Classification et caractérisation

- Classification
  - arbre de décision
  - classificateur Bayésien
  - réseau de neurones
  - …

- Caractérisation
  - Description des concepts
  - Généralisation des données
  - Induction orientée attribut
  - Analyse de la pertinence des attributs
  - Mesure de pertinence
  - Caractérisation analytique

Classification vs. Prédiction

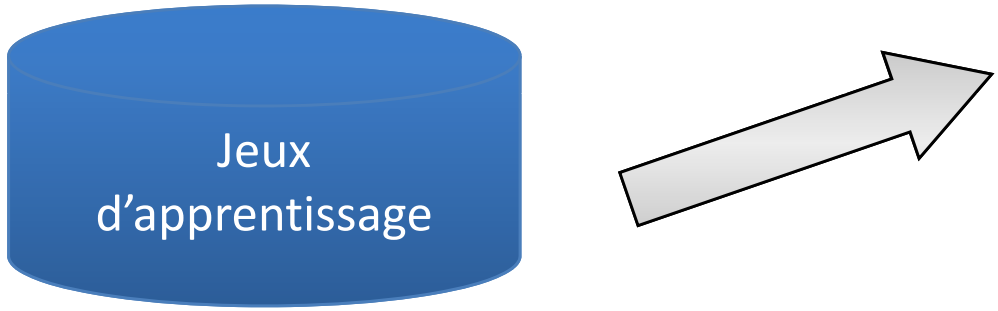
- Classification:
  - prédit la catégorie d'un objet
  - construit un modèle basé sur un jeu d'apprentissage et des valeurs (nom des catégories) et l'utilise pour classer des données nouvelles
- Prédiction:
  - modélise des données numériques pour prédire des valeurs inconnues ou manquantes
- Applications
  - diagnostique médical
  - séquences codantes
  - structure secondaire, tertiaire
  - …

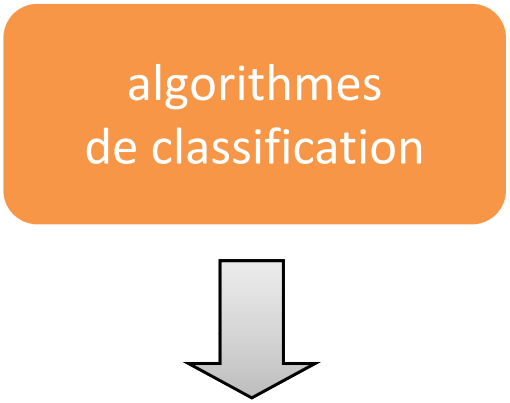
- Méthodes
  - arbre de décision
  - classificateur bayésien
  - réseau de neurones
  - plus proches voisins
  - règles d'association
  - algorithme génétique
  - machine à vecteur de support
  - modèle de Markov
  - …

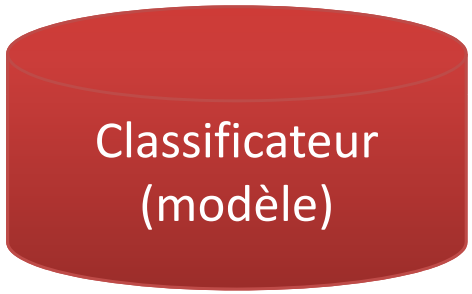
Classification : 2 étapes

- Construction du modèle
  - chaque objet appartient à une classe connue
  - jeu de données d'apprentissage : ensemble des objets utilisés pour la construction du modèle
- Utilisation du modèle pour classer des objets nouveaux ou inconnus
  - estimation de la précision du modèle
    - les classes connues du jeu d'apprentissage sont comparées à celles prédites
    - précision : pourcentage d'objets de jeu de test correctement classés
    - le jeu de test est indépendant du jeu d'apprentissage sinon risque de biais

Classification : construction du modèle



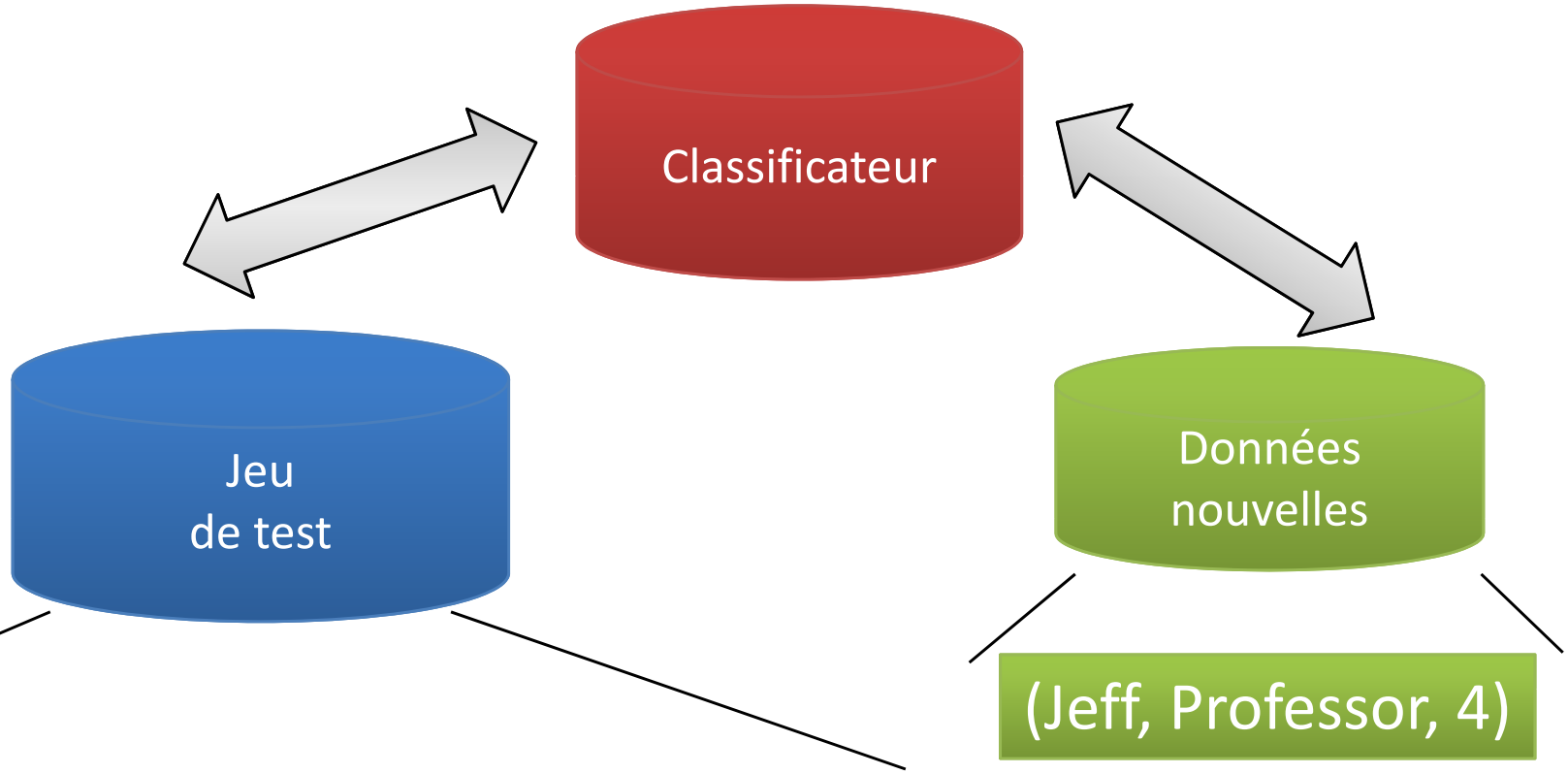




NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = 'professor'
OR years > 6
THEN tenured = 'yes'

Classification : utilisation du modèle



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Apprentissage supervisé ou non

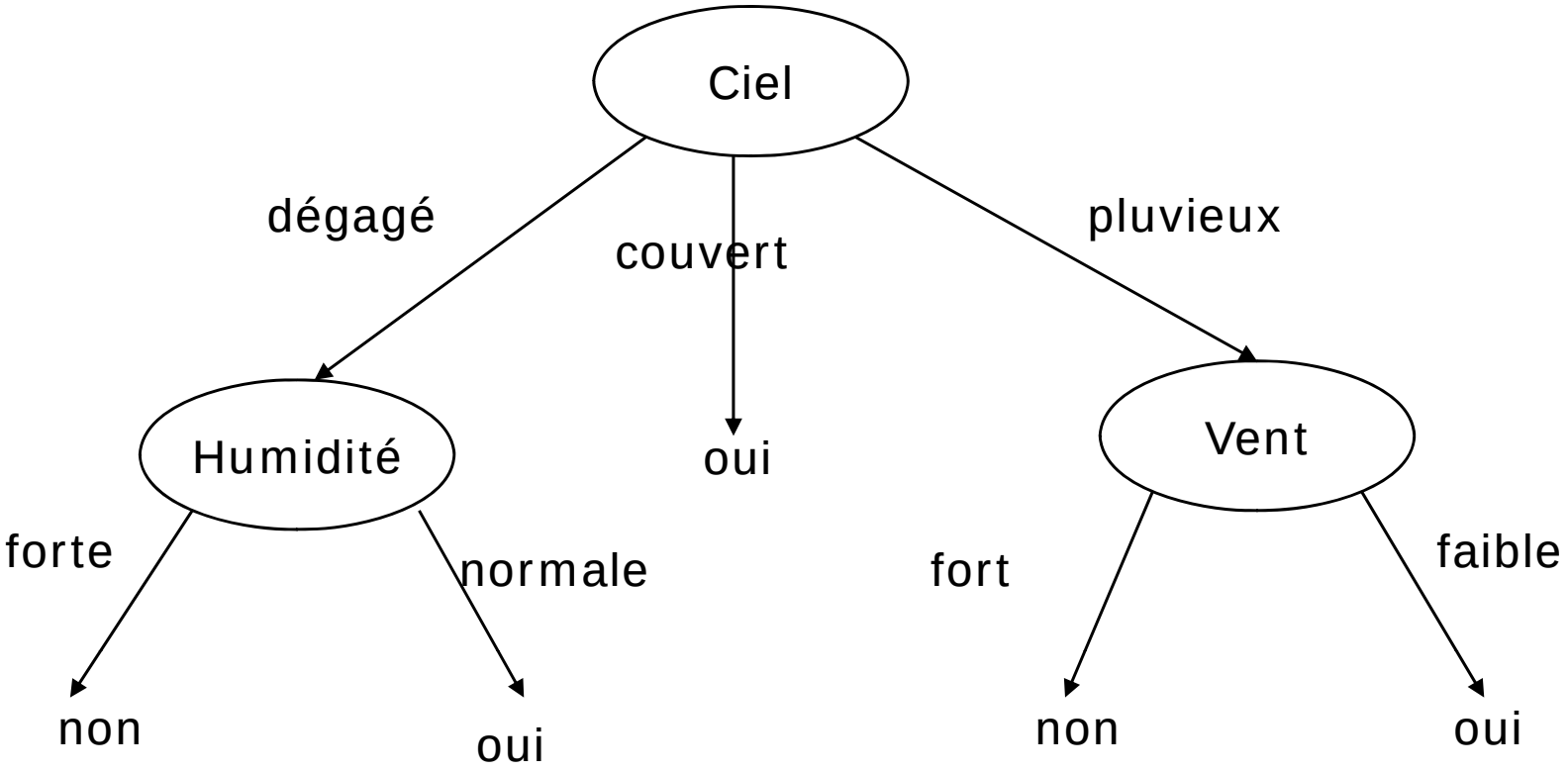
- Apprentissage supervisé (classification)
  - supervision : le jeu de données d'apprentissage fournit les classes des objets
  - les nouveaux objets sont classés en fonction du jeu d'apprentissage

- Apprentissage non supervisé (clustering)
  - Pas de classes définies
  - Étant donné un ensemble de mesures, d'observations, etc., essayer d'établir l'existence de classes ou de clusters dans les données

Considérations pour la classification

- Nettoyage des données
  - pré-traiter les données pour réduire le bruit et gérer les valeurs manquantes
- Analyse de pertinence
  - supprimer les attributs non pertinents ou redondants
- Transformation des données
  - généraliser ou normaliser les données
- Précision de la prédiction
- Efficacité et mise à l'échelle
  - pour construire le modèle
  - pour l'utiliser
- Robustesse
  - tolérance au bruit et aux données manquantes
- Interprétabilité
  - compréhension des données via le modèle
- Qualité des règles
  - taille de l'arbre de décision
  - règles de classification compactes

- Arbre de décision
  - nœuds internes : test sur un attribut
  - branches : résultat d'un test / valeur de l'attribut
  - feuilles : classe

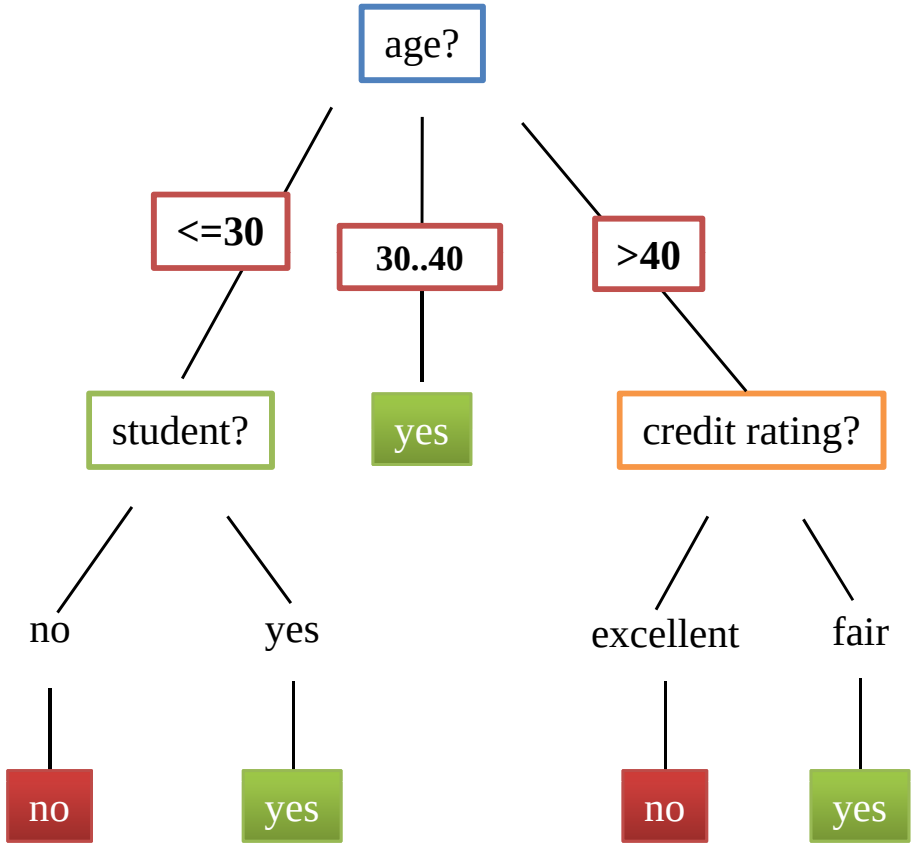


Classification par arbre de décision

- Génération de l'arbre en 2 étapes
  - Construction
    - au départ, tous les exemples du jeu d'apprentissage sont à la racine
    - partitionne récursivement les exemple en sélectionnant des attributs
  - Élagage
    - identification et suppression des branches correspondant à des exceptions ou du bruit
- Utilisation de l'arbre
  - teste les valeurs des attributs avec l'arbre de décision

Exemple : achète un ordinateur ?

age	income	student	credit_rating
<=30	high	no	fair
<=30	high	no	excellent
31…40	high	no	fair
>40	medium	no	fair
>40	low	yes	fair
>40	low	yes	excellent
31…40	low	yes	excellent
<=30	medium	no	fair
<=30	low	yes	fair
>40	medium	yes	fair
<=30	medium	yes	excellent
31…40	medium	no	excellent
31…40	high	yes	fair
>40	medium	no	excellent



Algorithme pour l'induction d'arbre de décision

- Algorithme glouton
  - approche descendante récursive *diviser pour régner*
  - au départ, tous les objets sont à la racine
  - attributs catégoriels (les valeurs continues sont discrétisées à l'avance)
  - les exemples sont partitionnés récursivement par la sélection d'attribut
  - les attributs sont sélectionnés sur la base d'une heuristique ou d'une mesure statistique

- Conditions d'arrêt
  - tous les exemples pour un nœud appartiennent à la même classe
  - plus d'attribut pour partitionner, dans ce cas la classe attribuées correspond à celle la plus représentée
  - plus d'exemple à classer

Mesure pour la sélection d'attribut. Exemple : gain d'information (ID3 et c4.5)

- Sélectionne l'attribut qui a le gain le plus élevé
- Soient 2 classes P et N
  - Soit un jeu d'apprentissage S qui contient p objets de classe P et n objets de classe N
  - La quantité d'information nécessaire pour décider si un objet de S appartient à P ou N est définie comme

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

- Les valeurs de A partitionnent S en {S1, …, Sv}
  - si S_i contient p_i exemples de P et n_i exemple de N, l'entropie ou l'information attendue nécessaire pour classer les objets dans tous les sous-arbres S_i est

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

- Le gain d'information de l'attribut A est $Gain(A) = I(p,n) - E(A)$

Exemple : achète un ordinateur ?

- Classe P : achète un ordinateur = oui
- Classe N : achète un ordinateur = non

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
30…40	4	0	0
>40	3	2	0.971

- I(p,n) = I(9,5) = 0.940

- Calcul de l'entropie

$$E(age) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.69$$

- Gain d'information

$$Gain(age) = I(p,n) - E(age)$$

- De même

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

Éviter de trop modéliser le jeu d'apprentissage (overfitting)

- L'arbre généré risque de trop refléter le jeu d'apprentissage
  - trop de branches, certaines peuvent représenter des anomalies
  - précision faible pour des données nouvelles
- 2 approches
  - pré-élagage : arrêter la construction de l'arbre tôt = ne pas partitionner un nœud si la mesure de qualité dépasse un seuil
    - difficulté de fixer le seuil
  - post-élagage : supprimer des branches d'un arbre totalement construit = obtenir une séquence d'arbres progressivement élagués
    - utiliser un jeu de données différents pour décider du meilleur arbre élagué

Améliorations de l'algorithme

- Attributs définis sur des valeurs continues
  - définir dynamiquement les valeurs pour partitionner les données
- Tolérance aux données manquantes
  - attribuer la valeur la plus fréquente
  - attribuer une probabilité pour chaque valeur possible

Classification Bayésienne

- Apprentissage probabiliste : calcule explicitement les probabilités des hypothèses, une des approches les plus pragmatiques pour certains types d'apprentissage

- Incrémental : chaque exemple met à jour la probabilité qu'une hypothèse est correcte. Des connaissances *a priori* peuvent être combinées avec des données d'observation.

- Prédiction probabiliste : prédit plusieurs hypothèses, pondérées par leur probabilité

Classification Bayésienne

- Le problème de classification peut être formulé en utilisant les probabilités *a posteriori* :
  - P(C|X) = probabilité que l'objet X = <x_1,…,x_k> est de la classe C
- Exemple : P(non|ciel=dégagé,vent=fort,…)

- Idée : attribuer à l'objet X la classe qui maximise P(C|X)

Estimation des probabilités *a posteriori*

- Théorème de Bayes
  - $P(C|X) = P(X|C) \cdot P(C) / P(X)$
- $P(X)$ est constant pour toutes les classes
- $P(C)$ = fréquence de la classe C
- C tel que $P(C|X)$ est maximum = C

  C tel que $P(X|C) \cdot P(C) / P(X)$ est maximum
- Problème : pas faisable en pratique !

Classificateur Bayésien naïf

- Assomption naïve : indépendance des attributs
  - $P(x_1,\ldots,x_k|C) = P(x_1|C)\cdot\ldots\cdot P(x_k|C)$

- si le i-ième attribut est catégoriel :

  $P(x_i|C)$ est estimée comme la fréquence des échantillons qui ont pour valeur x_i et qui sont de classe C

- si le i-ième attribut est continu :

  $P(x_i|C)$ est estimée avec une gaussienne

- calcul facile

Exemple

Ciel	Température	Humidité	Vent	Classe
dégagé	chaud	forte	faux	N
dégagé	chaud	forte	vrai	N
couvert	chaud	forte	faux	P
pluvieux	moyenne	forte	vrai	P
pluvieux	frais	normale	faux	P
pluvieux	frais	normale	vrai	N
couvert	frais	normale	vrai	P
dégagé	moyenne	forte	faux	N
dégagé	frais	normale	faux	P
pluvieux	moyenne	normale	faux	P
dégagé	moyenne	normale	vrai	P
couvert	moyenne	forte	vrai	P
couvert	chaud	normale	faux	P
pluvieux	moyenne	forte	vrai	N

Ciel	
P(dégagé|p) = 2/9	**P(dégagé|n) = 3/5**
P(couvert|p) = 4/9	**P(couvert|n) = 0**
P(pluvieux|p) = 3/9	**P(pluvieux|n) = 2/5**
température	
P(chaud|p) = 2/9	**P(chaud|n) = 2/5**
P(moyenne|p) = 4/9	**P(moyenne|n) = 2/5**
P(frais|p) = 3/9	**P(frais|n) = 1/5**
humidité	
P(forte|p) = 3/9	**P(forte|n) = 4/5**
P(normale|p) = 6/9	**P(normale|n) = 2/5**
vent	
P(vrai|p) = 3/9	**P(vrai|n) = 3/5**
P(faux|p) = 6/9	**P(faux|n) = 2/5**

P(p) = 9/14
P(n) = 5/14

- Un objet X = <pluvieux, chaud, forte, faux>
- P(X|p).P(p) =
  P(pluvieux|p)·P(chaud|p)·P(forte|p)·P(faux|p)·P(p) = 3/9·2/9·3/9·6/9·9/14 = 0.010582
- P(X|n)·P(n) =
  P(pluvieux|n)·P(chaud|n)·P(forte|n)·P(faux|n)·P(n) = 2/5·2/5·4/5·2/5·5/14 = **0.018286**
- X est classé comme n

L'hypothèse d'indépendance …

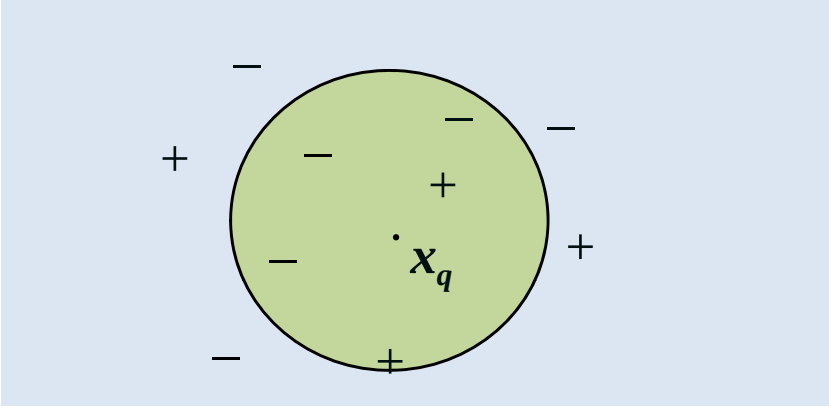
- … rend le calcul possible
- … mène à des classificateurs optimaux si elle est vérifiée
- … mais c'est rarement le cas, car les attributs sont souvent corrélés
- tentative de pallier cette difficulté :
  - réseaux Bayésiens : combinent le raisonnement Bayésien avec la relation causale entre les attributs
  - arbres de décision : considère un seul attribut à la fois, en commençant par le plus important

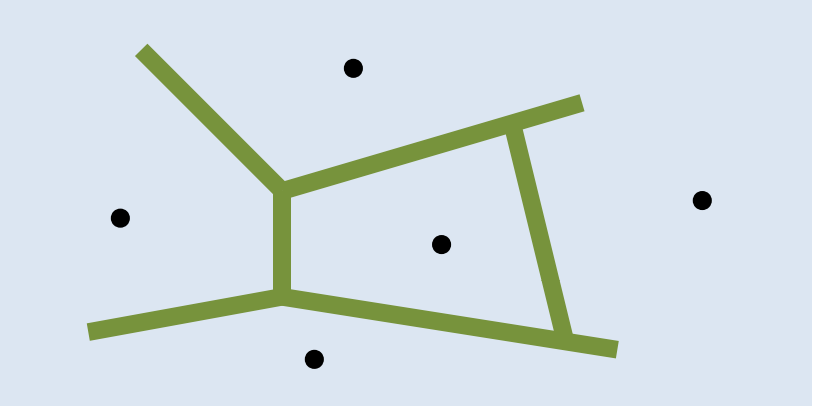
Méthodes basées sur des instances (Instance-Based Methods)

- Apprentissage basé sur des instances :
  - Stocke le jeu d'apprentissage et effectue le traitement quand une nouvelle instance doit être classée
- Approches typiques
  - *k* plus proches voisin (*k* nearest neighbors)
    - chaque objet représente un point dans l'espace
  - régression
    - loess, approximation locale

Algorithme des *k* plus proches voisins

- Chaque objet est représenté par un point dans un espace à *n* dimensions
- Les plus proches voisins sont définis en terme de distance (euclidienne, manhattan, …) ou dissimilarité
- La fonction de prédiction peut être discrète/nominale ou continue
  - discrète : valeur la plus fréquente
  - continue : moyenne
- Diagramme de Voronoï : surface de décision induite par le plus proche voisin





Discussion sur l'algorithme des *k* plus proches voisins

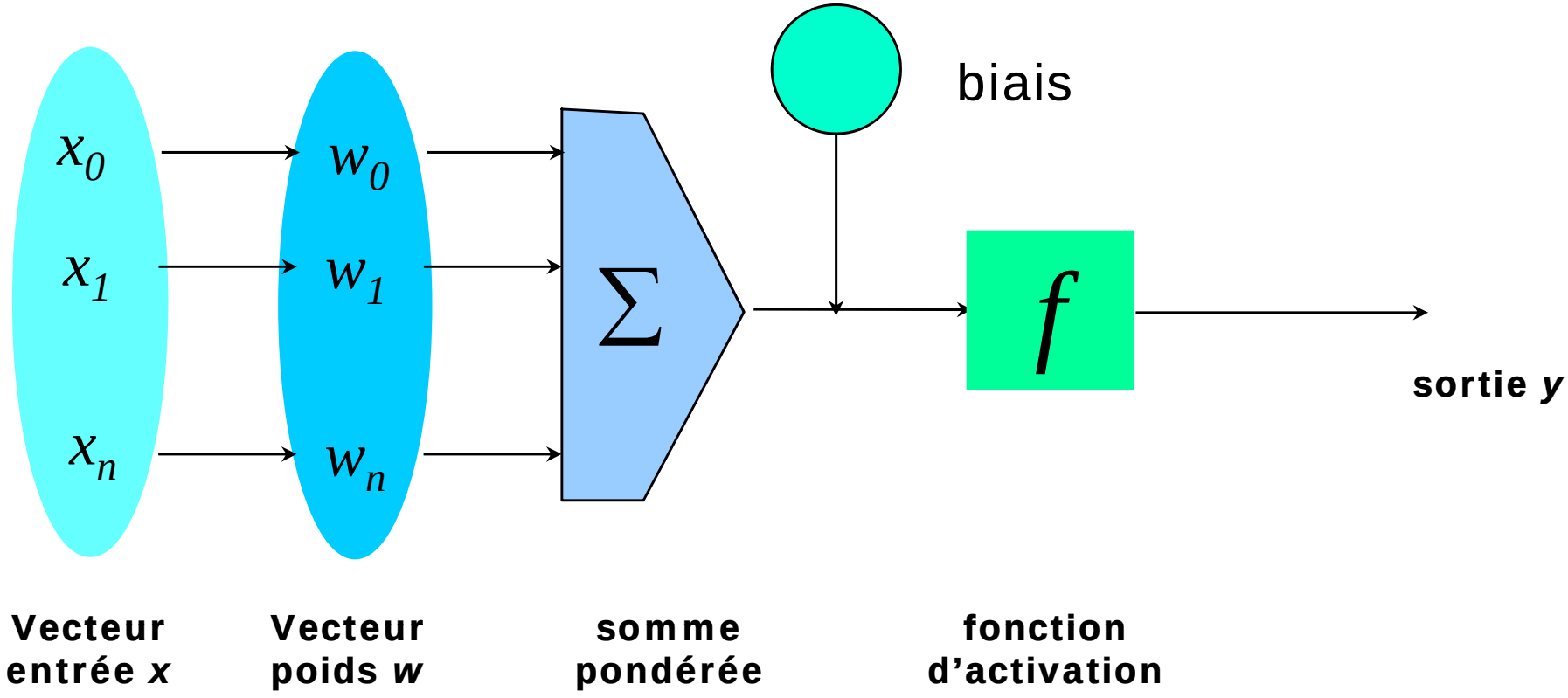
- Pondération en fonction de la distance
  - pondère la contribution de chacun des *k* voisins en fonction de sa distance à l'objet à classer
  - plus on est proche, plus on a de poids

$$w \equiv \frac{1}{d(x_q, x_i)^2}$$

- Robuste dans le cas de données bruitées
- Problème de dimensionnalité : la distance peut être dominée par des attributs non pertinents
  - solution : normalisation des dimensions ou élimination des attributs non pertinents

- Avantages
  - précision souvent élevée
  - robuste, marche lorsque le jeu d'apprentissage contient des erreurs
  - sortie peut être valeur discrète, continue, ou un vecteur de plusieurs valeurs discrètes ou continues
- Critiques
  - apprentissage long
  - difficile de comprendre le modèle
  - difficile d'incorporer des connaissances

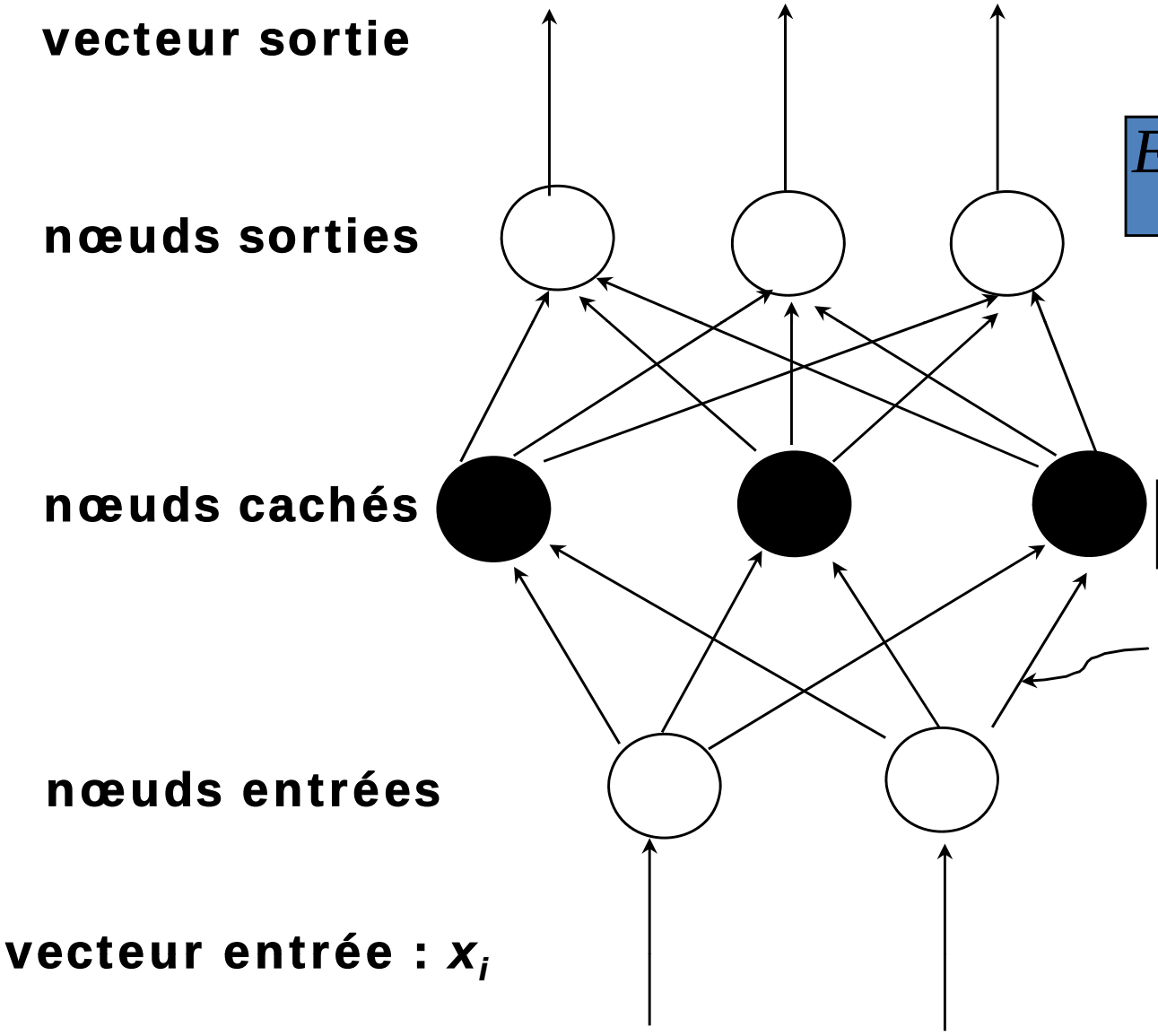
- vecteur *x* n-dimensionnel est intégré en *y* par le produit scalaire ($x_i.w_i$), le biais et la fonction d'activation



Entraînement

- Objectif
  - obtenir un ensemble de poids permettant de classer correctement (presque tous) les objets du jeu d'apprentissage
- Étapes
  - Initialiser les poids avec des valeurs aléatoires
  - Passer les objets au réseau un par un
  - pour chaque unité (neurone)
    - calculer l'entrée (combinaison linéaire de toutes les entrées)
    - calculer la sortie en utilisation la fonction d'activation
    - calculer l'erreur
    - mettre à jour le biais et les poids

Perceptron (multicouche)



$$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$$

$$\theta_j = \theta_j + (l)Err_j$$

$$w_{ij} = w_{ij} + (l)Err_j O_i$$

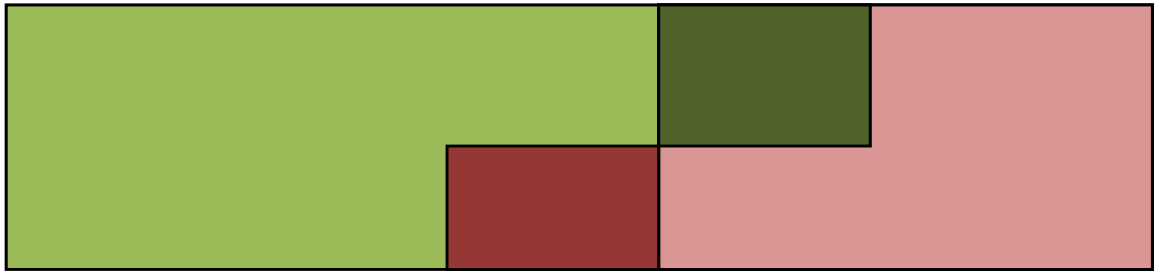
$$Err_j = O_j(1-O_j)(T_j - O_j)$$

$$O_j = \frac{1}{1+e^{-I_j}}$$

$$I_j = \sum_i w_{ij} O_i + \theta_j$$

Performances du classificateur

- Positif (P=•+•), Négatif (N=•+•),
  Prédit Positif (PP=•+•), Prédit Négatif (PN=•+•),
  Vrai Positif (VP=•), Faux Positif (FP=•),
  Vrai Négatif(VN=•), Faux Négatif (FN=•)
- Sensibilité=VP/P
- Spécificité=VN/N
- Précision=VP/(VP+FP) = VP/PP
- Exactitude=sensibilité.P/(P+N) + spécificité.N/(P+N) = (VP+VN)/(P+N)



Estimation du taux d'erreurs

- partitionnement :
  - utilisation de jeux indépendants : apprentissage (2/3), test (1/3)
- validation croisée :
  - diviser les données en k partitions
  - utiliser k-1 partitions pour l'apprentissage et la dernière pour le test
  - précision = nombre d'objets bien classés lors des k itérations / nombre d'objets
  - leave-one-out (validation croisée avec k = s)
- bootstrapping
  - tirage aléatoire avec remise des objets constituant le jeu d'apprentissage

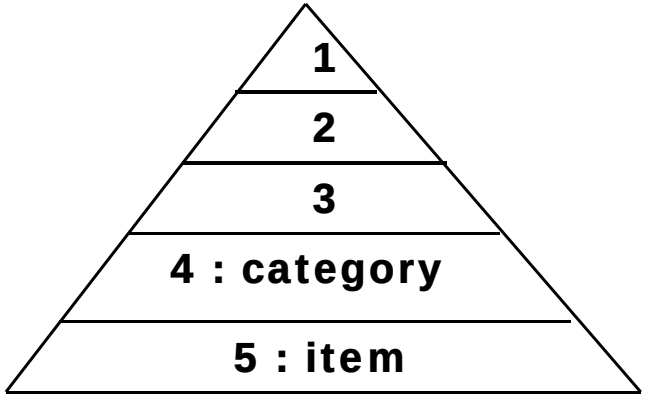
Classification et caractérisation

- Classification
  - arbre de décision
  - classificateur Bayésien
  - réseau de neurones
  - …

- Caractérisation
  - Description des concepts
  - Généralisation des données
  - Induction orientée attribut
  - Analyse de la pertinence des attributs
  - Mesures de pertinence
  - Caractérisation analytique

- Fouille de données descriptive ou prédictive
  - Fouille descriptive : décrits les concepts ou les données pertinentes pour l'analyse sous une forme concise, générale, informative, discriminative.
  - Fouille prédictive : construit un modèle des données basé sur leur analyse et prédit les tendances et les propriétés de données nouvelles ou inconnues
- Description des concepts
  - Caractérisation : fournit une généralisation concise et succincte d'un ensemble de données.
  - Comparaison : fournit des descriptions en comparant plusieurs ensembles de données.

- Généralisation des données
  - processus d'abstraction des données depuis un grand volume de données de bas niveau vers un niveau d'abstraction plus élevé
  - approches :
    - cube de données (OLAP)
    - Hiérarchie de concepts
    - induction orientée attribut (Attribute-oriented induction)



Niveaux conceptuels

Induction orientée attribut

- Proposée en 1989 (KDD ’89 Workshop)
- Ne se limite pas à des données nominales ni à des mesures particulières
- Principes :
  - sélection des données pertinentes
  - généralisation par **suppression d’attribut** ou **généralisation d’attribut**
  - agrégation par fusion des tuples généralisés identiques en conservant leurs effectifs
  - présentation interactive à l’utilisateur

Principes de l'induction orientée attribut

- Suppression d'attribut : supprime l'attribut A si le nombre de valeurs prises par cet attribut est grand et
  - il n'y a pas d'opérateur de généralisation ou
  - les niveaux supérieurs d'abstraction de A sont exprimés par d'autres attributs
- Généralisation d'attribut : s'il y a un grand nombre de valeurs distinctes pour A et qu'il existe un opérateur de généralisation
- Contrôle :
  - Seuil par attribut : en général de 2 à 8, spécifié par l'utilisateur ou par défaut (*attribute generalization threshold control*)
  - Seuil par taille de la table/relation : nombre de tuples maximal(*generalized relation threshold control*)

Algorithme simple

- RelationInitiale : table à généraliser
- PréGénéralisée : déterminer le plan de généralisation pour chaque attribut en fonction de l'analyse de ses valeurs : suppression ou jusqu'à quel niveau généraliser ?
- GénéraliséePrimaire : table résultant de l'application du plan
- Présentation : interaction de l'utilisateur :
  - ajustement des seuils (drill-down, roll-up)
  - représentation sous forme
    - de règles
    - de tableaux croisés
    - de graphiques…

Exemple

Name	Gender	Major	Birth-Place	Birth_date	Residence	Phone #	GPA
Jim Woodman	M	CS	Vancouver,BC, Canada	8-12-76	3511 Main St., Richmond	687-4598	3.67
Scott Lachance	M	CS	Montreal, Que, Canada	28-7-75	345 1st Ave., Richmond	253-9106	3.70
Laura Lee	F	Physics	Seattle, WA, USA	25-8-70	125 Austin Ave., Burnaby	420-5232	3.83
…	…	…	…	…	…	…	…
Removed	Retained	Sci,Eng, Bus	Country	Age range	City	Removed	Excl, VG,..

RelationInitiale

Gender	Major	Birth_region	Age_range	Residence	GPA	Count
M	Science	Canada	20-25	Richmond	Very-good	16
F	Science	Foreign	25-30	Burnaby	Excellent	22
…	…	…	…	…	…	…

GénéraliséePrimaire

Tableau croisé

Birth_Region / Gender	Canada	Foreign	Total
M	16	14	30
F	10	22	32
Total	26	36	62

- Relation généralisée
  - table avec quelques uns ou tous les attributs généralisés, avec effectifs ou autres valeurs d'agrégation accumulées
- Tableaux croisés
- Techniques de visualisation
  - camemberts, histogrammes, courbes, …
- Règles quantitatives caractéristiques

$$grad(x) \wedge male(x) \Rightarrow \begin{matrix} birth\_region(x) = "Canada"[t:53\%] \vee \\ birth\_region(x) = "foreign"[t:47\%]. \end{matrix}$$

relation généralisée

location	item	sales (in million dollars)	count (in thousands)
Asia	TV	15	300
Europe	TV	12	250
North_America	TV	28	450
Asia	computer	120	1000
Europe	computer	150	1200
North_America	computer	200	1800

tableau croisé

location \ item	TV		computer		*both_items*	
	sales	count	sales	count	sales	count
Asia	15	300	120	1000	135	1300
Europe	12	250	150	1200	162	1450
North_America	28	450	200	1800	228	2250
all_regions	45	1000	470	4000	525	5000

Analyse de la pertinence des attributs

- Pourquoi ?
  - quelles dimensions inclure ? à quel niveau d'abstraction ?
  - nombre d'attributs réduits : patterns plus faciles à comprendre
  - méthodes statistiques pour pré-traiter les données
    - filtrer les données non ou peu pertinentes
    - retenir et ordonner les attributs pertinents
  - pertinence relative des dimensions et niveaux
  - caractérisation analytique, comparaison analytique
- Comment ?
  - généralisation analytique : utiliser l'analyse du gain d'information (entropie ou autres mesures) pour identifier les dimensions et niveaux d'abstraction très pertinents
  - analyse de pertinence : trier et sélectionner les dimensions et niveaux les plus pertinents
  - induction orientée attribut sur les dimensions et niveaux sélectionnés

- Mesure quantitative de pertinence
  - détermine le pouvoir classificateur d'un attribut
- Méthodes
  - gain d'information (ID3)
  - ratio de gain (c4.5)
  - Information mutuelle

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p_1(x)\, p_2(y)} \right),$$

Approche basée sur la théorie de l'information

- Arbre de décision
  - les nœuds internes testent un attribut
  - les branches correspondent à une valeur d'attribut
  - les feuilles attribuent une classe
- Algorithme ID3
  - construit un arbre de décision basé sur un jeu de données d'apprentissage (les classes des objets sont connues)
  - ordonne les attributs selon la mesure de gain d'information
  - taille minimale
    - le moins de tests possibles pour classer un objet

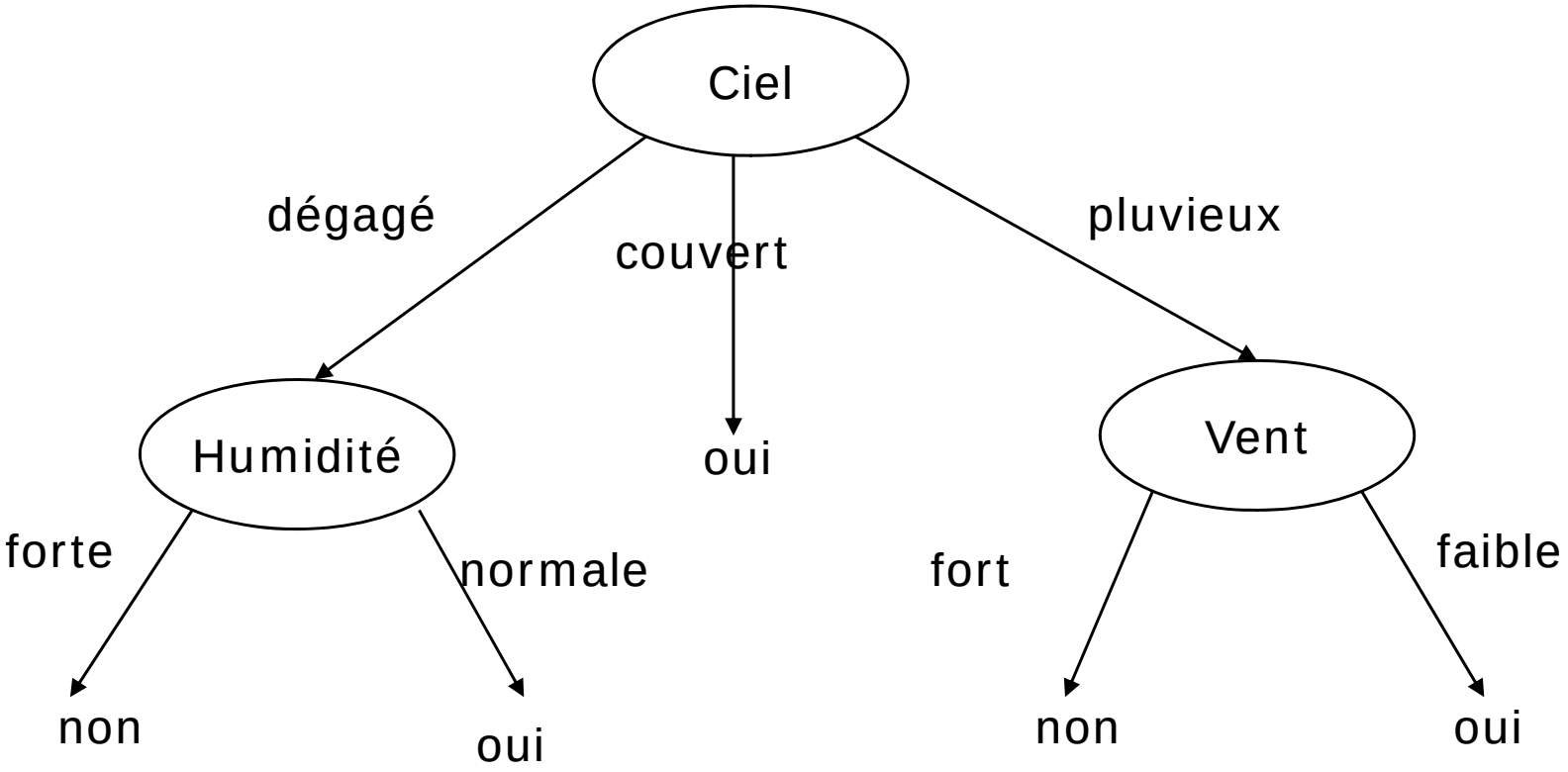
Exemple

CIEL	TEMPERATURE	HUMIDITE	VENT	JOUE ?
Dégagé	85	85	non	non
Dégagé	80	90	oui	non
Couvert	83	78	non	oui
Pluvieux	70	96	non	oui
Pluvieux	68	80	non	oui
Pluvieux	65	70	oui	non
Couvert	64	65	oui	oui
Dégagé	72	95	non	non
Dégagé	69	70	non	oui
Pluvieux	75	80	non	oui
Dégagé	75	70	oui	oui
Couvert	72	90	oui	oui
Couvert	81	75	non	oui
Pluvieux	71	90	oui	non

Induction descendante par arbre de décision

Attributs : {Ciel, Température, Humidité, Vent}

Classes : joue au tennis { oui , non }



Entropie et gain d'information

- S contient s_i tuples de classe C_i pour i = {1..m}
- L'information mesure la quantité d'information nécessaire pour classer un objet

$$I(s_1,s_2,\ldots,s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} log_2 \frac{s_i}{s}$$

- Entropie d'un attribut A ayant pour valeurs $\{a_1,\ldots,a_v\}$

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j}+\ldots+s_{mj}}{s} I(s_{1j},\ldots,s_{mj})$$

- Gain d'information en utilisant l'attribut A

$$Gain(A) = I(s_1,s_2,\ldots,s_m) - E(A)$$

Exemple : caractérisation analytique

- Tâche :
  - fouiller les caractéristiques générales qui décrivent les étudiants de 3ème cycle en faisant une caractérisation analytique
- Étant donné :
  - attributs : *name, gender, major, birth_place, birth_date, phone#*, and *gpa*
  - $Gen(a_i)$ = hiérarchies de concepts pour a_i
  - U_i = seuils analytiques d'attribut pour a_i
  - T_i = seuils de généralisation pour a_i
  - R = seuil de pertinence

Exemple : caractérisation analytique

- Données :
  - classe cible : étudiants 3ème cycle
  - classe contrastante : étudiant 1er et 2ème cycle
- Généralisation analytique
  - suppression d'attributs
    - supprime *name* et *phone#*
  - généralisation d'attribut
    - généralise *major*, *birth_place*, *birth_date* et *gpa*
    - accumule les effectifs
  - relation candidate : *gender*, *major*, *birth_country*, *age_range* et *gpa*

Exemple : caractérisation analytique

Name	Gender	Major	Birth-Place	Birth_date	Residence	Phone #	GPA
Jim Woodman	M	CS	Vancouver,BC, Canada	8-12-76	3511 Main St., Richmond	687-4598	3.67
Scott Lachance	M	CS	Montreal, Que, Canada	28-7-75	345 1st Ave., Richmond	253-9106	3.70
Laura Lee	F	Physics	Seattle, WA, USA	25-8-70	125 Austin Ave., Burnaby	420-5232	3.83
…	…	…	…	…	…	…	…
Removed	Retained	Sci,Eng, Bus	Country	Age range	City	Removed	Excl, VG,..

Relation initiale

Exemple : caractérisation analytique

gender	major	birth_country	age_range	gpa	count
M	Science	Canada	20-25	Very_good	16
F	Science	Foreign	25-30	Excellent	22
M	Engineering	Foreign	25-30	Excellent	18
F	Science	Foreign	25-30	Excellent	25
M	Science	Canada	20-25	Excellent	21
F	Engineering	Canada	20-25	Excellent	18

Relation candidate pour la classe cible : 3ème cycle (Σ=120)

gender	major	birth_country	age_range	gpa	count
M	Science	Foreign	<20	Very_good	18
F	Business	Canada	<20	Fair	20
M	Business	Canada	<20	Fair	22
F	Science	Canada	20-25	Fair	24
M	Engineering	Foreign	20-25	Very_good	22
F	Engineering	Canada	<20	Excellent	24

Relation candidate pour la classe contrastante : 1er et 2nd cycle (Σ=130)

- Analyse de pertinence
  - Calcul de l'information nécessaire pour classer un objet

$$I(s_1,s_2) = I(120,130) = -\frac{120}{250}log_2\frac{120}{250} - \frac{130}{250}log_2\frac{130}{250} = 0.9988$$

  - Calcul de l'entropie pour chaque attribut : ex : *major*

major="Science"	:	$s_{11}=84$	$s_{21}=42$	$I(s_{11},s_{21})=0.9183$
major="Engineering"	:	$s_{12}=36$	$s_{22}=46$	$I(s_{12},s_{22})=0.9892$
major="Business"	:	$s_{13}=0$	$s_{23}=42$	$I(s_{13},s_{23})=0$

Nombre de 3ème cycle en sciences

Nombre de 1er et 2nd cycle en sciences

Exemple : caractérisation analytique

- Calcul de l'information nécessaire pour classer un objet si S est partitionné selon l'attribut

$$E(major) = \frac{126}{250} I(s_{11}, s_{21}) + \frac{82}{250} I(s_{12}, s_{22}) + \frac{42}{250} I(s_{13}, s_{23}) = 0.7873$$

- Calcul du gain d'information pour chaque attribut

$$Gain(major) = I(s_1, s_2) - E(major) = 0.2115$$

  - gain pour tous les attributs

Gain(gender)	= 0.0003
Gain(birth_country)	= 0.0407
Gain(major)	= 0.2115
Gain(gpa)	= 0.4490
Gain(age_range)	= 0.5971

Exemple : caractérisation analytique

- Dérivation de RelationInitiale
  - R = 0.1
  - supprimer les attributs peu ou pas pertinents de la relation candidate : supprime *gender* et *birth_country*

major	age_range	gpa	count
Science	20-25	Very_good	16
Science	25-30	Excellent	47
Science	20-25	Excellent	21
Engineering	20-25	Excellent	18
Engineering	25-30	Excellent	18

- Effectuer l'induction orientée attribut