

Introduction au data Mining

Clustering et apprentissage
supervisé

Christel Dartigues-Pallez

*Équipe Sparks – Laboratoire I3S
dartigue@unice.fr*

De la Statistique ...

- Quelques centaines d'individus,
- Quelques variables,
- Fortes hypothèses sur les lois statistiques suivies,
- Importance accordée au calcul,
- Échantillon aléatoire.

... au Data mining

- Des millions d'individus,
- Des centaines de variables,
- Données recueillies sans étude préalable,
- Nécessité de calculs rapides,
- Pas un échantillon aléatoire.

Systemes d'information et data mining

- Les données forment le cœur des processus de base dans la plupart des entreprises.
- L'archivage des données crée la mémoire de l'entreprise.
- L'exploitation des données « data mining » crée l'intelligence de l'entreprise.

Définition de l'exploitation des données (data mining)

- L'exploration des données ou data mining est l'analyse de grandes quantités de données afin de découvrir des formes et des règles significatives en utilisant des moyens automatiques ou semi-automatiques basés sur l'analyse statistique et l'apprentissage automatique .
- (Berry et Linoff, 1997)

Terminologie

- KDD (Knowledge Discovery in Databases)
- Fouille de données (terme français)
- Extraction automatique de connaissances à partir de données (ECD)
- Recherche d'Information (Information Retrieval)

Domaines d'application du Data mining (1)

- Activités commerciales : grande distribution, vente par correspondance, banque, assurances
 - segmentation de la clientèle
 - détermination du profil du consommateur
 - analyse du panier de la ménagère
 - mise au point de stratégies de rétention de la clientèle
 - prédiction des ventes
 - détection des fraudes
 - identification de clients à risques

Domaines d'application du Data mining (2)

- Activités financières
 - recherche de corrélations entre les indicateurs financiers
 - maximiser le retour sur investissement de portefeuilles d'actions
- Activités de gestion des ressources humaines
 - prévision du plan de carrière
 - aide au recrutement

Domaines d'application du Data mining (3)

- Activités industrielles
 - détection et diagnostic de pannes et de défauts
 - analyse des flux dans les réseaux de distribution
- Activités scientifiques
 - diagnostic médical, santé publique : ex, étude du génome
 - analyse chimique, biologique et pharmaceutique
 - exploitation de données astronomiques
 - Recherche d'information dans les grands volumes de données multimédia

La découverte de connaissances dans les données

- Poser le problème
 - Comprendre le domaine d'application, les connaissances disponibles, la finalité
- Recherche des données
 - Les données existent déjà ou il faut les collecter (si oui quoi collecter et comment, quand, par qui?)
- Nettoyage des données
 - Doublons, erreurs de saisie, pannes de capteurs => valeurs aberrantes, manquantes...
 - Stratégies pour valeurs manquantes : ignorer, moyenne (pire solution), technique de régression
- Codage des données, actions sur les variables
 - Étape potentiellement discriminante (augmenter ou réduire le nombre de variables, moyen de les coder, standardisation, etc.)
- Recherche d'un modèle, de connaissances, d'information : Data mining
- Validation et interprétation du résultat, avec retour possible sur les étapes précédentes
- Intégration des connaissances apprises

Data mining

- Des algorithmes d'inspirations ...
 - Mathématiques : statistiques et Analyse de Données
 - Calculatoires
 - « Clustering »
 - Arbres de décision, forêts aléatoires
 - Règles d'association
 - Programmation dynamique
 - Machines à Vecteurs de Support (SVM)
 - Biologiques
 - Réseaux de neurones
 - Algorithmes génétiques

Data mining

- Des algorithmes :
 - Non Supervisés - Apprentissage a priori en mode Découverte
 - « Clustering »
 - Algorithmes génétiques
 - Règles d'association
 - Supervisés - Apprentissage a posteriori en mode Reconnaissance/Prédiction
 - Réseaux de neurones / Machines à Vecteurs de Support
 - Arbres de décision, forêts aléatoires
 - Programmation dynamique

Data mining

- Classification
- Estimation
- Recherche d'associations
- Clustering

Data mining : Classification

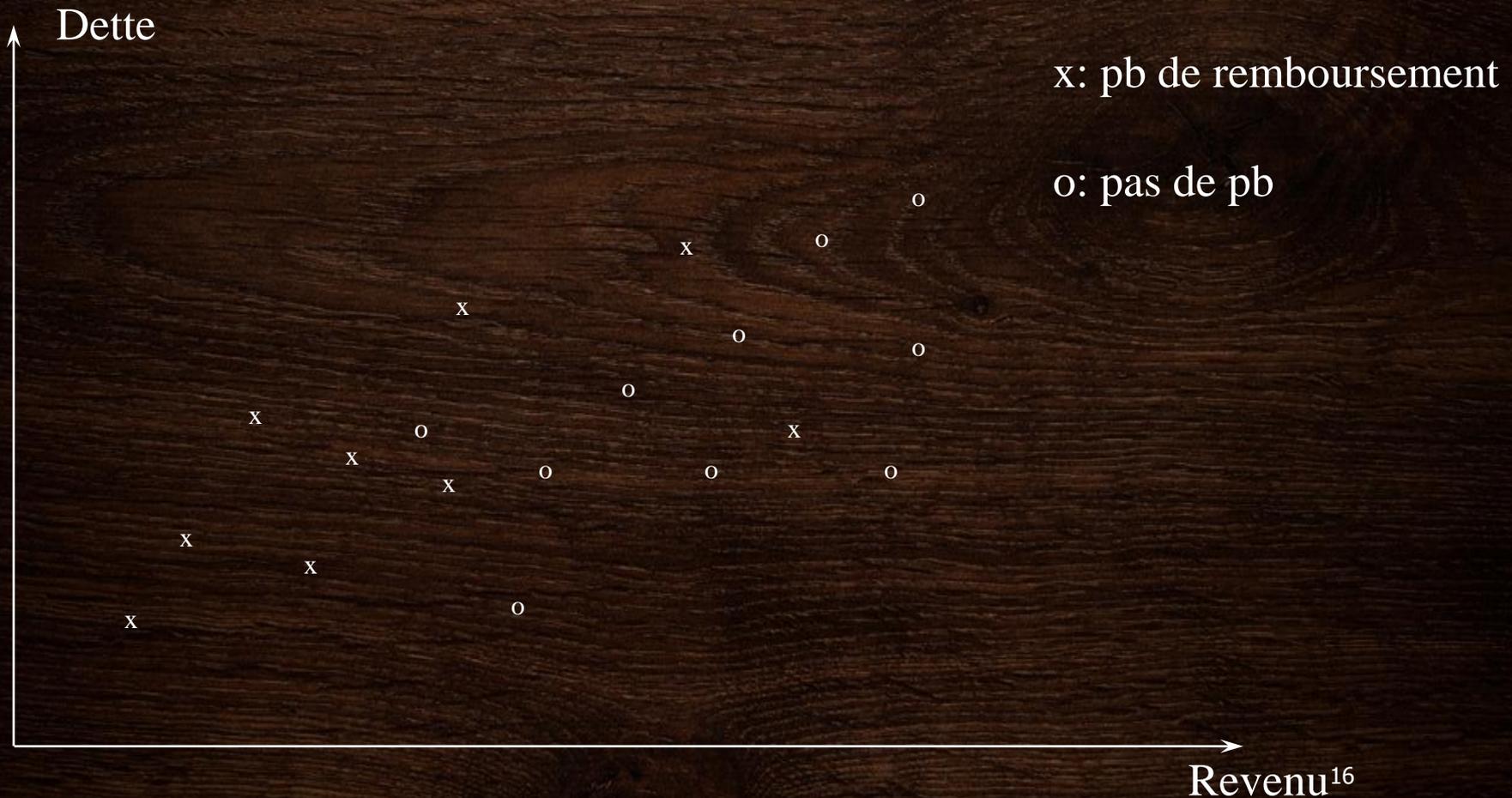
- Affecter un objet à une classe en fonction de ses caractéristiques A_1, \dots, A_n
- Exemple
 - Déterminer si un message est un mail de SPAM ou non (2 classes)
 - Affecter une page web dans une des catégories thématiques de l'annuaire Yahoo (multi-classes)
 - Diagnostic : risque d'accident cérébral ou non (2 classes)

Classification

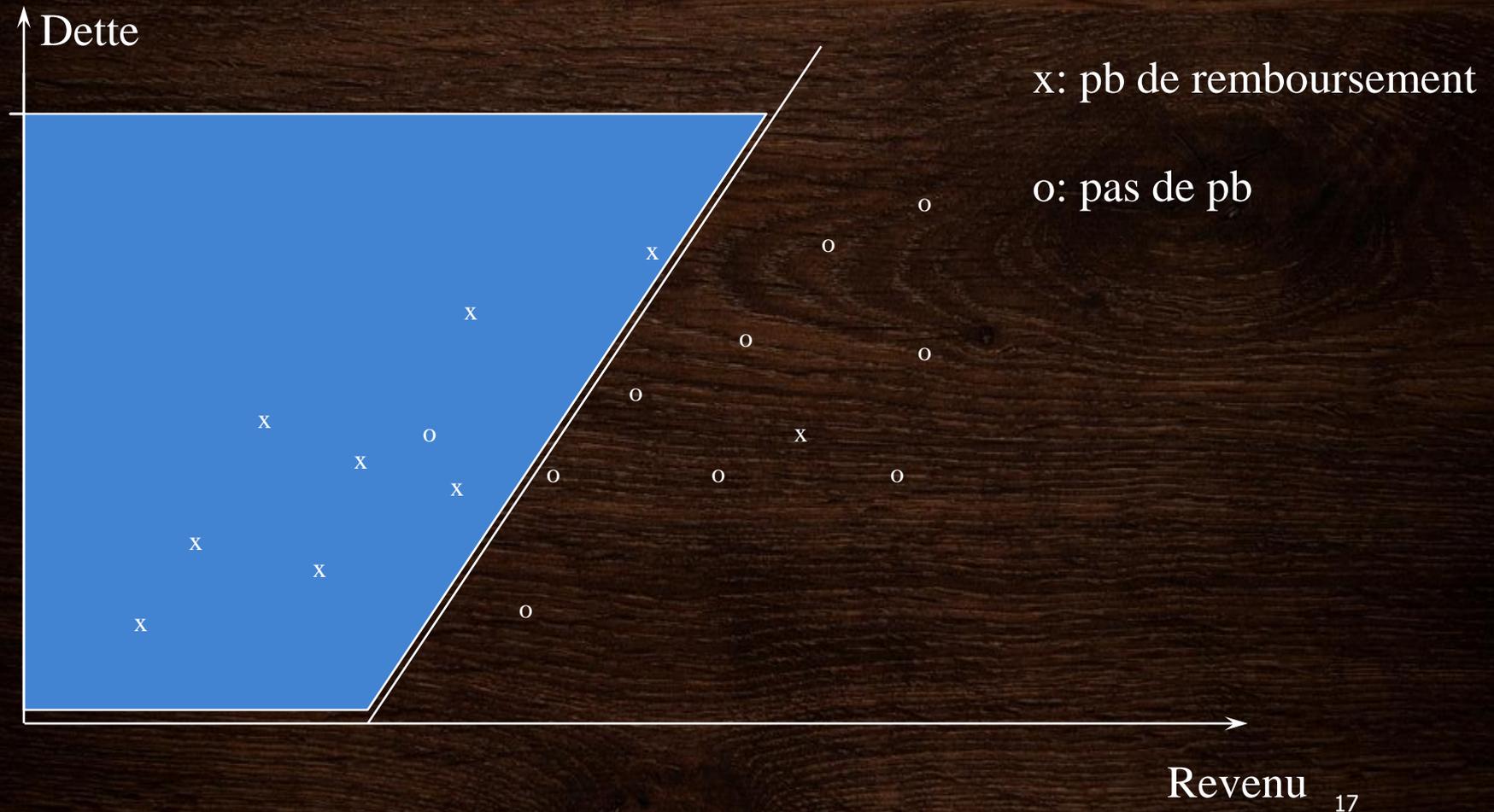
- Si pas de connaissance a priori pour définir la classe en fonction de A_1, \dots, A_n alors on étudie un ensemble d'exemples pour lesquels on connaît A_1, \dots, A_n et la classe associée et on construit un modèle
- Classe = $f(A_1, \dots, A_n)$
 - Analyse discriminante
 - Arbres de classification
 - Machines à noyaux

Un exemple schématique avec 2 classes

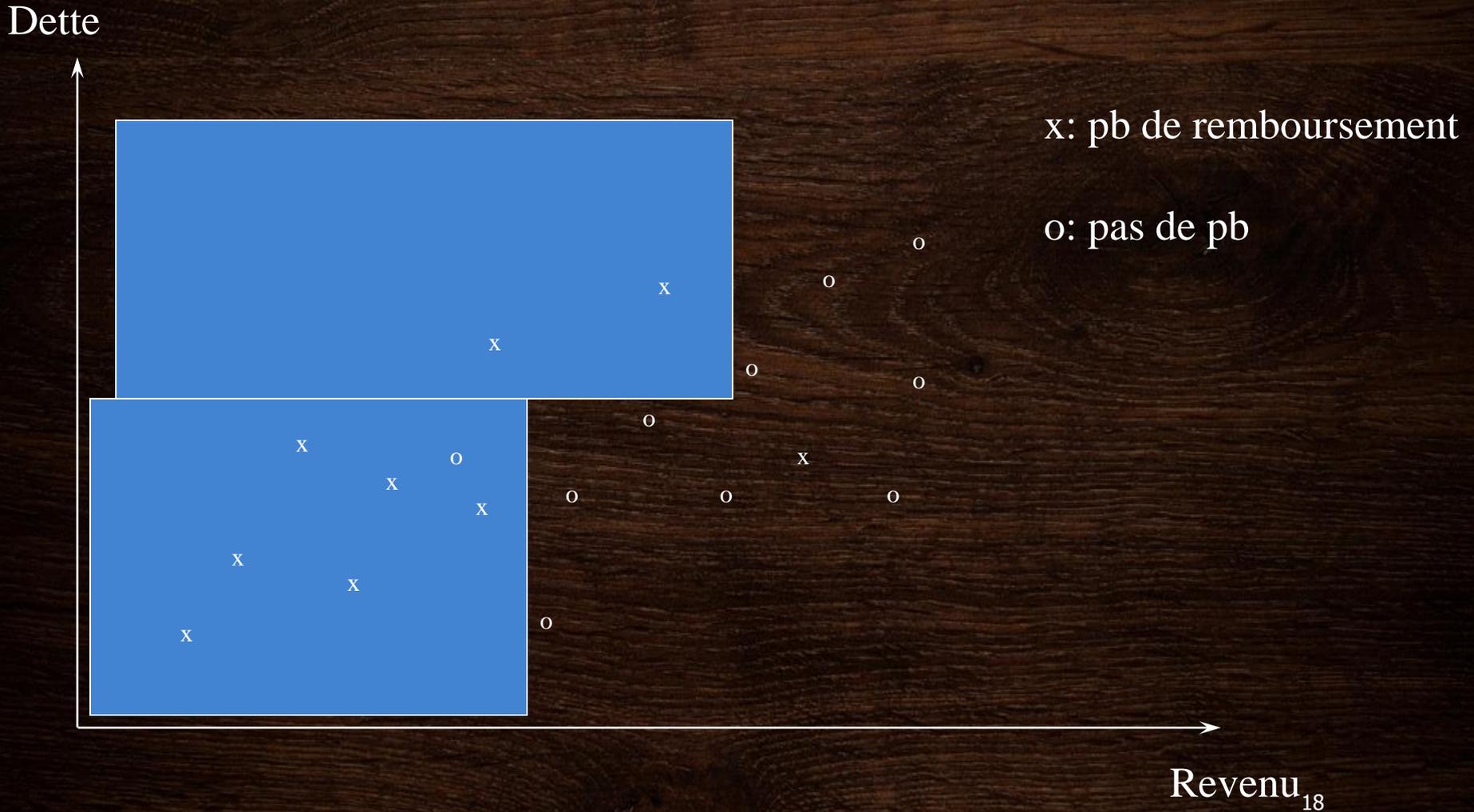
La classification c'est apprendre une fonction qui permet d'affecter un nouvel individu dans une classe ou une autre.



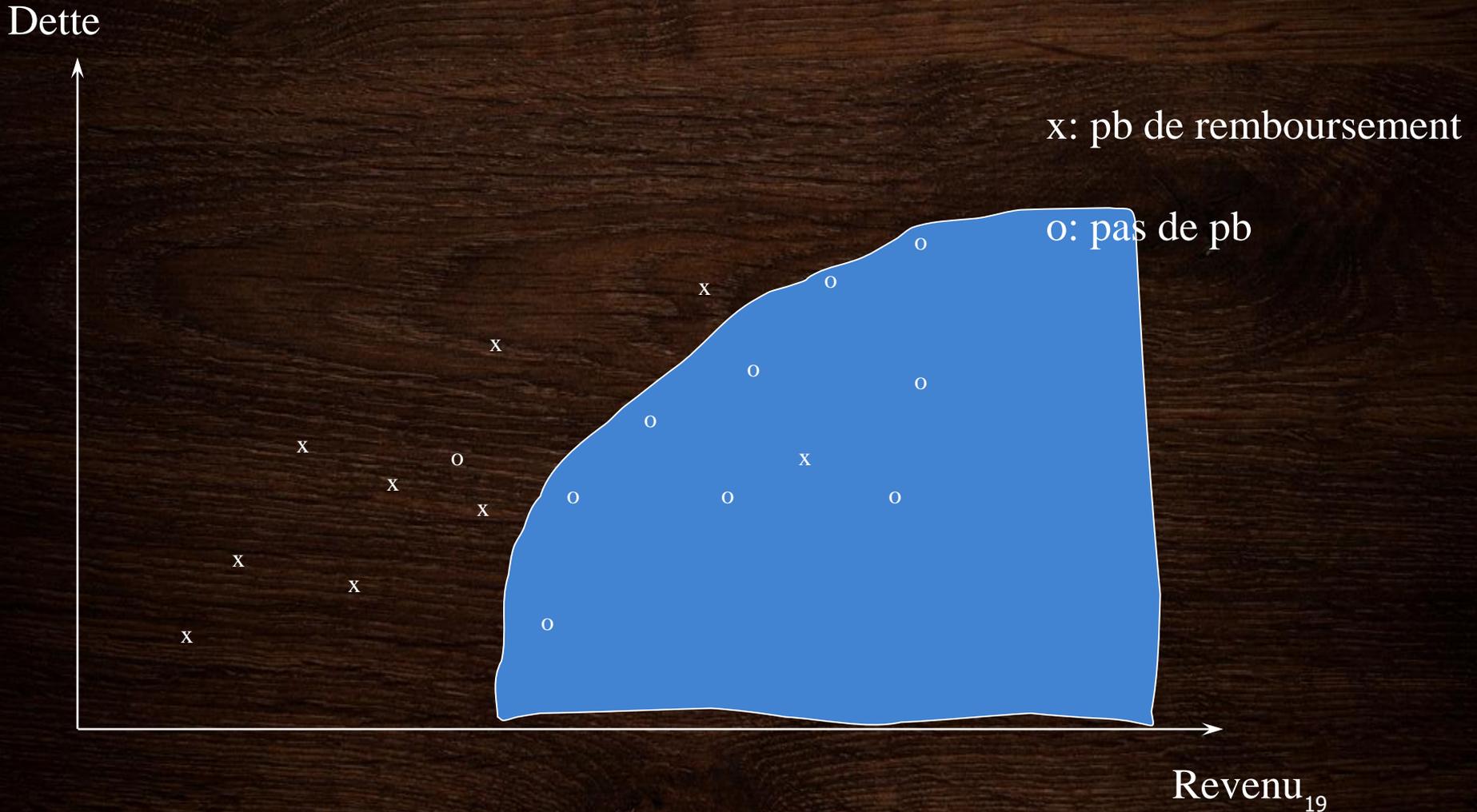
Classification par analyse discriminante



Classification par arbre de décision



Classification par machine à noyaux

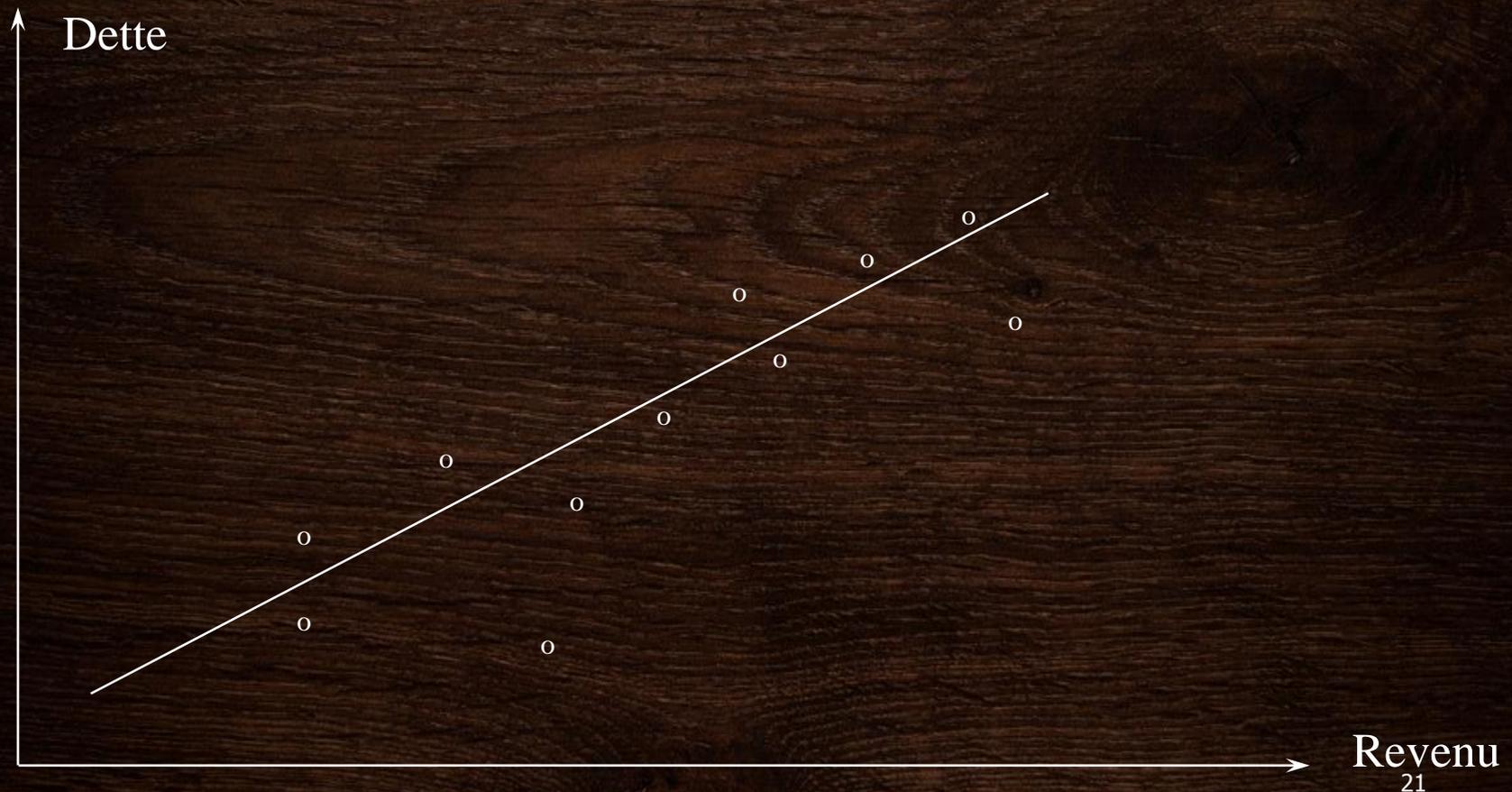


Data mining : Estimation

- Estimer (prédire) la valeur d'une variable à valeurs continues à partir des valeurs d'autres attributs
 - Régression
 - Machines à noyaux

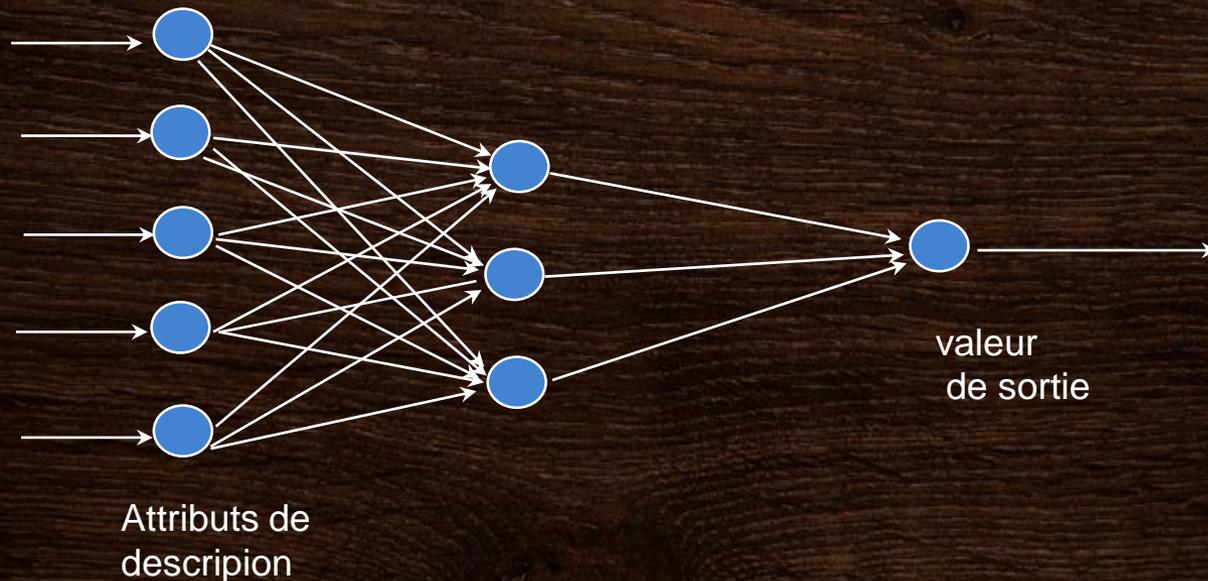
Régression linéaire simple

La régression explique les variations d'une variable par une fonction des autres variables : ici la dette est représentée comme une fonction du revenu, le résultat est médiocre car il y a peu de corrélation.



Machines à noyaux : Réseau de neurones & SVM

- La couche d'entrées correspond aux entrées, la couche de sortie(s) au résultat
- Système non-linéaire
- L'apprentissage va ajuster les poids des connexions mais l'architecture et le nombre de neurones dans la couche cachée est un choix arbitraire.



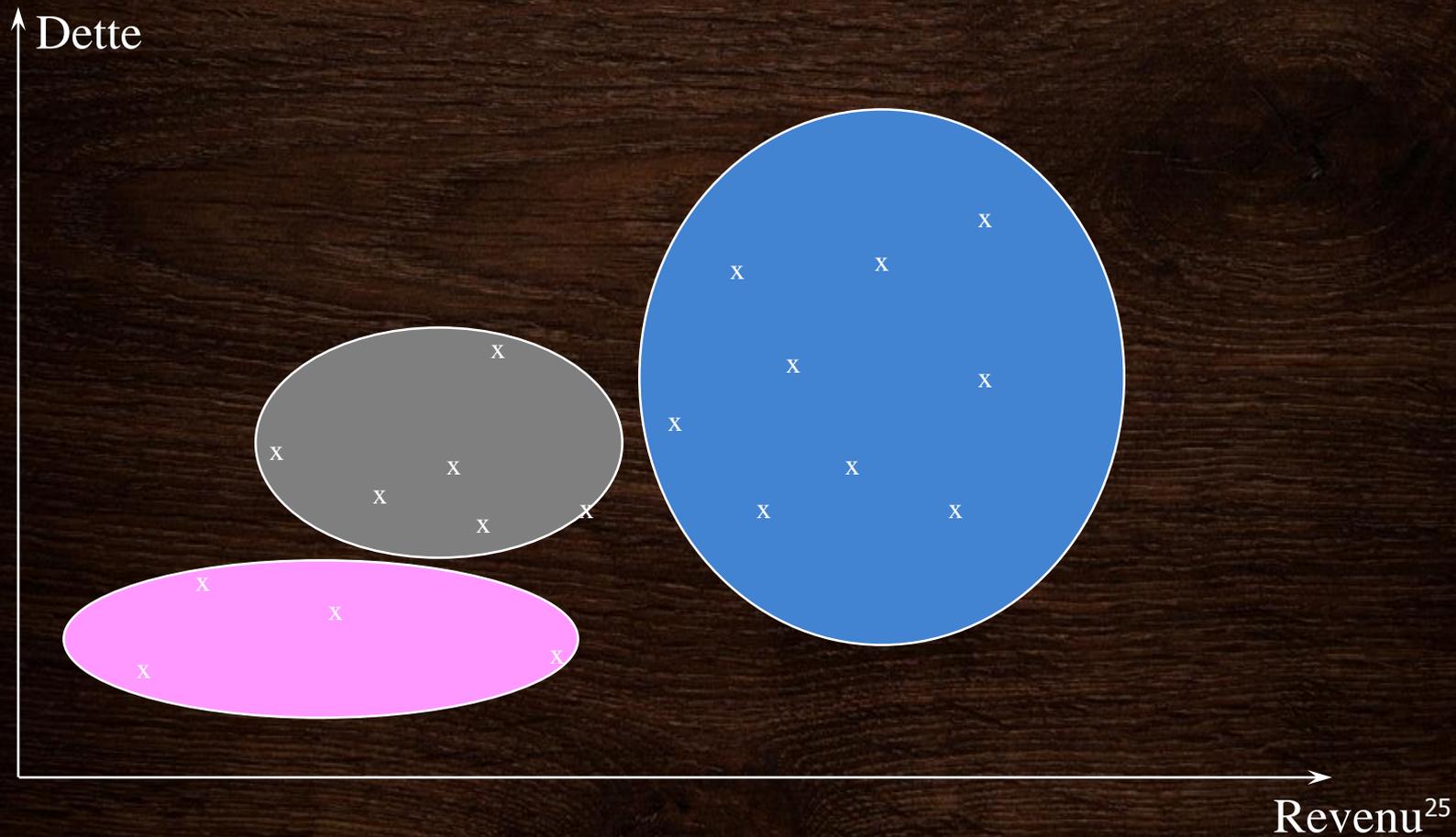
Data mining : Recherche de règles

- Règles d'associations : analyse du panier de la ménagère
 - « le jeudi, les clients achètent souvent en même temps des packs de bière et des couches. »
 - Y-a-t-il des liens de causalité entre l'achat d'un produit P et d'un autre produit P' ?

Data mining : Segmentation / partitionnement (clustering)

- Apprentissage non supervisé : les données ne sont pas classées, on isole des sous-groupes d'enregistrements similaires les uns aux autres (nuées dynamiques ou agrégation)
- Un fois les clusters détectés, on pourra appliquer des techniques de modélisation sur chaque cluster

Clustering *Pas d'affectation a priori, une classe connue au départ : on regroupe les individus par leur proximité en classes qui peuvent se recouper.*

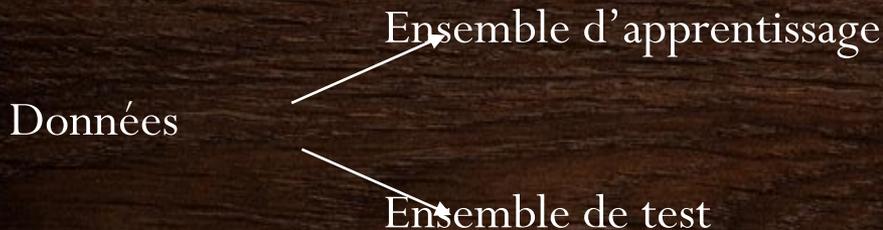


Supervisé vs NON supervisé

- La classification, la régression logistique sont des tâches supervisées
 - Data mining prédictif (on dispose d'une variable dépendante à prédire ou à estimer notée généralement par Y).
- Le clustering, la recherche de règles d'associations sont des tâches non supervisées
 - Data mining explicatif (on cherche plus à expliquer les relations entre les variables sans disposer d'une variable dépendante).

Validation dans le cas supervisé

- Validation par le test



Construction d'un modèle sur l'ensemble d'apprentissage et test du modèle sur le jeu de test pour lequel les résultats sont connus

- Évaluation quantitative (ne pas oublier les intervalles de confiance)

Intégration de la connaissance

- Prise de décision grâce aux connaissances extraites
- Les experts métiers sont essentiels pour donner du sens aux informations extraites !

Logiciels de Data Mining

- Commerciaux

- SAS Interprise Miner
(<http://www.sas.com/technologies/analytics/datamining/miner/>)
- SPSS Clementine (<http://www.spss.com/SPSSBI/Clementine/>)
- Insightful Miner (<http://www.insightful.com/products/iminer/>)
- ...

- Gratuits

- TANAGRA (<http://eric.univ-lyon2.fr/~ricco/tanagra/>)
- ORANGE (<http://www.ailab.si/orange>)
- WEKA (<http://www.cs.waikato.ac.nz/ml/weka/>)
- R (<http://www.r-project.org/>)

Clustering – K-means

Clustering

- Principes
 - Contexte non supervisé
 - « Révéler » l'organisation de motifs en groupes cohérents
 - Processus Subjectif
- Disciplines : Biologie, Zoologie, Psychiatrie, Sociologie, Géologie, Géographie...
- Synonymes : Apprentissage non supervisé, Taxonomie, Typologie, Partition

Qu'est-ce que le clustering ?

- Groupe ou “Cluster”: un ensemble d'objets ou d'individus
 - Semblables entre eux à l'intérieur d'un groupe
 - Différents d'un groupe à l'autre
- Segmentation ou “Cluster analysis”
 - Classement des individus ou objets dans différents groupes ou segments
- Le clustering est une technique non dirigée (c-à-d. il n'y a pas de variable “target”)

Quelques applications du clustering

- Reconnaissance de forme non supervisée
- Taxonomie (biologie, zoologie)
- Segmentation des marchés (marketing)
- Geo-segmentation
- WWW
 - classification des sites
 - classification des “Weblog” pour découvrir des profils d’accès semblables

Comment déterminer une bonne segmentation ?

- Un bon algorithme de classification fera en sorte qu'il y aura une :
 - petite variabilité intra-classe (c-à-d petite distance entre les individus d'un même groupe)
 - grande variabilité inter-classe (c-à-d grande distance entre les individus de groupes différents)
- La qualité des résultats de la classification dépendra de la mesure de distance utilisée et de l'algorithme choisi pour l'implanter

Approches de Clustering

- Algorithmes de Partitionnement: Construire plusieurs partitions puis les évaluer selon certains critères
- Algorithmes hiérarchiques: Créer une décomposition hiérarchique des objets selon certains critères
- Algorithmes basés sur la densité: basés sur des notions de connectivité et de densité

Algorithmes à partitionnement

- Construire une partition à k clusters d'une base D de n objets
- Les k clusters doivent optimiser le critère choisi
 - Global optimal: Considérer toutes les k -partitions
 - Heuristic methods: Algorithmes k -means et k -medoids
 - k -means (MacQueen'67) : Chaque cluster est représenté par son centre
 - k -medoids or PAM (Partition Around Medoids) (Kaufman & Rousseeuw'87) : Chaque cluster est représenté par un de ses objets

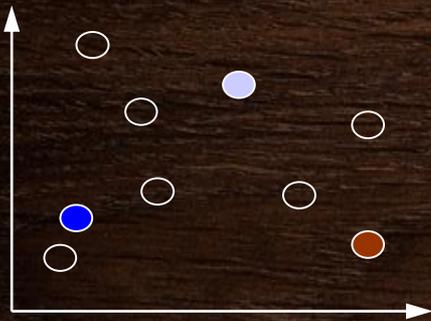
K-Means

- Méthode des K-moyennes (MacQueen'67)
 - choisir K éléments initiaux "centres" des K groupes
 - placer les objets dans le groupe de centre le plus proche
 - recalculer le centre de gravité de chaque groupe
 - itérer l'algorithme jusqu'à ce que les objets ne changent plus de groupe
- Encore appelée méthode des centres mobiles

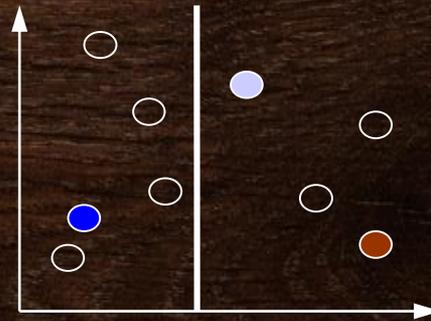
Algorithme

- Étapes:
 - fixer le nombre de clusters: k
 - choisir aléatoirement k tuples comme graines (centres)
 - assigner chaque tuple à la graine la plus proche
 - recalculer les k graines
 - tant que des tuples ont été changés
 - réassigner les tuples
 - recalculer les k graines
- C'est l'Algorithme le plus utilisé

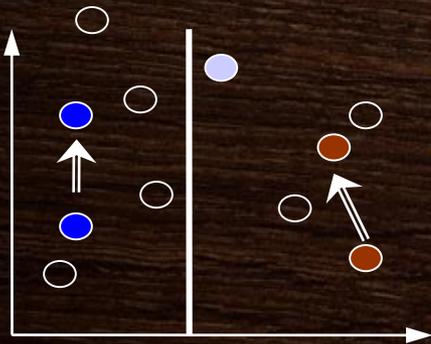
Exemple de K-Means (k=2)



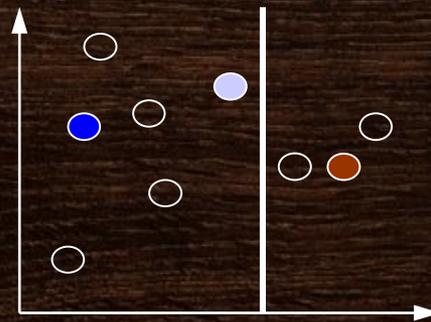
Choisir 2 graines



Assigner les tuples



Recalculer les centroïdes



Réassigner les tuples

K-means Clustering Demo

- Clustering demo:
 - http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/AppletKM.html

Avantages de K-means

- Rapidité : on ne compare pas toutes les observations entre elles mais par rapport aux centres de classes
- Permet de détecter les valeurs extrêmes et de les isoler
- Est pratique quand il y a un très grand nombre d'observations (des milliers)

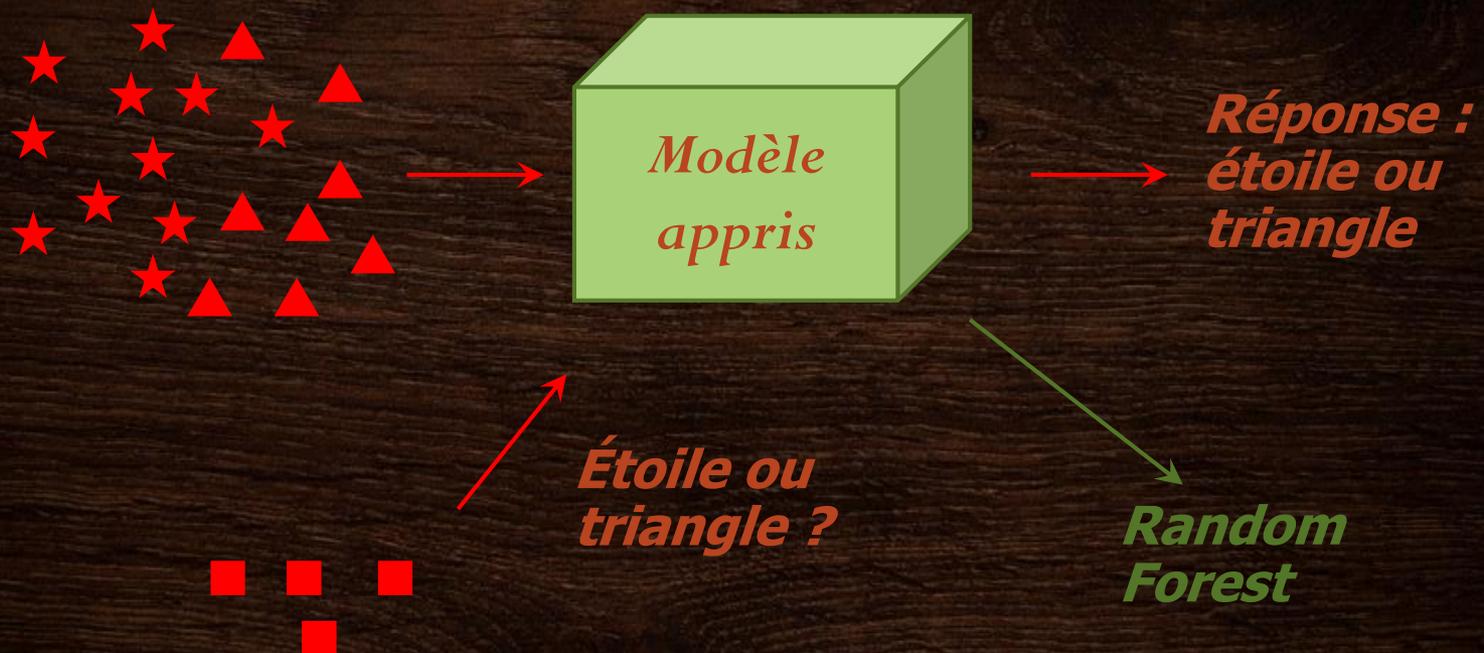
Inconvénients de K-means

- Obligation de fixer à priori le nombre de classes ou clusters
- Dépendance au choix des centres initiaux (seeds)
- Ne détecte bien que les formes convexes (surtout sphériques de même taille).
- Mauvaise prise en compte des "outliers"
 - points extrêmes en dehors des groupes
 - fausses les moyennes et donc les centres
- Convergence plus ou moins rapide
- Amélioration:
 - utilisation de points centraux (médoïdes)

Arbres de Décision

Apprentissage supervisé

- Construction d'un modèle (boîte noire) qui prend en entrée des données et qui fournit en sortie une réponse



Types de données

- Discrètes énumératives (couleur, code postal, etc.)
- Discrètes ordonnées (classe de salaire ou d'âge)
- Continues: revenu, temps, température
- Dates transformation en variables énumératives ordonnées
- Image et vidéo: extraction de "features"
- Données structurées: XML, etc.
- Textuelles: faire un histogramme par mots clés
- On peut transformer "facilement" continue \rightarrow discret \rightarrow binaire

Arbres de décision - Analogie

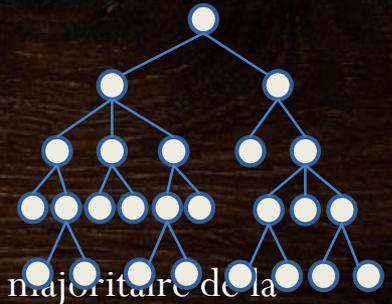


Généralités – Arbre de décision

- Arbre de décision : classifieur simple et graphique
 - Lisibilité
 - Rapidité d'apprentissage et d'exécution
- Objectifs
 - Répartir une population d'individus en groupes homogènes
 - Selon un ensemble de variables discriminantes
 - En fonction d'un objectif connu (apprentissage supervisé)
- Prédire les valeurs prises par la variable à prédire à partir d'un ensemble de descripteurs
 - Variable à prédire = objectif, variable cible, variable d'intérêt, attribut classe, variable de sortie
 - Descripteur = variables prédictives, variables discriminantes

Généralités – Arbre de décision

- Structure de données utilisée comme modèle pour la classification [Quinlan]
- Méthode récursive basée sur diviser-pour-régner pour créer des sous-groupes (plus) purs
 - Un sous-groupe est pur lorsque tous les éléments du sous-groupe appartiennent à la même classe)
- Construction du plus petit arbre de décision possible
 - Nœud = Test sur un attribut
 - Une branche pour chaque valeur d'un attribut
 - Les feuilles désignent la classe de l'objet à classer
- Taux d'erreur = proportion des instances qui n'appartiennent pas à la classe majoritaire de la branche
- Problèmes : Choix de l'attribut, terminaison



Construction d'un arbre

- Pour construire un arbre de décision, il faut :
 - Choisir, parmi les variables qui restent, la variable de segmentation du sommet courant
 - Lorsque la variable est continue, déterminer le seuil de coupure
 - Déterminer la bonne taille de l'arbre
 - Est-il souhaitable de produire absolument des feuilles pures selon la variable à prédire, même si le groupe correspondant correspond à une fraction très faible des observations?
 - Affecter la valeur de la variable à prédire aux feuilles

Algorithme de base

- $A = \text{MeilleurAttribut}(\text{Exemples})$
- Affecter A à la racine
- Pour chaque valeur de A , créer un nouveau nœud fils de la racine
- Classer les exemples dans les nœuds fils
- Si tous les exemples d'un nœud fils sont homogènes, affecter leur classe au nœud, sinon recommencer à partir de ce nœud

Terminaison

- Plusieurs critères possibles
 - Tous les attributs ont été considérés
 - Il n'est plus possible d'obtenir de gain d'information
 - Les feuilles contiennent un nombre prédéfini d'éléments majoritaires
 - Le maximum de pureté a été atteint
 - Toutes les instances sont dans la même classe
 - L'arbre a atteint une hauteur maximum

Combien d'arbres de décision?

- Considérons m attributs booléens
- Un arbre de décision possible pour chacun des m attributs
 - Il y a 2^m façons de donner des valeurs aux attributs
- 2^{2^m} arbres de décision possibles

→ Comment sélectionner le meilleur?

Théorie de l'information

- Besoin d'une méthode pour bien choisir l'attribut [Shannon & Weaver, 1949]
- À chaque étape, à chaque point de choix dans l'arbre, on va calculer le gain d'information
 - L'attribut avec le plus grand gain d'information est sélectionné
- Méthode ID3 pour la construction de l'arbre de décision

Météo et match de foot

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Attribut but

2 classes: yes et no

Prédire si un match de foot va avoir lieu ou non

Température est un nominal

Un exemple simple

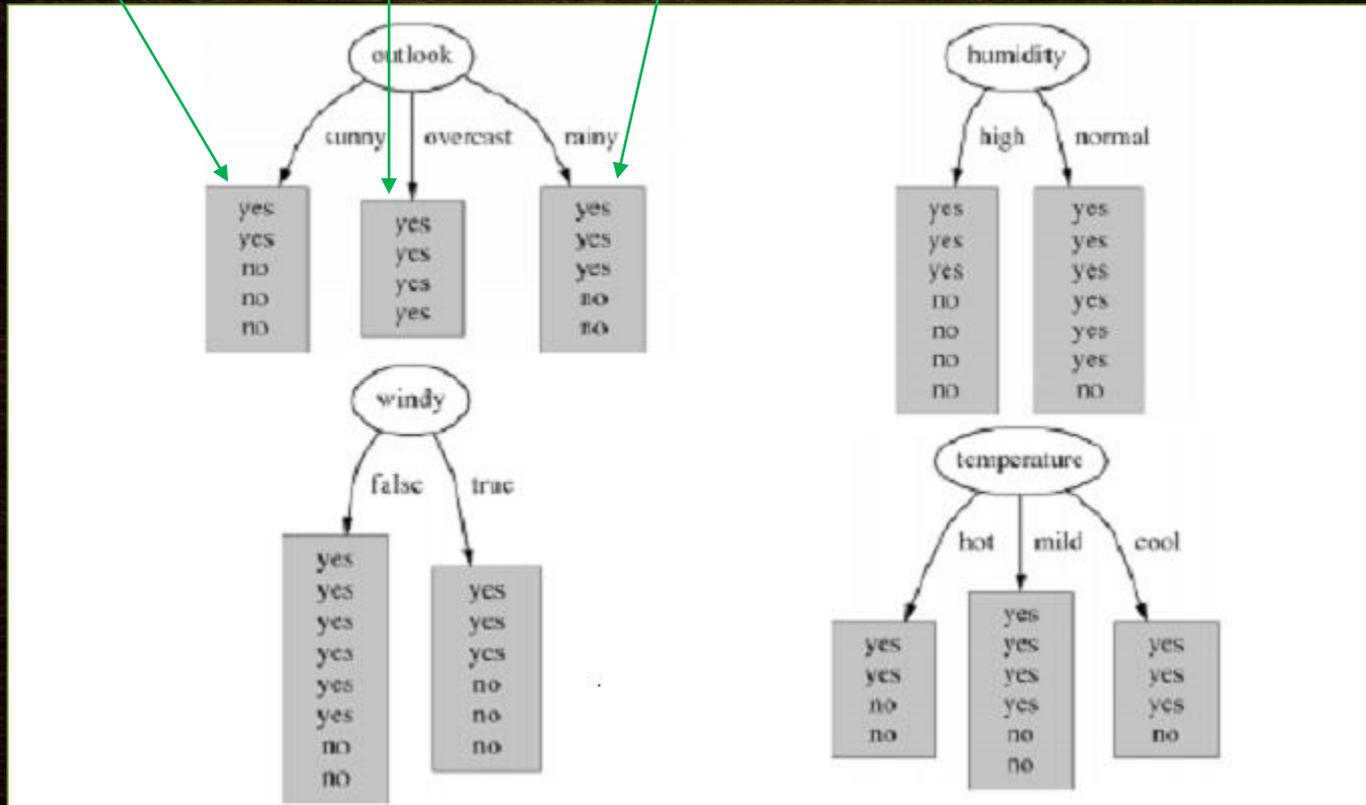
- Un ensemble de jours (un jour = un exemple)
- Chaque jour caractérisé par un numéro et ses conditions météorologiques (Outlook, Temperature, Humidity, Windy)
- Attribut cible : «Play?»
 - Valeurs possibles : oui et non (classification binaire)
- Exemples
 - 1, sunny, hot, high, false, no
 - 2, sunny, hot, high, yes, No
 - 3, overcast, hot, high, false, yes
- Une fois l'arbre de décision construit, on pourra classer une donnée correspondant à un nouveau jour pour savoir si on joue ou non ce jour-là

Quel attribut faut-il sélectionner?

Classe majoritaire : NO

Classe majoritaire : YES

Classe majoritaire: YES



Construction d'un arbre de décision

– quel attribut placer en racine?

- Entropie = quantité moyenne d'information de tous les messages dans un système
- Entropie de Shannon
 - Mesure l'hétérogénéité d'une population du point de vue de la classe de ses membres
- Considérons un ensemble X d'exemples dont une proportion p^+ sont positifs et une proportion p^- sont négatifs
- $(p^+) + (p^-) = 1$
- L'entropie de X est : $H(X) = -(p^+) \log_2(p^+) - (p^-) \log_2(p^-)$

Notion d'entropie – illustration

- Interprétation de l'entropie
 - Nombre minimum de bits nécessaires pour coder la classe d'un élément quelconque

Entropy in a nut-shell



Low Entropy



High Entropy

Entropie

- Cas simple (2 classes)
 - $0 \leq H(X) \leq 1$
 - Si $p_+ = 0$ ou $p_- = 0$, alors $H(X) = 0$ classe pure
 - Si $p_+ = p_- = 0,5$, alors $H(X) = 1$ (entropie maximale) : autant de positifs que de négatifs → pire cas : on ne peut rien dire
- Cas général (N classes)
 - Pour un attribut classe prenant N valeurs distinctes
 - p_i ($i \in \{1, N\}$) est la proportion d'exemples dont la valeur de cet attribut est i dans l'ensemble d'exemples considéré X
 - L'entropie de l'ensemble d'exemples X est :
$$H(X) = -(p_1 \log_2 p_1 + p_2 \log_2 p_2 + \dots + p_N \log_2 p_N)$$

Gain d'information et entropie

$$\mathbf{Info}([n, m]) = \mathbf{Entropy} \left(\frac{n}{n+m}, \frac{m}{n+m} \right)$$

$$\mathbf{Entropy}(\mathbf{p}_1, \mathbf{p}_2) = \frac{1}{\log 2} (-\mathbf{p}_1 \log \mathbf{p}_1 - \mathbf{p}_2 \log \mathbf{p}_2)$$

p_i est la probabilité de la classe i

$p_i = \text{nbre d'occurrences de } i / \text{nbre total d'occurrences}$

Cette formule est généralisable

Entropie pour 3 probabilités

$$\mathbf{Entropy}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = \frac{1}{\log 2} (-\mathbf{p}_1 \log \mathbf{p}_1 - \mathbf{p}_2 \log \mathbf{p}_2 - \mathbf{p}_3 \log \mathbf{p}_3)$$

$$\mathbf{Entropy}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = \mathbf{Entropy}(\mathbf{p}_1, \mathbf{p}_2 + \mathbf{p}_3) + ((\mathbf{p}_2 + \mathbf{p}_3) \mathbf{Entropy}\left(\left(\frac{\mathbf{p}_2}{\mathbf{p}_2 + \mathbf{p}_3}\right), \left(\frac{\mathbf{p}_3}{\mathbf{p}_2 + \mathbf{p}_3}\right)\right))$$

Propriété de l'entropie



Calcul du gain d'information pour un attribut

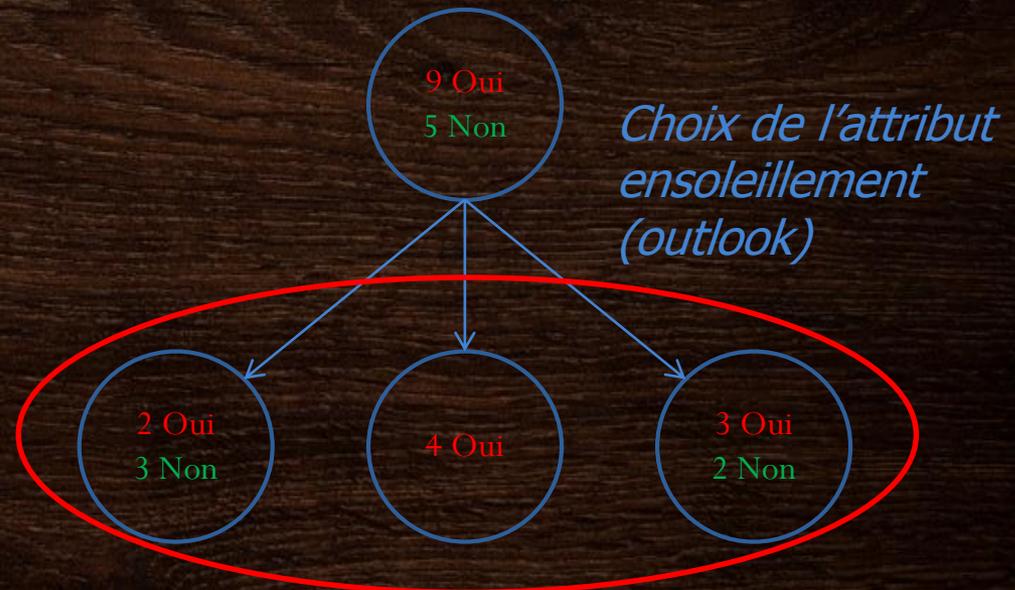
outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

- Principe général

– Gain(attribut) =

Info (avant le choix de l'attribut) –

Info (après le choix de l'attribut)



Info (avant choix d'ensoleillement)

– Info (après choix d'ensoleillement)

Gain d'information total pour Outlook

$$\text{gain}(\text{outlook}) = \mathbf{Info}([9,5]) - \mathbf{Info}([2,3],[4,0],[3,2])$$

$$\text{gain}(\text{outlook}) = 0.940 - 0.693$$

$$\text{gain}(\text{outlook}) = 0.247 \text{ bits}$$

De même:

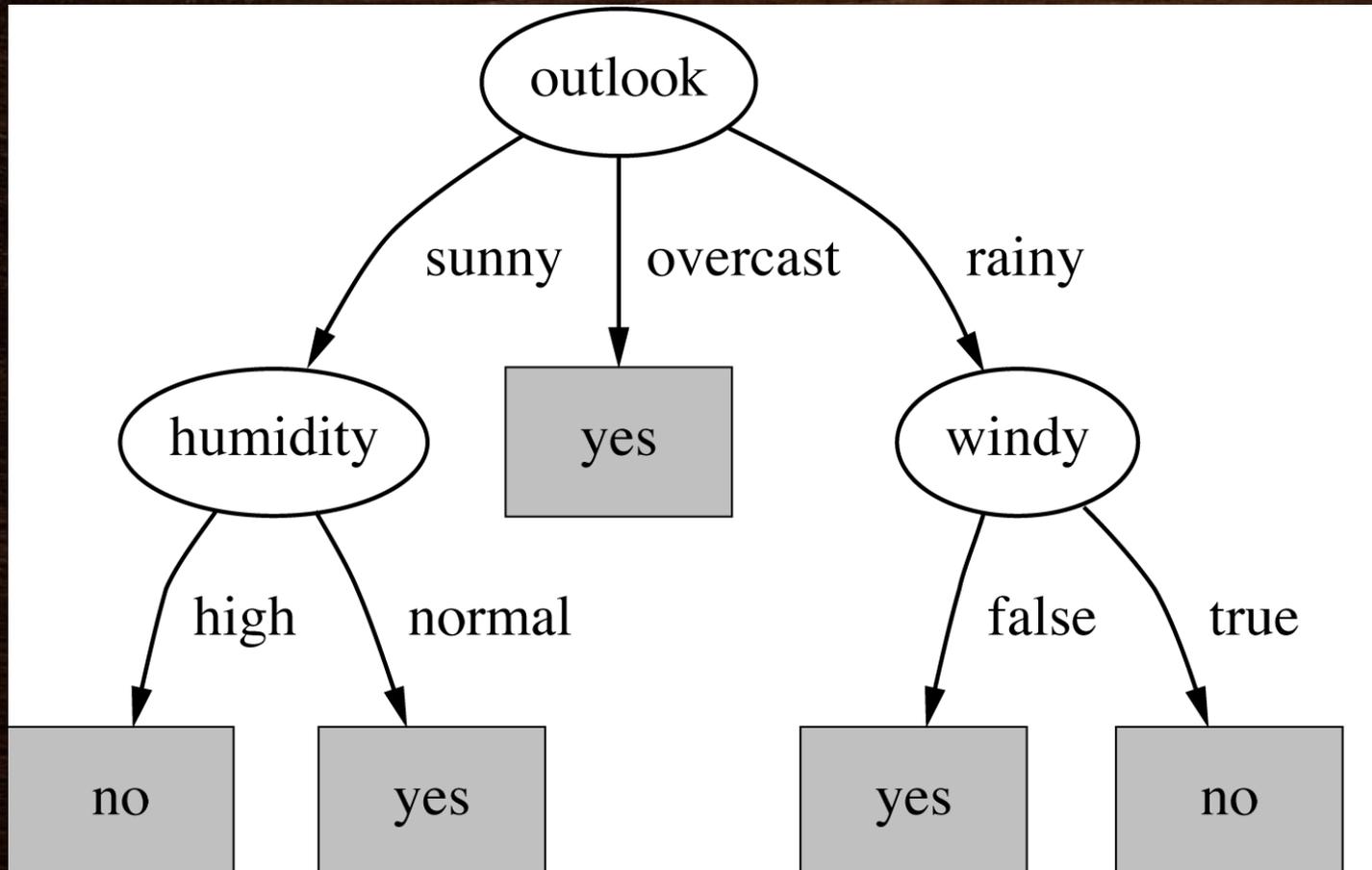
$$\text{gain}(\text{temperature}) = 0.029 \text{ bits}$$

$$\text{gain}(\text{humidity}) = 0.152 \text{ bits}$$

$$\text{gain}(\text{windy}) = 0.048 \text{ bits}$$

Outlook est choisi

Arbre de décision final



Extensions de l'algorithme

- Comment traiter:
 - Les attributs numériques
 - Les valeurs manquantes
- Comment simplifier le modèle pour éviter les bruits?
- Comment tolérer les bruits?
- Comment interpréter les arbres de décision?

Les valeurs manquantes

- Ignorer les instances avec des valeurs manquantes
 - Solution trop générale, et les valeurs manquantes peuvent être importantes
- Ignorer les attributs avec des valeurs manquantes
 - Peut-être pas faisable
- Traiter les valeurs manquantes comme des valeurs spéciales
 - Les valeurs manquantes ont un sens particulier
- Estimer les valeurs manquantes
 - Donner la valeur de l'attribut la plus répandue à l'attribut considéré
 - Donner une valeur en utilisant diverses méthodes
 - Exemple : régression

Définir la bonne taille de l'arbre

- Il y a un risque de surajustement du modèle : le modèle semble performant (son erreur moyenne est très faible) mais il ne l'est en réalité pas du tout
 - Les résultats sont bons sur les données apprises mais de nouvelles données (n'ayant pas servi à construire le modèle) seront mal classées
- Il faut trouver l'arbre le plus petit possible avec la plus grande performance possible (plus un arbre est petit et plus il sera stable dans ses prévisions futures)
- À performance comparable, on préférera toujours le modèle le plus simple, si l'on souhaite pouvoir utiliser ce modèle sur de nouvelles données totalement inconnues

Validation du modèle

- Apprentissage : construction du modèle sur un 1^{er} échantillon pour lequel on connaît la valeur de la variable cible ($\sim 70\%$)
- Test : Vérification du modèle sur un 2^{ème} échantillon pour lequel on connaît la valeur de la variable cible, que l'on compare à la valeur prédite par le modèle ($\sim 30\%$)
- Application du modèle à une nouvelle population pour laquelle on voudrait prédire la variable cible

Validation croisée

- Quand la population est trop petite pour en extraire un échantillon d'apprentissage et un autre de test, on a recours à la validation croisée (leave-one-out) :
 - La population est scindée en n échantillons de tailles égales
 - On utilise les $n-1$ premiers échantillons comme échantillon d'apprentissage et le restant comme échantillon test. On obtient ainsi un taux d'erreur en test
 - On répète $n-1$ fois la même opération en prenant à chaque fois tous les $n-1$ échantillons possibles comme échantillon d'apprentissage et le restant comme échantillon de test
 - On combine les n taux d'erreur obtenus

 *Échantillon d'apprentissage*

 *Échantillon de test*



Avantages des arbres de classification

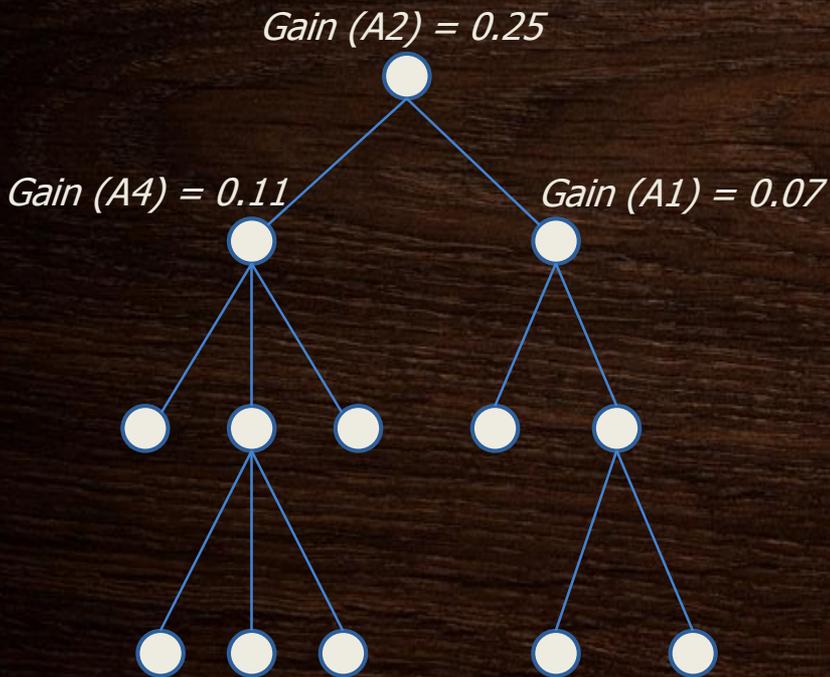
- Règles simples et facilement interprétables (contrairement aux réseaux de neurones par ex)
- Traitement sans recodification de données hétérogènes (spécialement CART)
- Traitement des valeurs manquantes
- Aucun modèle et aucun présupposé à satisfaire (méthode non paramétrique)
- Durée de traitement rapide

Désavantages

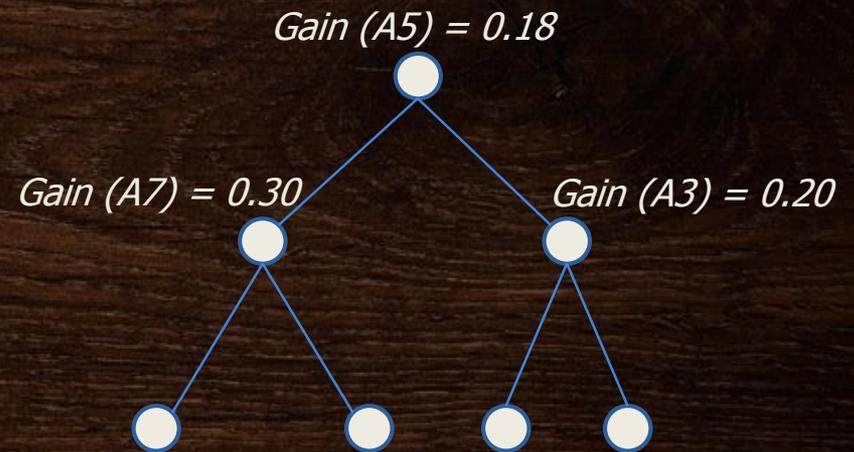
- Les nœuds du niveau $n+1$ dépendent fortement de ceux du niveau n (la modification d'une seule variable près du sommet peut entièrement modifier l'arbre)
- On choisit toujours les meilleurs attributs **locaux**, le meilleur gain d'information global n'est pas du tout garanti
- L'apprentissage nécessite un nombre suffisant d'individus
- Peu performants lorsqu'il y a beaucoup de classes
- Non incrémental: recommencer la construction de l'arbre si on veut intégrer de nouvelles données
- Sensible à de petites variations dans les données (instable)
- Trouver un arbre de décision d'erreur apparente minimale est, en général, un problème NPcomplet

Désavantages

- On choisi toujours les meilleurs attributs **locaux**, le meilleur gain d'information global n'est pas du tout garanti



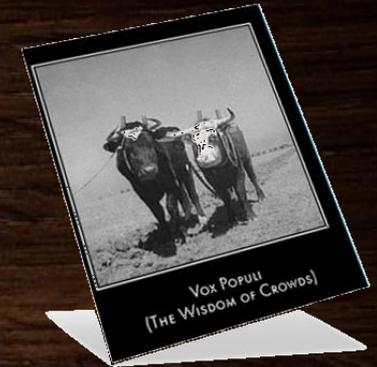
Solution choisie par l'algorithme



Solution qu'il faudrait pouvoir choisir

Fouille de Données et Forêts aléatoires

La sagesse des foules



- Francis Galton (1991)
 - En 1991 Francis Galton a observe 787 personnes essayant de deviner le poids d'un bœuf
 - Le poids de chaque individu et le vrai poids était supérieur à 1%
 - MAIS
 - La moyenne des 787 poids proposés était proche à 0,1% du vrai poids de l'animal
- Une étude à montré que si chaque individu a une probabilité supérieure à 0,5 de donner la bonne réponse, alors la moyenne des décisions de ces même individus tendra vers 100%

Vers une combinaison d'arbres de décision

- Net regain d'intérêt pour les arbres grâce aux méthodes d'agrégation de classifieurs (boosting, bagging, Random Forests, ...)
 - Permettent de tirer parti des avantages de la combinaison des prédicteurs
- Idée principale: on utilise le hasard pour améliorer les performances d'algorithmes de plus faibles performances

Historique

- Première apparition des travaux sur les forêts
 - 1992: [Ho 1992] met dans un premier temps en évidence le gain de performance que permettent d'obtenir des classifieurs qui utilisent plusieurs arbres de décision en combinaison parallèle
 - 1995: [Ho 1995] utilisation du terme de Decision Forest pour désigner de façon générale les ensembles d'arbres de décision,
 - 1996: [Breiman 1996] principe de Bagging (Bootstrap Aggregating), introduit par Breiman
 - Principe d'apprentissage d'ensemble de classifieurs qui peut être utilisé avec tout type de classifieur élémentaire,
 - L'efficacité est démontrée principalement avec les arbres de décision

Historique

- 1996: [Freund 1996] Freund et Schapire proposent leur algorithme de Boosting Adaboost, qui est depuis longtemps considéré comme le principe d'apprentissage d'ensemble de classifieurs le plus efficace
- 1996: [Amit 1996] Amit et Geman présentent le principe Random Feature Selection. Premiers principes de randomisation des arbres de décision

Bootstrap Aggregating

- Autrement connu sous le nom de BAGGING
 - Technique d'apprentissage (classification et régression) visant à
 - Améliorer la stabilité
 - Réduire la variance
 - Éviter le surapprentissage (overfitting)
 - Utilisable pour n'importe quel type de modèle
 - Utilisé surtout pour les arbres de décision
 - Principe
 - Étant donné un ensemble d'apprentissage D de taille n , on génère m nouveaux ensembles D_i de taille $n' \leq n$ en échantillonnant uniformément les exemples de D avec remise
 - Les m modèles sont entraînés en utilisant les m ensembles
 - Les réponses des modèles sont combinées (moyenne, vote, ...)

Random Feature Selection

- Ou Random Tree
- Introduit par Amit et Geman en 1996
- Création d'ensembles d'arbres aléatoires pour la reconnaissance d'écriture manuscrite
 - Trop de caractéristiques à prendre en compte → nécessité de réduire la complexité d'induction des arbres de décision
- Repris par Breiman qui lui a donné son nom

Random Feature Selection

- Idée
 - On introduit l'aléatoire dans le choix des règles de partitionnement à chaque nœud des arbres
 - Chaque règle n'est plus choisie à partir de l'ensemble des caractéristiques disponibles mais à partir d'un sous-ensemble de ces caractéristiques
- Principe
 - Sélectionner par tirage aléatoire sans remise un nombre K de caractéristiques et de choisir la meilleure des règles possibles utilisant ces K caractéristiques uniquement
 - L'algorithme d'induction correspondant est Random Tree

Définition

- Définition de Léo Breiman (2001)

- Une Forêt Aléatoire est un classifieur constitué d'un ensemble de classifieurs élémentaires de type arbres de décision, noté

$$\{ h(x, \Theta_k), \quad k = 1, \dots, L \}$$

- où $\{\Theta_k\}$ est une famille de vecteurs aléatoires indépendants et identiquement distribués, et au sein duquel chaque arbre participe au vote de la classe la plus populaire pour une donnée d'entrée x

Définition

- Force
 - Fiabilité de la forêt
- Corrélation
 - Taux de corrélation des classifieurs élémentaires entre eux
- Le taux d'erreur de la forêt est d'autant plus faible que l'ensemble des classifieurs élémentaires est fiable
- Le taux d'erreur de la forêt est d'autant plus faible que les classifieurs élémentaires sont décorrélés en terme de prédiction
- Tout l'enjeu réside donc dans le fait de produire un ensemble **d'arbres les plus performants possibles individuellement et les moins corrélés possible**

Principe des forêts aléatoires

- Avantages
 - Réduction de la variance (influence des données)
 - Simple à mettre en œuvre
 - Naturellement parallélisables
- Inconvénients
 - Temps de calcul plus important
 - Interprétabilité diminuée
- Introduction du caractère aléatoire : rendre les modèles (arbres) plus indépendant entre eux

Forest-RF

- ou Random Forests - Random Input
- Introduit par Breiman en 2001
- Algorithme de référence
- Utilise 2 principes de randomisation
 - Bagging
 - Random Feature Selection

Forest-RI - Algorithme

Algorithme 2 ForestRI

Entrée : T l'ensemble d'apprentissage

Entrée : L le nombre d'arbres dans la forêt

Entrée : K le nombre de caractéristiques à sélectionner aléatoirement à chaque nœud

Sortie : *foret* l'ensemble des arbres qui composent la forêt construite

1: **pour** l de 1 à L **faire**

2: $T_l \leftarrow$ ensemble bootstrap, dont les données sont tirées aléatoirement (avec remise) de T

3: *arbre* \leftarrow un arbre vide, *i.e.* composé de sa racine uniquement

4: *arbre.racine* \leftarrow $RndTree(\text{arbre.racine}, T_l, K)$

5: *foret* \leftarrow *foret* \cup *arbre*

6: **retour** *foret*;

Forest-R1 - Algorithme

Algorithme 3 RndTree

Entrée : n le nœud courant

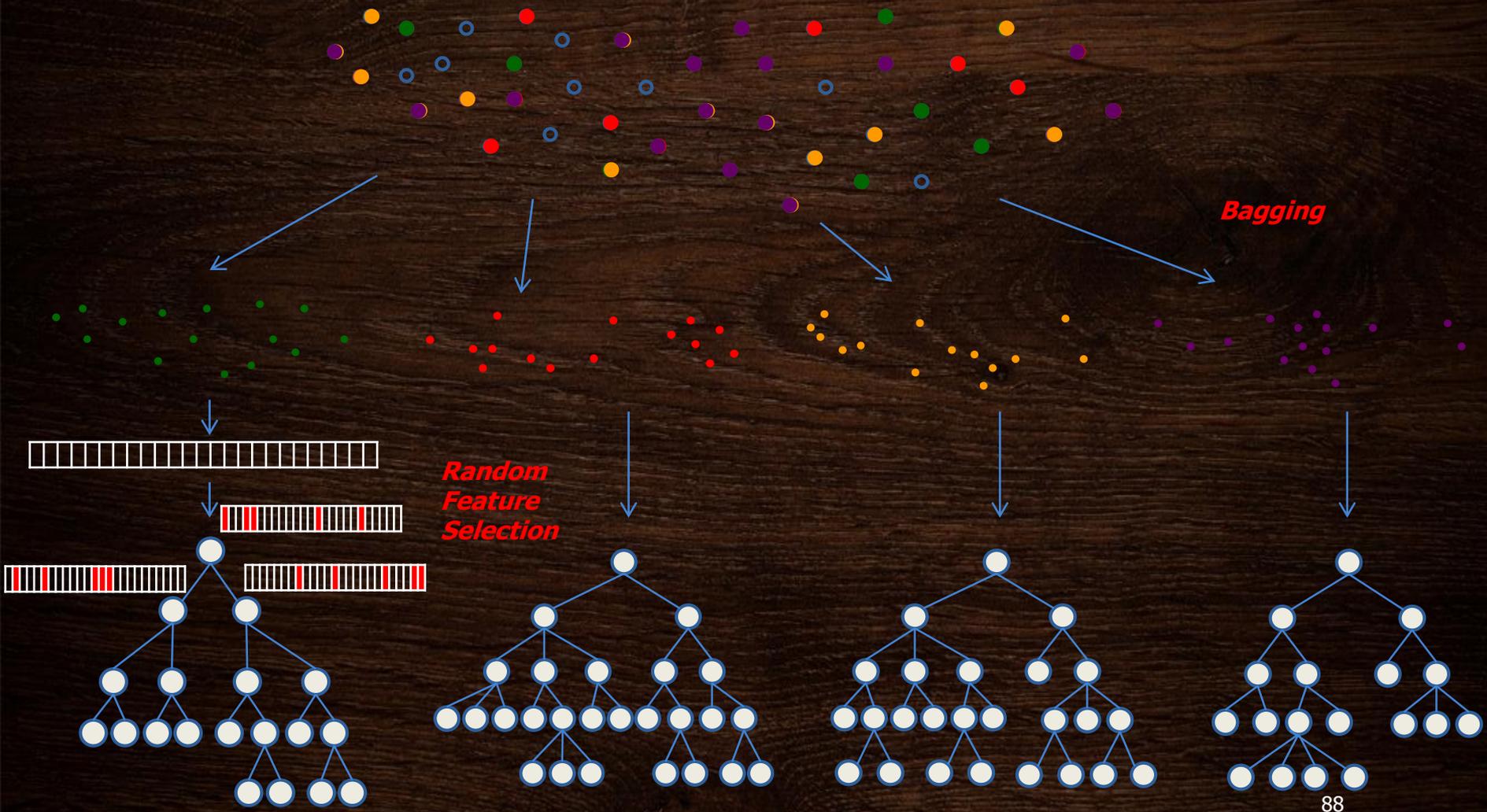
Entrée : T l'ensemble des données associées au nœud n

Entrée : K le nombre de caractéristiques à sélectionner aléatoirement à chaque nœud

Sortie : n le même nœud, modifié par la procédure

- 1: **si** n n'est pas une feuille **alors**
 - 2: $C \leftarrow K$ caractéristiques choisies aléatoirement
 - 3: **pour tout** $A \in C$ **faire**
 - 4: Procédure CART pour la création et l'évaluation (critère de Gini) du partitionnement produit par A , en fonction de T
 - 5: $partition \leftarrow$ partition qui optimise le critère Gini
 - 6: $n.ajouterFils(partition)$
 - 7: **pour tout** $fil \in n.noeudFils$ **faire**
 - 8: $RndTree(fil, fil.donnees, K)$
 - 9: **retour** n
-

Bagging et Random Feature Selection



Données importantes

- Les données à paramétrer
 - Nombre d'arbres de la forêt
 - Profondeur des arbres
 - Nombre de paramètres considérés pour construire chaque arbre
 - Choix des paramètres parmi l'ensemble des paramètres initiaux

Forest-RC (ou Random Forests - Random Combinations)

- Variante de Forest-RI, Définie par Breiman en 2001
- Permet de palier certaines limites de Forest-RI quand il y a peu de caractéristiques
 - Choisir pour K une fraction relativement importante de l'ensemble de ces caractéristiques peut entraîner une diminution de la diversité dans l'ensemble
- Au lieu de sélectionner la meilleure caractéristique parmi les K choisies au hasard, on calcule K combinaisons linéaires de F caractéristiques (avec F pas nécessairement égal à K)

Forest-RC (ou Random Forests - Random Combinations)

- Principe
 - On choisit aléatoirement F caractéristiques puis on les combine à l'aide de K jeux de coefficients également choisis aléatoirement dans l'intervalle $[-1, 1]$
 - Une fois les K combinaisons linéaires générées, on choisit celle qui offre le meilleur partitionnement
- Plus adapté que Forest-RI aux problèmes dont l'espace de description est de dimension faible
- Mais le gain de performance est souvent peu significatif pour un algorithme plus difficile à maîtriser que Forest-RI

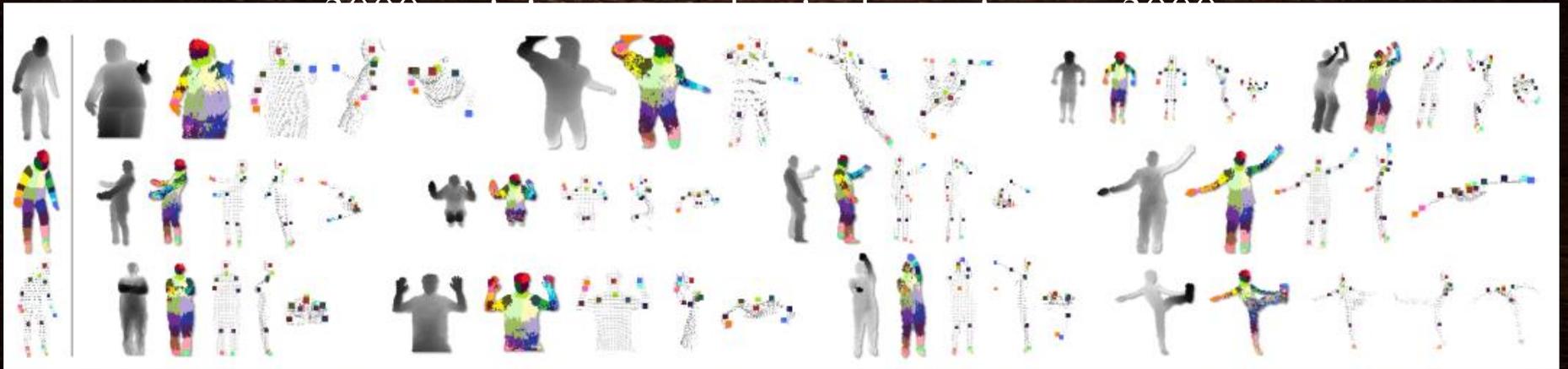
D'autres algorithmes

- Extremely Randomized RF (Geurts 2006)
- Weighted Voting Random Forest (Robnik-Sikonja 2004)
- Balanced Random Forests (Chen 2004)
- Weighted Random Forests (Chen 2004)
- Rotation Forest (Rodriguez 2006)
- Probabilistic Random Forest (Breitenbach 2002)
- "Meta" Random Forests (Boinee 2005)
 - Bagged Random Forest (BgRF)
 - AdaBoosted Random Forest (ABRF)
- On-line Random Forest,
- Hierarchical Random Forest, ...

Quelques exemples concrets

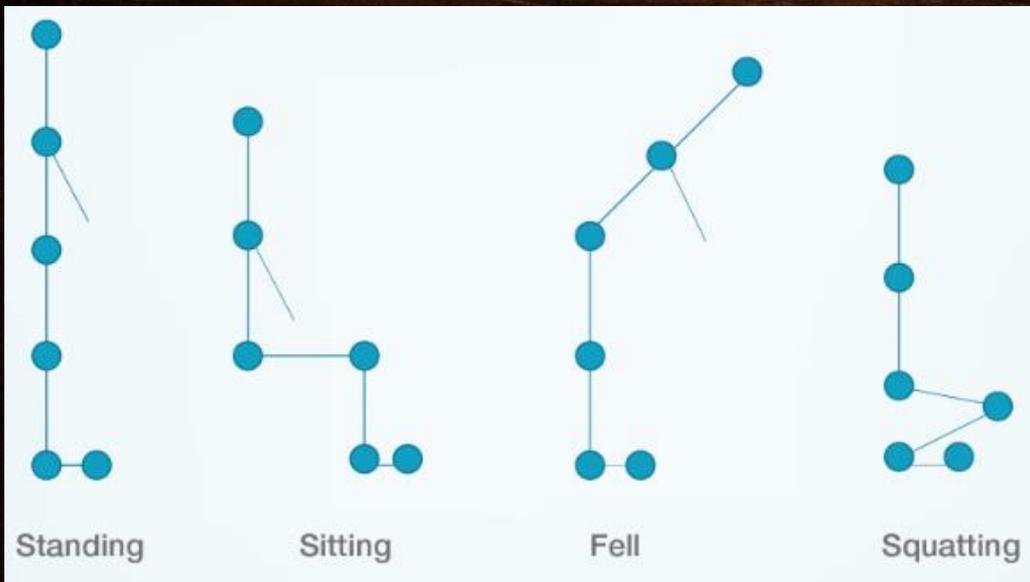
Implémentation des RF dans la Kinect

- Real-Time Human Pose Recognition in Parts from Single Depth Images (2011)
 - 3 arbres, profondeur de 20, 300 000 images par arbre, ~4 000 caractéristiques



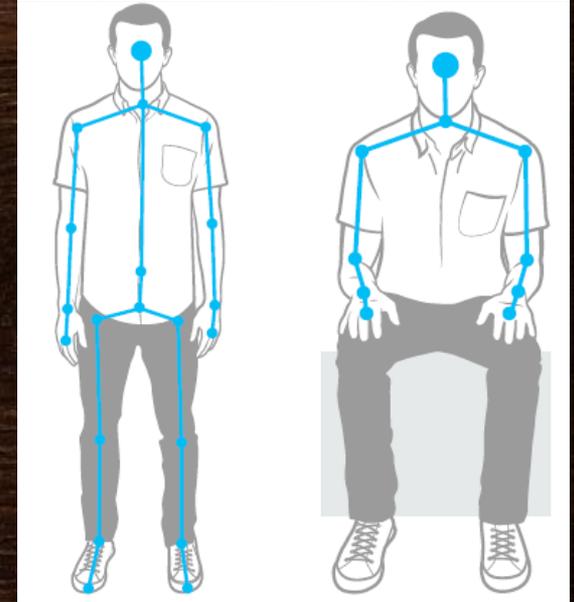
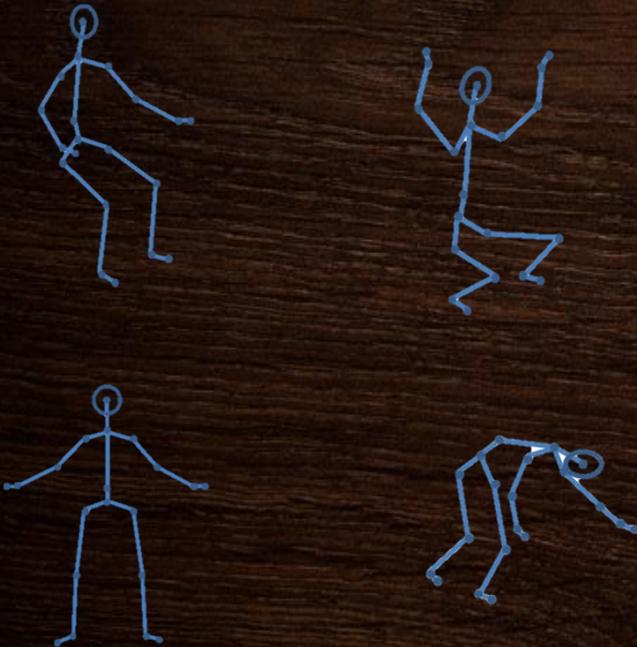
Reconnaissance de positions statiques

- Domaine de la reconnaissance de positions et de mouvements à partir de la Kinect
 - PFE Stéphanie Lopez – Perside Gbehounou



Reconnaissance de positions statiques

- À partir de la Kinect, squelette
 - 20 jointures



Reconnaissance de positions statiques

- Base d'apprentissage
 - 11 classes
 - 1690 exemples

	Positions	Annotations	P O S i t i o n s
Debout	Main droite levée	0	
	Main gauche levée	1	
	2 mains levées	2	
	2 mains baissées	3	
		4	
	Main droite levée	5	
	Main gauche levée	6	
	2 mains levées	7	
	2 mains baissées	8	
Accroupi	Main droite levée		
	Main gauche levée	9	
	2 mains levées		
	2 mains baissées	10	

Reconnaissance de positions statiques

- Question à se poser
 - Quel vecteur utiliser pour créer la forêt?
- Première idée
 - Coordonnées X, Y et Z de chaque jointure
 - Vecteur de 60 caractéristiques
- Est-ce suffisant?



Reconnaissance de positions statiques

- Pour construire la forêt, on prend, pour choisir chaque nœud, \sqrt{N} caractéristiques (N étant le nombre total de caractéristiques)
- Si on considère 60 caractéristiques au total, on obtient entre 7 et 8 caractéristiques
 - Pas suffisant
- Il faut donc trouver le moyen de renforcer le vecteur de caractéristiques
 - Mais ces nouvelles caractéristiques ne doivent pas être corrélées
- Comment les choisir?

Reconnaissance de positions statiques

- On considère toutes les distances entre chaque jointure
 - $20 * 19$ valeurs supplémentaires
- On rajoute tous les angles possibles entre les jointures
 - $20 * 19 * 18$ valeurs
- Au total on considère un vecteur de 7280 caractéristiques
- Démo

Reconnaissance d'actions

- Quelques exemples d'actions qu'on souhaite reconnaître
 - Téléphoner
 - Écrire à l'ordinateur
 - Boire
 - Écrire au tableau
 - Etc.



Reconnaissance d'actions

- Problème beaucoup plus complexe que la reconnaissance de positions statiques

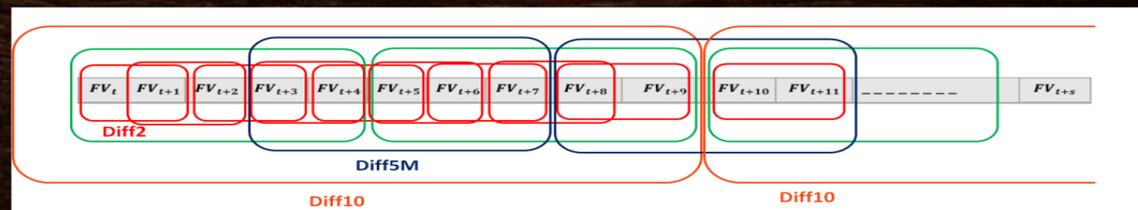
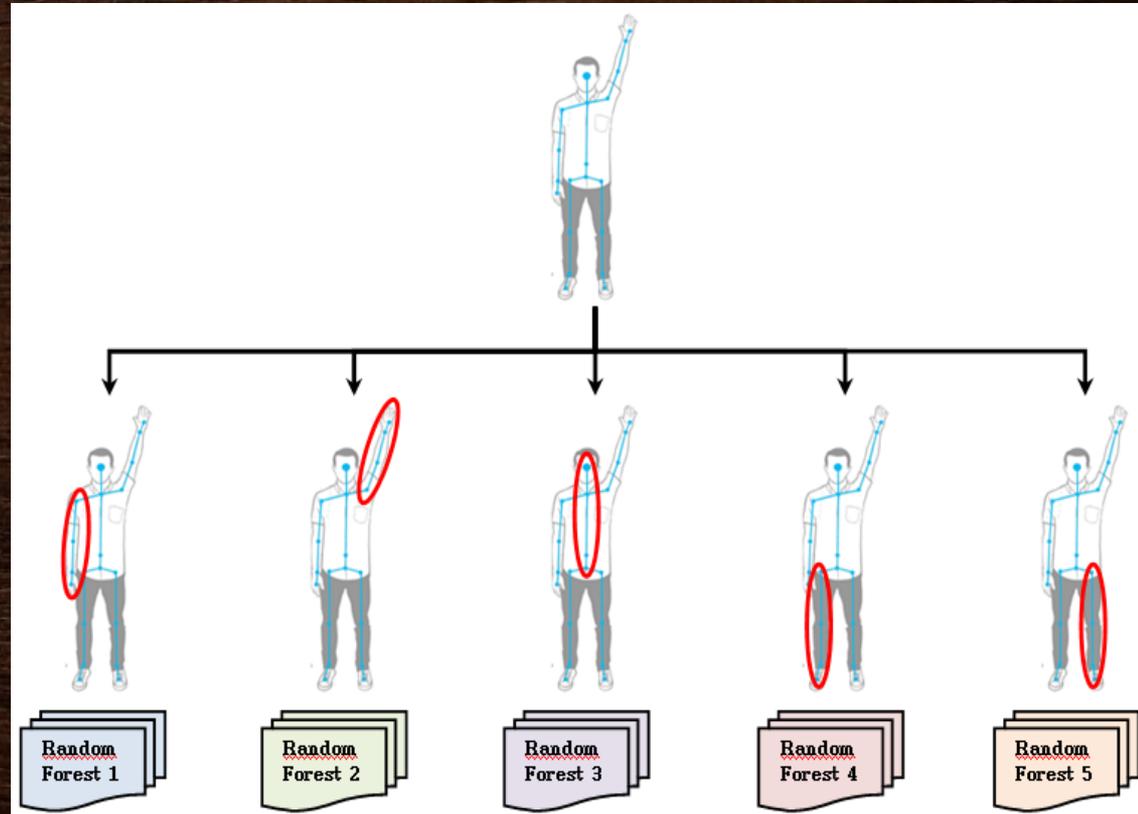
- On peut boire
 - Debout
 - Assis
 - Rapidement
 - Lentement



- On ne peut pas dire qu'une action est une suite fixe de positions statiques
 - Possibilité d'avoir des mouvements parasites
- Quel vecteur utiliser? Problème encore ouvert

Reconnaissance d'action

- Pistes :
 - Parti pris: garder uniquement les données du squelette pour avoir des données les plus simples possibles
 - Notion de fenêtres glissantes
 - Vecteur de plus de 46 000 caractéristiques
 - Utiliser des Random Forest hiérarchiques
 - Comment ensuite combiner les sous-forêts?
 - Comment combiner plusieurs Kinects?



Assemblage de composants

- Partenariat avec l'équipe Rainbow, plateforme Wcomp
 - PFE Marco Carta Gulung et Ahmed Ouertani
 - PFE et stage Master de Luis Gioanni
- Hypothèses de travail
 - Ensemble de services et dispositifs disponibles avec leur méta-données
 - L'infrastructure est dynamique
 - Ces objets communicants peuvent apparaître et disparaître de manière opportuniste
- Objectif
 - Apprendre « leur comportement » et l'influence ou l'impact de ce comportement sur la configuration finale de l'environnement

Assemblage de composants

- À quoi ça peut servir?
 - Monitoring d'activité à la maison
 - On veut apprendre à partir d'objets communicants tels qu'un téléphone, un chargeur de téléphone, un matelas muni de capteurs, une Kinect, etc.

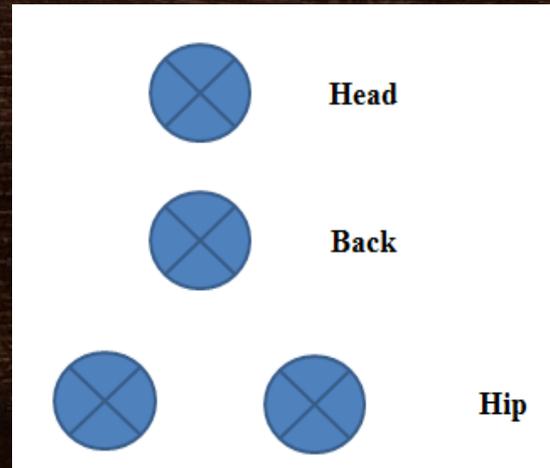
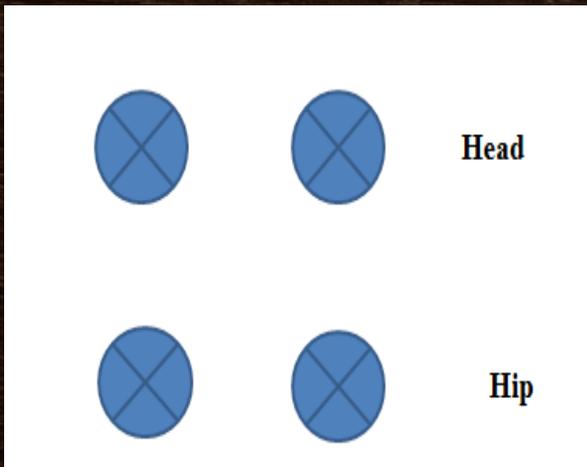
Assemblage de composants

- Travail préliminaire
 - Un matelas muni de capteurs
 - Différentes positions identifiées sur le matelas (couché au milieu, couché sur le côté droit du matelas, etc.).
 - Objectif final = monitoring de bébés couchés sur un matelas
 - Hypothèse de départ
 - 2 dispositions possibles de capteurs
 - Trouver la meilleur dispositin



Assemblage de composants

- Vecteur représentant les données
 - Données issues des 4 capteurs
- Problème
 - Vecteur bien trop petit pour que les RF soient vraiment efficaces

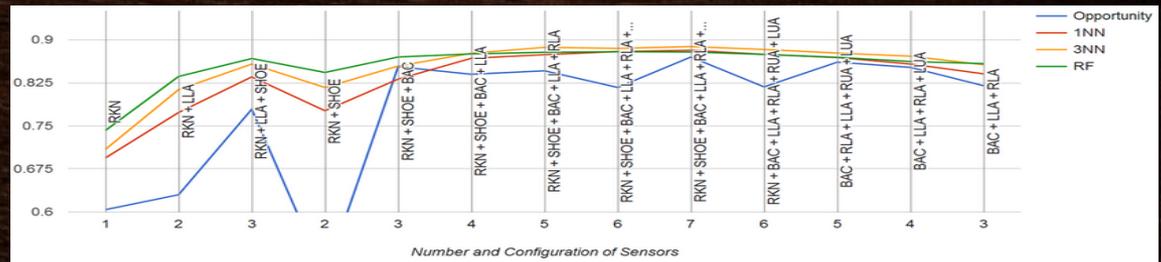
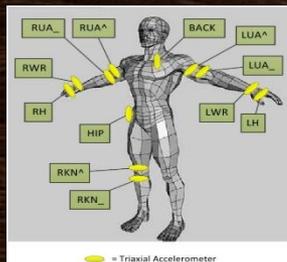
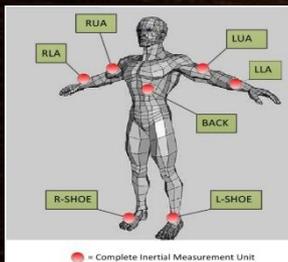


Assemblage de composants

- Solution
 - Renforcer le vecteur de caractéristiques avec de nouvelles caractéristiques non corrélées entre elles
 - On rajoute les moyennes 2 à 2 et 3 à 3 des différents capteurs
 - On obtient un vecteur avec 40 valeurs
- Pas encore suffisant pour avoir des valeurs cohérentes
 - Aucune logique dans les forêts créées, pas de convergence quand on rajoute des arbres
 - Les SVM offrent des résultats bien plus valables dans ce genre de problèmes

Assemblage de composants

- Hypothèses de travail
 - Ensemble de capteurs portés
 - Infrastructure dynamique : Apparition et disparition opportuniste des capteurs
- Apprendre des informations de haut niveau à partir une configuration qui évolue
 - Monitoring de personnes à domicile
- Comparaison des forêts avec d'autres classifieurs avec des configurations changeantes
 - Plus stables, meilleurs avec moins de capteurs



Reconnaissance d'action à partir de capteurs

- Chaque capteur apprend de son côté
- On combine les décisions des capteurs présents à un instant donné
- Questions soulevées
 - Quel est le bon algorithme d'apprentissage?
 - Quel vecteur créer?
 - Coordonnées des capteurs, moyenne, variance, notion de fenêtre glissante, ...
 - Comment combiner/composer les décisions de chaque capteur?
- Futurs travaux
 - Comment adapter le choix de l'algorithme d'apprentissage en fonction de la configuration des capteurs?

Assemblage de composants

- Questions ouvertes auxquelles on doit répondre
 - Comment choisir le meilleur algorithme d'apprentissage?
 - Que se passe-t-il si on a appris avec 5 composants et qu'un des composants disparaît?
 - Que se passe-t-il si on a appris avec Kinect, matelas, montre, lampe et qu'une 2ème montre apparaît?
 - Que se passe-t-il si on a appris avec la version 5 de l'Iphone et que la version 6 apparaît?
 - Comment la forêt peut-elle le plus rapidement se mettre à jour?
 - Etc.

Classification de texte courts

- Collaboration avec une entreprise qui fournit un outil pour donner des questionnaires à un grand nombre de personnes et qui veut regrouper les réponses effectuées
- Après chaque question posée, les réponses sont collectées et classées par le système
- Un pilote modifie cette classification à la main pendant que la question suivante est posée
- L'optimisation de l'algo de classification est donc un point essentiel



✓ Valider

⊘ Deselect

⚙ Run clustering

🔍 Answers : 21 (new : 0)

🔒 Freeze groups

🍃 Add new answers

n°28 : Que vous manque-t-il pour mieux faire ? Pour l'Entreprise Régionale CentreLoire

Coopération entre services

Une meilleure cohésion @

la confiance @

mutualiser @

Coopération entre services @

Des fonctions supports au service des opérationnels @

le travail d'équipe @

Décloisonner @

Une culture d'entreprise (Fierté d'appartenir à la famille Lyonnaise des Eaux) @

Coopération entre services(38%)

Autre(19%)

Homogénéisation de pratiques(14%)

Une stratégie claire(9%)

Un peu plus de moyens(9%)

Un poids dans les décisions(9%)

+++

Autre

adhésion d'une partie des agents de terrain et d'une partie d'encadrants @

un service client performant @

audit des contrats déficitaires @

Qu'on accepte qu'on a des choses que lesquelles on doit progresser @

Homogénéisation de pratiques

Homogénéisation de pratiques @

simplification du reporting interne @

^ v	09		# 7. Vous sentez-vous armé pour faire face aux enjeux de demain ,positionnez-vous sur un curseur de 0 à 10## Pour l'Entreprise Régionale Centre Loire :		 Editer	 Publier	 Suivre	Other ▾	
^ v	010		## Pour votre équipe :		 Editer	 Publier	 Suivre	Other ▾	
^ v	011		## Pour vous :		 Editer	 Publier	 Suivre	Other ▾	
^ v	017		# Vous sentez-vous arme pour faire face aux enjeux de demain ; positionnez-vous sur un curseur de 0 a 10## Pour l'Entreprise Regionale Centre Loire :	 Prêt à afficher	 Editer	Synthèse	Aperçu	 Diffuser	Other ▾
^ v	018		## Pour votre equipe :	 Prêt à afficher	 Editer	Synthèse	Aperçu	 Diffuser	Other ▾
^ v	019		## Pour vous :	 Prêt à afficher	 Editer	Synthèse	Aperçu	 Diffuser	Other ▾
^ v	020		# 1. Vous allez prochainement participer au 1er séminaire résidentiel COMEXLyonnaise des Eaux Centre Loire à l'île de Ré, dans quel état d'esprit êtesvous ?## Une idée par champ	 Prêt à afficher	 Editer	 Gérer groupes	Aperçu	 Diffuser	Other ▾

Classification de textes courts

- Problème

- Textes très courts: 2/3 mots
- Pas de contexte
- Un pilote qui améliore la classification initiale
- Vecteur représentatif vide
 - ~20 000 colonnes remplies de 0

- Solutions

- Enrichir le vecteur représentatif avec des mots liés sémantiquement
- Réduire le vecteur représentatif
- Intégrer la sémantique dans la construction de la forêt
- Apprendre les actions du pilote

Home Valider Deselect Run clustering Answers : 21 (new : 0)

n°28 : Que vous manque-t-il pour mieux faire ? Pour l'Entreprise Régionale CentreLoire

Coopération entre services

- Une meilleure cohésion @
- la confiance @
- mutualiser @
- Coopération entre services @
- Des fonctions supports au service des opérationnels @
- le travail d'équipe @
- Décloisonner @
- Une culture d'entreprise (Fierté d'appartenir à la famille Lyonnaise des Eaux) @

Autre

- adhésion d'une partie des agents de terrain et d'une partie d'encadrants @
- un service client performant @
- audit des contrats déficitaires @
- Qu'on accepte qu'on a des choses que lesquelles on doit progresser @

Homogénéisation de pratiques

- Homogénéisation de pratiques @
- simplification du reporting interne @

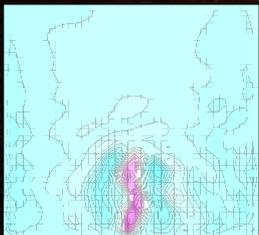
116

Prédiction du courant en haute mer

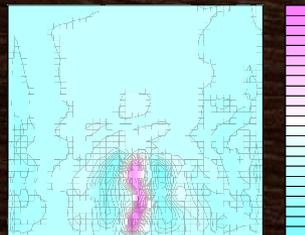
Domain $\mathcal{D} = \{(x, y, z) \mid x, y \in \mathbf{R}, t \geq 0, -H \leq z \leq \zeta\}$.

$$\begin{aligned} \partial_t U_h + U \cdot \nabla U_h - \nu \Delta U_h + 2(\vec{\Omega} \wedge U)_h + \nabla_h p &= 0, \\ \partial_z p &= -\rho g, \\ \nabla_h \cdot U_h + \partial_z w &= 0. \end{aligned}$$

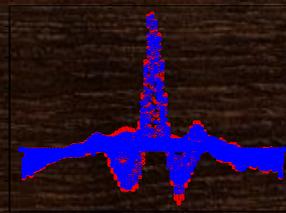
- Modèle mathématique complexe qui génère des données de courant vs Random Forest
- 1 an de données générées
- Prédictions correctes $> 94\%$



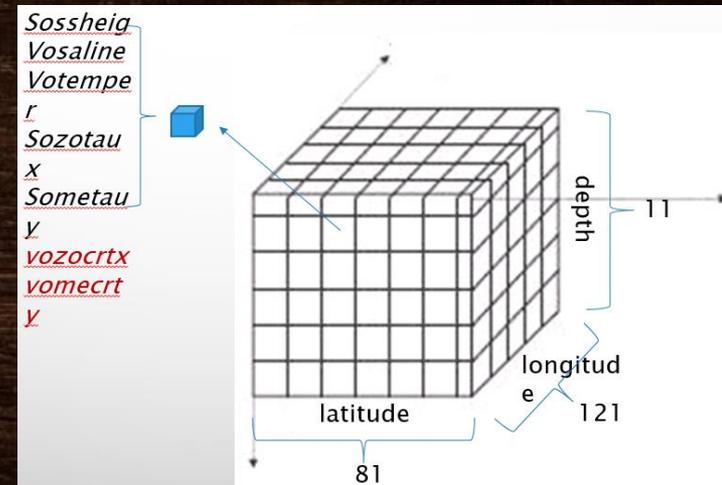
Modèle
mathématique



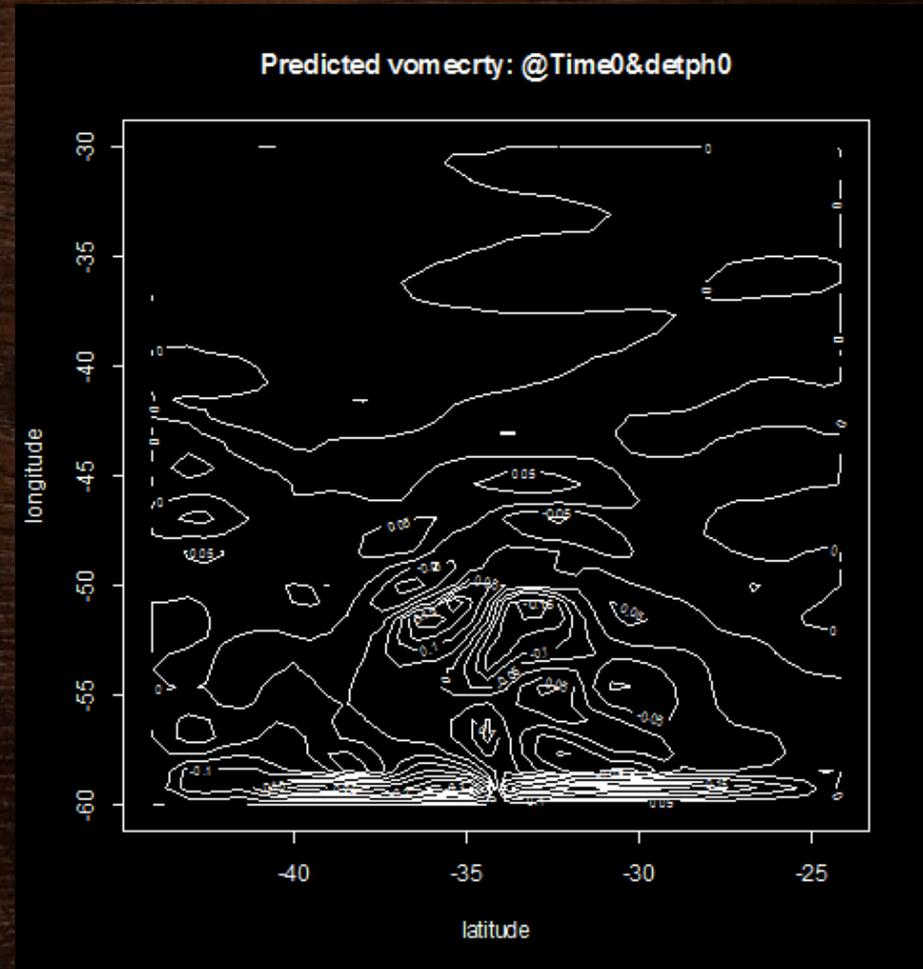
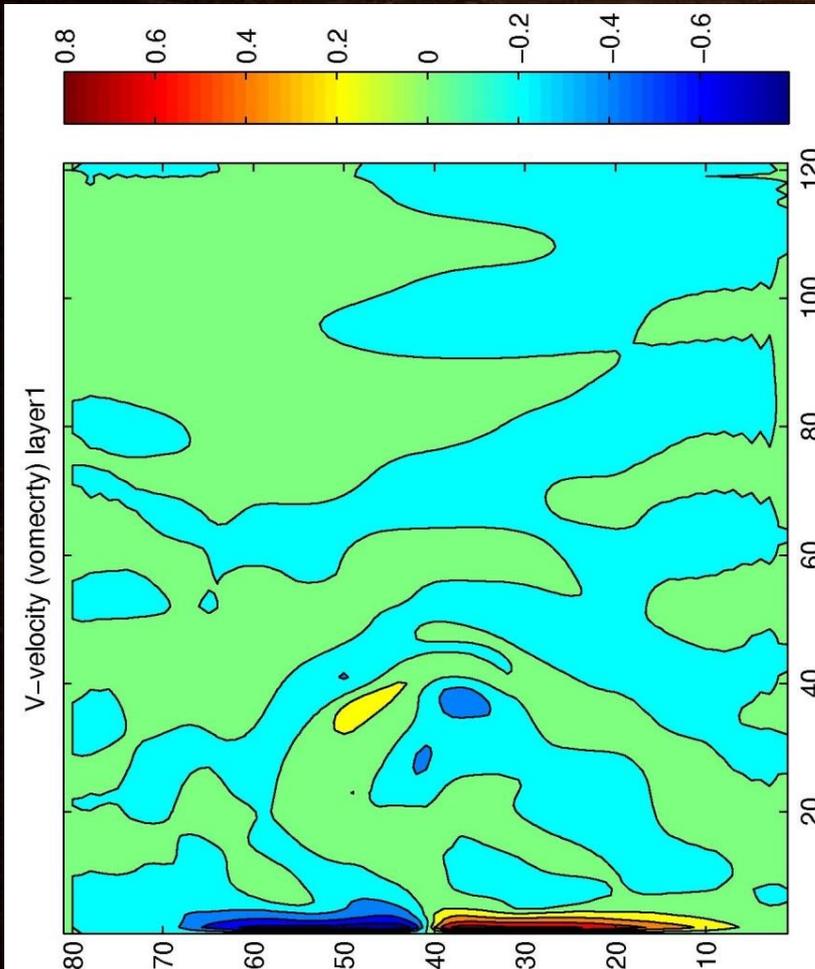
Random Forest



Blues: predictions of testing set
Reds: observations of testing set



Prédiction de courant

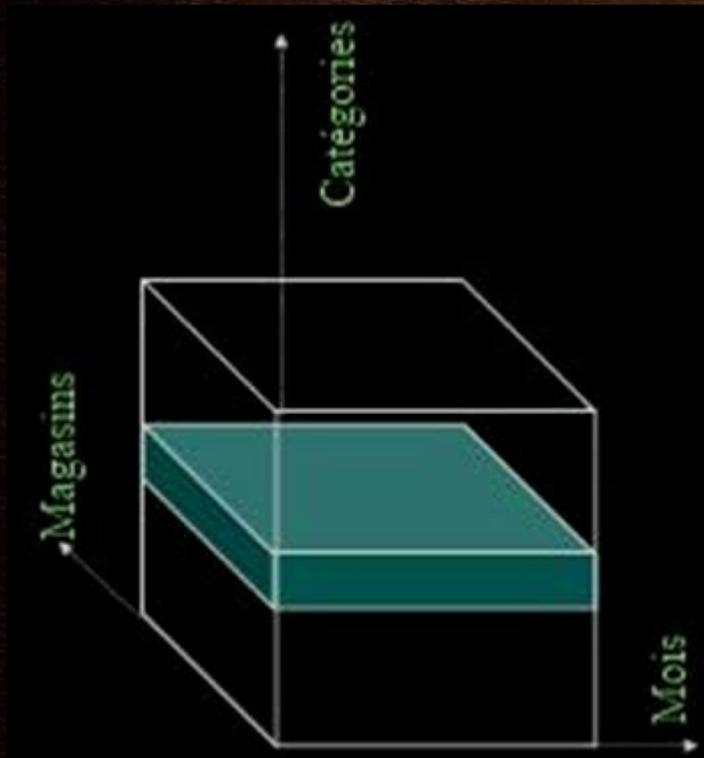


Prédiction de courant

- Données prises en compte pour l'apprentissage
 - time_counter: from 0 to 359, represents 360 recording moments, during one year
 - depth: from 0 to 10, represents 11 depth levels of the ocean. in fact, depth_t=depth_v=depth_u=depth
 - nav_lat: dim = 81×121 , the coordinate of grid along the latitude
 - nav_lon: dim = 81×121 , the coordinate of grid along the longitude
 - sossheig: dim = $360 \times 81 \times 121$, pressure of the surface of ocean
 - vosaline: dim = $360 \times 11 \times 81 \times 121$, the salinity
 - votemper: dim = $360 \times 11 \times 81 \times 121$, the temperature
 - sozotaux: dim = $360 \times 81 \times 121$, wind stress along i-axis
 - sometauy: dim = $360 \times 81 \times 121$, win stress along j-axis
 - vozocrtx: dim = $360 \times 11 \times 81 \times 121$, zonal current
 - vomecrty: dim = $360 \times 11 \times 81 \times 121$, meridional current

Prédiction de courant

- Différentes actions possibles
 - On apprend sur toutes les données et on essaye de prédire une couche de données à l'instant $t+1$

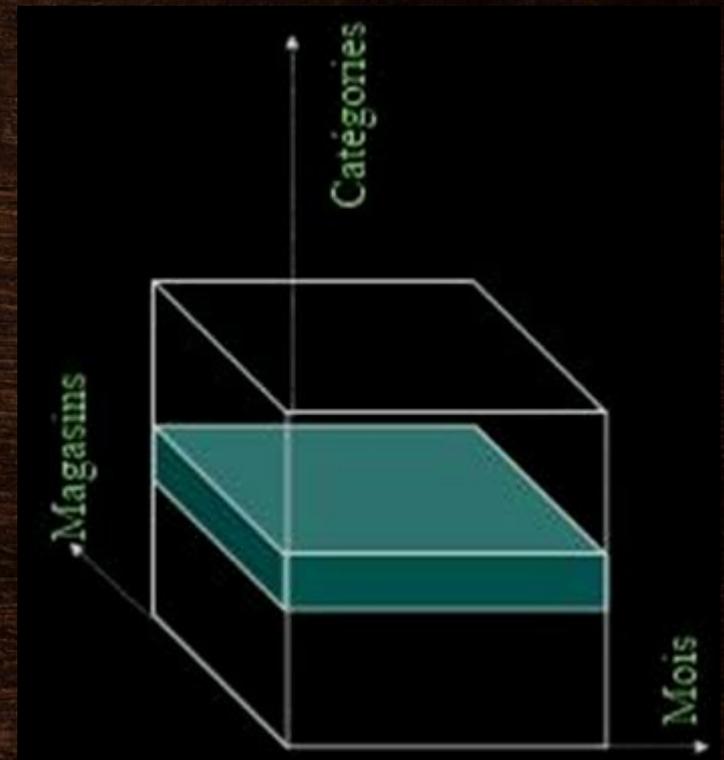
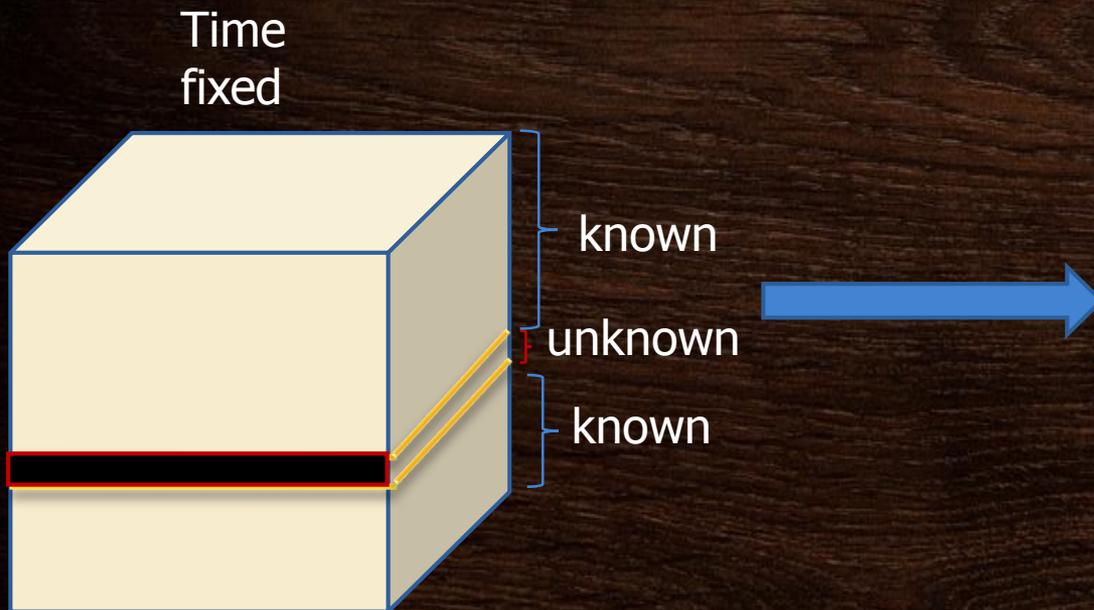


$T = T+1$



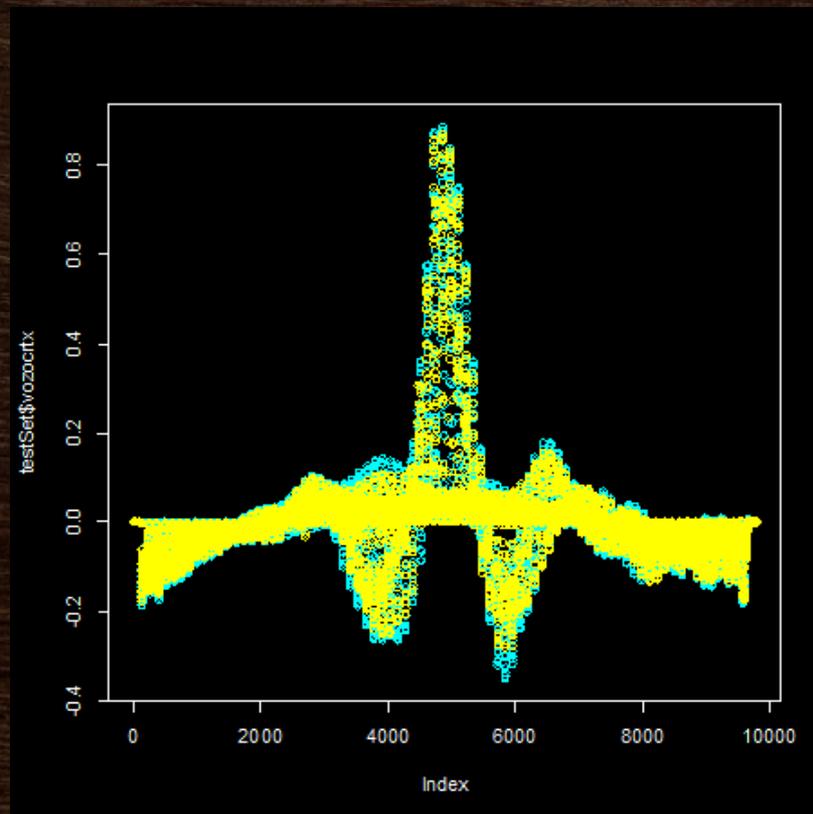
Prédiction de courant

- Différentes actions possibles
 - On apprend avec 80 couches et on essaye de prédire la 81ème



Prédiction de courant

- Des résultats
 - On apprend avec 10 instants consécutifs et on prédit sur le 11ème

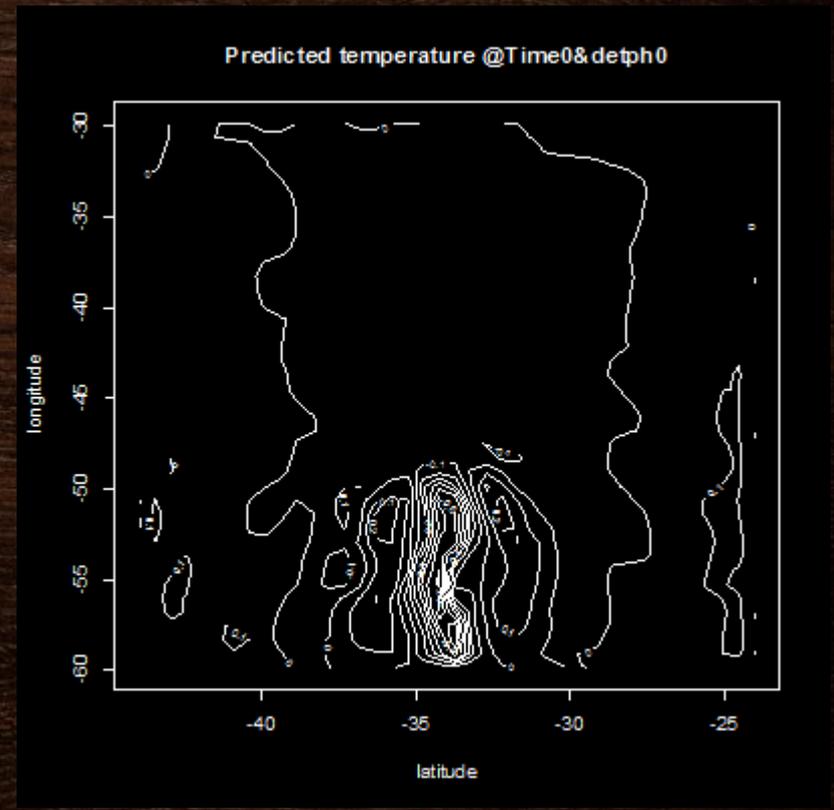
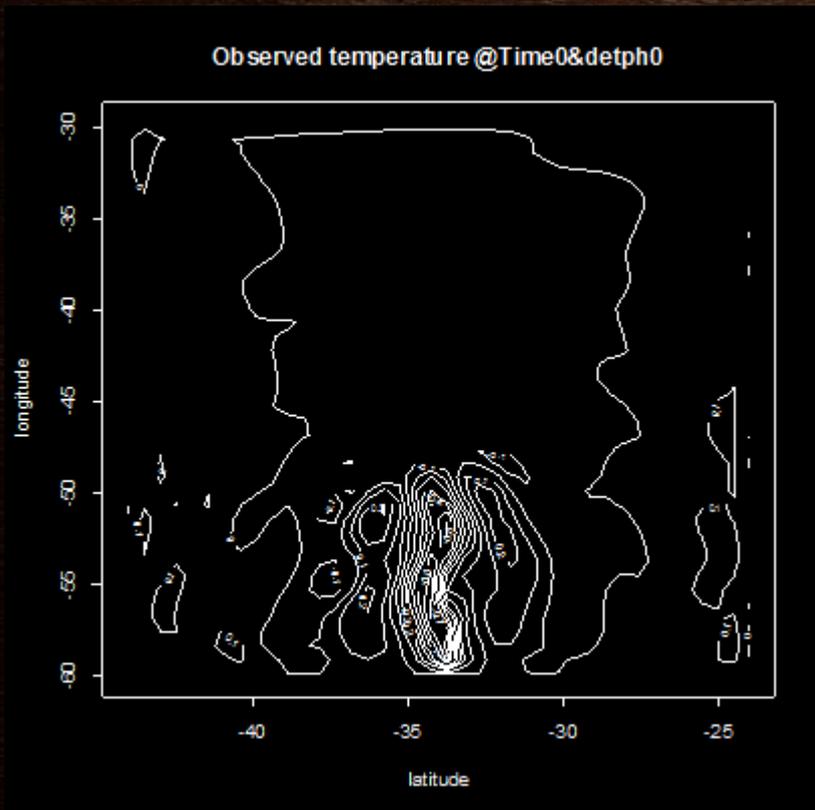


Blues: predictions of testing set

Reds: observations of testing set

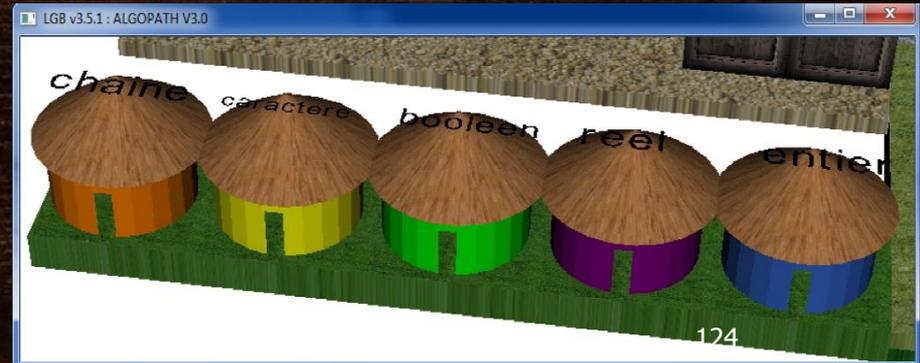
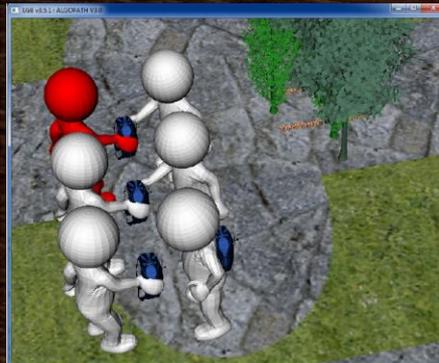
Prédiction de courant

- Prédiction de la température



Apprentissage dans les serious games

- Logiciel AlgoPath : apprentissage de l'algorithmique
- Capture des actions des élèves
- Objectifs
 - Évaluer automatiquement le niveau d'un élève
 - Proposer un parcours éducatif adaptatif
 - Comparer clustering et apprentissage supervisé



D'autres travaux



- Parc National de Port Cros
 - Besoin : prédiction en temps réel du seuil optimal de fréquentation des îles
 - À partir de données de capteurs
 - Température, courant marin, pluviométrie, etc.
 - Données issues de questionnaires
 - Données de comptage
 - 10 ans de données, plus de 8 Go de données textuelles
 - Trouver des liens entre des attributs (une centaine d'attributs)
 - Objectif à terme
 - Doter les visiteurs de logiciels sur leur smartphone ou sur des équipements de prêt en contrepartie de services offerts (guide de visite, ...) pour collecter des données environnementales géolocalisées