Supervised approaches: SVM & Random Forests

Machine learning basic concepts

Gilles Le Chenadec ✉

ENSTA BRETAGNE

Lab-STICC

Objectifs du cours

Introduction

Apprentissage supervisé

- Régression
- Régression linéaire
- Classification supervisée
- Régression logistique
- Neural Networks
- Réseaux de neurones
- Support Vector Machines (SVM)
- Decision trees
- Random forests

Apprentissage supervisé

Support Vector Machines (SVM)

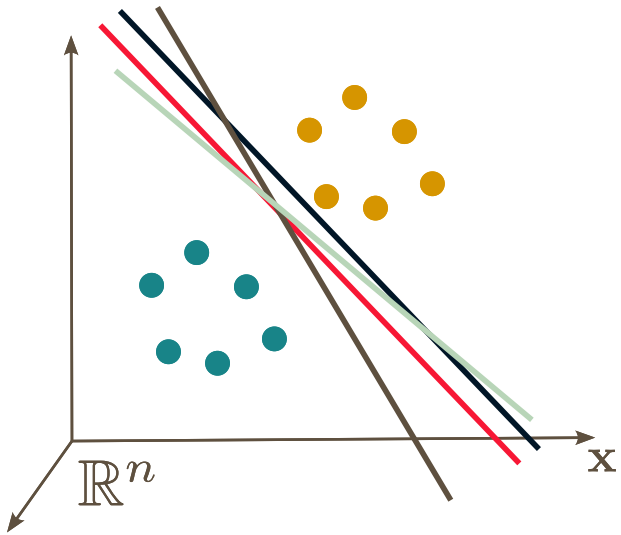
Classifieur: Support Vector Machine

- Approche
  - classifieur binaire (deux classes)
  - Les nuages des classes sont représentées par quelques exemples (les vecteurs/points supports).
  - A l'issue de l'entraînement, on peut ne garder que quelques points.
  - Au centre de cette approche, est la notion de Marge
- Pour découvrir cette approche, on va successivement aborder les cas de la classification
  - binaire linéaire
  - binaire linéaire avec tolérance
  - binaire non-linéaire
- Pour la classification multi-classe, vous vous baserez sur les différentes stratégies pour transformer un classifieur binaire en classifieur multiclasse

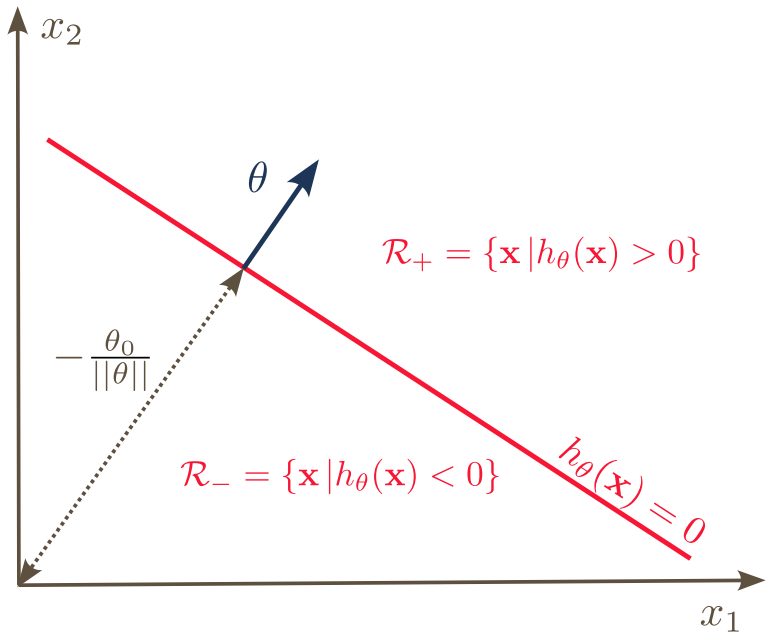
SVM: linear separable case

- Hypothèse forte: le problème de classification est linéaire
- Différentes surfaces de décisions (**hyperplans** dans l'espace à n dim.; **droite** $n = 2$) possibles
- Approche SVM consiste à maximiser la **marge**

Binary supervised classification

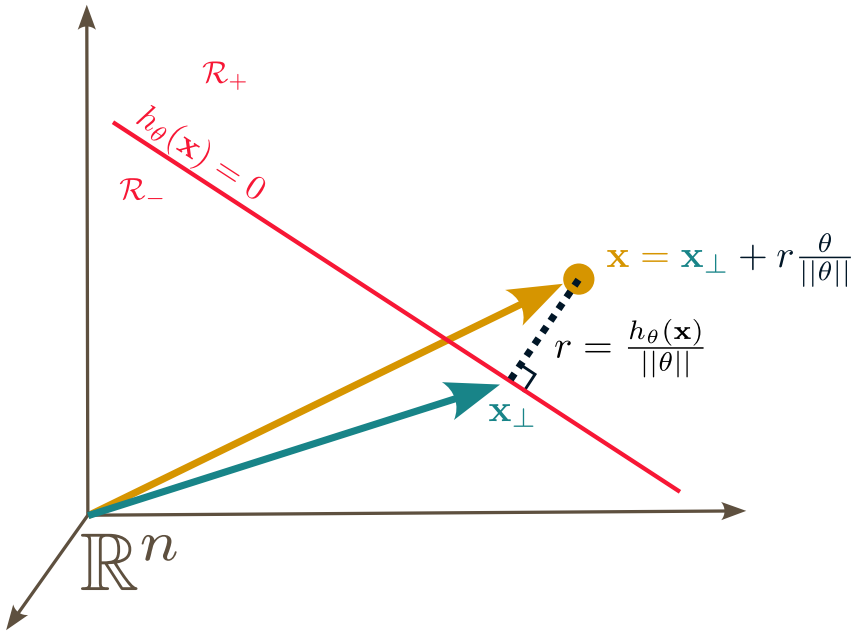


Géométrie de la classification binaire linéaire (1/2)



- La droite (hyperplan si $n > 2$) $h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^\top \mathbf{x} + \theta_0$ définit les deux demi-espaces $\mathcal{R}_+$ et $\mathcal{R}_-$
- Le vecteur $\boldsymbol{\theta}$ est perpendiculaire à la droite $h_{\boldsymbol{\theta}}(\mathbf{x})$[1]
- La distance entre la droite et l'origine est $\frac{\boldsymbol{\theta}^\top \mathbf{x}}{||\boldsymbol{\theta}||} = -\frac{\theta_0}{||\boldsymbol{\theta}||}$

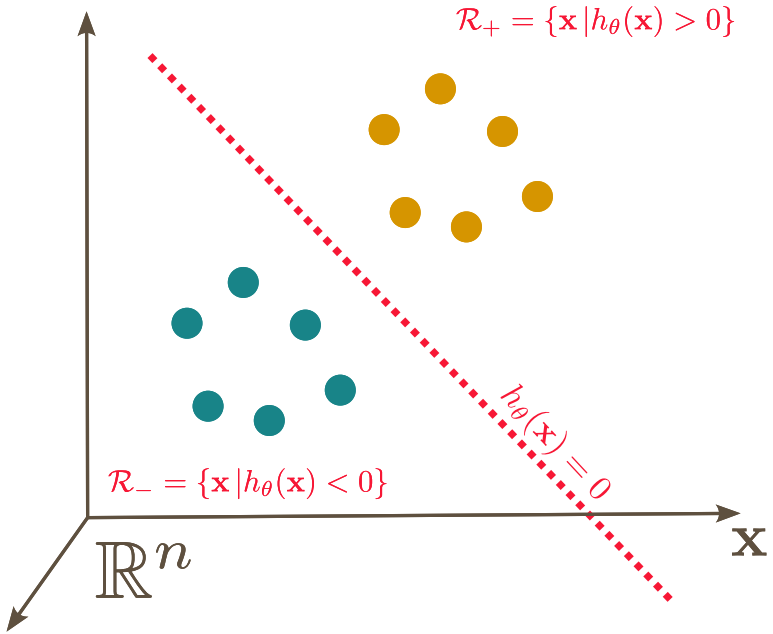
Géométrie de la classification binaire linéaire (2/2)



Soit $\mathbf{x}\ in\mathbb{R}^n$

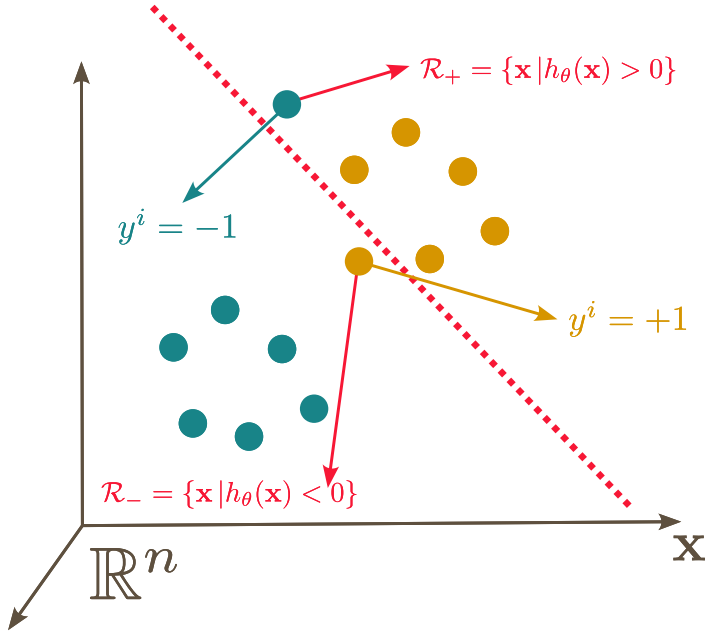
- alors $\mathbf{x} = \mathbf{x}_\perp + r\frac{\boldsymbol{\theta}}{||\boldsymbol{\theta}||}$
- avec $r = \frac{h_{\boldsymbol{\theta}}(x)}{||\boldsymbol{\theta}||}^2$

SVM pour un problème linéaire binaire (1/8)



- **ATTENTION, par convention:** la variable cible a forcément deux valeurs possibles qui sont obligatoirement $y^i \in \{-1, +1\}$

SVM pour un problème linéaire binaire (2/8)



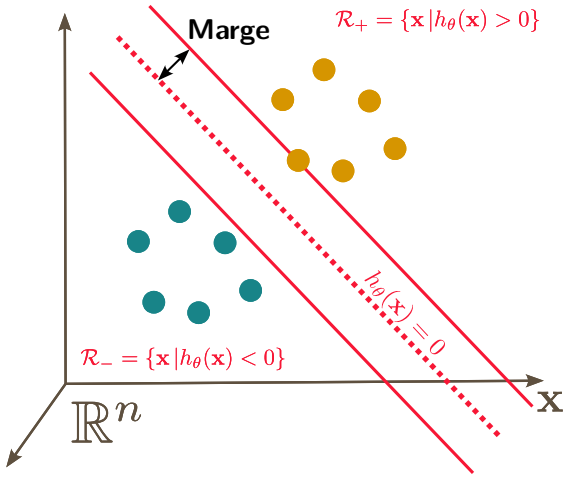
- La distance signée pour un exemple $(\mathbf{x}_n, t_n)$ est:

$$\frac{y^i.h_{\boldsymbol{\theta}}(\mathbf{x}^i)}{||\boldsymbol{\theta}||} = \frac{y^i.(\boldsymbol{\theta}^\top \mathbf{x}^i + \theta_0)}{||\boldsymbol{\theta}||}$$

- si il y a erreur de classification pour $\mathbf{x}^i$, la distance signée $\frac{y^i.h_{\boldsymbol{\theta}}(\mathbf{x}^i)}{||\boldsymbol{\theta}||}$ est **négative**
  - si $y^i = -1$, $h_{\boldsymbol{\theta}}(\mathbf{x}^i)$ doit être négatif sinon erreur de classification
  - si $y^i = +1$, $h_{\boldsymbol{\theta}}(\mathbf{x}^i)$ doit être positif sinon erreur de classification

SVM pour un problème linéaire binaire (3/8)

La **marge** est la distance signée entre la surface de décision et les entrées de l'ensemble d'entraînement/



- i.e. la marge est la distance signée pour un exemple $(\mathbf{x}^i, y^i)$ est:

$$\frac{y^i.h_{\boldsymbol{\theta}}(\mathbf{x}^i)}{||\boldsymbol{\theta}||} = \frac{y^i.(\boldsymbol{\theta}^\top \mathbf{x}^i + \theta_0)}{||\boldsymbol{\theta}||}$$

SVM pour un problème linéaire binaire (4/8)

- L'approche SVM cherche à estimer $\boldsymbol{\theta}$ et θ_0 qui maximise la marge

$$\arg\max_{\boldsymbol{\theta},\theta_0} \left[\frac{1}{||\boldsymbol{\theta}||} \min_i \left[y^i.(\boldsymbol{\theta}^\top \mathbf{x}^i + \theta_0) \right] \right]$$

- Le $\min_i [.]$ détermine le point le plus proche de la surface de décision.

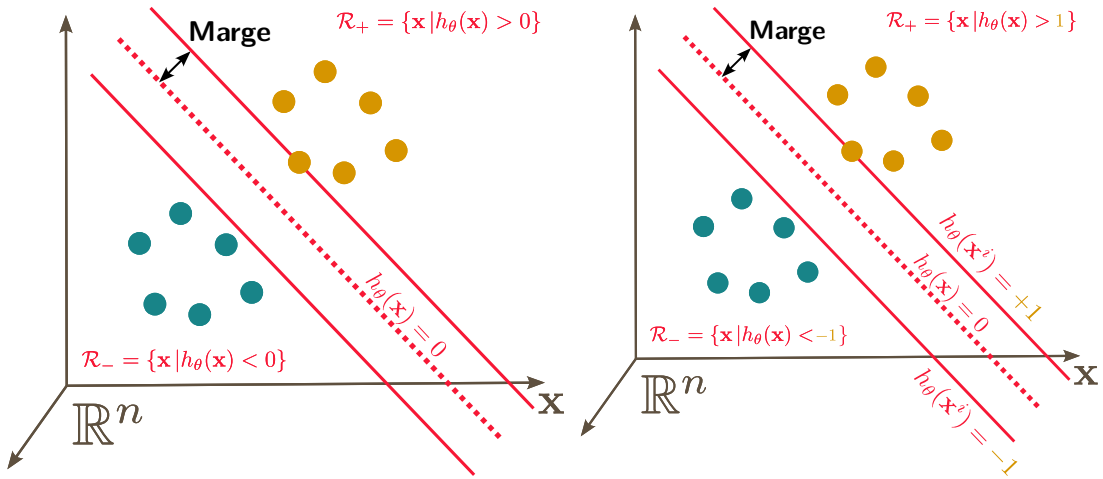
SVM pour un problème linéaire binaire (5/8)

- La marge est la même si on multiplie $\boldsymbol{\theta}$ et θ_0 par une constante a

$$\frac{y^i.(a\boldsymbol{\theta}^\top \mathbf{x}^i + a\theta_0)}{a||\boldsymbol{\theta}||} = \frac{y^i.(\boldsymbol{\theta}^\top \mathbf{x}^i + \theta_0)}{||\boldsymbol{\theta}||}$$

- On peut contraindre le point le plus proche de la surface de décision $(\mathbf{x}^i, y^i)$ tel que:

$$h_{\boldsymbol{\theta}}(\mathbf{x}^i) = y^i \left(\mathbf{w}^\top \mathbf{x}^i + \theta_0\right) = 1$$



SVM pour un problème linéaire binaire (6/8)

- En supposant l'ensemble d'entraînement linéairement séparable, l'approche SVM cherche à estimer $\boldsymbol{\theta}$ et θ_0 qui maximise la marge

$$\arg\max_{\boldsymbol{\theta},\theta_0} \left[\frac{1}{||\boldsymbol{\theta}||} \min_i \left[y^i.(\boldsymbol{\theta}^\top \mathbf{x}^i + \theta_0) \right] \right] = \arg\max_{\boldsymbol{\theta},\theta_0} \left[\frac{1}{||\boldsymbol{\theta}||}.1 \right]$$

- Le problème d'optimisation équivalent est :

$$\arg\min_{\boldsymbol{\theta},\theta_0} \left[\frac{||\boldsymbol{\theta}||^2}{2} \right] \text{ avec les } m \text{ contraintes: } y^i.(\boldsymbol{\theta}^\top \mathbf{x}^i + \theta_0) \geq 1 \ \forall i \in \{1, \ldots, m\}$$

SVM pour un problème linéaire binaire (7/8)

- Une solution à ce problème d'optimisation quadratique est de former le lagrangien avec les multiplicateurs $a^i \geq 0 \ \forall i \in \{1, \ldots, \grave{u}\}$:

$$L\left(\boldsymbol{\theta}, \theta_0, \mathbf{a}\right) = \frac{1}{2}||\boldsymbol{\theta}||^2 - \sum_{i=1}^{m} a^i \left[y^i \left(\boldsymbol{\theta}^\top \mathbf{x}^i + \theta_0\right) - 1\right]$$

- En annulant les dérivées par rapport à $\boldsymbol{\theta}$ et θ_0, on obtient:

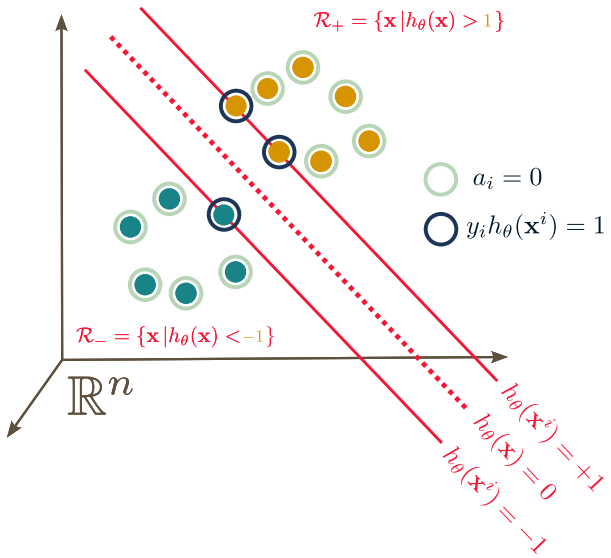
$$\boldsymbol{\theta} = \sum_{i=1}^{m} a^i y^i \mathbf{x}^i \text{ et } 0 = \sum_{i=1}^{\grave{u}} a^i y^i$$

SVM pour un problème linéaire binaire (8/8)

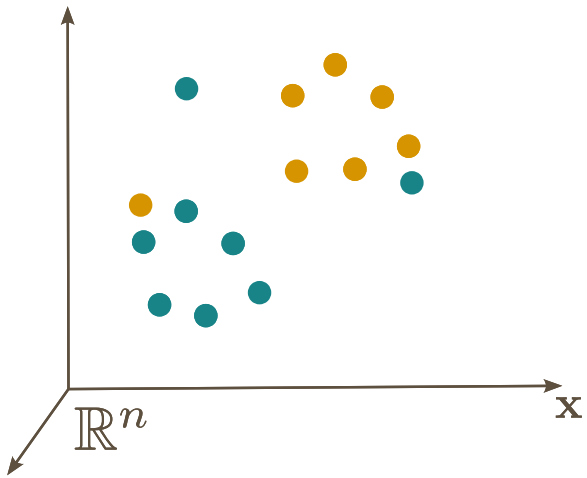
- On peut démontrer que la solution satisfait $\forall i \in \{1, \ldots, m\}$

$$a^i \geq 0$$
$$y^i h_{\boldsymbol{\theta}}(\mathbf{x}^i) - 1 \geq 0$$
$$a^i \left[y^i h_{\boldsymbol{\theta}}(\mathbf{x}^i) - 1 \right] = 0$$

- Pour la dernière condition et pour chaque point $\mathbf{x}^i$
  - soit $y^i h_{\boldsymbol{\theta}}(\mathbf{x}^i) = 1$ et le point $\mathbf{x}^i$ tel que $a^i > 0$ est appelé **vecteur support**.
  - soit $a^i = 0$ et le point x^i n'est pas un vecteur support (et peut être ignoré par la suite)

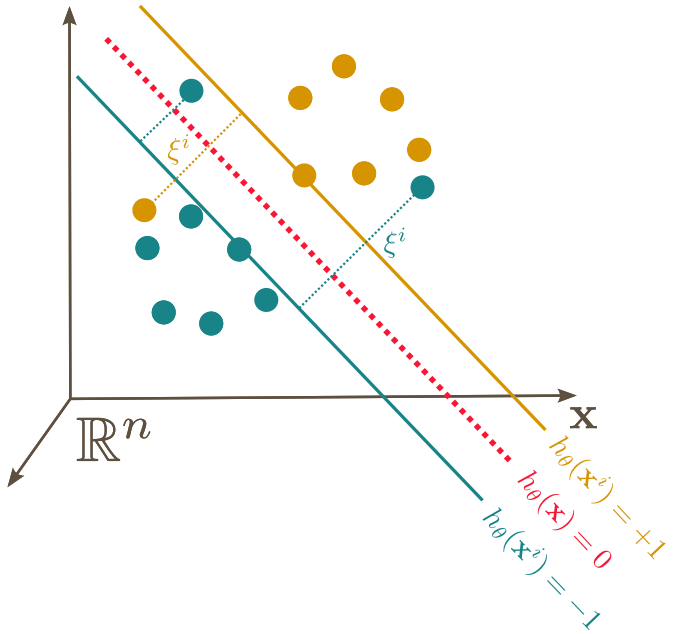


SVM pour un problème linéaire binaire non séparable (1/4)



- S'il y a chevauchement des deux classes, l'hypothèse de classification linéaire n'est plus valide

SVM pour un problème linéaire binaire non séparable (2/4)



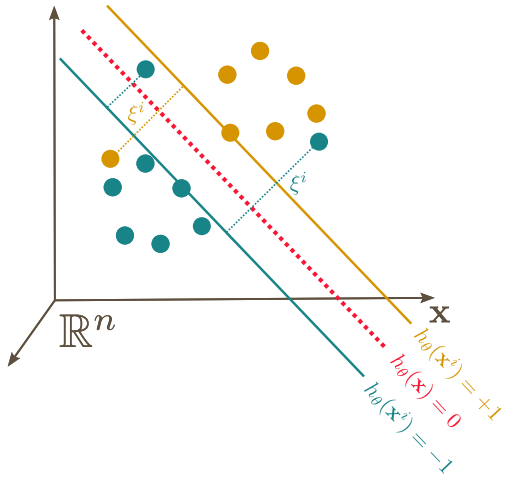
On relâche les contraintes en introduisant des variables *ressorts* $\xi^i,\ \forall i = 1, \ldots, n$ qui :

- correspondent aux violations des contraintes de marge.
- pénalisent l'erreur commise

Si:

- $0 \leq \xi^i \leq 1$, $\mathbf{x}^i$ est quand même bien classé
- $\xi^i \geq 1$, $\mathbf{x}^i$ est mal classé

SVM pour un problème linéaire binaire non séparable (3/4)



On cherche à trouver $\boldsymbol{\theta}^i$, θ_0 et ξ^i en minimisant la fonction de coût:

$$\underset{\boldsymbol{\theta},\theta_0,\xi^i}{\arg\min} \left[\frac{||\boldsymbol{\theta}||^2}{2} + C \sum_{i=1}^{n} \xi_n \right]$$

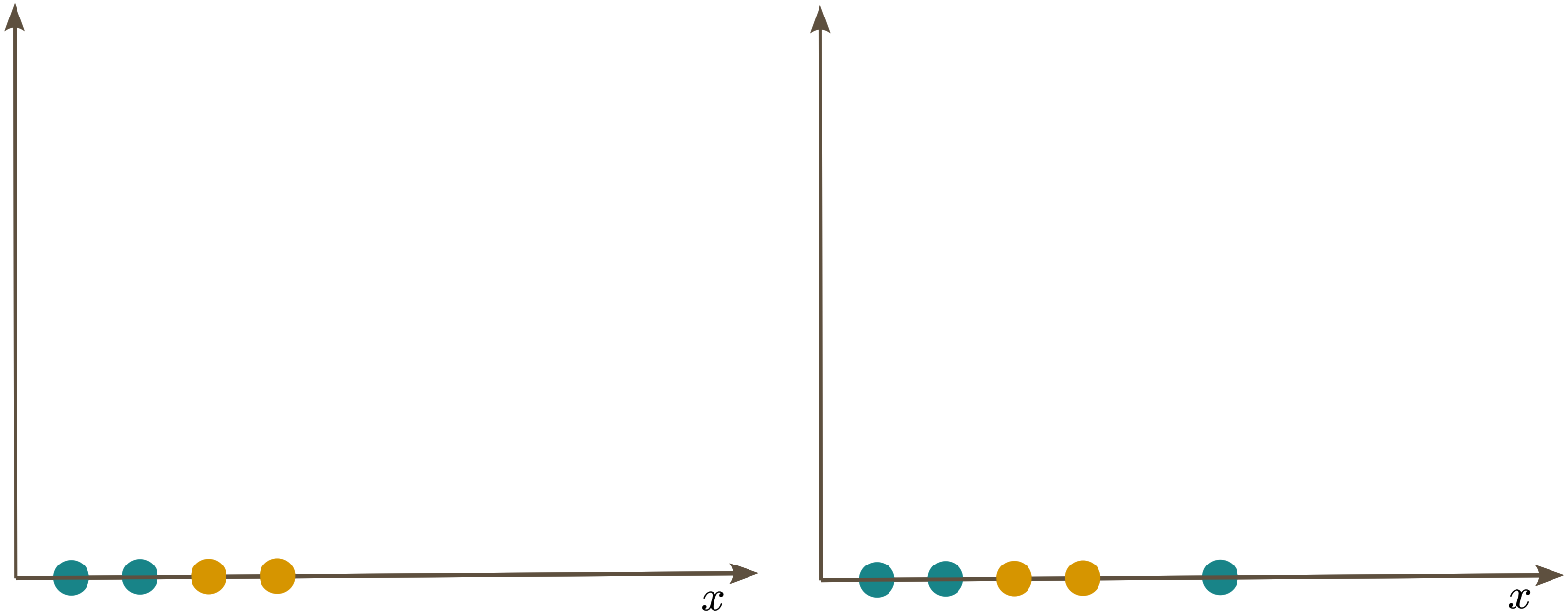
sous les contraintes $\begin{bmatrix} y^i \left(\theta^\top \mathbf{x}^i + \theta_0\right) \geq 1 - \xi^i & \forall i = 1, \ldots, n \\ \xi^i \geq 0 & \forall i = 1, \ldots, n \end{bmatrix}$

SVM pour un problème linéaire binaire non séparable (4/4)

Hyper-paramètre: $C > 0$

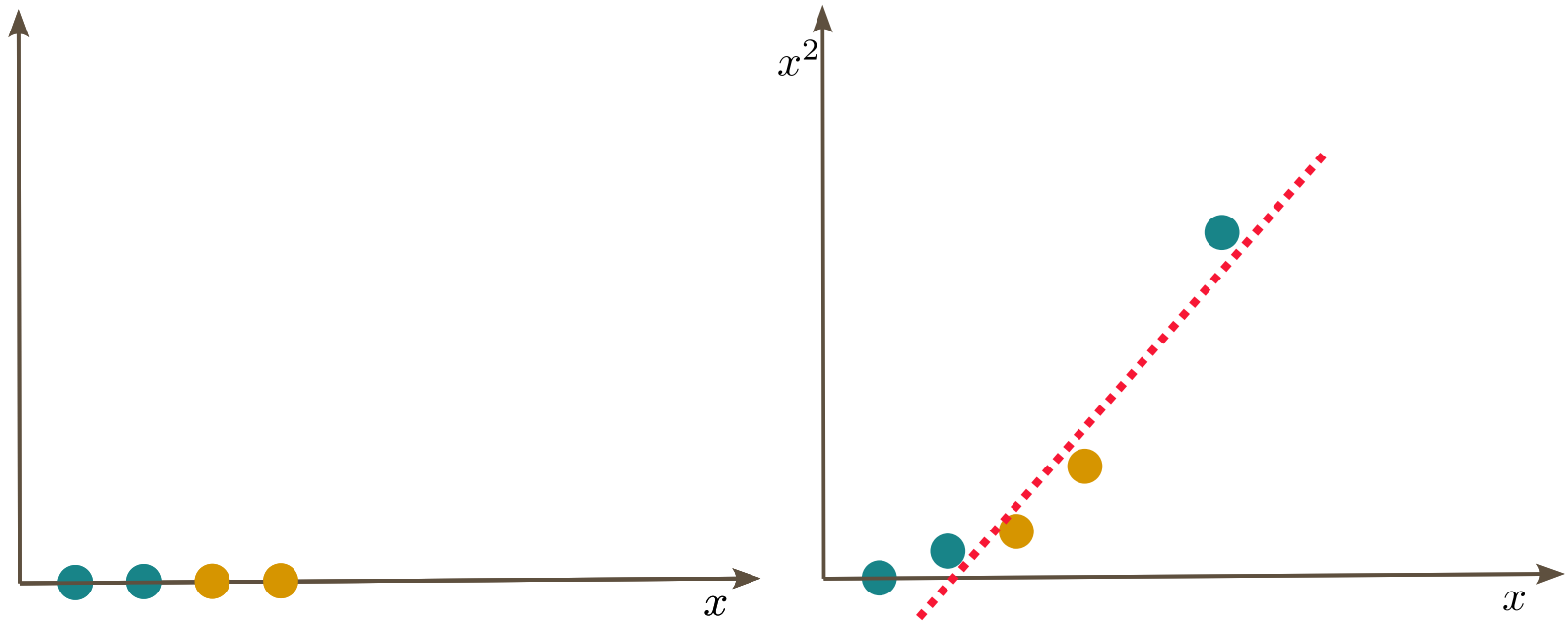
- plus C est petit, plus la capacité diminue (peu de pénalisation des erreurs)
- plus C est grand, plus le risque d'overfitting est grand

SVM pour un problème non-linéaire binaire non séparable (1/4)



Est-ce qu'un SVM peut correctement classer ces données?

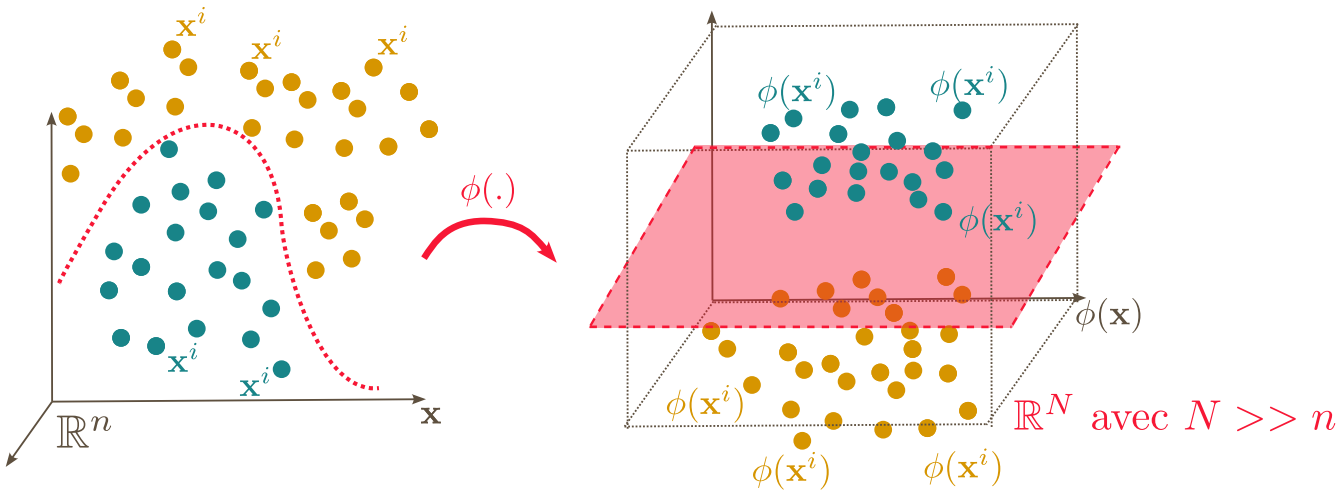
SVM pour un problème non-linéaire binaire non séparable (2/4)



Et maintenant?

SVM pour un problème non-linéaire binaire non séparable (3/4)

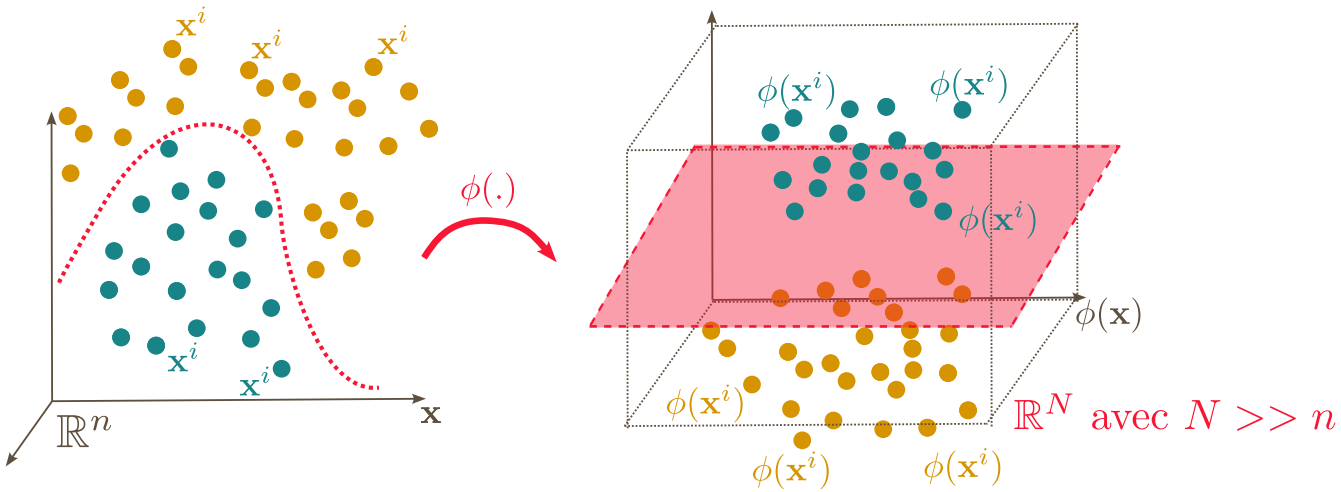
L'espace des descripteurs originaux ($\mathbf{x}$) peut être projeté pour un autre espace de descripteurs ($\phi(\mathbf{x})$) de plus grande dimension.



- $\mathbf{x}^i$ est remplacé dans les équations précédentes par $\phi(\mathbf{x}^i)$

SVM pour un problème non-linéaire binaire non séparable (4/4)

L'espace des descripteurs originaux ($\mathbf{x}$) peut être projeté pour un autre espace de descripteurs ($\phi(\mathbf{x})$) de plus grande dimension.



- Pour des raisons de mémoire, les $\mathbf{x}^i$ ne sont pas réellement projetés dans un espace de plus grande dimension. ϕ est en général uniquement défini via un noyau $k(.,.)$
- Choix possibles pour $k(\mathbf{x}, \mathbf{x}') =$
  - linéaire : $\mathbf{x}^\top \mathbf{x}'$
  - polynomial: $(\mathbf{x}^\top \mathbf{x}' + 1)^d$
  - rbf : $\exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$ où $\gamma > 0$
  - sigmoid : $\tanh(\gamma \cdot \mathbf{x}^\top \mathbf{x}' + r)$
- **Hyper-paramètre supplémentaire**, le noyau $k(.,.)$ et son/ses paramètre/s

Résumé SVM

Modèle $h_{\boldsymbol{\theta}}(\mathbf{x})$	$\boldsymbol{\theta}^\top \mathbf{x} + \theta_0$	$\boldsymbol{\theta}^\top \phi(\mathbf{x}) + \theta_0$
Fonction de coût	$\underset{\boldsymbol{\theta},\theta_0,\xi^i}{\arg\min} \left[\frac{\|\|\boldsymbol{\theta}\|\|^2}{2} + C \sum_{i=1}^m \xi^i \right]$	$\underset{\boldsymbol{\theta},\theta_0,\xi^i}{\arg\min} \left[\frac{\|\|\boldsymbol{\theta}\|\|^2}{2} + C \sum_{i=1}^m \xi^i \right]$
Contraintes $\forall i = 1, \ldots, m$	$\begin{cases} y^i \left(\boldsymbol{\theta}^\top \mathbf{x}^i + \theta_0 \right) \geq 1 - \xi^i \\ \xi^i \geq 0 \end{cases}$	$\begin{cases} y^i \left(\boldsymbol{\theta}^\top \phi(\mathbf{x}^i) + \theta_0 \right) \geq 1 - \xi^i \\ \xi^i \geq 0 \end{cases}$
Hyper-paramètres	C	C et $\phi()$ et ses params
Prédiction	$\mathcal{C}_1$ si $h_{\boldsymbol{\theta}}(\mathbf{x}^i) \geq 0$; sinon $\mathcal{C}_2$	$\mathcal{C}_1$ si $h_{\boldsymbol{\theta}}(\mathbf{x}^i) \geq 0$; sinon $\mathcal{C}_2$

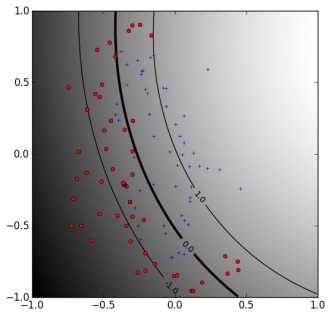
SVM: influence de γ

source ou ici aussi

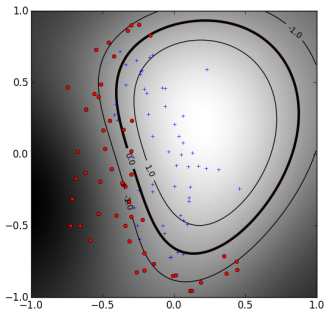
σ est le paramètre du noyau gaussien.

- Plus la valeur de σ est petite plus le modèle va être capable d'apprendre avec précision les données d'entrainement.
- Plus on fait décroître la valeur de σ plus le modèle va être sujet au phénomène de surapprentissage.

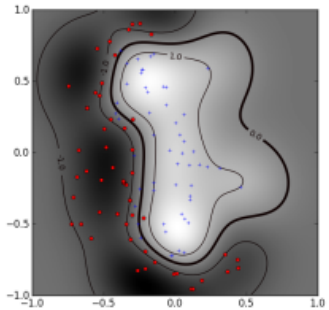
$\gamma = 3.16$



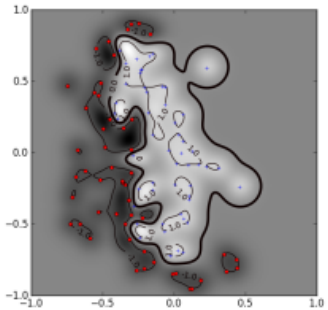
$\gamma = 1$



$\gamma = 0.316$



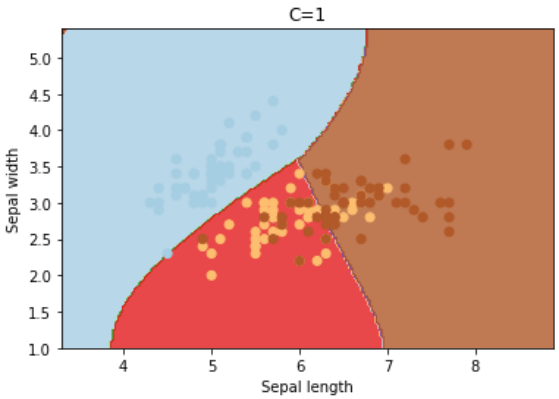
$\gamma = 0.1$

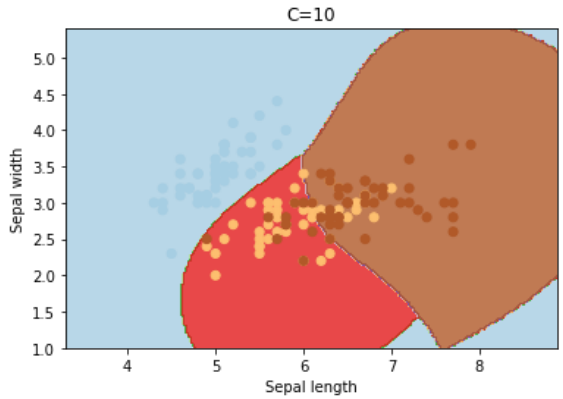


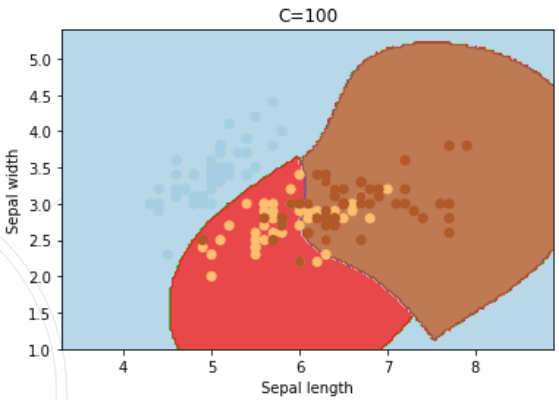
SVM: influence de C

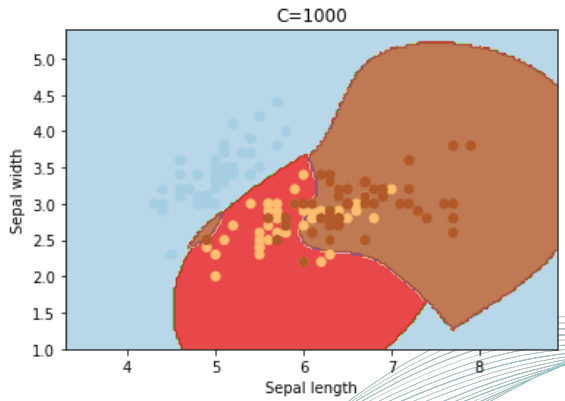
source ou ici aussi

C est le paramètre de pénalité du terme d'erreur et contrôle le compromis entre une frontière de décision régulière (smooth) et la classification correcte des points d'entrainements. Plus la valeur de C augmente, plus il y a un risque d'overfitting sur les données d'entraînement.









SVM pour un problème multi-classe (1/1)

Cf. cours 2: Multi-class classification with binary classifier

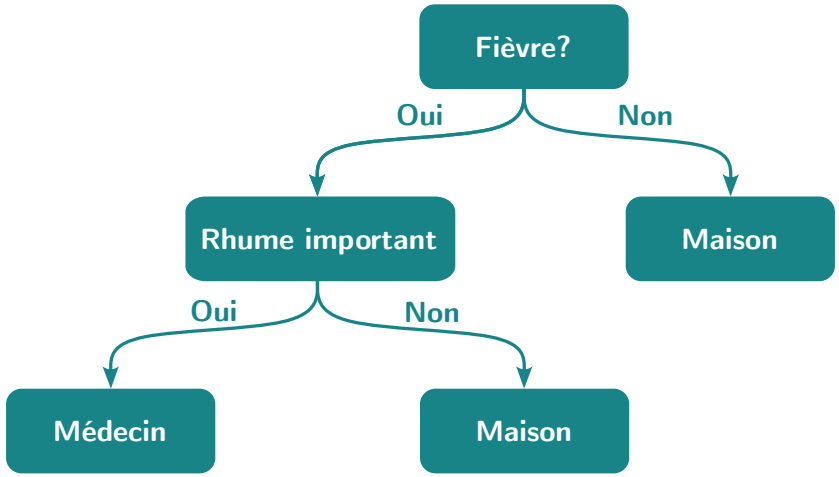
- Possiblement: stratégie One-Vs-All
- Dans Scikit-learn: “one-versus-one

Apprentissage supervisé

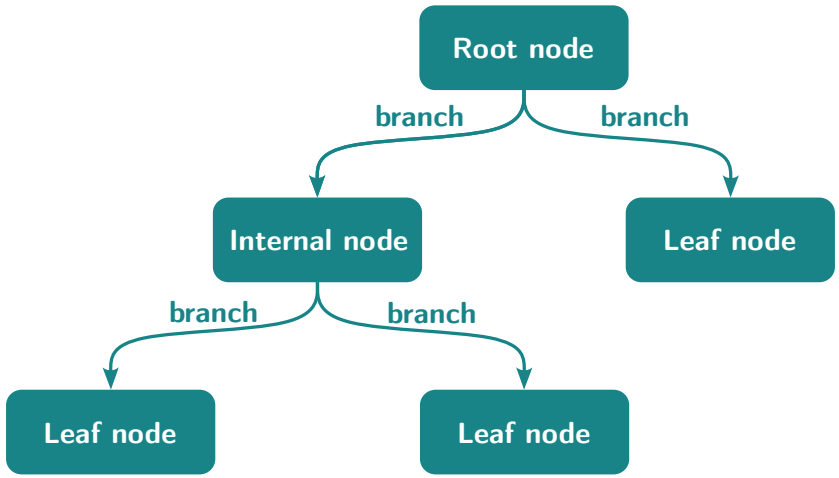
Decision trees

Que sont ces arbres de décisions

- Un **arbre de décisions** est un **modèle de machine learning** qui ressemble à une structure arborescente et hierarchique.
- Il s'agit d'un moyen de **prendre des décisions** basées sur des **règles apprises** à partir des données.
- Supposons que vous souhaitiez décider si vous devez aller chez le médecin ou rester à la maison. Cet arbre de décisions pourrait être:



Anatomie d'un arbre



- **Nœud racine: test initial ou décision initiale**, l'ensemble des **données** commence à être **divisé** en fonction de la **branche** prise.
- **Nœud interne:** tests/décisions intermédiaires. des branches en sortent.
- **(Nœuds) Feuilles** - nœuds terminaux, ces nœuds représentent la sortie où la décision finale est prise.
- **Branches:** lignes reliant les nœuds, représentant le résultat d'un test et menant au nœud suivant.
- **Une règle de décision**
  - chemin partant du nœud racine et se terminant à une feuille.est composée de décisions simples formant une règle plus complexe

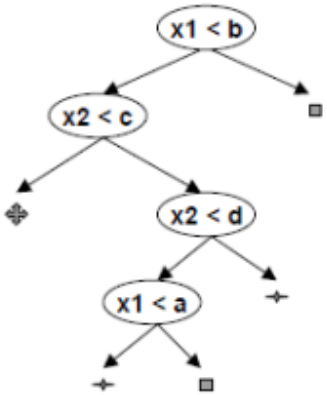
Type d'arbres de décision (1/1)

Il existe deux types d'arbre de décisions:

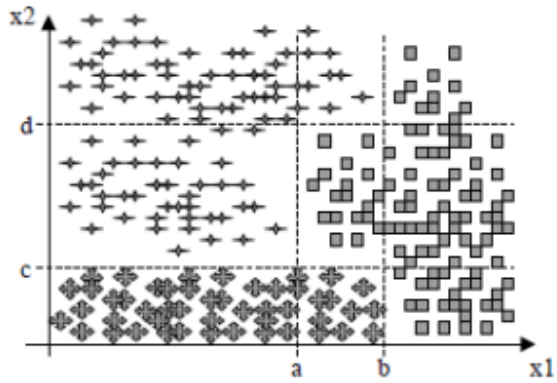
- Arbres de classification
- Arbres de régression (pas étudié)

Comment ces arbres fonctionnent?

Vue arbre



Vue données



En partant de la racine,

- l'arbre réalise un test/une décision basé sur un descripteur
- i.e. **divise linérairement l'espace des descripteurs** dans le but de séparer les données
- i.e. chaque test définit une **fonction discriminante dans l'espace des descripteurs le divisant en plus petites régions**
- Chaque division/chaque région de l'espace vise à créer des sous-ensembles de données les plus **homogènes en terme de classe**
- Une **feuille** définit une région localisée de l'espace où les données ont la même classe

Construction d'un arbre de décisions pour la classification

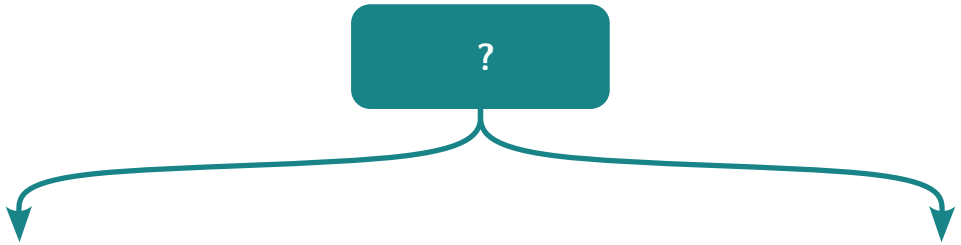
Construisons sur un exemple:

Jour	Temps	Température	Humidité	Vent	Classe
1	Soleil	Chaude	85	Non	No tennis
2	Soleil	Chaude	90	Oui	No tennis
3	Couvert	Chaude	78	Non	Tennis
4	Pluie	Douce	96	Non	Tennis
5	Pluie	Fraiche	80	Non	Tennis
6	Pluie	Fraiche	70	Oui	No tennis
7	Couvert	Fraiche	65	Oui	Tennis
8	Soleil	Douce	95	Non	No tennis
9	Soleil	Fraiche	70	Non	Tennis
10	Pluie	Douce	80	Non	Tennis
11	Soleil	Douce	70	Oui	Tennis
12	Couvert	Douce	90	Oui	Tennis
13	Couvert	Chaude	75	Non	Tennis
14	Pluie	Douce	80	Oui	No tennis

Attention aux variables

- **Nature des variables**? qualitative, quantitative
- Temps, Température, Humidité, Vent, Classe
- **Descripteurs (features)**?, **cibles/labels ?**
- Impact sur les règles de décision?
- Sur le **type d'arbre**? regression, classification

Apprentissage à partir des données (1/13)

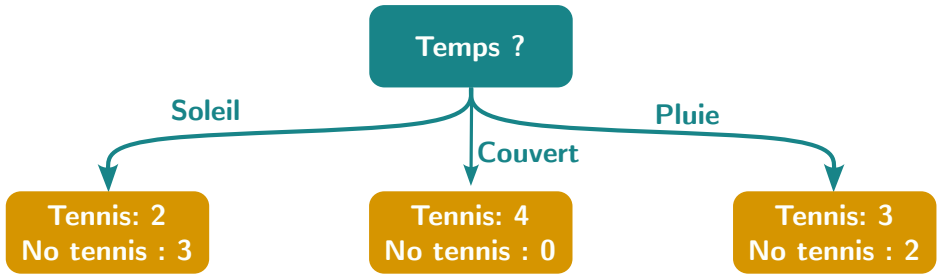


- première chose à décider: test du nœud racine qui créera la division initiale de l'ensemble de données.
- i.e. **selection automatique de descripteurs** (Feature selection)
- point fort des arbres de décisions
- examen de chacun des descripteurs selon un critère
- Critère:
  - le descripteur qui prédit le mieux la cible sera celui qui rend le plus homogène/pur les sous-ensembles de données après division.
  - Comment quantifier ce critère?

Apprentissage à partir des données (2/13)

Testons le premier descripteur

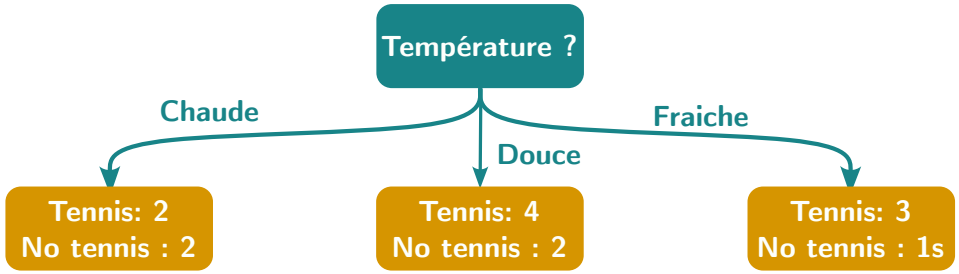
Num	Temps	Classe
1	Soleil	No tennis
2	Soleil	No tennis
3	Couvert	Tennis
4	Pluie	Tennis
5	Pluie	Tennis
6	Pluie	No tennis
7	Couvert	Tennis
8	Soleil	No tennis
9	Soleil	Tennis
10	Pluie	Tennis
11	Soleil	Tennis
12	Couvert	Tennis
13	Couvert	Tennis
14	Pluie	No tennis



Apprentissage à partir des données (3/13)

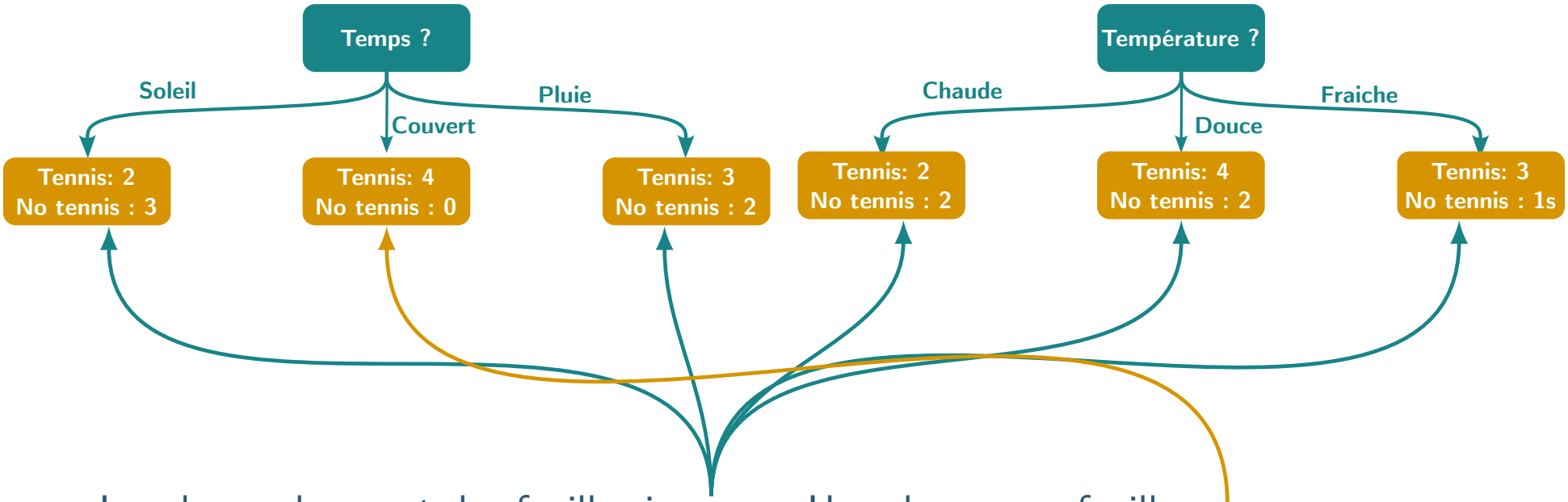
Testons le second descripteur

Num	Température	Classe
1	Chaude	No tennis
2	Chaude	No tennis
3	Chaude	Tennis
4	Douce	Tennis
5	Fraiche	Tennis
6	Fraiche	No tennis
7	Fraiche	Tennis
8	Douce	No tennis
9	Fraiche	Tennis
10	Douce	Tennis
11	Douce	Tennis
12	Douce	Tennis
13	Chaude	Tennis
14	Douce	No tennis



Apprentissage à partir des données (4/13)

Quel arbre founit le meilleur descripteur pour rendre homogène?



- Les deux arbres ont des feuilles impures. Un arbre a une feuille pure.
- Comment quantifier la feuille la plus impure ?
- Comment quantifier l'arbre dans son ensemble qui est le plus impur ?
- Il existe un grand nombre de mesures (entropie, χ^2, etc.), mais la plus célèbre est l'impureté de Gini.

Apprentissage à partir des données (5/13)

Impureté de Gini:

$$Gini = 1 - \sum_{k=C_1}^{C_k} p_k^2$$

où p_k est la proportion de données de la classe k dans le sous-ensemble considéré.

Quelques exemples:

Pour deux classes

- $p = [1, 0]$ alors $Gini = 0$ pur
- $p = [1/2, 1/2]$ alors $Gini = 0.5$ impur

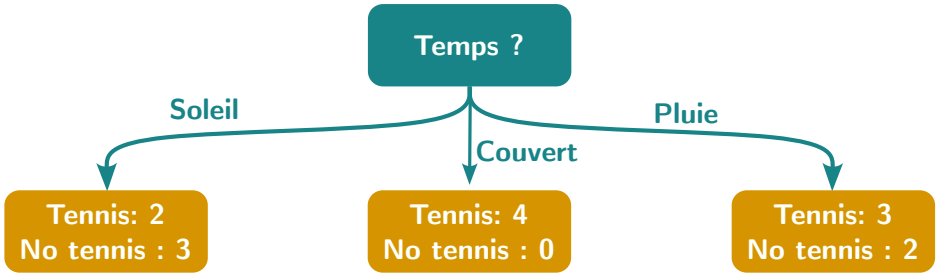
Pour trois classes

- $p = [1, 0, 0]$ alors $Gini = 0$ pur
- $p = [1/2, 1/2, 0]$ alors $Gini = 0.5$ impur
- $p = [1/3, 1/3, 1/3]$ alors $Gini = 0.666$ impur

Apprentissage à partir des données (6/13)

Principe

- Calcul de l'indice de Gini pour chaque branche
- Calcul de l'indice de Gini pour un descripteur: moyenne pondérée des modalités par effectifs

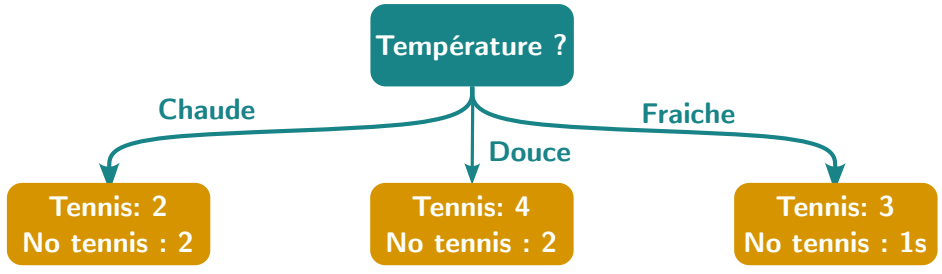


Branche de/du

- **gauche**: $Gini = 1 - \frac{2}{5}^2 - \frac{3}{5}^2 = 0.48$
- **milieu**: $Gini = 1 - \frac{4}{4}^2 - \frac{0}{4}^2 = 0.00$
- **droite**: $Gini = 1 - \frac{3}{5}^2 - \frac{2}{5}^2 = 0.48$

- **Total**:

$Gini = \left(\frac{5}{14}\right)0.48 + \left(\frac{4}{14}\right)0.00 + \left(\frac{5}{14}\right)0.48 = 0.17$



Branche de/du

- **gauche**: $Gini = 1 - \frac{2}{4}^2 - \frac{2}{4}^2 = 0.50$
- **milieu**: $Gini = 1 - \frac{4}{6}^2 - \frac{2}{6}^2 = 0.44$
- **droite**: $Gini = 1 - \frac{3}{4}^2 - \frac{1}{4}^2 = 0.375$

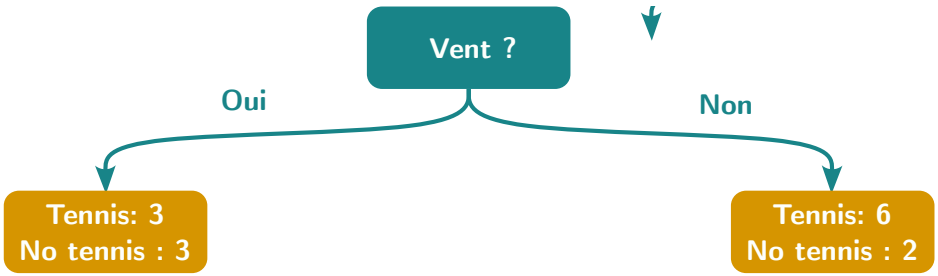
- **Total**:

$Gini = \left(\frac{4}{14}\right)0.50 + \left(\frac{6}{14}\right)0.44 + \left(\frac{4}{14}\right)0.375 = 0.44$

Apprentissage à partir des données (7/13)

Pour le descripteur: Vent

Jour	Vent	Classe
1	Non	No tennis
2	Oui	No tennis
3	Non	Tennis
4	Non	Tennis
5	Non	Tennis
6	Oui	No tennis
7	Oui	Tennis
8	Non	No tennis
9	Non	Tennis
10	Non	Tennis
11	Oui	Tennis
12	Oui	Tennis
13	Non	Tennis
14	Oui	No tennis



Branche de/du

- **gauche**: $Gini = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 = 0.375$
- **droite**: $Gini = 1 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2 = 0.48$

- **Total**:
  $Gini = \left(\frac{4}{14}\right) 0.375 + \left(\frac{10}{14}\right) 0.48 = 0.45$

Apprentissage à partir des données (8/13)

Pour le descripteur quantitatif: humidité

Apprentissage à partir des données (9/13)

Calculer l'indice de Gini pour une variable quantitative

- Trier les valeurs par ordre croissant
- Calculer la valeur moyenne
- Calculer l'indice de Gini total pour ce seuil
- Répéter pour le calcul pour tous les seuils

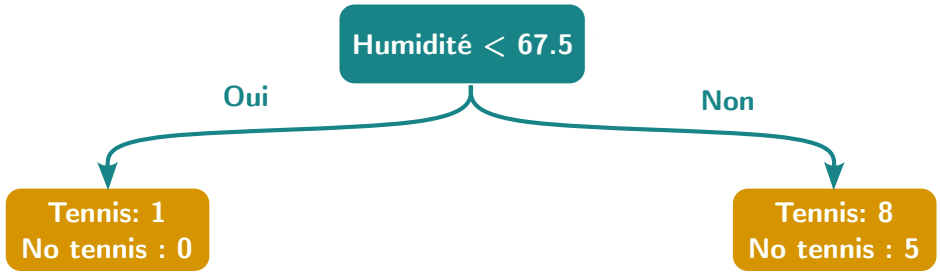
Jour	Humidité	Classe
1	85	No tennis
2	90	No tennis
3	78	Tennis
4	96	Tennis
5	80	Tennis
6	70	No tennis
7	65	Tennis
8	95	No tennis
9	70	Tennis
10	80	Tennis
11	70	Tennis
12	90	Tennis
13	75	Tennis
14	80	No tennis

Jour	Humidité	Humidité moyenne	Classe
7	65	-	Tennis
6	70	67.5	No tennis
9	70	67.5	Tennis
11	70	67.5	Tennis
13	75	72.5	Tennis
3	78	76.5	Tennis
5	80	79	Tennis
10	80	79	Tennis
14	80	79	No tennis
1	85	82.5	No tennis
2	90	87.5	No tennis
12	90	87.5	Tennis
8	95	92.5	No tennis
4	96	95.5	Tennis

Apprentissage à partir des données (10/13)

Calcul de l'indice de Gini pour ce seuil:

Jour	Humidité	Humidité moyenne	Classe
7	65	-	Tennis
6	70	67.5	No tennis
9	70	70.	Tennis
11	70	70.	Tennis
13	75	72.5	Tennis
3	78	76.5	Tennis
5	80	79	Tennis
10	80	80.	Tennis
14	80	80.	No tennis
1	85	82.5	No tennis
2	90	87.5	No tennis
12	90	90.	Tennis
8	95	92.5	No tennis
4	96	95.5	Tennis



Branche de/du

- **gauche**: $Gini = 1 - \frac{1}{1}^2 - \frac{0}{1}^2 = 0.00$
- **droite**: $Gini = 1 - \frac{8}{13}^2 - \frac{5}{13}^2 = 0.47$

- **Total**:
  $Gini = \left(\frac{1}{14}\right) 0.00 + \left(\frac{13}{14}\right) 0.47 = 0.44$

Apprentissage à partir des données (11/13)

On réitère le calcul pour tous les seuils:

Jour	Humidité	Humidité moyenne	Classe	Gini impurity
7	65	-	Tennis	-
6	70	67.5	No tennis	0.44
9	70	70.	Tennis	0.45
11	70	70.	Tennis	0.45
13	75	72.5	Tennis	0.45
3	78	76.5	Tennis	0.43
5	80	79	Tennis	0.40
10	80	80.	Tennis	0.37
14	80	80.	No tennis	0.37
1	85	82.5	No tennis	0.39
2	90	87.5	No tennis	0.44
12	90	90.	Tennis	0.46
8	95	92.5	No tennis	0.45
4	96	95.5	Tennis	0.44

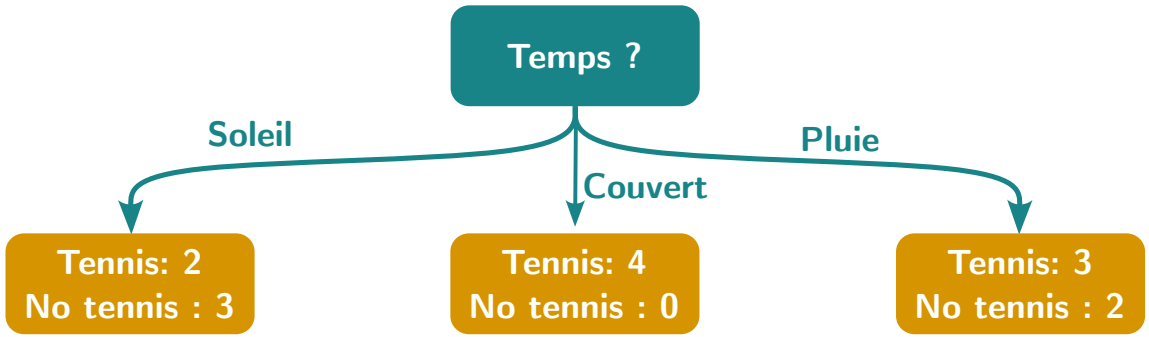
On choisit l'indice de Gini minimale sur les lignes. Ici le seuil 80 donne le plus faible indice de 0.37 **Pour le descripteur Humidité et le seuil 80, l'indice de Gini est de 0.37.**

Apprentissage à partir des données (12/13)

Choix du premier descripteur: synthèse des résultats

	Temps	Température	Humidité < 80	Vent
Indice de Gini	0.17	0.44	0.37	0.45

- donc le noeud racine est le temps et l'arbre débute de cette manière:



- On essayer d'homégénéiser le sous-ensemble de gauche puis de droite...on recommence

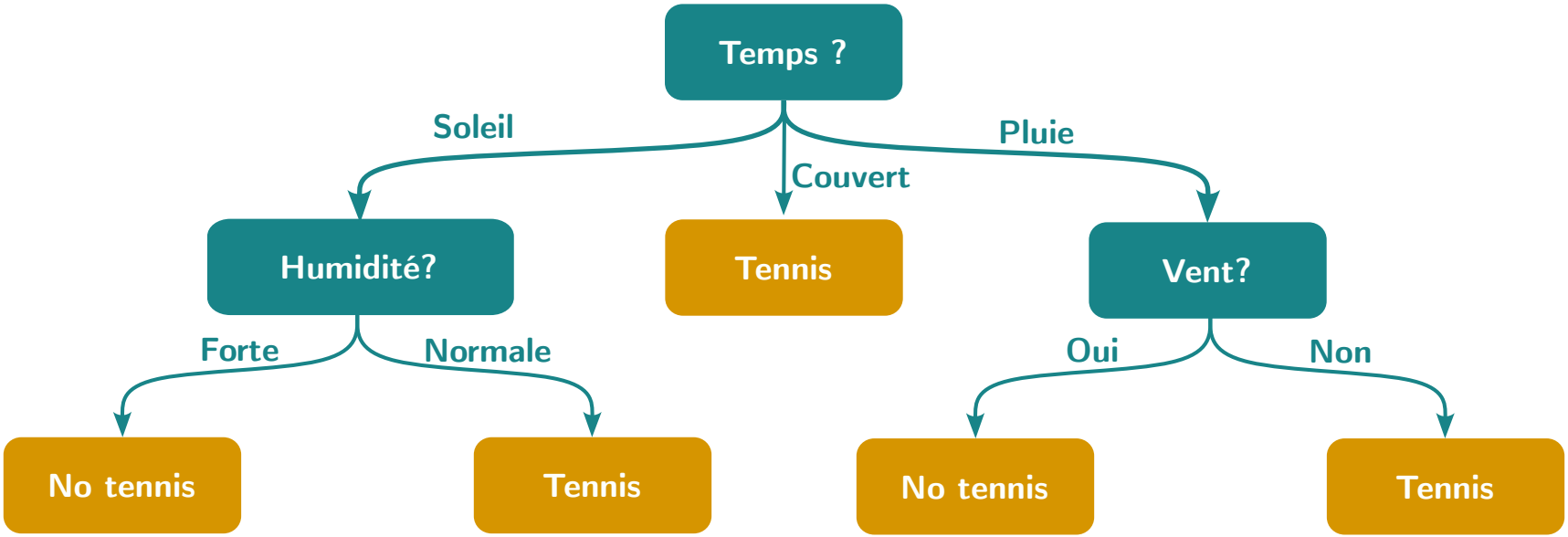
Jour	Tempér	Humidi	Vent	Classe
1	Chaude	85	Non	No tennis
2	Chaude	90	Oui	No tennis
8	Douce	95	Non	No tennis
9	Fraiche	70	Non	Tennis
11	Douce	70	Oui	Tennis

Jour	Tempér	Humidi	Vent	Classe
3	Chaude	78	Non	Tennis
7	Fraiche	65	Oui	Tennis
12	Douce	90	Oui	Tennis
13	Chaude	75	Non	Tennis

Jour	Tempér	Humidi	Vent	Classe
4	Douce	96	Non	Tennis
5	Fraiche	80	Non	Tennis
6	Fraiche	70	Oui	No tennis
10	Douce	80	Non	Tennis
14	Douce	80	Oui	No tennis

Apprentissage à partir des données (13/13)

L'arbre final est alors:



Conclusions

Avantages

- Modèle
  - interprétable: Arbre = modèle graphique de décisions
  - peu d'hypothèses préalables
- Feature selection
  - Adaptée au cas où les descripteurs sont nombreux
  - permet l'utilisation de descripteurs de tous types (Continues, discrètes, catégorique)
- Data
  - Peu de perturbation des individus extrêmes (Isolés dans des petites feuilles)
  - Efficace pour un nombre important d'individus
  - performances correctes

Inconvénients

- Modèle:
  - Frontières linéaires de décision
- Feature:
  - Utilisation des descripteurs non simultanée mais séquentielle (voir multivariable decision tree)
- Complexité:
  - Temps de calculs importants (Recherche des critères de division)
- **Sur-apprentissage possible**

Sur apprentissage ou overfitting (1/1)

- Problème du sur-apprentissage
  - Apprendre trop bien les données d'apprentissage
  - Avoir un modèle trop complexe pour les données d'apprentissage
- Pour un problème de classification à deux classes et deux descripteurs

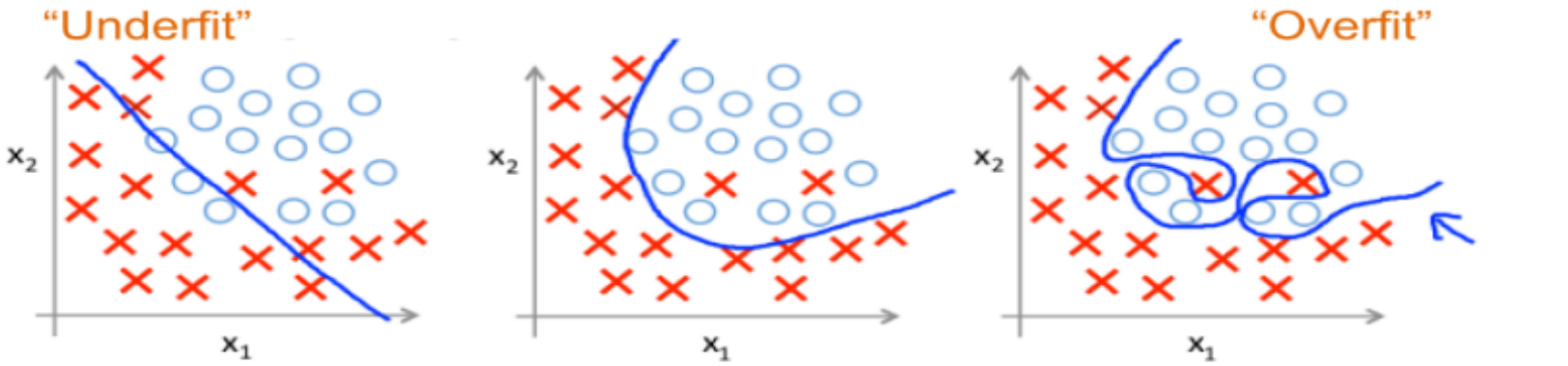


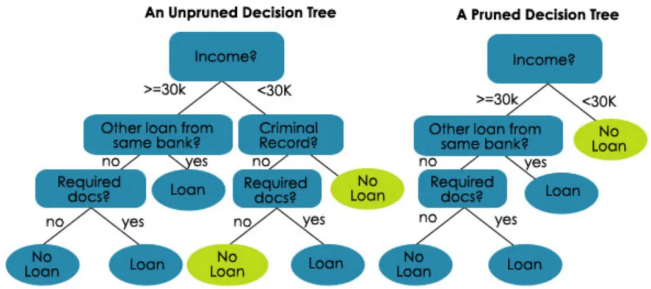
Image from Andrew Ng

- Savoir si on est face à du sur-apprentissage nécessite au minimum d'avoir deux ensembles de données:
  - un pour évaluer la capacité d'apprentissage du modèle
  - un pour évaluer la capacité de généralisation du modèle

Limiter le sur-apprentissage pour les arbres de décisions (1/2)

Plusieurs techniques possibles:

- **Pruning (elagage)**



- supprimer branches ou nœuds qui ne contribuent pas beaucoup aux performances de l'arbre.
- avant (prepruning) ou après (postpruning) la construction complète de l'arbre
- prepruning: contraintes lors de la constuction de l'arbre (par exemple limiter la profondeur, le nombre de nœuds ou le nombre minimum d'échantillons requis pour une division)
- post-pruning: critères tels que le gain d'information, le taux d'erreur ou l'intervalle de confiance.

Vous pourrez analyser pendant le TP, les options `max_depth`, `min_samples_split`, `min_samples_leaf`, `min_weight_fraction_leaf`, `max_leaf_nodes`, `min_impurity_decrease` de l'algorithm `DecisionTreeClassifier` de la bibliothèque `scikit-learn`

Limiter le sur-apprentissage pour les arbres de décisions (2/2)

Plusieurs techniques possibles:

- **méthodes d'ensemble**:
  - combiner plusieurs arbres de décision en un seul modèle
  - peut contribuer à améliorer la précision et la robustesse du modèle
  - en réduisant le biais (erreurs d'apprentissage) et la variance (erreurs de généralisation) des arbres individuels

Apprentissage supervisé

Random forests

Random forests

Le modèle des random forests

- est un classifieur de type “ensemble”
- qui regroupent un grand nombre de (d’arbres de) décisions.
- permet une prédiction/inférence obtenue par vote majoritaire sur l’ensemble des prédictions des arbres de décisions
- combine l’idée de **sélection de descripteurs et de règles de décisions** et de **Bagging** de Breiman.

Bagging

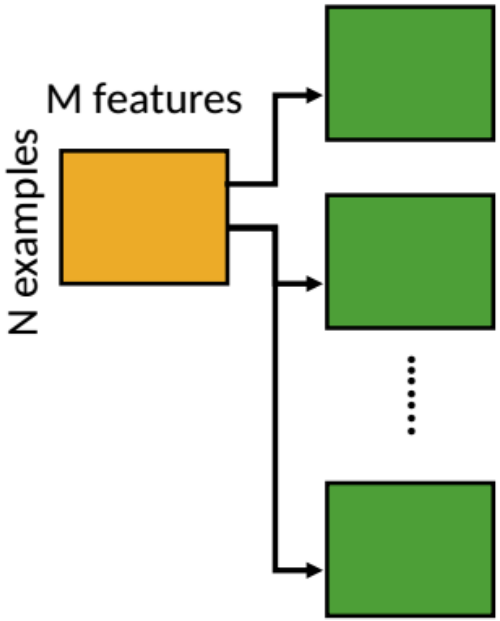
Le Bagging (ou agrégation de type bootstrap)

- technique pour réduire la variance de la prédiction (i.e. améliorer la capacité de généralisation du classifieur) .
- Pour la classification, un ensemble d'arbres réalise un vote pour prédire la classe.
- Entraînement:
  - Soit un ensemble de données S, à chaque itération i, un ensemble d'entraînement S_i est échantillonné par remplacement à partir de S (i.e. bootstraping)
  - Un classifieur C_i est appris pour chaque S_i
- Classification: Soit un exemple non vu X,
  - Chaque classifieur C_i renvoie une prédiction de classe
  - Le classifieur H comptent les votes et assigne la classe majoritaire à X.

Random Forest Classifier

Supposons qu'on ait un ensemble de données d'apprentissage (N exemples x M descripteurs)

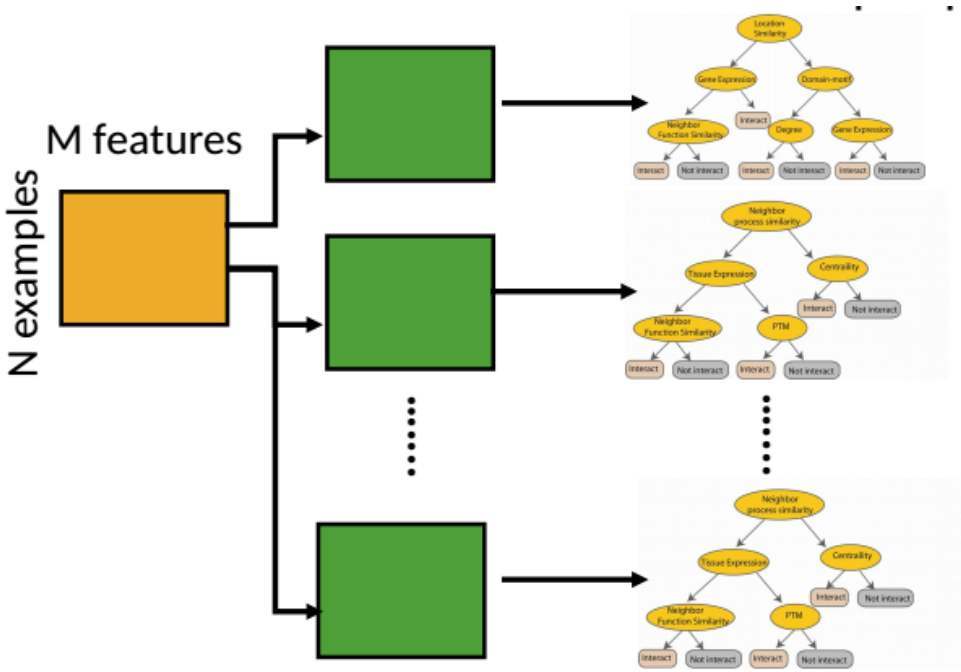
Random Forest Classifier



Génération des données

- tirer aléatoirement des ensembles de données avec remplacement à partir des données d'apprentissage
- chaque ensemble généré ayant généralement la même taille que l'ensemble de formation original (N exemples x M descripteurs)

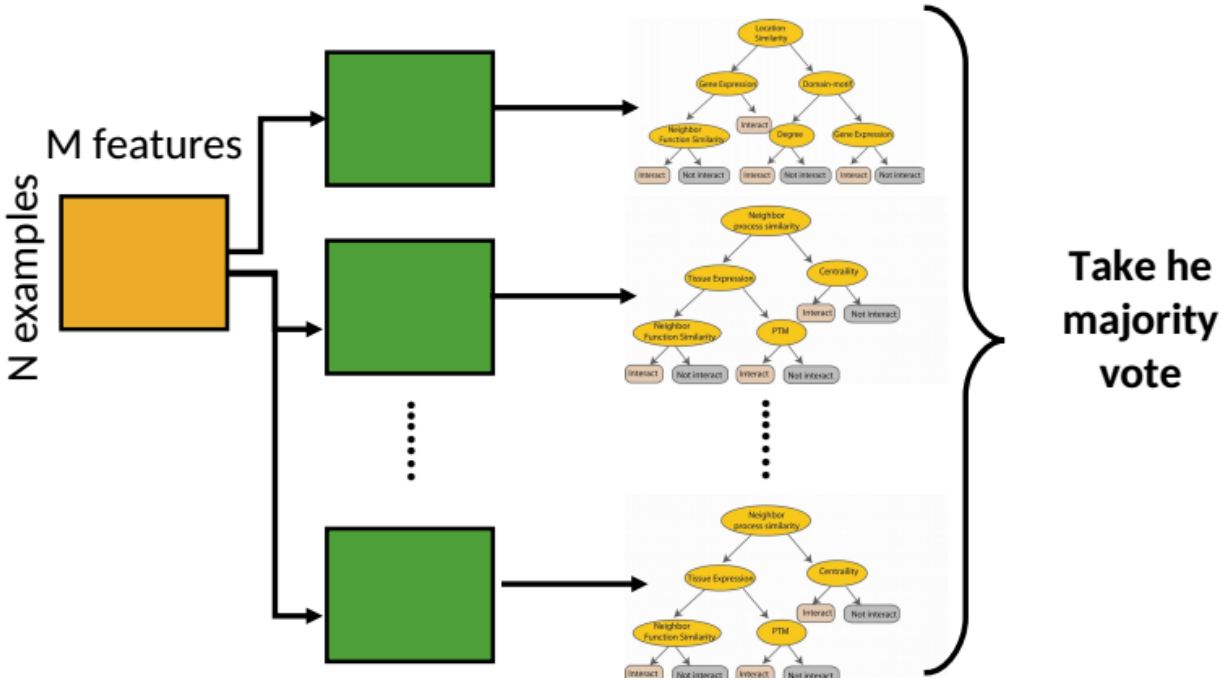
Random Forest Classifier



Apprentissage à partir des données

- Apprendre les arbres de décisions pour chaque ensemble de données

Random Forest Classifier



Classement de nouvelle donnée

- pour 1 exemple, le nombre d'arbres appris donne le nombre de décisions proposé
- la décision finale sur la classe pour un exemple est donnée par vote majoritaire

Conclusions

- arbres de décisions et random forests : outils puissants de machine learning
- arbres de décisions: performances correctes mais surtout utile pour l'interprétation des décisions
- randoms forests: améliore sensiblement les performances
  - algorithme même meilleur que les réseaux de neurones pour les données sans structure ou temporelles (signaux, vidéos) ou spatiales (images, vidéos)

TP5 Seabed Classification (1/1)

Mettre en pratique!

- Toutes les étapes:
  - preprocessing: normalisation, encodage des labels, etc.
  - apprentissage des classifieurs et estimation des hyperparamètres
  - évaluation des performances
- utilisation de scikit-learn
- utilisation jupyter notebook (conda install+tuto) et éventuellement de google colab (tuto)