n$$

- Now becomes:

$$y_i(\omega^\top x_i + b) \geq 1 - \xi_i \quad \forall i = 1, \ldots, n$$

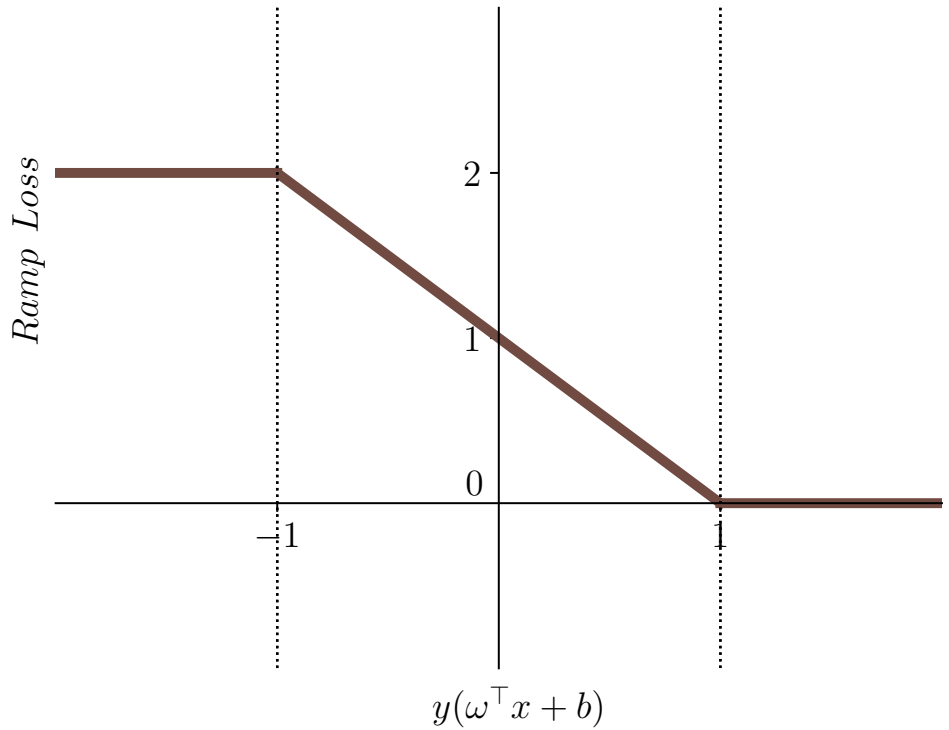
Hinge Loss

$$\ell(x, y) = \max(0, 1 - y(\omega^\top x + b)) = \begin{cases} 0 & \text{if } y(\omega^\top x + b) \geq 1 \\ 1 - y(\omega^\top x + b) & \text{if } y(\omega^\top x + b) \leq 1 \end{cases}$$



Ramp Loss

$$\ell(x, y) = \begin{cases} 0 & \text{if } y(\omega^\top x + b) \geq 1 \\ 1 - y(\omega^\top x + b) & \text{if } -1 \leq y(\omega^\top x + b) \leq 1 \\ 2 & \text{if } y(\omega^\top x + b) \leq -1 \end{cases}$$



Example: Type A data

Both classes have the identity matrix as covariance.

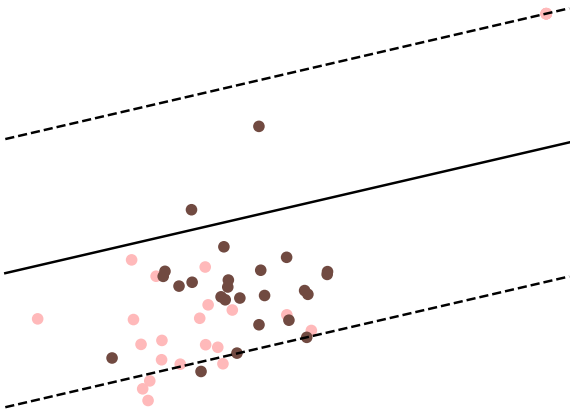
- Class $+1$: mean is the origin.
- Class -1: mean is $(2/(d), \ldots, 2/(d))$.

The training data sets are contaminated with outliers. Outlier observations are sampled for Class $+1$ using a Gaussian distribution with covariance matrix 0.001 times the identity matrix and with a mean $(10/(d), \ldots, 10/(d))$.

Brooks, J.P. *Support vector machines with the ramp loss and the hard margin loss*. Operations Research: 59(2), 467-479 (2011)

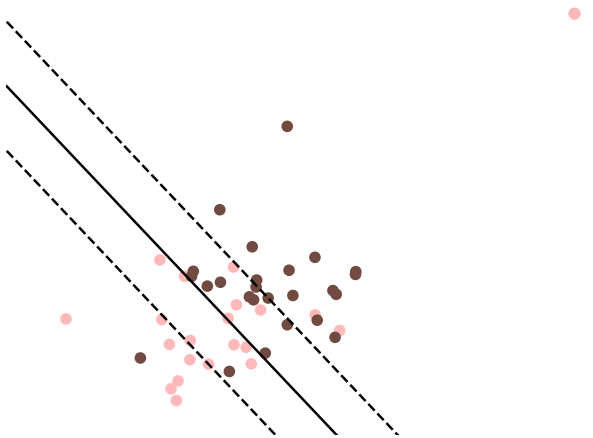
Robustness to outliers

SVM with the Hinge Loss



Accuracy of 44%

SVM with the Ramp Loss



Accuracy of 78%

SVM with the Ramp Loss

$$\min_{\omega,b,\xi,z} \frac{1}{2}\sum_{j=1}^{d}\omega_j^2 + \frac{C}{n}\left(\sum_{i=1}^{n}\xi_i + 2\sum_{i=1}^{n}(1-z_i)\right)$$

s.t.

$$\boxed{(y_i(\omega^\top x_i + b) - 1 + \xi_i)\cdot z_i \geq 0 \quad \forall i = 1,\dots,n}$$

$$0 \leq \xi_i \leq 2 \quad \forall i = 1,\dots,n$$
$$z \in \{0,1\}^n$$
$$\omega \in \mathbb{R}^d$$
$$b \in \mathbb{R}.$$

The kernel trick

- Soft-margin SVM[1]:

$$\min_{\omega,\, b,\, \xi} \ \frac{1}{2}\sum_{j=1}^{d}\omega_j^2 + \frac{C}{n}\sum_{i=1}^{n} g_i(\xi_i)$$

s.t.

$$\begin{aligned} y_i(\omega^\top \phi(x_i) + b) &\geq 1 - \xi_i \qquad \forall i = 1, \ldots, n \\ \omega &\in \mathbb{R}^d \\ b &\in \mathbb{R} \\ \xi_i &\geq 0 \qquad \forall i = 1, \ldots, n, \end{aligned}$$

(2)

- Kernel trick. Example: $x \in \mathbb{R}^3, \phi(x) \in \mathbb{R}^{10}$

$$\phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_3, x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3)$$

[1] Cortes, C., Vapnik, V. *Support-vector networks*. Machine learning, 20(3), 273-297.

Definition (Kernel)

$k : X \times X \to \mathbb{R}$ is a kernel if

1. k is symmetric: $k(x_1, x_2) = k(x_2, x_1)$.
2. k is positive semi-definite, i.e., $\forall x_1, x_2, \ldots, x_n \in X$, the "Gram Matrix" K defined by $K_{ij} = k(x_i, x_j)$ is positive semi-definite. (A matrix $M \in \mathbb{R}n \times n$ is positive semi-definite if $\forall a \in \mathbb{R}^n$, $a'Ma \geq 0$.)

Kernel $k(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle = \phi(x_1)^\top \phi(x_2)$. Most popular kernels:

- Linear

$$k(x_1, x_2) = \langle x_1, x_2 \rangle$$

- Radial Basis Function (RBF)

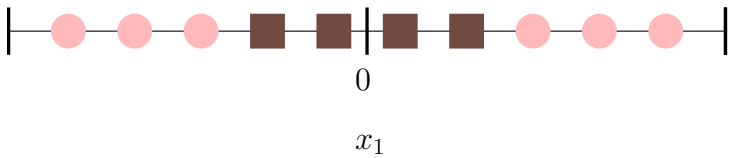
$$k(x_1, x_2) = e^{-\gamma \|x_1 - x_2\|^2}$$

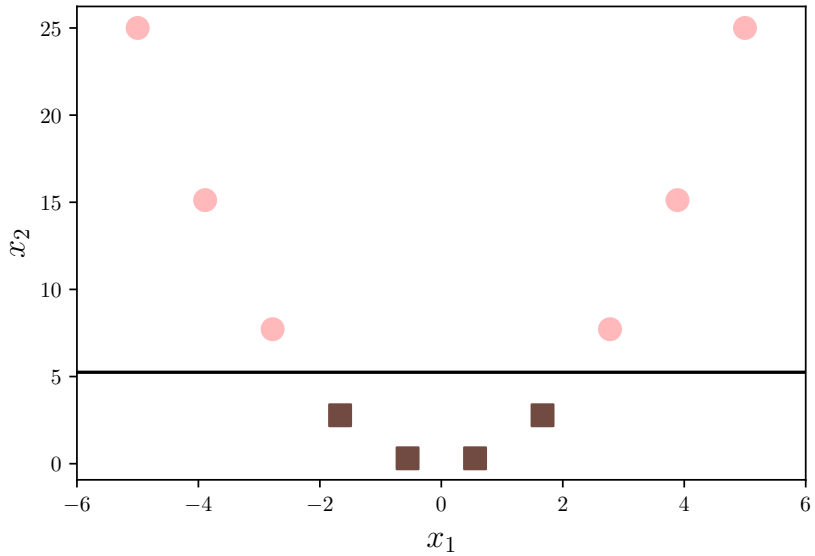
- Polynomial kernel (of dimension d):

$$k(x_1, x_2) = (x_1^\top x_2 + c)^d$$

How do we know kernels help separating data?

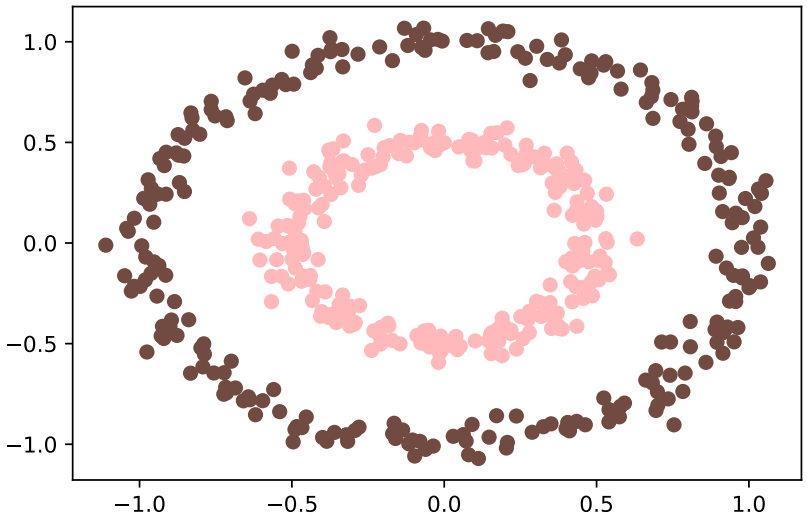
- In $\mathbb{R}^d$, any d independent vectors are linearly separable.
- If k is positive definite $\rightarrow$ data linearly separable.
- Example: $x_1 \in \mathbb{R}$, $\Phi(x_1) = (x_1, x_1^2) \in \mathbb{R}^2$

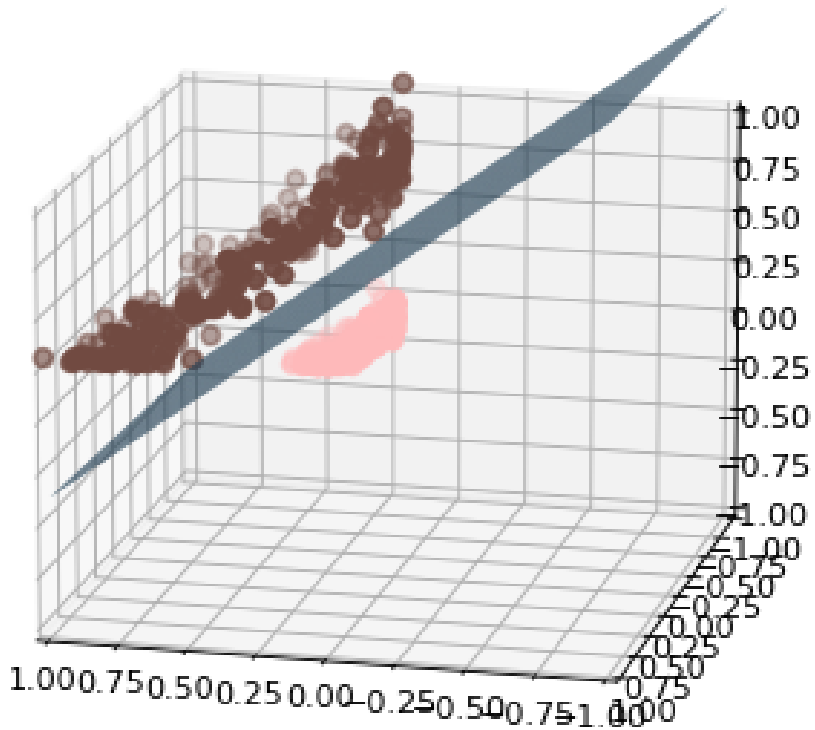


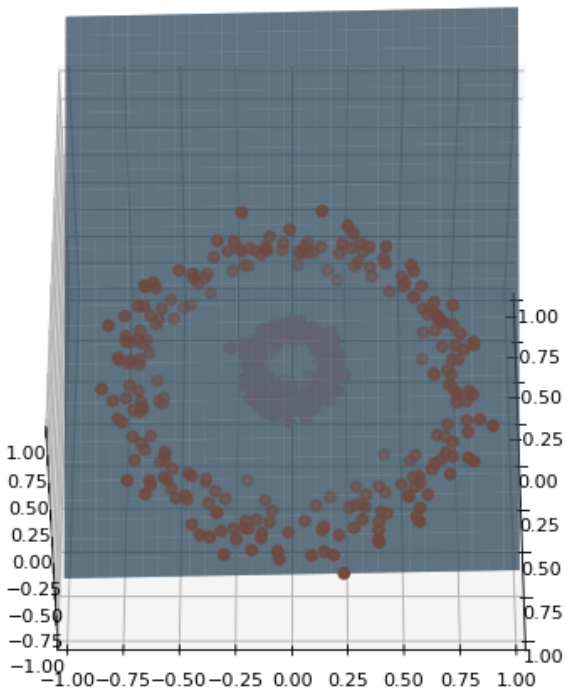


Example: $\mathbb{R}^2 \rightarrow \mathbb{R}^3$

$$x \in \mathbb{R}^2, \phi(x) \in \mathbb{R}^3, \phi(x) = (x_1^2, \sqrt{2}x_1, x_2, x_2^2)$$






1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00
1.00
0.75
0.50
0.25
0.00
0.25
0.50
0.75
1.00

Parameter selection

- Important step of the ML cycle.
- Parameters: C, kernel parameters.
- Example:

$$\gamma \text{ in } e^{-\gamma \|x-y\|^2}$$

$$c, d \text{ in } (x^\top y + c)^d$$

- How to select them?

And as well:

- How to select kernels? RBF, polynomial,...
- How to select methods? SVM, decision trees,...

Performance evaluation

- In practice:
  available data $\rightarrow$ training, testing and validation
- Train in the training.
- Test in the testing.
- Report in the validation.
- K-fold cross-validation.

K-fold cross-validation

For each k,

(i) Split the dataset into training, testing and validation sets.

(ii) For each C,

Solve the SVM in the training set and obtain the solution (ω^C, b^C, ξ_i^C).

(iii) Choose the optimal C^* in the testing set.

Report the quality metric for the classifier $(\omega^{C^*}, b^{C^*}, \xi_i^{C^*})$ in the validation set.

Multi-class classification

- k classes
- One-against-all: train k binary SVMs

$$\begin{array}{ccc} \text{1st class} & \text{vs.} & (2-k)\text{th class} \\ \text{2nd class} & \text{vs.} & (1,3-k)\text{th class} \\ & \vdots & \end{array}$$

- k decision functions

$$(\omega^1)^\top x + b^1$$

$$\vdots$$

$$(\omega^k)^\top x + b^k$$

- Prediction

$$\arg\max_j (\omega^j)^\top x + b^j$$

- Reason: If the 1st class, then we should have

$$(\omega^1)^\top x + b^1 \geq 0$$
$$(\omega^2)^\top x + b^2 \leq 0$$
$$\vdots$$
$$(\omega^k)^\top x + b^k \leq 0$$

- One-against-one: train $k(k-1)/2$ binary SVMs
- Example: 4 classes $\rightarrow$ 6 binary SVMs

$y_i = +1$	$y_i = -1$	Decision functions
Class 1	Class 2	$f^{12}(x) = (\omega^{12})^\top x + b^{12}$
Class 1	Class 3	$f^{13}(x) = (\omega^{13})^\top x + b^{13}$
Class 1	Class 4	$f^{14}(x) = (\omega^{14})^\top x + b^{14}$
Class 2	Class 3	$f^{23}(x) = (\omega^{23})^\top x + b^{23}$
Class 2	Class 4	$f^{24}(x) = (\omega^{24})^\top x + b^{24}$
Class 3	Class 4	$f^{34}(x) = (\omega^{34})^\top x + b^{34}$

- In the testing dataset, we predict all binary SVMs

Classes		winner
1	2	1
1	3	1
1	4	1
2	3	2
2	4	4
3	4	3

- Select the one with the largest vote

Class	1	2	3	4
#votes	3	1	1	1

Quality of a classifier

- Accuracy: percentage of objects correctly classified.
- Sensitivity (True Positive Rate): the proportion of those who received a positive prediction out of those who actually belong to the positive class.
- Specificity (True Negative Rate): the proportion of those who received a negative prediction out of those who actually belong to the negative class.
- Sparsity (vs. complexity): $\frac{\#(w_j=0, j=1,\ldots,d)}{d} \cdot 100$

And fairness!!!

Equal opportunity, demographic parity, predictive parity, predictive equality, p%-rule, unawareness...

The COMPAS dataset

- A dataset from the investigation by Propublica about a commercial tool made by Northpointe, Inc. to assess the criminal defendant's likelihood of becoming a recidivist.
- `https://github.com/propublica/compas-analysis`
- Sample size: $n = 7214$.
- Dimension: $d = 52$.
  - 2 continuous features.
  - 17 discrete features.
  - 33 categorical features.

How do we deal with categorical features?

Race

The categorical feature race has the following categories: African-American, Caucasian, Hispanic, Asian, Native American, Other.

- We binarize the categorical feature *race* into 6 binary features.

i	Race	African-American	Caucasian	Hispanic	Asian	Native American	Other
1	Asian	0	0	0	1	0	0
2	Other	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Bibliography

- Cortes, C., Vapnik, V. *Support-vector networks*. Machine learning, 20(3), 273-297.
- Brooks, J.P. *Support vector machines with the ramp loss and the hard margin loss*. Operations Research: 59(2), 467-479 (2011).