Données multimédia et spatio-temporelles (NFE205)

Introduction à l'apprentissage statistique

Michel Crucianu
(prenom.nom@cnam.fr)

Département Informatique
Conservatoire National des Arts & Métiers, Paris, France

1 octobre 2021

Plan du cours

Quelques exemples

- Prédiction de la densité du trafic automobile
- Prédiction de la crue d'un cours d'eau
- Prédiction de volume de ventes
- Prédiction du nombre d'arrêts d'abonnements à un service
- Prédiction du cours d'une matière première (ou autre actif)
- Prédiction du classement d'un film dans l'ordre de préférence d'un abonné
- Prédiction : un prêt bancaire sera remboursé ou non ?
- Prédiction de la pollution de l'océan à 10 mètres de profondeur
- Prédiction : cette partie d'image représente un piéton, un poteau ou une chaise ?
- Prédiction : quel sera le comportement du piéton ?

→ Sens élargi du mot « prédiction » : concerne un fait non observé car à venir ou non observable/caché au système

Un exemple

- Comment un banquier du dix-septième siècle décidait-il d'accorder un prêt à un armateur pour monter une expédition ?

  Décision d'accorder le prêt si prédiction de bon remboursement
  - Que prédit le devin ? ☺
  - L'armateur a de quoi garantir le prêt en cas d'échec de l'expédition ?
  - L'armateur a déjà bien remboursé un autre prêt ?
  - L'armateur a un navire récent avec un capitaine expérimenté ?
  - (Un peu) plus complexe : si l'armateur n'a pas de quoi garantir 100% du prêt mais a déjà bien remboursé un autre prêt et a un navire récent avec un capitaine expérimenté, a-t-il de quoi garantir 50% du prêt ?
  - ...

  ! Envisageable de prendre en compte plus de facteurs et établir des règles plus complexes, mais au-delà d'un certain niveau de complexité un traitement algorithmique devient nécessaire

Comment décider (2)

- Comment obtenir une règle permettant de décider (ici, d'accorder ou non un prêt) ?
  1. A partir d'une parfaite compréhension du problème
     - Exemple : prêt accordé si le débiteur donne des gages à hauteur de 100% du montant
     - On ne peut avoir une « parfaite compréhension » que si le nombre de facteurs considérés est très (trop ?) réduit et des aléas exclus
     - Les règles résultantes peuvent être très limitatives (ex. prêt rarement accordé)
  2. A partir de données : constats réalisés sur un (grand, si possible) nombre de cas antérieurs, pour lesquels on connaît à la fois les valeurs prises par les facteurs qui comptent (par ex. taux de garantie, prêts déjà remboursés, âge du capitaine) et l'issue (remboursement ou défaillance)
     - « Facteurs qui comptent » : variables explicatives (ou prédictives, ou d'entrée)
     - Issue : variable expliquée (ou prédite, ou de sortie)

Expédition	Taux de garantie	Age du capitaine	Remboursement
Colomb	10	35	oui
Surcouf 1	10	30	non
Surcouf 2	40	31	oui
⋮	⋮	⋮	⋮

Prédiction et décision

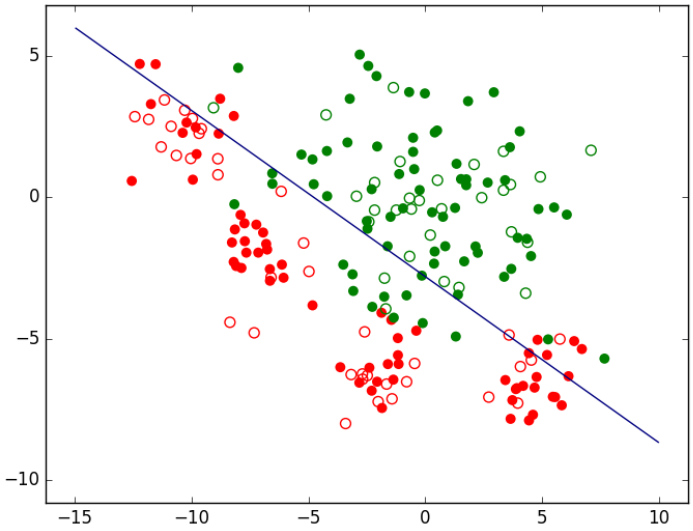
- En général, une décision se base sur une prédiction
  - On décide d'accorder un prêt si on prévoit son bon remboursement
  - On décide d'acheter un actif si on prévoit une hausse de son cours
- Le passage de prédiction à décision n'est pas toujours simple
  - La prédiction peut être un score (valeur entre 0 et 100%) pour une décision binaire (par ex. issue de la comparaison du score à un seuil)
  - La décision peut être humaine à partir d'une prédiction obtenue par une règle ou un algorithme de calcul
- Malgré cela, on parle souvent indifféremment de modèle prédictif ou décisionnel
- Par modèle on entend une règle ou un algorithme plus complexe qui, à partir des valeurs (observées) des variables explicatives, prédit une valeur (non observée) pour la variable expliquée

Types de problèmes de prédiction

1. Classification (ou discrimination, plus traditionnellement appelé « classement » : classer dans une « case ») : la variable expliquée est une variable nominale, chaque observation possède une modalité (appelée en général classe)
   - Exemple : le prêt sera remboursé ou non, une partie d'image représente une personne ou un poteau ou une chaise, etc.
2. Régression : la variable expliquée est une variable quantitative (domaine $\subset \mathbb{R}$)
   - Exemple : densité de trafic, volume de ventes, cours d'un actif
3. *Ranking* (« classement » en français, possibilité de confusion avec classification ci-dessus) : la variable expliquée est une variable ordinale
   - Exemple : classement d'un article dans l'ordre de préférence d'un utilisateur
4. Prédiction structurée : la variable expliquée prend des valeurs dans un domaine de données structurées (les relations entre parties comptent)
   - Exemple : extraire une entité nommée d'une phrase

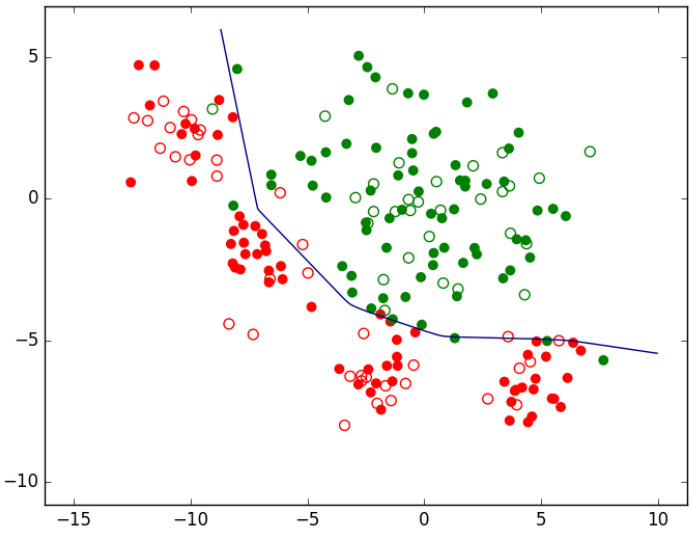
Classification

- Modèle : règle (ou algorithme) de classification, par ex. frontière de discrimination (trait bleu foncé)
- Par ex., pour chaque observation, 2 variables explicatives : taux de garantie en abscisse (X) et âge du capitaine en ordonnée (Y)



Classification

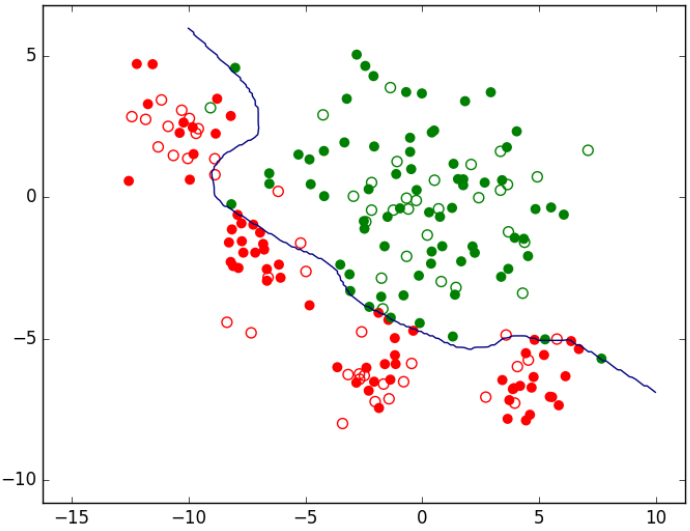
- Modèle : règle (ou algorithme) de classification, par ex. frontière de discrimination (trait bleu foncé)
- Par ex., pour chaque observation, 2 variables explicatives : taux de garantie en abscisse (X) et âge du capitaine en ordonnée (Y)



Prédiction : pourquoi et comment | Types de problèmes de prédiction 7 / 56

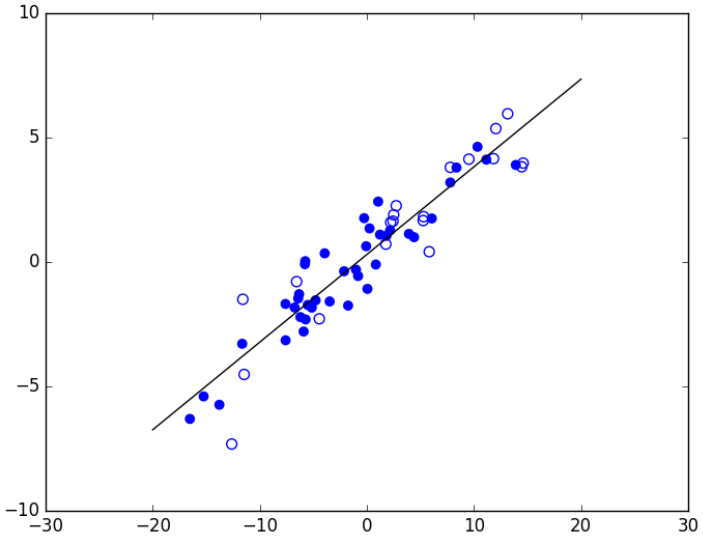
Classification

- Modèle : règle (ou algorithme) de classification, par ex. frontière de discrimination (trait bleu foncé)
- Par ex., pour chaque observation, 2 variables explicatives : taux de garantie en abscisse (X) et âge du capitaine en ordonnée (Y)



Régression

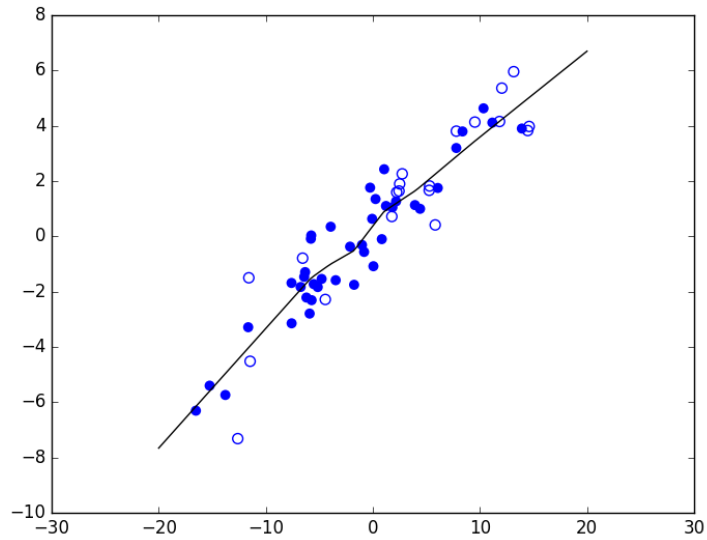
- Modèle : règle de prédiction (trait noir dans la figure)
  - Par ex. $y = ax + b$ pour modèle linéaire
- Par ex., pour chaque observation : taux de garantie comme variable explicative en abscisse (X), taux de remboursement comme variable expliquée en ordonnée (Y)



Prédiction : pourquoi et comment | Types de problèmes de prédiction 8 / 56

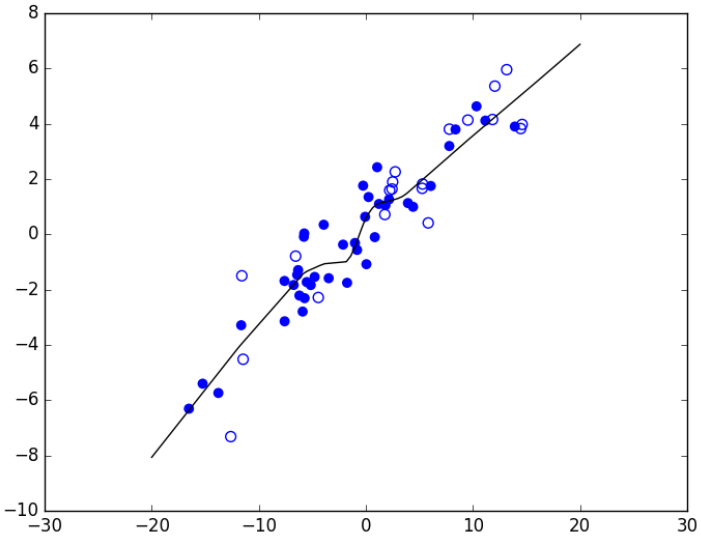
Régression

- Modèle : règle de prédiction (trait noir dans la figure)
- Par ex., pour chaque observation : taux de garantie comme variable explicative en abscisse (X), taux de remboursement comme variable expliquée en ordonnée (Y)



Régression

- Modèle : règle de prédiction (trait noir dans la figure)
- Par ex., pour chaque observation : taux de garantie comme variable explicative en abscisse (X), taux de remboursement comme variable expliquée en ordonnée (Y)



Prédiction structurée

- Modèle : règle de prédiction
- Exemples :
  1. Déterminer que dans la phrase « La Maison Blanche a démenti ces informations. » il y a une entité nommée qui est La Maison Blanche
     - L'inclusion d'un mot dans une entité nommée est liée à celle des autres mots composant l'entité

Prédiction structurée

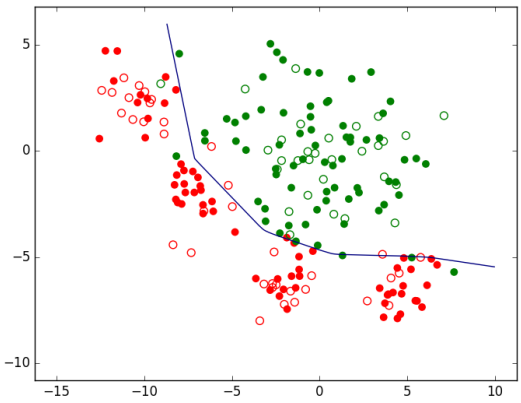
- Modèle : règle de prédiction
- Exemples :
  1. Déterminer que dans la phrase « La Maison Blanche a démenti ces informations. » il y a une entité nommée qui est La Maison Blanche
     - L'inclusion d'un mot dans une entité nommée est liée à celle des autres mots composant l'entité
  2. Délimiter la région correspondant aux pantalons dans l'image

     - L'affectation d'un pixel à une classe est liée aux affectations des pixels voisins (et de certains plus éloignés)

Plan du cours

Qu'est-ce qu'un modèle prédictif ?

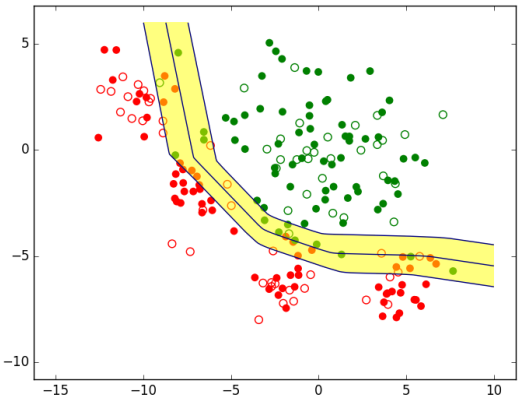
- Modèle = règle (ou algorithme) de prédiction (ou de décision)
- Exemple : frontière de discrimination pour problème de classement à 2 classes



Modélisation prédictive à partir de données | Qu'est-ce qu'un modèle prédictif | 10 / 56

Qu'est-ce qu'un modèle prédictif ?

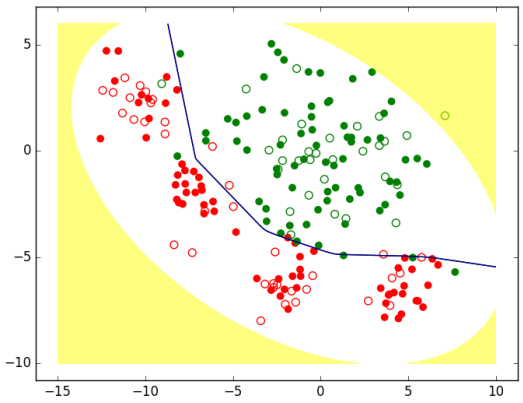
- Modèle = règle (ou algorithme) de prédiction (ou de décision)
- Exemple : frontière de discrimination pour problème de classement à 2 classes



- Éventuellement complété par des critères de rejet (refus d'affectation)
  1. Refus de classer les données trop proches de la frontière (rejet d'ambiguïté)

Qu'est-ce qu'un modèle prédictif ?

- Modèle = règle (ou algorithme) de prédiction (ou de décision)
- Exemple : frontière de discrimination pour problème de classement à 2 classes



- Éventuellement complété par des critères de rejet (refus d'affectation)

2. Refus de classer les données trop éloignées des données connues (rejet de non représentativité)

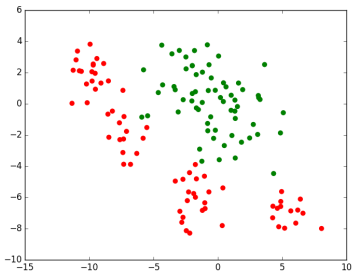
Construction analytique ou à partir des données ?

1. Construction analytique, à partir d'une parfaite connaissance du phénomène
   - Exemple : le prêt sera remboursé si le débiteur donne des gages à hauteur de 100% du montant

   → Peut être très contraignant, même inapplicable, si on se limite aux cas où l'issue souhaitée doit être certaine

   → Néglige souvent l'impact de variables non contrôlables (par ex. la valeur des terres agricoles données en gages diminue suite à une inondation)
2. A partir de données passées : ensemble d'observations pour lesquelles les valeurs des variables explicatives et des variables expliquées sont connues

   → Apprentissage supervisé : à partir d'observations étiquetées, c'est à dire pour lesquelles les valeurs de la variable expliquée sont également connues



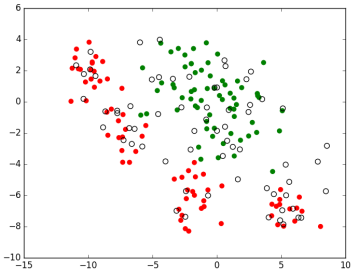
Construction analytique ou à partir des données ?

1. Construction analytique, à partir d'une parfaite connaissance du phénomène
   - Exemple : le prêt sera remboursé si le débiteur donne des gages à hauteur de 100% du montant

   → Peut être très contraignant, même inapplicable, si on se limite aux cas où l'issue souhaitée doit être certaine

   → Néglige souvent l'impact de variables non contrôlables (par ex. la valeur des terres agricoles données en gages diminue suite à une inondation)
2. A partir de données passées : ensemble d'observations pour lesquelles les valeurs des variables explicatives et des variables expliquées sont connues

   → Apprentissage supervisé : à partir d'observations étiquetées, c'est à dire pour lesquelles les valeurs de la variable expliquée sont également connues



   - Apprentissage semi-supervisé : exploite aussi des observations pour lesquelles les valeurs de la variable expliquée sont inconnues

Étapes générales dans la construction de modèle

1. Préparation des données : préparation du recueil, recueil de données, nettoyage et vérification des données, compléter éventuellement les données manquants, transformation des données
2. Choix des objectifs et de la nature des erreurs à minimiser
3. Choix de la (ou des) technique(s) de modélisation : arbres de décision, machines à vecteurs de support (SVM), forêts aléatoires, réseaux de neurones, etc.
4. Construction du (des) modèle(s) : optimisation des paramètres suivant la méthode adaptée à la technique de modélisation
5. Sélection de modèle : si plusieurs modèles sont développés (différentes techniques, différentes valeurs de hyper-paramètres), choisir entre ces candidats
6. Estimation des performances futures qui peuvent être attendues du modèle sélectionné
7. Utilisation (et surveillance) du modèle pour prédire les valeurs de la variable expliquée pour de nouvelles données pour lesquelles seules sont connues les valeurs des variables explicatives

Préparation des données

- Préparation du recueil des données (si elles ne sont pas déjà recueillies) :
  - Quelles variables sont utiles (pertinence)
  - Où recueillir des données (couverture, représentativité)
  - Comment les recueillir (précision des mesures, fiabilité du processus)
  - S'assurer des droits d'utilisation des données
- Recueil des données : surveiller et éventuellement ajuster le recueil
- Nettoyage et vérification des données : uniformiser codage, vérifier distributions des valeurs des variables, vérifier pour un échantillon les valeurs de la variable expliquée
- Compléter les données manquantes ? Si certaines observations sont incomplètes (les valeurs de certaines variables manquent) il est envisageable de les estimer avant (ou éventuellement pendant) la modélisation
- Transformation des données : si on « soupçonne » une dépendance non linéaire spécifique (par ex. $y = k_1 \log(x)$), il est préférable de transformer les données (ici appliquer le $\log$) et ainsi chercher ensuite un modèle plus simple

Objectifs et fonction(s) d'erreur

- Au-delà de la performance du modèle, d'autres objectifs peuvent être
  - Faible coût de calcul pour chaque prédiction et/ou pour une mise à jour du modèle
  - Respect de critères de non discrimination
  - Lisibilité/explicabilité/interprétabilité du modèle : capacité pour un humain à comprendre comment chaque prédiction est faite
- Choix du (des) critère(s) de performance
  - Suivant la nature du problème : classification (par ex. taux d'erreur), régression (par ex. erreur quadratique)...
  - Suivant l'importance de chaque type d'erreur : précision (faux positifs coûteux), rappel (faux négatifs coûteux)
  - Déséquilibre des classes : simple taux d'erreur inadapté (par ex. si une classe correspond à 5% des données, en prédisant toujours l'autre classe on obtient un taux d'erreur de seulement 5%)
- Le critère à optimiser peut être une combinaison entre plusieurs critères mentionnés, auxquels s'ajoute un terme de régularisation afin de maîtriser la complexité du modèle résultant

Techniques de modélisation

- Grande diversité, domaine en évolution permanente (et rapide)
- Quelques critères de choix de la (des) technique(s) de modélisation candidate(s)
  - Adéquation à la complexité du problème
  - Accès (et coût d'accès) aux compétences spécifiques
  - Exigences d'explicabilité/interprétabilité du modèle résultant
  - Volume de données étiquetées et coût de calcul exigés par la construction d'un modèle
  - Disponibilité sur la plate-forme de développement employée
  - Sans oublier : avec une technique qu'on maîtrise bien on peut obtenir de meilleurs résultats qu'avec une nouvelle technique qu'on découvre
- Les techniques les plus performantes sont basées sur le développement automatique, lors de l'apprentissage, de nouvelles représentations pour les données (apprentissage de représentations), mais leur application est souvent conditionnée par la disponibilité d'un très grand nombre d'observations étiquetées
- Chaque modèle est défini par les valeurs que prennent ses paramètres
  - Par ex., le modèle $y = ax + b$ qui prédit le taux de remboursement y à partir du taux de garantie x est défini par les valeurs de a et b

Construction d'un modèle

- Étape qui consiste à trouver les paramètres du modèle qui permettent d'optimiser le critère retenu, qui inclut au moins un critère de performance
- Technique de modélisation + critère à optimiser $\rightarrow$ choix algorithme d'optimisation
  - Dans de nombreux cas, c'est la descente de gradient :

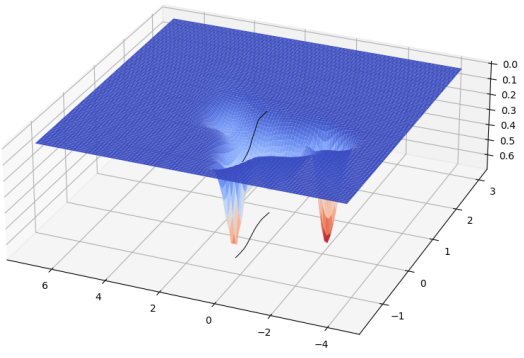
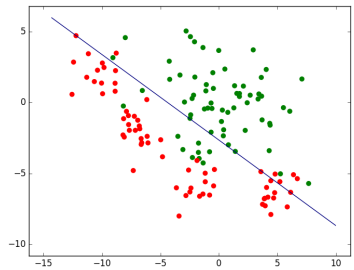
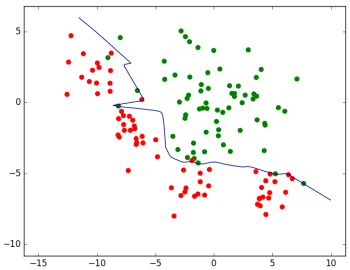
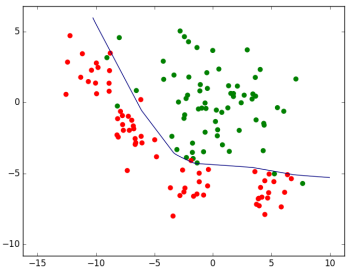


FIG. – Cas simple où les paramètres à optimiser sont représentes par les axes du plan horizontal et la valeur du critère à optimiser par l'axe vertical. La descente de gradient est un processus itératif où on applique aux paramètres, à chaque itération, une modification dans la direction opposée au gradient du critère par rapport aux paramètres. La trajectoire dans le plan représente les itérations successives.

Construction d'un modèle (2)

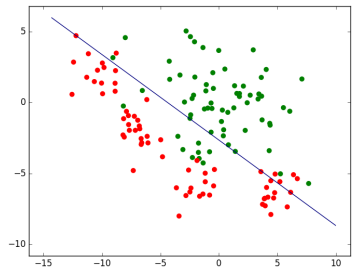
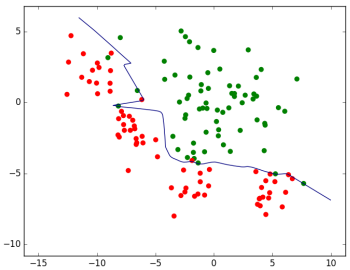
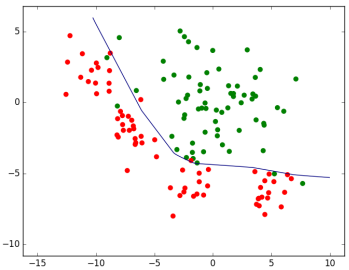
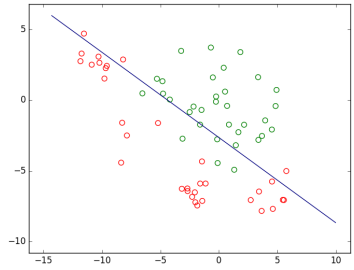
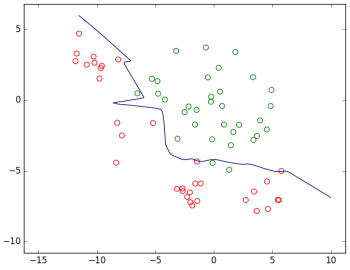
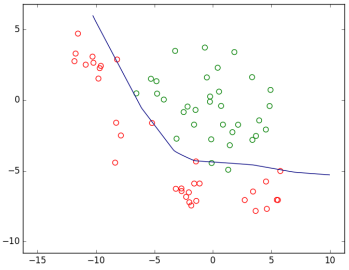
- Le critère peut être mesuré et optimisé sur les observations passées (apprentissage)
- Mais on vise les meilleures performances sur les observations futures (généralisation)

	Modèle 1	Modèle 2	Modèle 3
Apprentissage			
	Erreur app. 12%	2,3%	4,5%

Modélisation prédictive à partir de données | Comment trouver le modèle | 17 / 56

Construction d'un modèle (2)

- Le critère peut être mesuré et optimisé sur les observations passées (apprentissage)
- Mais on vise les meilleures performances sur les observations futures (généralisation)

	Modèle 1	Modèle 2	Modèle 3
Apprentissage			
	Erreur app. 12%	2,3%	4,5%
Test			
	Erreur test 14%	6%	4,6%

Construction d'un modèle (3)

- Observations disponibles employées :
  - Apprentissage : obs. étiquetées passées sur lesquelles on optimise le critère choisi
  - Test : autres obs. étiquetées passées sur lesquelles on estime les performances futures
- Constats
  - Modèle 1 : trop « simple », a une erreur d'apprentissage élevée : sous-apprentissage
  - Modèle 2 : trop « complexe », erreur d'apprentissage la plus faible mais l'erreur de test est sensiblement supérieure : surapprentissage
  - Modèle 3 : l'erreur de test est la plus faible même si l'erreur d'apprentissage (proche de celle de test) n'est pas la plus faible

  → Le modèle qui minimise l'erreur d'apprentissage n'est pas celui qui généralise le mieux

→ Important de maîtriser la complexité du modèle, en général à travers des techniques de régularisation
  - Terme de régularisation introduit dans le critère à optimiser (en plus du terme d'erreur)
  - Le poids relatif de ce terme est critique et doit être bien choisi

Sélection de modèle

- Sur les mêmes observations d'apprentissage on obtient plusieurs modèles
  - Emploi de plusieurs techniques de modélisation qui satisfont les critères de choix
  - Pour chaque technique, plusieurs valeurs différentes des hyper-paramètres (par ex. le poids du terme de régularisation) → plusieurs modèles
    - Exploration de l'espace des hyper-paramètres : « en grille » ou aléatoire à budget donné
- Quel modèle choisir pour utilisation ultérieure ?
  - Le modèle qui présente l'erreur de test la plus faible ⇒ ce n'est plus possible d'estimer ses performances de généralisation
    - En servant la sélection, ces observations ont contribué à trouver le modèle, estimer la généralisation à partir de ces mêmes observations donnerait un résultat optimiste

  → On met de côté un autre ensemble d'observations passées étiquetées, l'ensemble de validation, et on choisit le meilleur modèle sur cet ensemble

  → On peut ensuite se servir des observations de test pour estimer les performances de généralisation du modèle sélectionné

Estimation des performances futures et utilisation du modèle

- Observations étiquetées passées : 3 ensembles disjoints
  1. Apprentissage : modification paramètres modèle pour optimiser critère choisi
  2. Validation : choisir entre plusieurs modèles candidats
  3. Test : estimer les performances de généralisation du modèle retenu
  - Les résultats avec un seul partitionnement peuvent varier $\rightarrow$ validation croisée
- En général, la distribution des données et parfois la nature même du phénomène évoluent dans le temps
  - Certaines données deviennent plus fréquentes, au dépens d'autres données (changement de distribution)
  - Les intervalles de variation de certaines variables changent
  - De nouvelles variables interviennent, d'autres perdent en importance

  $\Rightarrow$ surveiller la performance du modèle et opérer les ajustements nécessaires

Plan du cours

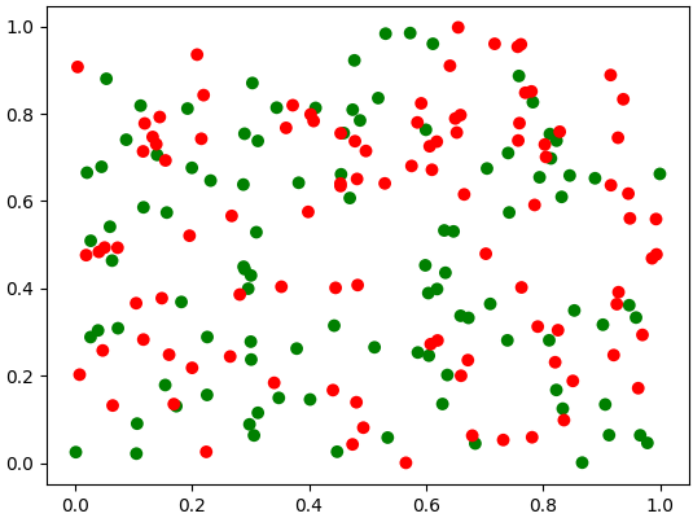
Difficultés posées par les données

- Au-delà de la difficulté intrinsèque d'un problème, c'est la qualité des données qui limite les performances des modèles obtenus à partir de ces données
- Quelques types de difficultés :
  1. Données inadaptées
  2. Données non représentatives
  3. Données aberrantes
  4. Données manquantes
  5. Classes déséquilibrées
  6. Nombre très élevé de variables

Données inadaptées

- Les observations ne sont pas informatives → aucun modèle utile ne peut en sortir (*garbage in, garbage out*)
- Exemple : quelle frontière de complexité limitée peut séparer les rouges des verts ?

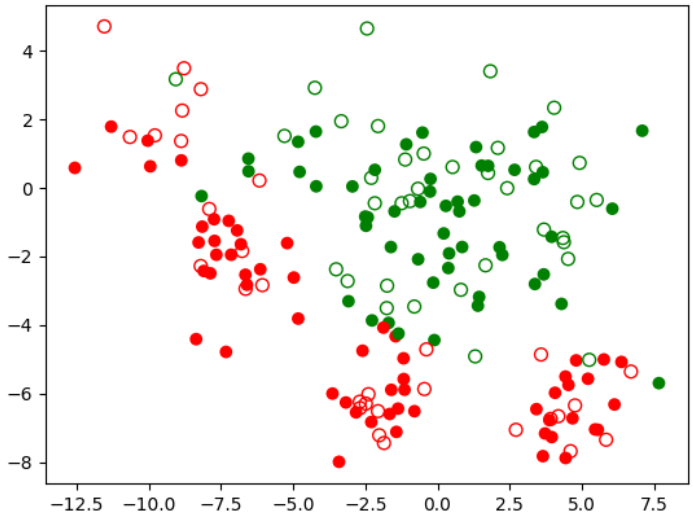
→ Nouvelle analyse du problème et nouvelle collecte



Données non représentatives

- Certains domaines utiles sont (très) peu couverts par les observations étiquetées
- Exemple : l'âge de départ à la retraite des capitaines de bateaux a été repoussé (âge en ordonnée, y), pour les nouvelles observations (points creux) les valeurs y couvrent un intervalle plus large que pour les observations d'apprentissage (points pleins)

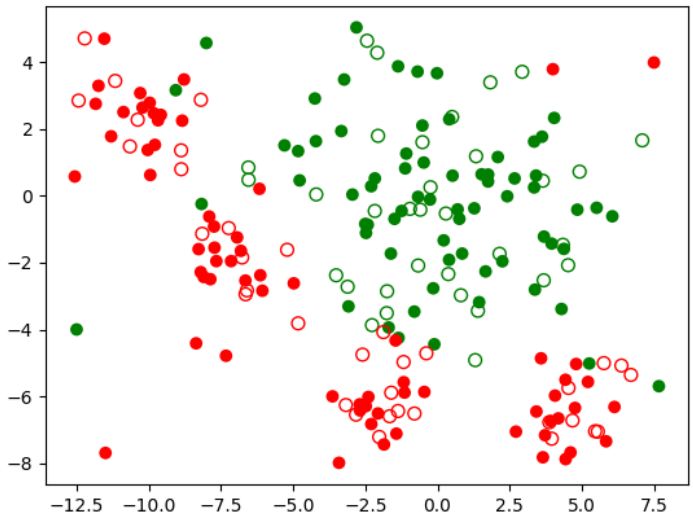
→ Compléter la collecte ; en attendant, limiter l'usage au domaine couvert



Données aberrantes

- Données éloignées des autres ou seulement de celles de leur classe
- Exemples : points ● en haut à droite et en bas à gauche, point ● en bas à gauche
- Erreurs de mesure/enregistrement/étiquetage ou phénomène significatif ?

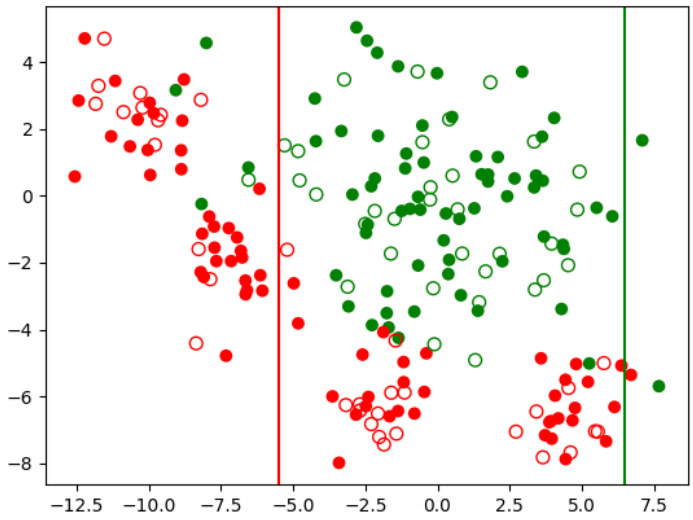
→ Détecter et corriger (ou ignorer) si erreurs, modéliser si phénomène significatif



Données manquantes

- Pour certaines observations, les valeurs de certaines variables manquent
- Exemple : capitaines qui refusent de communiquer leur âge (observations représentées par lignes verticales car seules les abscisses et étiquettes sont connues)

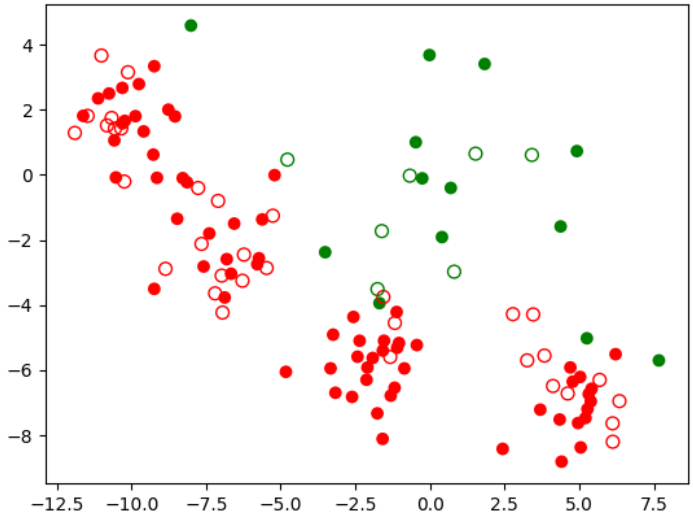
→ Ignorer observations incomplètes ou plutôt estimer (imputer) valeurs manquantes ?



Classes déséquilibrées

- Une classe contient nettement moins d'observations que l'autre → sans contre-mesures, la classe minoritaire est mal traitée

→ Sur-pondérer classe minoritaire, sous-échantillonner classe majoritaire, générer des observations synthétiques pour la classe minoritaire (par ex. avec SMOTE), reformuler le problème comme une détection d'*outliers*



Nombre très élevé de variables d'entrée

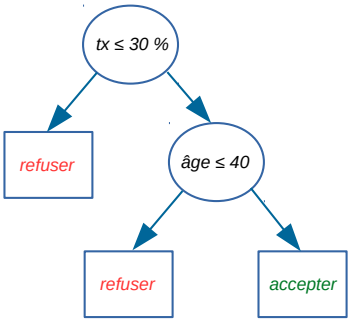
- Difficultés liées à la « malédiction de la dimension » (déjà mentionnées)
  - Proximité des hypersurfaces externes → faible sélectivité des requêtes par similarité
  - « Concentration des mesures » (toutes les observations à ∼égale distance les unes des autres) → inefficacité des indexes multidimensionnels, perte de signification de la similarité
- Difficultés de construction d'un (bon) modèle
  - Le nombre de variables d'entrée (ou explicatives) est un facteur de la complexité du modèle → avec trop de variables il peut être difficile d'éviter le surapprentissage !

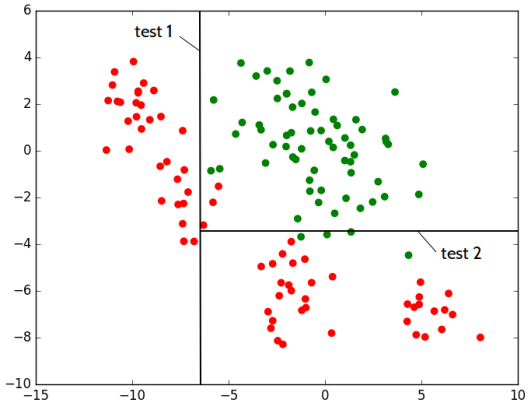
Plan du cours

Arbres de décision

- Idée : obtenir la prédiction (ou décision) par l'application d'une succession de règles organisées hiérarchiquement
- Nœuds :
  - Racine, intermédiaires : test appliqué à la valeur prise par une variable explicative ; les branches correspondent aux résultats possibles du test
  - Feuilles : nœuds terminaux correspondant aux prédictions
- Exemple (accorder ou non un prêt pour nouvelle expédition) :

```
test1 : taux de couverture <= 30%
  si vrai alors refuser prêt
  sinon test2 : âge capitaine <= 40 ans
    si vrai alors refuser prêt
    sinon accepter prêt
```



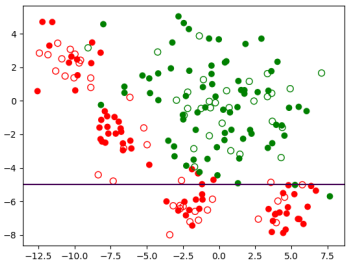


Arbres de décision : construction à partir des données

- Principe de construction
  - Partition hiérarchique données apprentissage : un test appliqué dans chaque nœud
  - Choix à chaque étape de l'attribut (= variable explicative) et du test effectué : optimisation de la séparation entre les classes
  - Arrêt : toutes les feuilles sont « pures » ou la profondeur maximale est atteinte
- Critères employés pour le choix des attributs dans les nœuds
  - Entropie : mesure la capacité d'un attribut à séparer les classes ; plus l'entropie est faible, mieux l'attribut sépare
  - Gain d'information : mesure l'amélioration de la séparation entre classes que peut apporter un attribut ; plus le gain est élevé, meilleur est l'apport de l'attribut
  - Impureté de Gini : mesure la capacité d'un attribut à séparer les classes (variante de l'entropie) ; plus l'impureté est faible, mieux l'attribut sépare
  - Réduction de la variance (de la variable quantitative expliquée) : mesure la réduction de la variance que peut apporter un attribut

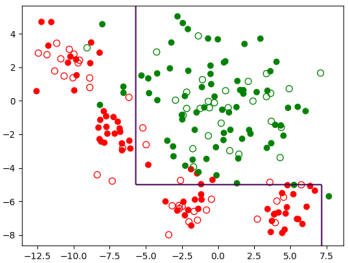
Arbres de décision : résultats illustratifs

Profondeur 1 :



Erreur test 33,33%

Profondeur 2 :

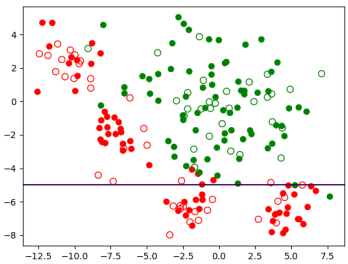


Erreur test 10%

30 / 56

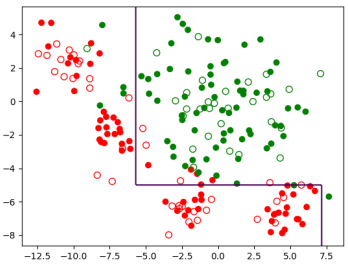
Arbres de décision : résultats illustratifs

Profondeur 1 :



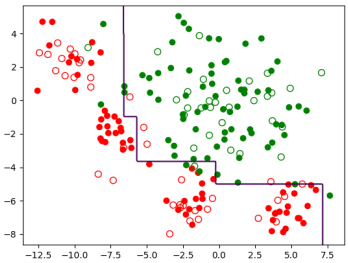
Erreur test 33,33%

Profondeur 2 :



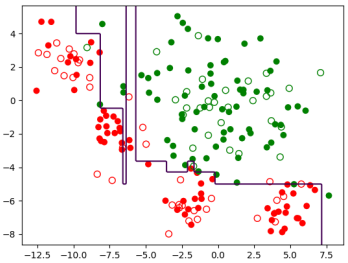
Erreur test 10%

Profondeur 4 :



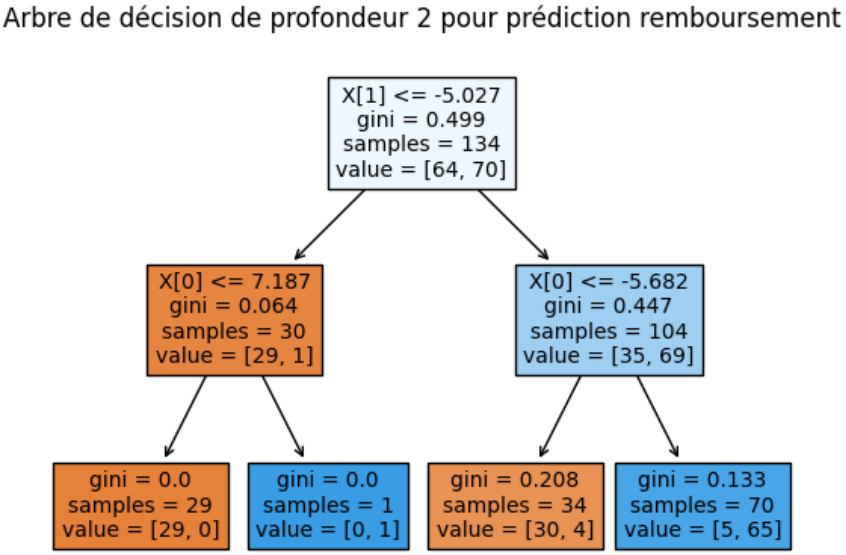
Erreur test 11%

Profondeur 8 :



Erreur test 10%

Arbres de décision : résultat illustratif pour profondeur = 2



- `X[i] <= …` : test effectué, la branche de gauche est pour résultat positif
- `gini` : valeur de l'indice de pureté de Gini
- `samples` : nombre d'observations d'apprentissage classées dans le nœud
- `value = […]` : nombre d'observations par classe
- Couleur : classe majoritaire dans le nœud ; la saturation indique la pureté

Arbres de décision : versions, extensions

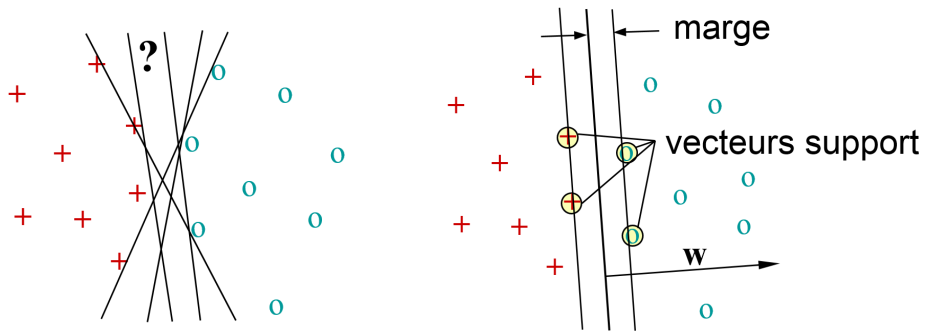
- Variantes les plus connues
  - ID3 (*Iterative Dichotomiser* 3) : seulement variables à valeurs discrètes, chaque variable est examinée à un seul niveau, un nœud peut avoir autant de branches qu'il y a de valeurs différentes pour la variable testée
  - C4.5 (successeur de ID3) : variables à valeurs discrètes ou continues (discrétisées dynamiquement), accepte les valeurs manquantes, ré-examen de l'arbre après construction en vue de sa simplification
  - CART (*Classification And Regression Tree*) : variables à valeurs discrètes ou continues ($\rightarrow$ recherche de seuils), peut traiter des problèmes de classification ou de régression, nœuds binaires
- Extensions :
  - Forêts aléatoires
    - Apprentissage d'un ensemble d'arbres de décision ($\rightarrow$ forêt) par double échantillonnage : sur variables explicatives, sur observations
    - Prédiction : vote majoritaire pour la classification, moyenne pour la régression
    - Inconvénient : perte de la facilité à comprendre les prédictions
  - *Gradient Boosted Trees* : ensemble d'arbres de décision construit incrémentalement, chaque nouvel arbre cherche à corriger les erreurs de l'ensemble précédent ; inconvénient : perte de la facilité à comprendre les prédictions

Arbres de décision : avantages et inconvénients

- Avantages
  - Peu de préparation nécessaire pour les données
  - Applicables directement à des variables quantitatives et nominales, pour des problèmes de classification ou de régression
  - Processus de prédiction facilement compréhensible (règles simples)
- Inconvénients
  - Algorithmes gloutons (*greedy*) basés sur des choix localement et non globalement optimaux
  - Approximation d'une frontière non linéaire par une succession de segments parallèles aux axes (en 2D)
  - Cas d'instabilité : modifications légères de certaines observations $\Rightarrow$ changements importants dans l'arbre
  - Certains problèmes sont difficiles à résoudre par une hiérarchie de tests

Machines à vecteurs de support (SVM)

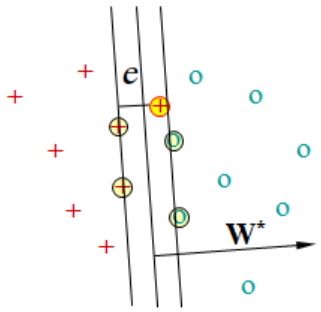
- *Support Vector Machines* (voir [5]) : « séparateurs à vastes marges » (pas les seuls « séparateurs » qui maximisent une marge !), « machines à vecteurs (de) support »
- Idée : lorsque plusieurs séparations (et même une infinité) sont possibles, préférer celle qui maximise la marge



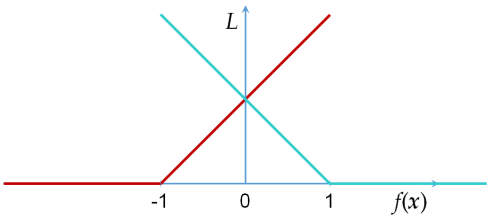
- Pourquoi chercher à maximiser la marge : meilleures garanties de généralisation (réduction démontrable de l'écart entre erreur d'apprentissage et erreur de test)

SVM : séparation linéaire

- Frontière linéaire : hyperplan d'équation $\mathbf{w}^T\mathbf{x} + b = 0$ ($\mathbf{w}$ étant le vecteur normal)



- Fonction d'erreur : pénalité dès qu'une donnée est au-delà du « bord » de sa classe (même si du bon côté de la frontière) $\rightarrow$ erreur « charnière » (*hinge*)



SVM : séparation linéaire (2)

- Critère à minimiser : compromis entre
  1. maximisation de la marge (terme 1) : terme de régularisation
  2. minimisation de l'erreur d'apprentissage (terme 2)

$$\min_{\mathbf{w},b} \left(\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} e_i \right)$$

  - Plus C est faible, plus la régularisation est forte
  - Minimisation par descente de gradient
- Fonction de décision résultante, f^*, appliquée à une observation $\mathbf{x}$ est

$$f^*(\mathbf{x}) = \mathbf{w}^{*T}\mathbf{x} + b^*$$

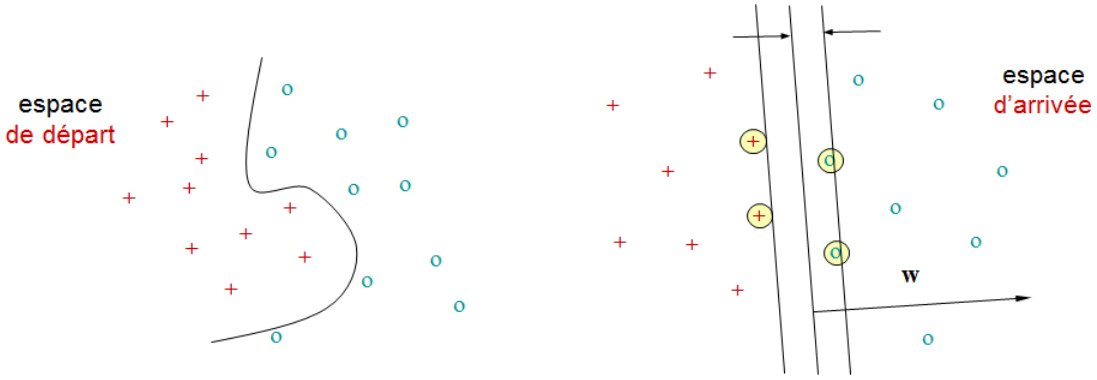
f^* s'annule sur la frontière de séparation ; * désigne les paramètres optimaux (et la fonction optimale) trouvés par résolution du problème d'optimisation

- Sorties binaires : appliquer fonction signe ou fonction de Heaviside à $f^*(\mathbf{x})$

SVM : séparation non linéaire

- Que faire si les données ne sont (vraiment) pas linéairement séparables ?

→ Transposer les données dans un autre espace (en général de dimension supérieure) dans lequel elles deviennent (presque) linéairement séparables : $\phi : \mathbb{R}^p \to \mathcal{H}$, $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = K(\mathbf{x}_i, \mathbf{x}_j)$

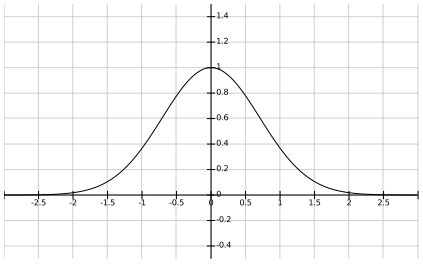


- Grâce à la fonction noyau K, tout calcul impliquant des produits scalaires dans l'espace d'arrivée peut être réalisé en utilisant K dans l'espace de départ

SVM : séparation non linéaire (2)

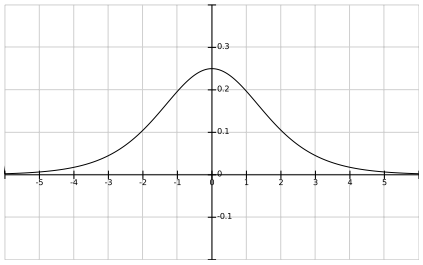
- Exemples de noyaux :

Gaussien (RBF)



$$\phi(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2}$$

Logistique



$$\phi(u) = \frac{e^{-|u|}}{\left(1+e^{-|u|}\right)^2}$$

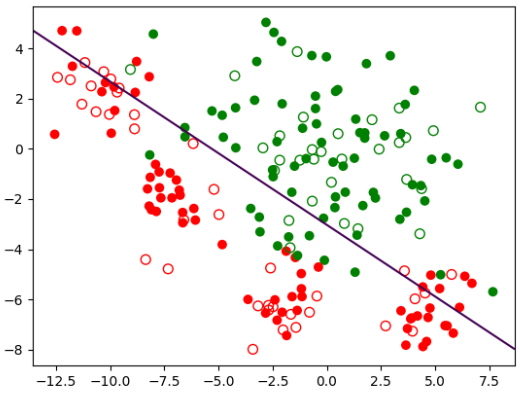
- L'optimisation dans l'espace d'arrivée peut devenir impossible ou peu pratique ⇒ optimisation quadratique sous contraintes d'inégalité dans l'espace de départ
- Fonction de décision résultante :

$$f^*(\mathbf{x}) = \sum_{i=1}^{v} \alpha_i^* y_i K(\mathbf{x}, \mathbf{x}_i) + b^*$$

$\mathbf{x}_i$ étant les vecteurs de support et v leur nombre

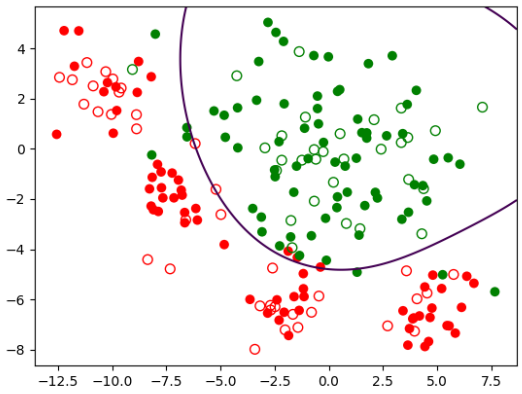
SVM : résultats illustratifs

Linéaire :



Erreur apprentissage 16.4%
Erreur test 10.6%

Noyau RBF :

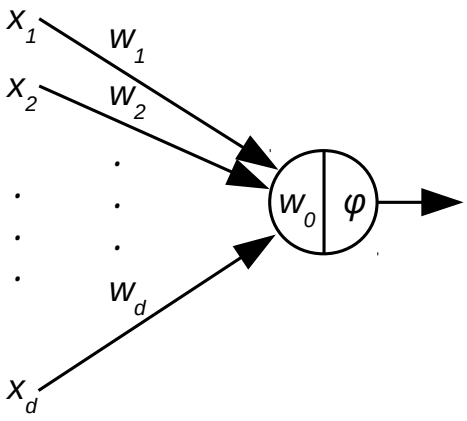


Erreur apprentissage 7.5%
Erreur test 4.5%

SVM : avantages et inconvénients

- Avantages
  - Résultats robustes en dimension élevée avec relativement peu d'observations
  - Généralité : applicables à tous types de problèmes (classification, régression, classement, etc.), utilisation directe de variables explicatives complexes grâce à la construction de noyaux adéquats
- Inconvénients
  - Cas non linéaire : coût potentiellement élevé de chaque prédiction (proportionnel au nombre de vecteurs support)
  - Prédictions en général opaques (hors cas linéaires particuliers)

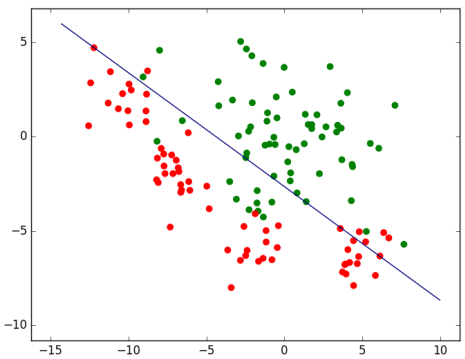
Neurone « formel »



- Introduit dans [3] comme modèle simplifié d'un neurone réel
  - Entrées $x_j, 1 \leq j \leq d$
  - Poids $w_j, 1 \leq j \leq d$, « seuil » w_0 ; w_j est le poids de la connexion avec l'entrée j
  - Fonction d'activation ϕ (dans [3] : fonction « marche » θ de Heaviside)
  - Sortie $\hat{y} = \phi\left(\sum_{j=1}^{d} w_j x_j - w_0\right)$ (fonction linéaire suivie par application de ϕ)

Perceptron et Adaline

- Apprendre les poids pour obtenir une association entrées $\rightarrow$ sortie désirée
  - Perceptron (Rosenblatt, 1957) : $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\,\mathbf{x}$ ssi $\mathbf{x}$ mal classé
  - Adaline [7] : $\mathbf{w}_{t+1} = \mathbf{w}_t + \eta(y - \hat{y})\mathbf{x}$ (règle obtenue en minimisant l'erreur quadratique par descente de gradient)
- Le Perceptron et l'Adaline permettent de résoudre seulement des problèmes linéairement séparables

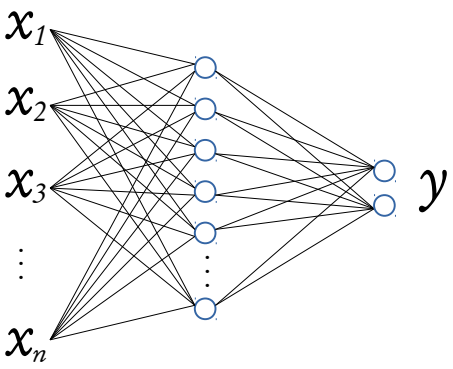


Perceptron multi-couches (PMC)

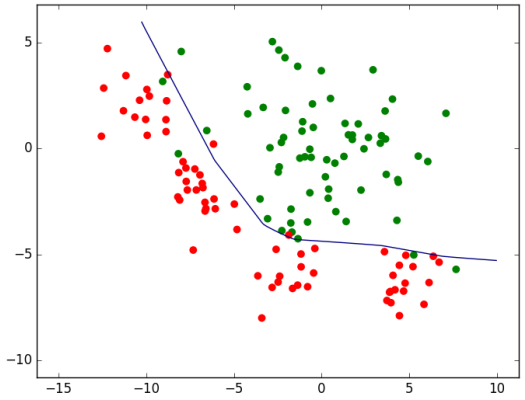
- Comment aller au-delà des séparations linéaires ?

→ En ajoutant une (ou plusieurs) couche(s) cachée(s)

- Les neurones d'au moins une des couches cachées doivent avoir des fonctions d'activation non linéaires
  - L'activation des neurones se propage de l'entrée vers la sortie



$\Rightarrow$



Perceptron multi-couches (2)

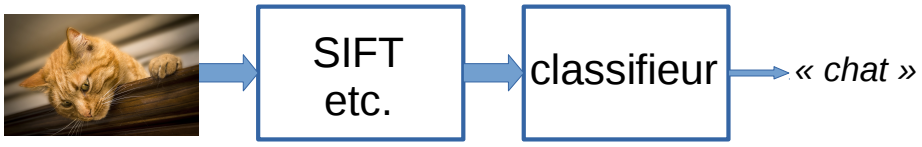
- Résultat d'approximation universelle : quelle que soit une fonction continue sur des compacts de $\mathbb{R}^n$, une approximation aussi bonne que souhaité peut être obtenue avec un PMC ayant un nombre fini de neurones dans la couche cachée et une fonction d'activation non polynomiale
- Erreur régularisée : $E(\hat{y}(\mathbf{w}), y) = L(\hat{y}(\mathbf{w}), y) + \alpha R(\mathbf{w})$, avec
  - L la fonction de « perte » (*loss*), par ex. $L(\hat{y}(\mathbf{w}), y) = (\hat{y} - y)^2$
  - R le terme de régularisation, par ex. $R(\mathbf{w}) = \|\mathbf{w}\|^2$ (appelé oubli), pondéré par α
- Comment apprendre les poids dans ce cas ?

→ Toujours par descente de gradient, en calculant le gradient (les dérivées partielles) de l'erreur régularisée en sortie E par rapport aux poids de toutes les couches

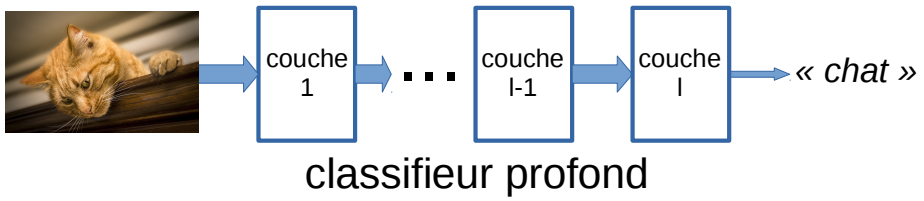
- Dérivation de fonction composée : $E = E(\hat{y}(\mathbf{w}), y) \Rightarrow \frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial \hat{y}} \frac{\partial y}{\partial w_{ij}}$ ...
- Passage d'une couche à la précédente : $\frac{\partial E}{\partial o_j} = \sum_{l \in \text{couche}+1} \frac{\partial E}{\partial o_l} \phi' w_{jl}$
  (le calcul des dérivées partielles est fait de la sortie vers l'entrée)
- A la fin, modification des poids par descente de gradient de la fonction d'erreur régularisée : $\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$, η étant le vitesse d'apprentissage

Représentations apprises *vs* représentations non apprises

- Approche classique : représentations *handcrafted* + classifieur
  - Représentations de bas niveau sémantique
  - Représentations qu'on ne peut pas optimiser par rapport à une tâche, des données



- Apprentissage profond : représentations apprises dans classifieur profond
  - Représentations dont le niveau sémantique progresse entre entrée et sortie
  - Représentations développées (donc optimisées) par apprentissage



Comment apprendre des représentations ?

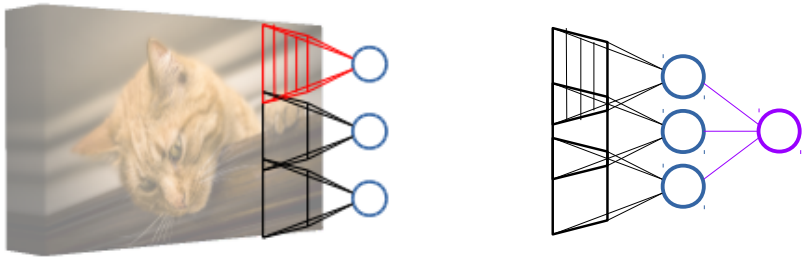
- PMC (connectivité complète entre couches successives) profond ⇒ **très** grand nombre de paramètres
  - Par ex., en entrée image de $1000 \times 1000 = 10^7$ pixels, suivie de 10 couches de 1000 neurones $\rightarrow 1000 \times 1000 \times 1000 + 10 \times 1000 \times 1000 \sim 10^9$ connexions

  → Une bonne généralisation exige un **très** grand volume de données d'apprentissage

  → Les invariances spécifiques au domaine doivent être apprises…

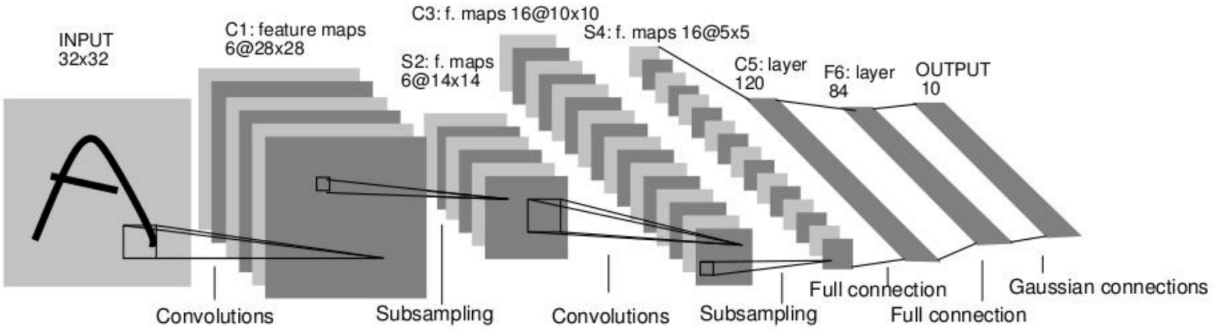
→ Tirer profit des spécificités du domaine pour réduire fortement la connectivité
  - Un « détecteur » devrait être invariant à la translation de l'objet dans l'image

  ⇒ connectivité locale, poids partagés entre neurones (convolution + nonlinéarité) suivie de couche d'agrégation spatiale (*pooling*)



Réseaux convolutionnels

- Introduits dès 1989 [1, 2]
  - LeNet 5 :



- Principe : succession de blocs [convolution + nonlinéarité + *pooling* (*subsampling*)] + couche(s) à interconnexion complète en sortie

Réseaux convolutionnels (2)

- Explosion du nombre de couches cachées (avec amélioration des performances) : 6 (LeNet5) $\rightarrow$ 7 (AlexNet 2012) $\rightarrow$ 18 (VGG, 2014) $\rightarrow$ 22 (GoogLeNet, 2014)) $\rightarrow$ 152 (ResNet, 2015)...
- Permise par de nombreuses évolutions, parmi lesquelles
  - Données : très grands volumes (ImageNet 14×10^6 images annotées, 20000 classes) ; *data augmentation*
  - Activation : ReLU plutôt que sigmoïde $\rightarrow$ éviter *vanishing gradient*
  - Convolution : fenêtres moins larges mais plusieurs couches ; plusieurs largeurs de fenêtre
  - *Pooling* : `max` plutôt que `avg`, recouvrement
  - Architecture : couches de classification intermédiaires $\rightarrow$ régularisation, éviter *vanishing gradient* ; court-circuits (*shortcuts*)
  - Régularisation : *dropout*, normalisation par *batch*

Réseaux profonds et représentation des données multimédia

- Réseau convolutionnel (CNN) profond qui apprend à résoudre un problème assez général sur un très grand corpus (par ex. reconnaissance sur ImageNet) → représentations discriminantes, plutôt générales et d'assez haut niveau sémantique développées dans les couches cachées proches de la sortie

⇒ Il est possible de se servir de ces représentations pour d'autres tâches

- Recherche d'images par similarité combinant aspect visuel et sémantique
- Classement de nouvelles images dans de nouvelles classes
- *Semantic segmentation* : segmentation avec reconnaissance

- Par ex., pour classement de nouvelles images dans de nouvelles classes :
  1. Remplacement de la couche de sortie par une autre (ou par SVM), apprentissage de cette seule couche sur les nouvelles classes
  2. Rétropropagation dans les couches cachées avec vitesse d'apprentissage faible (*fine-tuning*, seulement si la nouvelle base n'est pas trop petite)

Plan du cours

Pourquoi ?

- Chercher à mieux comprendre les prédictions d'un modèle
  1. Pouvoir faire confiance aux prédictions, surtout dans des domaines critiques (médecine, défense, transports, etc.)
  2. Pouvoir identifier des problèmes *a priori* ou corriger des erreurs *a posteriori*
  3. Pouvoir justifier des décisions lorsque le besoin se manifeste
- Pourquoi est-ce difficile de comprendre les prédictions
  1. Complexité : nombre élevé de variables d'entrée (explicatives)
  2. Complexité : frontière compliquée pour la séparation entre classes
  3. Opacité : niveaux multiples de re-représentation interne des observations (surtout *deep learning*)
- Contradiction : les meilleures performances exigent souvent des modèles complexes $\leftrightarrow$ les humains ne comprennent que des relations comparativement simples
  - $\Rightarrow$ Expliciter sous une forme « simple » les relations qui comptent le plus pour un modèle en général, pour une prédiction en particulier, pour une composante/étape d'une prédiction

Un peu de terminologie

- Plusieurs communautés, points de vue multiples $\rightarrow$ différents termes
  - Transparence : accès à des informations faciles à comprendre concernant le fonctionnement interne
  - Intelligibilité : possibilité pour un humain de comprendre le fonctionnement, de prédire l'effet d'un changement
  - Interprétabilité : possibilité pour un humain de conceptualiser la façon dont les prédictions sont réalisées
  - Explicabilité : capacité à fournir une justification ou explication de chaque prédiction
- Nous remarquons une certaine ambiguïté, un degré parfois élevé de relativité dans les définitions (qu'est-ce qui est « compréhensible », pour qui), ainsi qu'une utilisation parfois interchangeable de certains termes (voir par ex. [4])
- Le néologisme « explicabilité » (*eXplainable Artificial Intelligence*, XAI) est souvent employé pour désigner cette problématique générale

Première approche : construire directement des modèles explicables

1. Modèles « simples » : limiter le nombre de variables et choisir comme méthode de modélisation les modèles « lisibles » (par ex. les arbres de décision)
   - Limiter le nombre de variables : sélection parmi les variables de départ, ou construction d'un faible nombre de variables dérivées optimisant un critère (par ex. l'indépendance entre variables)
   - Limiter la complexité des modèles a souvent pour conséquence la limitation de leurs performances
2. Construction de variables explicatives dérivées, permettant d'employer à la suite des modèles simples avec de bonnes performances
   - Revient à isoler une partie de la complexité dans des variables dérivées interprétables (par ex. rapports, comme l'indice de masse corporelle)
3. Modèles modulaires : fonctionnalités décomposables, avec
   - Composantes individuellement explicables
   - Interactions interprétables entre composantes

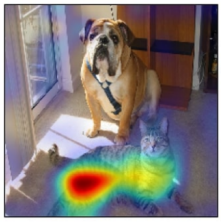
Seconde approche : extraire *a posteriori* des explications

A. Méthodes globales (pour l'ensemble du domaine couvert par le modèle)
   1. Approximation globale du modèle complexe original par un modèle plus explicable : la simplification peut mettre en évidence des relations dominantes entre (groupes de) variables mais subsistent des écarts au niveau de certaines prédictions individuelles
   2. Techniques de visualisation adaptées : relations entre variables et prédictions, représentations intermédiaires apprises (*deep learning*)

B. Méthodes locales (pour chaque prédiction individuelle)
   1. Approximation locale, autour d'une observation, par un modèle plus explicable : le modèle simplifié peut s'avérer performant localement (mais ne doit pas être considéré valable au-delà du voisinage de l'observation)
   2. Comparaison de l'observation à ses k plus proches voisins : quelles différences entre valeurs des variables correspondent aux éventuelles différences de classe
   3. Techniques de visualisation adaptées : variables (ou représentations intermédiaires) importantes pour la prédiction individuelle

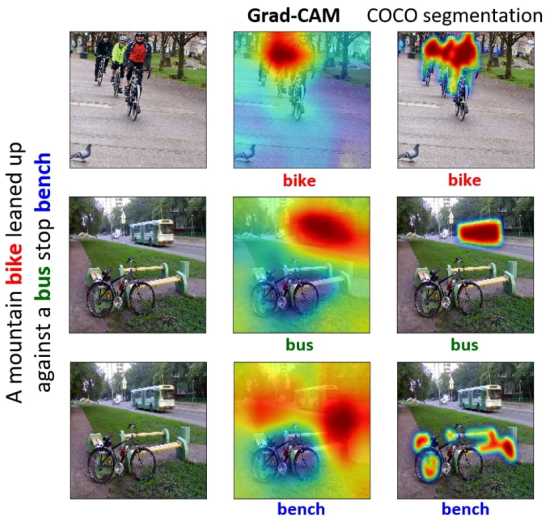
Un exemple : Grad-CAM [6]

- Objectif : visualiser quelle partie d'une image est utilisée par un modèle (réseau de neurones profond) pour prédire chaque mot (nom commun) d'une légende
- Méthode : variables dont le changement a le plus d'impact sur la sortie spécifique
- Permet également de comprendre les erreurs du modèle et de les corriger en complétant les données d'apprentissage, voir par ex.
  `https://towardsdatascience.com/understand-your-algorithm-with-grad-cam-d3b62fce353`



(c) Grad-CAM 'Cat'

(i) Grad-CAM 'Dog'



Références I

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel.
Back-propagation applied to handwritten zip code recognition.
Neural Computation, 1(4) :541–551, 1989.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner.
Gradient-based learning applied to document recognition.
Proceedings of the IEEE, 86(11) :2278–2324, November 1998.

[3] W. S. McCulloch and W. Pitts.
A logical calculus of the ideas immanent in nervous activity.
Bulletin of Mathematical Biophysics, 5 :115–133, 1943.

[4] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu.
Definitions, methods, and applications in interpretable machine learning.
Proceedings of the National Academy of Sciences, 116(44) :22071–22080, 2019.

[5] B. Schölkopf and A. Smola.
Learning with Kernels.
MIT Press, 2002.

[6] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra.
Grad-CAM : Visual explanations from deep networks via gradient-based localization.
International Journal of Computer Vision, 128(2) :336–359, Oct 2019.

Références II

[7] B. Widrow and M. E. Hoff.
Adaptive switching circuits.
In *1960 IRE WESCON Convention Record, Part 4*, pages 96–104, New York, 1960. IRE.