

Règles de décision prédictives

Modélisation prédictive à l'aide d'une base de règles

Ricco RAKOTOMALALA

Université Lumière Lyon 2



PLAN

1. L'induction de règles pour l'analyse prédictive
2. Approche divide-and-conquer - Les arbres de décision
3. Approche separate-and-conquer (stratégies top-down)
 - a. Construction des listes de décision
 - b. Construction de règles non imbriquées
4. Autres approches pour l'induction de règles
 - a. Stratégie bottom-up
 - b. Algorithmes génétiques pour l'induction de règles
 - c. Règles d'association prédictives
 - d. Règles prédictives floues
5. Bilan
6. Logiciels – Tutoriels
7. Références bibliographiques



Apprentissage supervisé

Modélisation à l'aide d'une base de règles

Pourquoi ? Quel intérêt ?



Contexte – Apprentissage supervisé

Variable cible à prédire : $Y \in \{1, 2, \dots, K\}$

Variables prédictives : (X_1, X_2, \dots)

(≈ Quinlan, 1993 ;
page 18)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	Yes
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	No
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Weak	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Objectif : Construire une fonction de prédiction $f(\cdot)$ reliant Y aux (X_1, X_2, \dots) , paramétrée, qui soit la plus **efficace** possible.

Problème : On travaille à partir d'un échantillon, mais on souhaite que la fonction soit performante en généralisation sur la population.

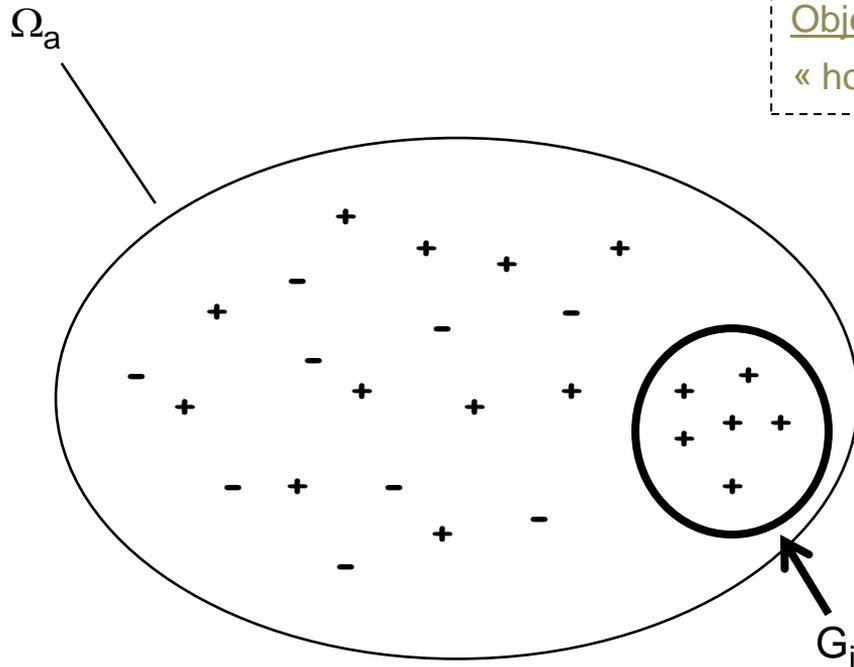
$$Y = f(X_1, X_2, \dots ; \alpha)$$

Comment choisir $f(\cdot)$?

Comment estimer les paramètres à partir des données ?



Induction de règles prédictives



Objectif : isoler un groupe d'individus qui soit le plus « homogène » possible du point de vue de la variable cible Y

Exemple : la variable qualitative Y prend ses valeurs dans {+,-}

Le sous-groupe G_i est complètement pur du point de vue de Y, il ne possède que des individus portant la valeur + de Y

si ($\omega \in G_i$) **alors** ($Y = +$)

L'idée est de trouver le plus rapidement Possible (avec le moins de variables) des groupes où $P(Y=+) \# 1$



Un groupe est décrit par une conjonction de propositions « attribut = valeur »



Un modèle = une base de règles c.-à-d. un ensemble de règles



On souhaite produire le plus petit ensemble de règles qui soit le plus performant possible (principe de parcimonie)



Règle et sa caractérisation



(r) Si outlook = rain et wind = weak Alors Play = yes

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	Yes
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	No
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Weak	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



Play	antecedent	not (antecedent)	total
no	n_1r = 1	4	n_1. = 5
yes (k*)	n_2r = 2	7	n_2. = 9
total	n_.r = 3	11	n = 14

Quelques mesures d'évaluation des règles

Couverture absolue

$$\text{cov}(r) = n_{.r} = 3$$

Couverture relative

$$\text{cov}'(r) = \frac{n_{.r}}{n} = \frac{3}{14}$$

Dans le cadre des règles d'association, on parlerait de « support de l'antécédent » de la règle

Confiance (précision)

$$\text{conf}(r) = \frac{n_{k^*r}}{n_{.r}} = \frac{2}{3}$$

Pour « lisser » les estimations sur les petits effectifs. Estimateur laplacien des probabilités (cf. [Wikipedia](http://fr.wikipedia.org/wiki/Estimateur_de_Laplace))

$$\text{conf}'(r) = \frac{n_{k^*r} + 1}{n_{.r} + K} = \frac{2 + 1}{3 + 2}$$

Entropie de Shannon

$$S(r) = \sum_k \frac{n_{kr}}{n_{.r}} \log \frac{n_{kr}}{n_{.r}} = \frac{1}{3} \log \frac{1}{3} + \frac{2}{3} \log \frac{2}{3}$$

Indique le degré d'incertitude (d'homogénéité) dans le groupe délimité par la règle. Peut être calculé également sur l'estimateur laplacien des probas.

J-Measure

$$J(r) = \frac{n_{.r}}{n} \sum_k \frac{n_{kr}}{n_{.r}} \log \frac{n_{kr}}{n_{k.}} = \frac{3}{14} \times \left[\frac{1}{5} \log \frac{1}{5} + \frac{2}{9} \log \frac{2}{9} \right]$$

Indique à la fois l'importance (support) de la règle et l'information qu'elle porte (distorsion entre les distributions globales et locales des classes)



Base de règles et sa caractérisation

- (1) Si outlook = sunny Alors play = yes (3/5)
- (2) Si outlook = overcast Alors play = yes (4/4)
- (3) Si outlook = rain Alors play = no (3/5)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	Yes
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	No
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Weak	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



Des mesures faisant un arbitrage entre la complexité de la base de règles (nombre de règles, nombre total de propositions) et leurs précisions (des règles) existent. Mais elles ne se sont jamais imposées.



Une approche simple consiste à considérer la base de règles comme un modèle prédictif et à mesurer ses performances sur un échantillon test (matrice de confusion + indicateurs)



- Quelle stratégie adopter si un individu à classer déclenche plusieurs règles ?
- Que faire si un individu à classer ne déclenche aucune règle ?



Avantages et inconvénients – Modélisation par règles

Avantages

- **Modèle intelligible**, possibilité d'interprétation
- Sélection automatique des propositions (et variables) pertinentes
- Possibilité d'intervention et d'amélioration par expertise
- **Règle déclenchée identifiable en classement**
- Modèle facilement implémenté dans les systèmes d'information

- Techniques non paramétriques
- Robustesse par rapport aux points aberrants

Inconvénients

- Stabilité sur des petites bases (observations) – Groupes avec très peu d'individus
- Seuils de discrétisation (un peu) arbitraires pour les variables quantitatives
- Peu adapté au « scoring » (les individus d'un même groupe présentent le même score)



Approche divide and conquer

Arbres de décision



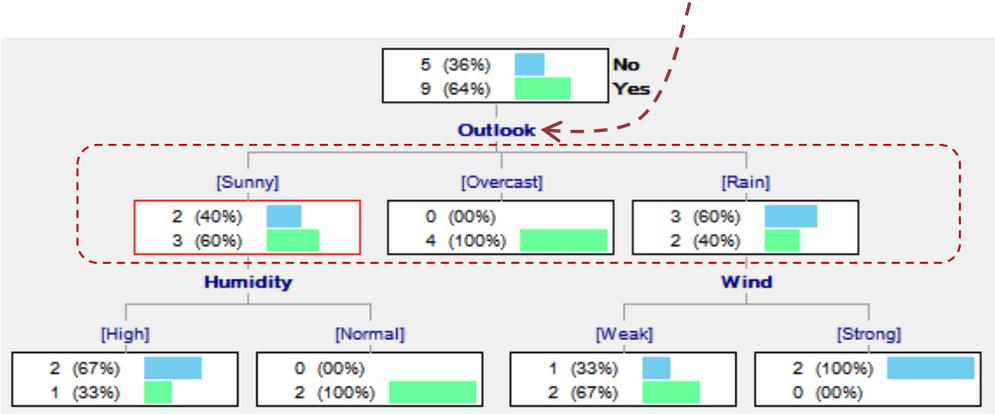
Principe « divide-and-conquer » : les arbres de décision

- Partitionnement successif des observations : algorithme récursif
1. Subdivision des observations avec une variable prédictive
 2. On souhaite que les sous-groupes soient (en moyenne) les plus purs possibles
 3. Pour chaque sous-groupe, si non pur (ou règle d'arrêt non déclenchée), retour en 1. Sinon, arrêt de l'algorithme.

➡ Typiquement, c'est le principe de construction des arbres de décision

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	Yes
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	No
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Weak	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

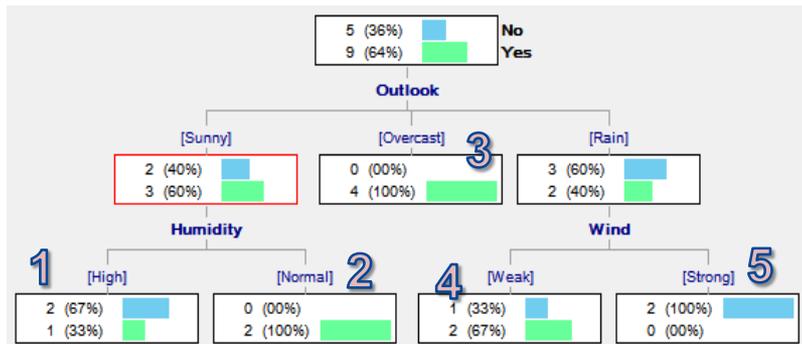
Variable qui emmène le meilleur « gain moyen » de pureté (homogénéité des classes dans les sous-groupes)



Voir le support de cours : « [Introduction aux arbres de décision](#) »



Conversion des arbres en base de règles (A)



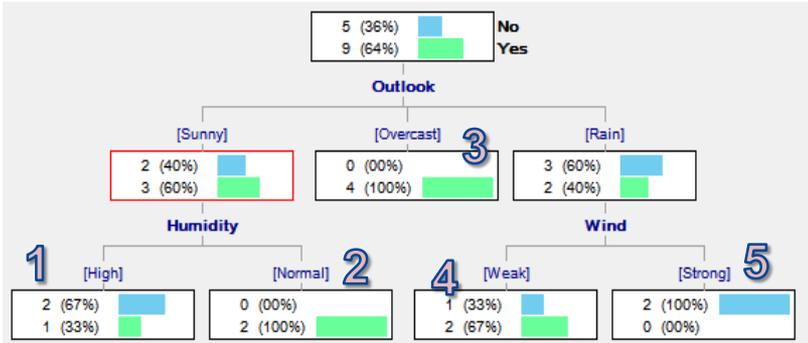
Lecture sous la forme de règles indépendantes : tous les chemins partant de la racine aux feuilles.

- (1) si outlook = sunny et humidity = high alors play = no
- (2) si outlook = sunny et humidity = normal alors play = yes
- (3) si outlook = overcast alors play = yes
- (4) si outlook = rain et wind = weak alors play = yes
- (5) si outlook = rain et wind = strong alors play = no

- Les règles sont considérées de qualité équivalente, mais ce n'est pas un problème car...
- Règles mutuellement exclusives → un individu à classer ne déclenche qu'une seule règle
- Règles exhaustives → un individu à classer déclenche forcément une règle



Conversion des arbres en base de règles (B)



Lecture sous la forme d'une cascade de conditions SI imbriquées

Si outlook = sunny
...Alors
 Si humidity = high
 ...Alors play = no (1)
 ...Sinon play = yes (2)
...Sinon
 Si outlook = overcast
 ...Alors play = yes (3)
 ...Sinon
 Si wind = weak
 ...Alors play = yes (4)
 ...Sinon play = no (5)

Chaque condition SI ne contient qu'une seule proposition.

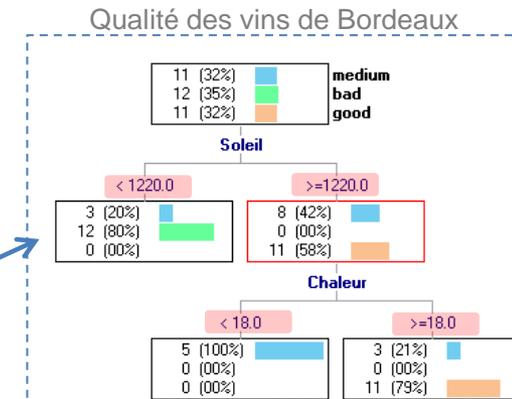
Facile à reprendre dans les langages de programmation utilisés en déploiement (ex. fonction SI d'Excel, Java, Python, C++, etc.)



Avantages et inconvénients des arbres de décision

Avantages

- Pas de conflits de règles
- Une règle unique identifiée en classement de chaque individu
- Discrétisation des variables continues intégrée dans le processus



Inconvénients

- Structure d'arbre introduit une contrainte supplémentaire dans la recherche de règles
 - Cela implique un risque de fragmentation accru des données
 - Optimisation simultanée dans les sous-groupes issues de la segmentation est contraignant.
- Gain moyen (segmentation) peut être un frein pour isoler rapidement les « meilleures » règles

« C4.5 rules » propose un post-traitement des règles (règles indépendantes) issues de l'arbre pour dépasser ces inconvénients...



Idée : Post-traitement des règles issues de l'arbre (règles indépendantes)

Traitement de chaque règle

- Pour chaque règle, retirer (pas à pas) des propositions de manière à réduire l'erreur pessimiste
- Après coup, retirer les éventuelles règles redondantes

Traitement de la base de règles

- Grouper les règles par conclusion (classe prédite). Pour chaque groupe, trouver le sous-ensemble de règles permettant de maximiser un critère MDL (théorie de la longueur de description minimale des messages)
- Ordonner les groupes de règles selon leur précision (min. de faux positifs)
- Prédiction par défaut (quand aucune règle n'est déclenchée) correspond à la classe la plus représentée parmi les observations qui ne déclenchent aucune règle

Pour éviter les conflits de décision



Assurer l'universalité de la décision



Algorithme lent et implémenté nulle part (sauf C5.0 rules, version commerciale, algo. non publié)
Mais l'idée qu'induction de règles et arbres de décision ne sont pas forcément synonymes est acquise



Approches separate and conquer

Induction de règles prédictives



Principe de l'approche « separate-and-conquer »

Séquence d'opérations :

1. Construire une règle qui prédit au mieux une des classes (**conquête**)
2. Retirer de la base d'apprentissage les individus couverts par la règle (**séparation**)
3. S'il reste suffisamment d'observations, retour en 1.

A partir de ce schéma type, plusieurs stratégies possibles :

- Approche top-down par spécialisation successive d'une règle (adjonction de propositions)
- Approche bottom-up par généralisation, départ un individu, retrait itératif de propositions
- Ciblage d'une classe particulière ou non durant la conquête
- Critère d'évaluation des règles durant la phase de conquête
- Retrait de tous les individus couverts ou des individus positifs durant la séparation



Présentation de 2 approches top-down (simples, rapides) dans ce qui suit



Listes de décision



Les listes de décision (Decision List)

Liste de décision = base de règles ordonnées et mutuellement exclusives

Condition(s) sont des conjonctions de propositions

Si condition_1 Alors conclusion_1

Sinon Si condition_2 Alors conclusion_2

Sinon Si condition_3 Alors conclusion_3

Sinon ...

Sinon (Règle par défaut) conclusion_m

En classement, la première règle est évaluée. Si elle est déclenchée, on s'en tient à sa conclusion. Sinon, on passe à la suivante, etc. Si aucune règle déclenchée, utilisation de la règle par défaut. L'exhaustivité est assurée par cette règle par défaut.



Listes de décision – Un exemple (1)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	Yes
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	No
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Weak	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Classifier characteristics

Data description

Target attribute: PlayTennis (2 values)
descriptors: 4

Number of rules = 5

Knowledge-based system

Antecedent	Consequent	Distribution
IF Outlook in [Overcast]	PlayTennis in [Yes]	(0; 4)
ELSE IF Temperature in [Hot]	PlayTennis in [No]	(2; 0)
ELSE IF Outlook in [Sunny]	PlayTennis in [Yes]	(0; 3)
ELSE IF Wind in [Strong]	PlayTennis in [No]	(2; 0)
ELSE IF Outlook in [Rain]	PlayTennis in [Yes]	(1; 2)
ELSE (DEFAULT RULE)	PlayTennis in [Yes]	(0; 0)

Components

Data visualization	Statistics	Nonparametric statistics
Feature construction	Feature selection	Regression
PLS	Clustering	Spv learning
Spv learning assessment	Scoring	Association
Correlation scatterplot	Scatterplot	View dataset
Export dataset	Scatterplot with label	View multiple scatterplot

Distribution dans les groupes délimités par les règles

Règle par défaut : classe majoritaire parmi les observations restantes ou, à défaut, classe majoritaire dans l'échantillon initial



Listes de décision – Un exemple (« Congress Vote » dataset)

Supervised Learning 1 (Decision List)

Classifier characteristics

Data description

Target attribute: Class (2 values)
descriptors: 16

Number of rules = 3

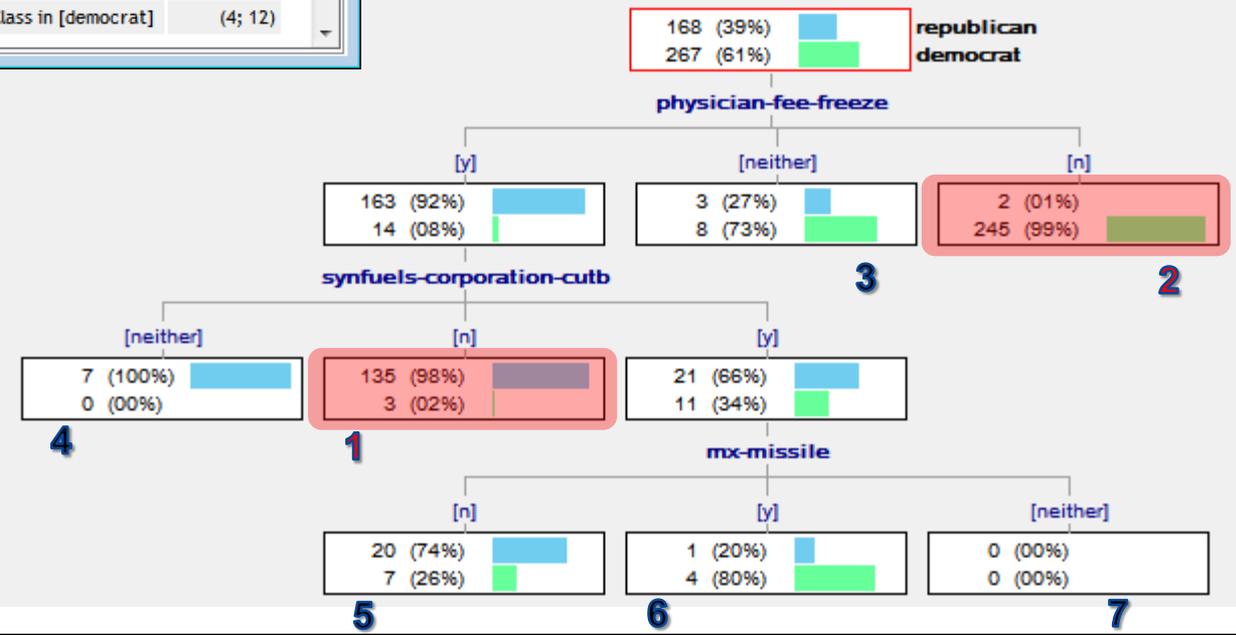
Knowledge-based system

Antecedent	Consequent	Distribution
IF physician-fee-freeze in [y] -- synfuels-corporation-cutb in [n]	Class in [republican]	(135; 3)
ELSE IF physician-fee-freeze in [n]	Class in [democrat]	(2; 245)
ELSE IF physician-fee-freeze in [y] -- mx-missile in [n]	Class in [republican]	(27; 7)
ELSE (DEFAULT RULE)	Class in [democrat]	(4; 12)

A
B
C

(A = 1) ; (B = 2) ; on voit que (C ≈ 5), mais la règle C s’est affranchie de la contrainte « synfuels... », elle est de fait plus précise ; (3, 4, 6, 7) sont en plus dans l’arbre (superflues ?) . **La liste de décision est plus compacte.**

Arbre de décision (C4.5)



DecisionList (var.cible, var.prédicatives, exemples)

Base de règles = \emptyset

Répéter

Règle = **Spécialiser** (var.cible, var.prédicatives, exemples)

Si (Règle != NULL) Alors

 Base de Règles = Base de Règles + {Règle}

 exemples = exemples - {Individus couverts par la Règle}

Fin Si

Jusqu'à (Règle = NULL)

Base de Règles = Base de Règles + {Règle par défaut (exemples)}

Renvoyer (Base de règles)

Spécialiser (var.cible, var.prédicatives, exemples)

Règle = NULL

Max.Mesure = $-\infty$

Tant Que (Règle.Arrêt(Règle) == FAUX)

 Ref.Mesure = $-\infty$

 Pour Chaque Proposition Candidate

 Mesure = **Evaluation** (Règle, Proposition)

 Si (Mesure > Ref.Mesure)

 Alors

 Ref.Mesure = Mesure

 Proposition* = Proposition

 Fin Si

 Fin Pour

 Si (Ref.Mesure > Max.Mesure)

 Règle = Règle x Proposition*

 Max.Mesure = Ref.Mesure

 Fin Si

Fin Tant Que

Renvoyer (Règle)

Boucle externe.

Elaboration de la base de règles à partir de l'échantillon complet. A chaque nouvelle règle définie, les individus couverts sont retirés de la base.

Boucle interne.

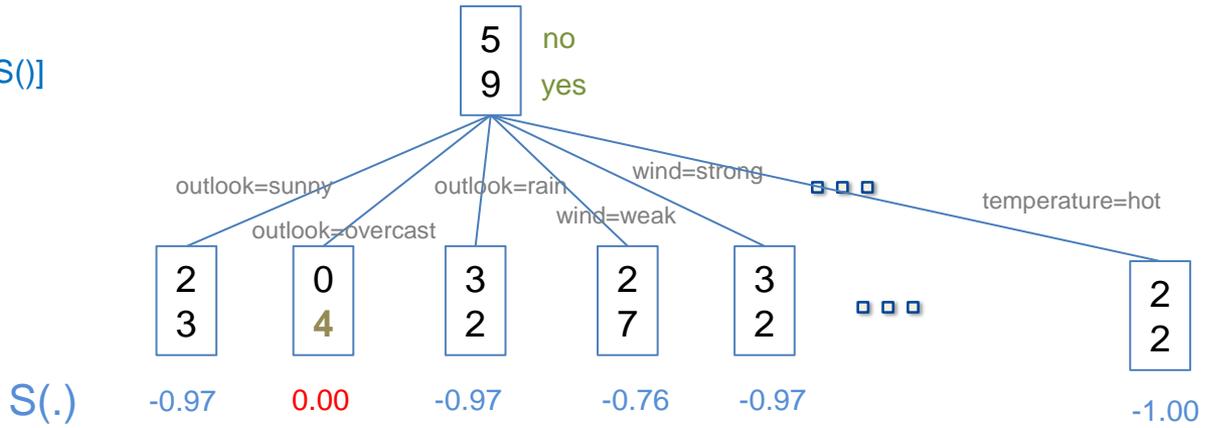
Elaboration d'une règle à partir d'un échantillon passé en paramètre. Algorithme glouton par spécialisations successives. Optimisation du critère calculé avec la fonction « **Evaluation** » : Entropie de Shannon ou J-Measure.



Illustration (1)

Construction première règle [Shannon – S()]

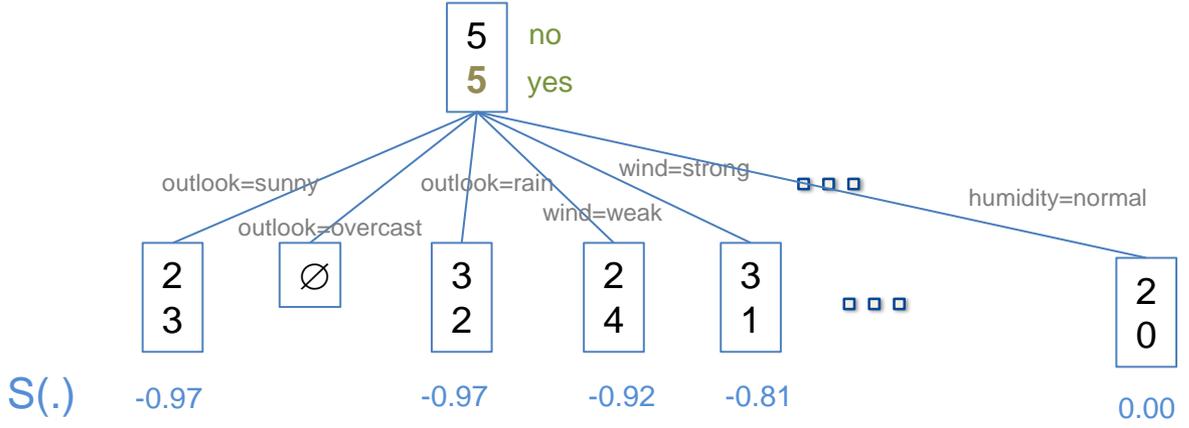
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	Yes
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	No
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Weak	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



Meilleure spécialisation, pas de spécialisation additionnelle puisque groupe pur. La 1^{ère} règle est validée.

Construction deuxième règle après exclusion des 4 individus couverts par la 1^{ère} règle

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	Yes
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	No
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Weak	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



Meilleure spécialisation, pas de spécialisation additionnelle puisque groupe pur. La 2^{ème} règle est validée.

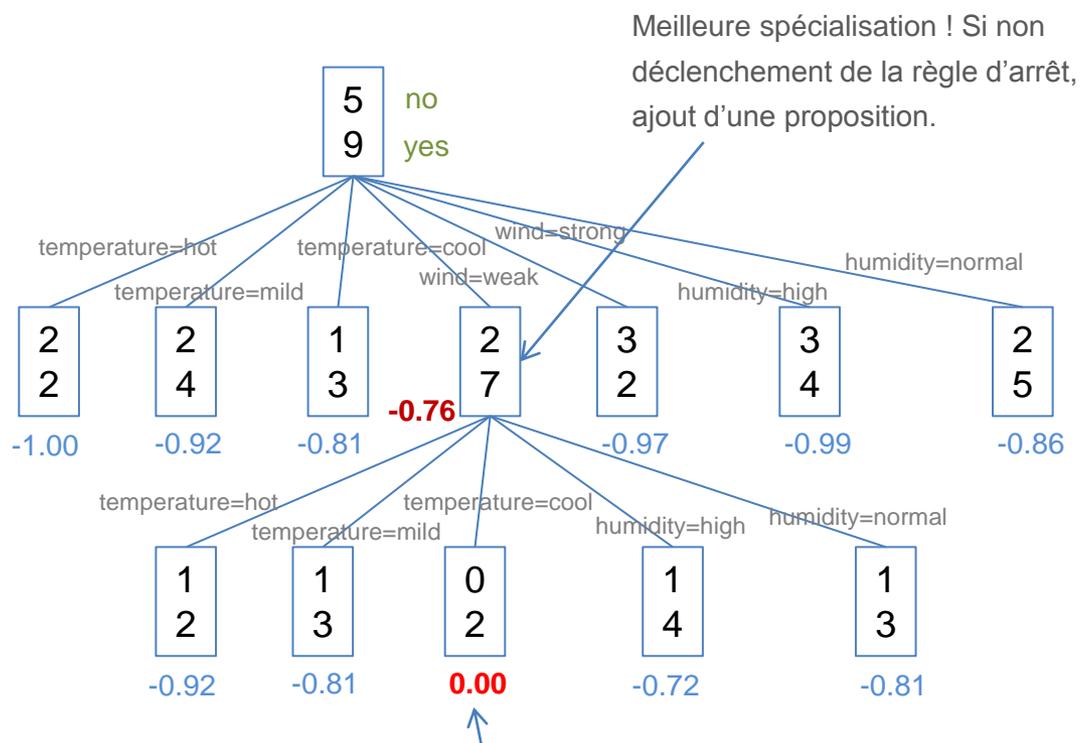
Etc.



Illustration (2) – La variable « outlook » n'est pas utilisée

Construction première règle (Shannon)

Day	Temperature	Humidity	Wind	PlayTennis
D1	Hot	High	Weak	No
D2	Hot	High	Strong	No
D3	Hot	High	Weak	Yes
D4	Mild	High	Weak	Yes
D5	Cool	Normal	Weak	Yes
D6	Cool	Normal	Strong	No
D7	Cool	Normal	Strong	Yes
D8	Mild	High	Weak	Yes
D9	Cool	Normal	Weak	Yes
D10	Mild	Normal	Weak	No
D11	Mild	Normal	Strong	Yes
D12	Mild	High	Weak	Yes
D13	Hot	Normal	Weak	Yes
D14	Mild	High	Strong	No



Meilleure spécialisation ! Si non déclenchement de la règle d'arrêt, ajout d'une proposition.

Meilleure spécialisation ! La règle est pure, arrêt. Retrait des 2 individus couverts. Puis recherche de la règle suivante.

Au final, nous obtenons la base de règles.



Number of rules = 3

Knowledge-based system

Antecedent	Consequent	Distribution
IF Wind in [Weak] -- Temperature in [Cool]	PlayTennis in [Yes]	(0; 2)
ELSE IF Wind in [Weak] -- Humidity in [High] -- Temperature in [Mild]	PlayTennis in [Yes]	(0; 3)
ELSE IF Humidity in [High] -- Wind in [Strong]	PlayTennis in [No]	(2; 0)
ELSE (DEFAULT RULE)	PlayTennis in [Yes]	(3; 4)



Règle d'arrêt de la spécialisation d'une règle

La règle doit couvrir suffisamment d'observations, elle doit être aussi suffisamment précise.

Critère de support

Nombre minimum d'individus couverts par la règle (ex. $n.min = 2$)

Significativité

Est-ce que l'adjonction d'une proposition améliore significativement la connaissance des valeurs prises par la variable cible – Test du KHI-2 – Risque du test (niveau de signification).

Exemple :

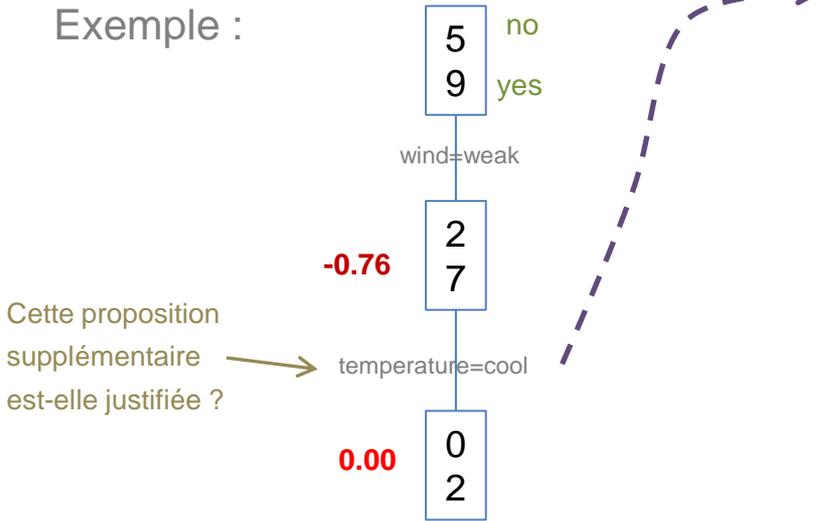


Tableau de contingence. Association entre la proposition et la variable cible ?

Play	temp = cool	not(temp=cool)	total
no	0	2	2
yes	2	5	7
total	2	7	9

KHI-DEUX = 0.735

p-value = 0.391

Que l'on compare au seuil α (paramètre de l'algorithme). Ex. $\alpha = 1\%$, etc.



Avantages et inconvénients des listes de décision

Avantages

- Permet de dépasser la contrainte de représentation par arbre
- De fait, les bases de règles obtenues sont plus compactes
- Et la fragmentation des individus est moindre

Inconvénients

- Discrétisation préalable des variables prédictives quantitatives (cf. « [La discrétisation des variables quantitatives](#) »)
- Lisibilité par rapport aux arbres ? Une règle est lue en fonction des précédentes.
- Moindre rapidité et capacité à traiter des très grandes bases (vs. arbres)
- Peu présent dans les logiciels, moins connu, travail pédagogique à faire auprès des non-experts
- Pas de logiciels proposant le traitement interactif (vs. arbres)



Base de règles non imbriquées



Règles non-imbriquées

Base de règles non imbriquées = règles non-ordonnées et non mutuellement exclusives

Condition(s) sont des conjonctions de propositions

Si condition_1 Alors conclusion_1

Si condition_2 Alors conclusion_2

Si condition_3 Alors conclusion_3

...

Sinon (Règle par défaut) conclusion_m

En classement, toutes les règles doivent être évaluées. Si plusieurs règles sont déclenchées, il faut les faire coopérer. Si aucune règle déclenchée, utilisation de la règle par défaut. L'exhaustivité est assurée par cette règle par défaut.



Règles non-imbriquées – Un exemple (1)

Dataset (tanBF49.txt)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	Yes
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	No
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Weak	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Classifier characteristics

Data description

Target attribute: PlayTennis (2 values)
descriptors: 4

Number of rules = 2

Knowledge-based system

Antecedent	Consequent	Distribution
IF Outlook in [Rain] -- Wind in [Strong]	PlayTennis in [No]	(2; 0)
IF Temperature in [Hot] -- Outlook in [Sunny]	PlayTennis in [No]	(2; 0)
(DEFAULT RULE)	PlayTennis in [Yes]	(1; 9)

Components

Nonparametric statistics	Instance selection	Feature construction
Factorial analysis	PLS	Clustering
Spv learning assessment	Scoring	Association

Feature selection

Spv learning	Regression
Meta-spv learning	Meta-spv learning

Tools: Correlation scatterplot, Scatterplot with label, Export dataset, View dataset, Scatterplot, View multiple scatterplot

Distribution dans les groupes définis par les règles

Règle par défaut : classe majoritaire parmi les observations restantes ou, à défaut, classe majoritaire dans l'échantillon initial



Règles non-imbriquées – Un exemple (« Congress Vote »)

Classifier characteristics

Data description

Target attribute	Class (2 values)
# descriptors	16

Number of rules = 1

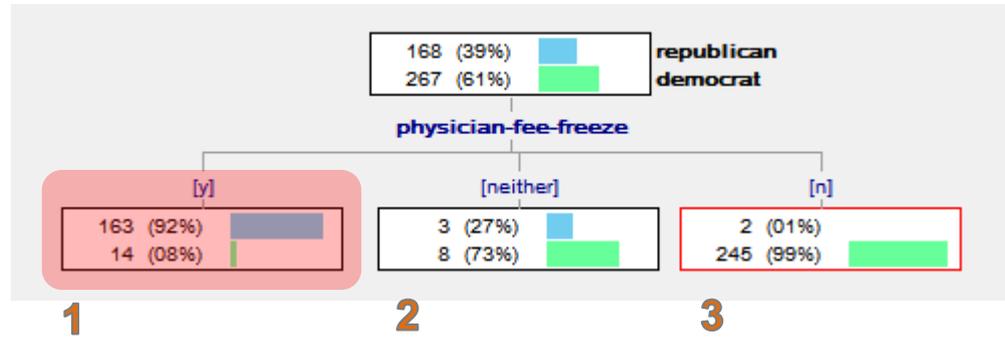
Knowledge-based system

Antecedent	Consequent	Distribution
IF physician-fee-freeze in [y]	Class in [republican]	(163; 14)
(DEFAULT RULE)	Class in [democrat]	(5; 253)

A

(A = 1) ; (Règle par défaut = 2 et 3).
 La base de règles est totalement équivalente à l'arbre, avec une expression plus compacte.

Arbre de décision à 1 seul niveau



Induction de règles - Algorithmme

Induction Règles (var.cible, var.prédicatives, exemples)

Base de règles = \emptyset

Trier les classes par ordre de fréquence croissante

Pour chaque classe « c » sauf la dernière

Sample = Exemples

Tant que Règle != NULL

Règle = **Spécialiser** (c, var.cible, var.prédicatives, sample)

Si (Règle != NULL) Alors

Base de Règles = Base de Règles + {Règle}

Sample = Sample - {Individus « positifs » couverts par la Règle}

Fin Si

Fin Tant Que

Fin Pour

Exemples = Exemples - {Individus couverts par au moins une règle}

Base de Règles = Base de Règles + {Règle par défaut (exemples)}

Renvoyer (Base de règles)

Spécialiser (classe, var.cible, var.prédicatives, exemples)

Règle = NULL

Max.Mesure = $-\infty$

Répéter

Ref.Mesure = $-\infty$

Proposition* = NULL

Pour Chaque Proposition Candidate

Mesure = **Evaluation** (classe, Règle, Proposition)

Si (Mesure > Ref.Mesure)

Alors

Ref.Mesure = Mesure

Proposition* = Proposition

Fin Si

Fin Pour

Si (Ref.Mesure > Max.Mesure)

Règle = Règle x Proposition*

Max.Mesure = Ref.Mesure

Fin Si

Jusqu'à (Proposition* == NULL)

Si **Significative**(Règle) == VRAI Alors **Renvoyer (Règle)** Sinon **Renvoyer(NULL)**

Elaboration de la base de règles à partir de l'échantillon complet. **On cherche en priorité les règles pour les classes rares.** A chaque nouvelle règle définie, les individus couverts « positifs » (c.-à-d. correspondant à la conclusion de la règle) sont retirés de la base. La règle par défaut est élaborée à partir des individus qui n'ont déclenché aucune règle.

A partir de l'échantillon passé en paramètres, recherche gloutonne par spécialisation des propositions de la règle. La mesure d'évaluation tient compte explicitement de la « classe » (modalité de la variable à prédire) ciblée.



Induction de règles - Mesures

	Antécédent	Non (Antécédent)	Effectif
Classe cible « c »	n_{ac}		n_c
Non (classe cible)			
Effectif	n_a		n

n_a nombre d'individus couverts par la règle

n_{ac} nombre d'individus « positifs » (corresp. à « c ») couverts par la règle

n_c nombre d'individus corresp. à la classe « c »

n nombre d'observations

Statistique de confiance

$$ZC(c, a) = \frac{n_{ac} - n_a n_c / n}{\sqrt{\frac{(n_a (n - n_a) / n)(n_c (n - n_c) / n)}{n - 1}}}$$

Statistique du contre-exemple

$$ZI(c, a) = -\frac{(n_a - n_{ac}) - (n - n_c) n_a / n}{\sqrt{\frac{(n - n_c) n_a}{n}}}$$

J-Mesure asymétrique

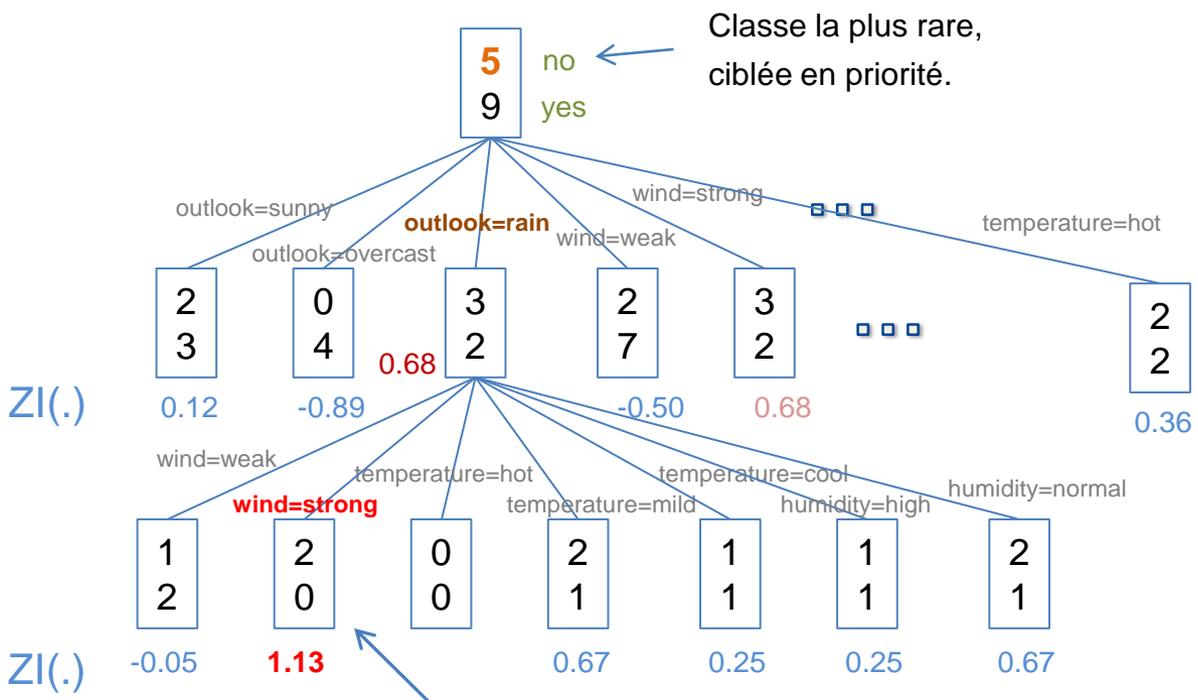
$$J_\beta(c, a) = \left(\frac{n_a}{n}\right)^{\frac{1}{\beta}} \times \left[\frac{n_{ac}}{n_a} \log \frac{n_{ac}}{n_a} - \frac{n_c - n_{ac}}{n - n_a} \log \frac{n_c - n_{ac}}{n - n_a} \right]$$

β est paramètre permettant de faire un arbitrage entre couverture et spécialisation (précision). $\beta \nearrow$, on privilégie la précision.



Construction première règle (ZI)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	Yes
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	No
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Weak	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

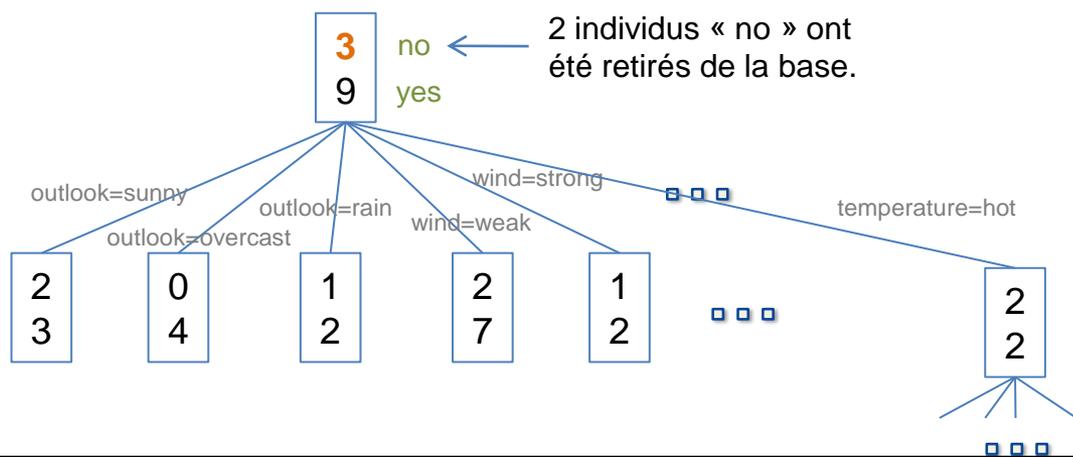


Classe la plus rare, ciblée en priorité.

Groupe pur. Arrêt de la spécialisation. Règle « outlook = rain et wind = strong » validée. Retrait des 2 individus corresp. à « play = no ». Construction de la règle suivante.

Construction règle suivante (ZI)

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	Yes
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	No
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Weak	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



2 individus « no » ont été retirés de la base.



Règle d'arrêt de la spécialisation d'une règle

La règle doit couvrir suffisamment d'observations, elle doit être aussi suffisamment précise.

Critère de support

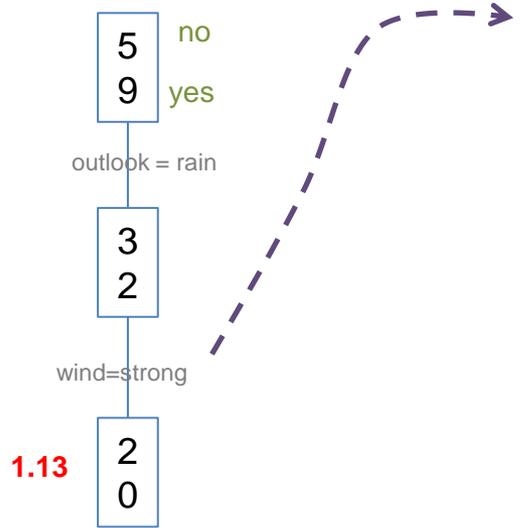
Nombre minimum d'individus couverts par la règle (ex. $n.min = 2$)

Significativité

Est-ce que l'adjonction d'une proposition améliore significativement la connaissance des valeurs prises par la variable cible – [1] Certaines mesures (ZC, ZI) suivent (très approximativement) une loi normale. [2] La J-Mesure asymétrique peut décroître lorsqu'on rajoute des propositions non-pertinentes.

Exemple :

Cette proposition supplémentaire est-elle justifiée ?



ZI : La proportion des contre-exemple à la règle est-elle significativement plus faible par rapport à la distribution initiale ?

ZI = 1.13
p-value = 0.1284

Que l'on compare au seuil α (paramètre de l'algorithme). Ex. $\alpha = 1\%$, etc.



Ex. $Y \in \{+, -\}$

Principe : additionner les effectifs couverts par les règles et prédire la classe la plus fréquente

{ Si condition 1 Alors $Y = +$ (150, 25)
Si condition 2 Alors $Y = +$ (45, 15)
Si condition 3 Alors $Y = -$ (12, 21)

Prédiction : $Y = +$ (207, 61)



Le dispositif fonctionne même si l'individu déclenche des règles contradictoires !!!



L'implémentation de cette stratégie dans les systèmes d'information est peu aisée, des calculs additionnels sont nécessaires pour chaque individu à classer.



Avantages et inconvénients des règles non imbriquées

Avantages

- Permet de dépasser la contrainte de représentation par arbre
- De fait, les bases de règles obtenues sont plus compactes
- Et la fragmentation des individus est moindre
- La non-imbriation des règles rend la lecture du modèle plus facile...

Inconvénients

- ...mais quid de la lisibilité de la décision quand un individu déclenche plusieurs règles ?
- Implémentation peu aisée de la gestion des conflits dans les systèmes d'information
- Discrétisation préalable des variables prédictives quantitatives (cf. « [La discrétisation des variables quantitatives](#) »)
- Moindre rapidité et capacité à traiter des très grandes bases (vs. arbres)
- Peu présent dans les logiciels, moins connu, travail pédagogique à faire auprès des non-experts
- Pas de logiciels proposant le traitement interactif (vs. arbres)



Autres approches pour l'induction des règles



Stratégies bottom-up



Stratégies bottom-up pour l'induction de règles

Idée : Toujours dans un cadre « separate-and-conquer », procéder par généralisation

1. On part d'un individu pris au hasard = règle
 - A. Retirer la proposition qui produit la meilleure généralisation (→ groupe pur)
 - B. Procéder ainsi jusqu'à ce qu'une bonne généralisation ne soit plus possible
2. Si règle validée, retirer les individus couverts.
3. FIN si arrêt de recherche déclenchée, sinon retour en 1.



Bottom-up – Un exemple

Entropie de Shannon calculée sur l'estimateur laplacien des probabilités

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	Yes
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	No
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Weak	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Départ D7

Outlook	Temp.	Humidity	Wind	Play
Overcast	Cool	Normal	Strong	Yes (0, 1)

↓
S'(.)

1^{ère} généralisation possible : retrait de « temperature = cool »

Overcast	Cool	Normal	Strong	Yes (1, 1)	-1.0
Overcast	Cool	Normal	Strong	Yes (0, 1)	-0.92
Overcast	Cool	Normal	Strong	Yes (0, 1)	-0.92
Overcast	Cool	Normal	Strong	Yes (0, 1)	-0.92

2^{ème} généralisation possible : retrait de « wind = strong »

Overcast		Normal	Strong	Yes (1, 2)	-0.97
Overcast		Normal	Strong	Yes (0, 1)	-0.92
Overcast		Normal	Strong	Yes (0, 2)	-0.81

D'où l'intérêt d'utiliser l'estimateur laplacien des probas. Les deux derniers cas sont indiscernables avec une entropie de Shannon calculée à partir des fréquences usuelles.

3^{ème} généralisation possible : retrait de « humidity = normal »

Overcast		Normal		Yes (2, 5)	-0.92
Overcast		Normal		Yes (0, 4)	-0.65 *

→ Si outlook = overcast Alors Play = Yes (couv. = 4/14, conf. = 4/4)



Bottom-up – Avantages et inconvénients

(par rapport aux approches top-down notamment)

Avantages

Peut trouver des règles indétectables avec une approche top-down (ex. problème xor)
Notamment parce que l'approche démarre à partir d'individus observés

Inconvénients

Extrêmement lent, impraticable sur des bases importantes
Produit généralement un très grand nombre de règles
Dépendant du choix (au hasard) des individus de départ
Introuvable dans les logiciels habituellement accessibles



Algorithme génétique pour l'induction de règles

Dépasser les limites des algorithmes gloutons (greedy)



Principe des algorithmes génétiques

(Algorithme évolutionnaire utilisé en optimisation - [Wikipédia](#))

Population de base générée aléatoirement n chaînes de caractères ou de bits. 1 individu correspond à 1 chromosome.

Évaluation à chaque chaîne, une note (**fitness**) correspondant à son adaptation au problème.

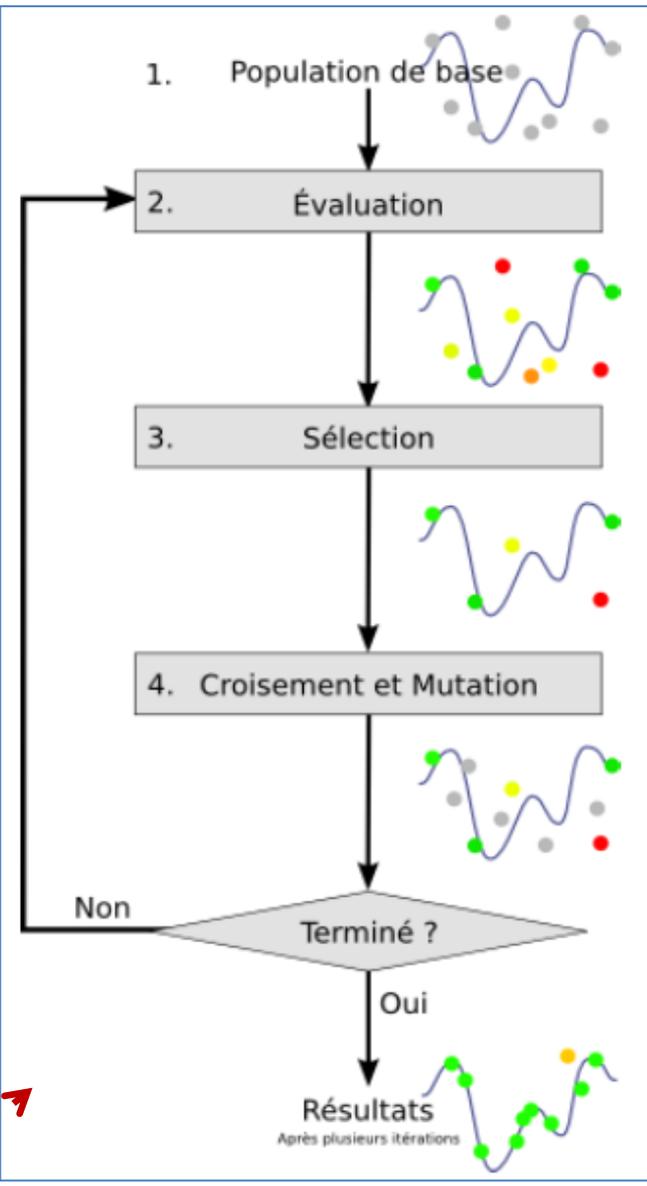
Sélection tirage au sort de n/2 couples de chaînes sur une roue biaisée. Chaque chaîne a une probabilité d'être tirée proportionnelle à son adaptation au problème. Optimisation possible : si l'individu le plus adapté n'a pas été sélectionné, il est copié d'office dans la génération intermédiaire à la place d'un individu choisi aléatoirement.

Croisement et mutation Chaque couple donne 2 chaînes filles.

Enjambement (croisement, hybridation). Probabilité : 70 %. Emplacement de l'enjambement choisi aléatoirement. Exemple : Chaînes parents : A : 00110100 ; B : 01010010 Chaînes filles : A' : 00010010 ; B' : 01110100 Croisement en 2 points plus efficace.

Mutations des chaînes filles. Probabilité : de 0,1 à 1 %. Inversion d'un bit au hasard ou remplacement au hasard d'un caractère par un autre. Probabilité fixe ou évolutive (auto-adaptation). On peut prendre probabilité = 1/nombre de bits.

Les algorithmes génétiques reprennent la théorie de [Darwin](#) : [sélection naturelle](#) de variations individuelles : les individus **les plus adaptés** (cf. **fitness**) tendent à survivre plus longtemps et à se reproduire plus aisément.



On a un processus itératif qui continue jusqu'à « convergence »



A.G. – Transformation d'une base de données

Tableaux de données initial

$A \in \{a1, a2\}$

$B \in \{b1, b2\}$

$C \in \{c1, c2\}$

$Y \in \{1, 0\}$

ID	A	B	C	Y
1	a1	b1	c1	1
2	a1	b1	c2	1
3	a1	b2	c1	1
4	a2	b1	c1	1
5	a1	b2	c2	0
6	a2	b1	c2	0
7	a2	b2	c1	1
8	a2	b2	c2	1



Transformation du tableau en chaînes de bits (représentation compatible avec les algorithmes génétiques)

ID	a1	a2	b1	b2	c1	c2	Y
1	1	0	1	0	1	0	1
2	1	0	1	0	0	1	1
3	1	0	0	1	1	0	1
4	0	1	1	0	1	0	1
5	1	0	0	1	0	1	0
6	0	1	1	0	0	1	0
7	0	1	0	1	1	0	1
8	0	1	0	1	0	1	1



A.G. – Codage des règles (chromosomes)

ID	A	B	C	Y
1	a1	b1	c1	1
2	a1	b1	c2	1
3	a1	b2	c1	1
4	a2	b1	c1	1
5	a1	b2	c2	0
6	a2	b1	c2	0
7	a2	b2	c1	1
8	a2	b2	c2	1

Codage binaire d'une règle

Une règle peut couvrir plusieurs individus.

Une règle peut être qualifiée par sa qualité (fitness). On choisit « conf' » dans notre cas.

Si A=a1 et B=b2 et C=c1 Alors Y = 1

a1	a2	b1	b2	c1	c2	Y	Conf'
1	0	0	1	1	0	1	0.67

Si A=a1 et B=b2 Alors Y = 1

a1	a2	b1	b2	c1	c2	Y	Conf'
1	0	0	1	1	1	1	0.50

↑
Pour signifier : « qu'importe la valeur prise par C ».

Si A=a1 Alors Y = 1

a1	a2	b1	b2	c1	c2	Y	Conf'
1	0	1	1	1	1	1	0.67

« Qu'importe les valeurs prises par B et C ».

Certaines représentations ne correspondent pas à une règle, nous les pénalisons fortement avec un fitness nul.

a1	a2	b1	b2	c1	c2	Y	Conf'
0	0	1	0	1	0	1	0.00

???



A.G. [1] – Génération de la population initiale

ID	A	B	C	Y
1	a1	b1	c1	1
2	a1	b1	c2	1
3	a1	b2	c1	1
4	a2	b1	c1	1
5	a1	b2	c2	0
6	a2	b1	c2	0
7	a2	b2	c1	1
8	a2	b2	c2	1

[1.b] Calcul du fitness de la population initiale

[1.a] Génération d'une population de (p = 5) règles – « p » est un paramètre de l'algorithme

	a1	a2	b1	b2	c1	c2	Y	Conf'
H1	1	0	0	1	1	0	1	0.67
H2	0	0	1	1	0	1	0	0.00
H3	1	0	1	1	0	1	0	0.50
H4	1	0	1	0	0	1	1	0.67
H5	1	1	1	0	1	0	1	0.75

Exemple

H5 : Si B = b1 et C = C1 Alors Y = 1



2 individus couverts
Et les deux sont Y = 1



$$conf'(r) = \frac{n_{k*r} + 1}{n_r + K} = \frac{2 + 1}{2 + 2} = 0.75$$



A.G. [2] – Sélection

Idée : sélection **stochastique** d'un sous-ensemble d'hypothèses (règles) performantes

Plusieurs stratégies possibles. Il faut que les règles qui ont un fitness élevé aient plus de chances d'être sélectionnées (ex. probabilité de tirage proportionnel au fitness, ou sélection par tournoi, etc.)

Ex. Sélectionner $p' = 3$
règles parmi les $p = 5$
(p' est paramétrable)



Dans notre exemple, on garde le meilleur, et on en choisit 2 en effectuant un tournoi parmi les 4 restants...

	a1	a2	b1	b2	c1	c2	Y	Conf'
H1	1	0	0	1	1	0	1	0.67
H2	0	0	1	1	0	1	0	0.00
H3	1	0	1	1	0	1	0	0.50
H4	1	0	1	0	0	1	1	0.67
H5	1	1	1	0	1	0	1	0.75



H5 est conservée
H1 vs. H2 → H1 est conservée
H3 vs. H4 → H4 est conservée



	a1	a2	b1	b2	c1	c2	Y	Conf'
H1	1	0	0	1	1	0	1	0.67
H4	1	0	1	0	0	1	1	0.67
H5	1	1	1	0	1	0	1	0.75

On dispose d'une
population de 3 règles



A.G. [3] – Croisement

Idée : croiser les paires d'hypothèses [échanges de parties du code génétique] pour faire apparaître des descendants

Plusieurs stratégies possibles, tout dépend du masque de croisement (intensité du brassage, cf. [croisement multipoints vs. croisement aléatoires](#)). Le nombre de paires à croiser est un paramètre de l'algorithme.

Ex. croisement à 1 point des hypothèses H1 et H4

H1	1	0	0	1	1	0	1
----	---	---	---	---	---	---	---

H6	1	0	0	1	0	1	1
----	---	---	---	---	---	---	---

H4	1	0	1	0	0	1	1
----	---	---	---	---	---	---	---

X

H7	1	0	1	0	1	0	1
----	---	---	---	---	---	---	---

Ex. On retrouve une population à 5 individus



	a1	a2	b1	b2	c1	c2	Y	Conf'
H1	1	0	0	1	1	0	1	0.67
H4	1	0	1	0	0	1	1	0.67
H5	1	1	1	0	1	0	1	0.75
H6	1	0	0	1	0	1	1	0.25
H7	1	0	1	0	1	0	1	0.67



A.G. [4] – Mutation

Idée : modifier (faire muter) aléatoirement un ou plusieurs gènes des chromosomes pour assurer une certaine diversité dans les hypothèses explorées

Objectif : permettre à l'algorithme d'explorer d'autres solutions et ne pas s'enfermer dans un optimum local. La probabilité de mutation doit être très faible pour laisser la prépondérance à la sélection + croisement, sinon l'algorithme équivaudrait à une recherche aléatoire.

Tableau de données

ID	A	B	C	Y
1	a1	b1	c1	1
2	a1	b1	c2	1
3	a1	b2	c1	1
4	a2	b1	c1	1
5	a1	b2	c2	0
6	a2	b1	c2	0
7	a2	b2	c1	1
8	a2	b2	c2	1

Population issue
du croisement

	a1	a2	b1	b2	c1	c2	Y	Conf'
H1	1	0	0	1	1	0	1	0.67
H4	1	0	1	0	0	1	1	0.67
H5	1	1	1	0	1	0	1	0.75
H6	1	0	0	1	0	1	1	0.25
H7	1	0	1	0	1	0	1	0.67

Population après mutation
de quelques (rares) gènes.

	a1	a2	b1	b2	c1	c2	Y	Conf'
H1	1	0	0	1	1	0	1	0.67
H4'	1	0	1	0	0	0	1	0.75
H5'	1	1	1	1	1	0	1	0.83
H6	1	0	0	1	0	1	1	0.25
H7'	1	0	0	0	1	0	1	0.00

Ah, ah ! Une règle pas mal du tout
« apparaît » : Si C = c1 Alors Y = 1



On passe à l'itération suivante. On recommence : sélection, croisement, etc... jusqu'à « convergence » de l'algorithme...



A.G. dans la pratique – La méthode GABIL (DeJong et al., 1993)

Décrit dans l'ouvrage de Tom Mitchell, « Machine Learning », 1997; [chapitre 9](#), page 251).

$GA(Fitness, Fitness_threshold, p, r, m)$

- *Initialize*: $P \leftarrow p$ random hypotheses
- *Evaluate*: for each h in P , compute $Fitness(h)$
- While $[\max_h Fitness(h)] < Fitness_threshold$

1. **Select**: Probabilistically select $(1 - r)p$ members of P to add to P_s .

$$\Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^p Fitness(h_j)}$$

2. **Crossover**: Probabilistically select $\frac{r \cdot p}{2}$ pairs of hypotheses from P . For each pair, $\langle h_1, h_2 \rangle$, produce two offspring by applying the Crossover operator. Add all offspring to P_s .
3. **Mutate**: Invert a randomly selected bit in $m \cdot p$ random members of P_s
4. **Update**: $P \leftarrow P_s$
5. **Evaluate**: for each h in P , compute $Fitness(h)$

- Return the hypothesis from P that has the highest fitness.

p : nombre d'hypothèses à générer (100 à 1000, **argh !**)
 r : fraction de population à remplacer par crossover (0.6)
 m : taux de mutation des gènes (0.001)

On s'arrête dès que la meilleure règle est « suffisamment » (paramétrable) bonne

Sélection avec probabilités inégales

En pratique, l'évaluation (calcul de la fonction fitness impliquant un scan de la base de données, très coûteux) ne se fait que lorsqu'une nouvelle génération de la population est engendrée.

A l'issue des calculs, on extrait « la » meilleure règle. On sépare les individus couverts et on recommence sur les individus restants. Nous sommes bien dans un cadre « separate-and-conquer ».



Bottom-up – Avantages et inconvénients

(par rapport aux approches top-down notamment)

Avantages

Optimisation plus performante, peut générer des règles introuvables par les autres approches

Inconvénients

Extrêmement lent, impraticable sur des bases importantes (nombre de scans trop élevé)

Gestion de la règle de convergence pas toujours évidente (« suffisamment » bonne ?)

Introuvable dans les logiciels habituellement accessibles



Règles d'association prédictives

Voir le support « [Les règles d'association](#) »



Principe des règles d'association prédictives

Reproches faits aux systèmes usuels d'induction de règles :

- On souhaite obtenir un système de classement précis
- Mais on souhaite également interpréter les résultats c.-à-d. comprendre la nature des relations entre (X_1, X_2, \dots) et Y



Les choix effectués (la sélection de variables) opérée par les algorithmes de règles masque le rôle des variables qui ne sont pas présents dans le modèle définitif.



Idée des règles d'association prédictive : règles d'association restreintes aux conséquents portant sur la variable cible.



De fait, nous disposons d'une description plus complète des relations entre les (X_1, X_2, \dots) et Y



Exemple

Association rule software - [Association rule viewer (Beta = 1.00)]

Analysis Rule management Window Help

Selected attributes: Outlook, Temperature, Humidity, Wind, PlayTennis

Learning Parameters: Min support = 0.1000, Min confidence = 0.7500, Max rule length = 5, Max cons. length = 1

Id	Antecedent	Consequent	Len...	Support	Confi...	Recall	F-meas...	Lift	Convi...
45	Outlook=Sunny & PlayTennis=No	Temperature=Hot	3	0.1429	1.0000	0.5000	0.6667	3.5000	2.7500
17	PlayTennis=Yes & Temperature=Hot & Wind=Weak	Outlook=Overcast	4	0.1429	1.0000	0.5000	0.6667	3.5000	2.7500
46	Humidity=High & Outlook=Sunny & PlayTennis=No	Temperature=Hot	4	0.1429	1.0000	0.5000	0.6667	3.5000	2.7500
16	PlayTennis=Yes & Temperature=Hot	Outlook=Overcast	3	0.1429	1.0000	0.5000	0.6667	3.5000	2.7500
23	Outlook=Rain & Wind=Strong	PlayTennis=No	3	0.1429	1.0000	0.4000	0.5714	2.8000	2.5000
24	Outlook=Sunny & Temperature=Hot	PlayTennis=No	3	0.1429	1.0000	0.4000	0.5714	2.8000	2.5000
22	Humidity=High & Wind=Strong	PlayTennis=No	3	0.1429	1.0000	0.4000	0.5714	2.8000	2.5000
21	Humidity=High & PlayTennis=No & Temperature=Hot	Outlook=Sunny	4	0.1429	1.0000	0.4000	0.5714	2.8000	2.5000
18	Humidity=Normal & PlayTennis=No	Outlook=Rain	3	0.1429	1.0000	0.4000	0.5714	2.8000	2.5000
20	PlayTennis=No & Temperature=Hot	Outlook=Sunny	3	0.1429	1.0000	0.4000	0.5714	2.8000	2.5000
25	Humidity=High & Outlook=Sunny & Temperature=Hot	PlayTennis=No	4	0.1429	1.0000	0.4000	0.5714	2.8000	2.5000
19	PlayTennis=No & Temperature=Mild	Outlook=Rain	3	0.1429	1.0000	0.4000	0.5714	2.8000	2.5000
47	Humidity=High & Outlook=Rain	Temperature=Mild	3	0.1429	1.0000	0.3333	0.5000	2.3333	2.2500
3	Outlook=Sunny & Temperature=Hot	Humidity=High	3	0.1429	1.0000	0.2857	0.4444	2.0000	2.0000
7	Outlook=Sunny & PlayTennis=No & Temperature=Hot	Humidity=High	4	0.1429	1.0000	0.2857	0.4444	2.0000	2.0000
14	Temperature=Cool & Wind=Weak	Humidity=Normal	3	0.1429	1.0000	0.2857	0.4444	2.0000	2.0000
15	PlayTennis=Yes & Temperature=Cool & Wind=Weak	Humidity=Normal	4	0.1429	1.0000	0.2857	0.4444	2.0000	2.0000
4	PlayTennis=No & Temperature=Hot	Humidity=High	3	0.1429	1.0000	0.2857	0.4444	2.0000	2.0000

63 règles extraites, avec les paramètres suivants

23 règles avec « PlayTennis » dans le conséquent. Pour « playtennis = no », la n°23 et n°24 correspondent à ceux extraits par l'induction de règles (page 28). Mais d'autres règles apparaissent également.

Id	Antecedent	Consequent	Length	Support	Confidence	Recall	F-measure	Lift	Conviction	
23	Outlook=Rain & Wind=Strong	PlayTennis=No	3	0.1429	1	1	0.4	0.5714	2.8	2.5
24	Outlook=Sunny & Temperature=Hot	PlayTennis=No	3	0.1429	1	1	0.4	0.5714	2.8	2.5
22	Humidity=High & Wind=Strong	PlayTennis=No	3	0.1429	1	1	0.4	0.5714	2.8	2.5
25	Humidity=High & Outlook=Sunny & Temperature=Hot	PlayTennis=No	4	0.1429	1	1	0.4	0.5714	2.8	2.5
33	Humidity=Normal & Outlook=Sunny	PlayTennis=Yes	3	0.1429	1	0.2222	0.3636	1.5556	1.5	1.5
36	Outlook=Overcast & Temperature=Hot	PlayTennis=Yes	3	0.1429	1	0.2222	0.3636	1.5556	1.5	1.5
38	Outlook=Sunny & Temperature=Mild	PlayTennis=Yes	3	0.1429	1	0.2222	0.3636	1.5556	1.5	1.5
29	Humidity=High & Outlook=Overcast	PlayTennis=Yes	3	0.1429	1	0.2222	0.3636	1.5556	1.5	1.5
37	Outlook=Overcast & Wind=Weak	PlayTennis=Yes	3	0.2143	1	0.3333	0.5	1.5556	1.875	1.875
42	Humidity=High & Temperature=Mild & Wind=Weak	PlayTennis=Yes	4	0.2143	1	0.3333	0.5	1.5556	1.875	1.875
26	Outlook=Overcast	PlayTennis=Yes	2	0.2857	1	0.4444	0.6154	1.5556	2.25	2.25
41	Humidity=High & Outlook=Overcast & Wind=Weak	PlayTennis=Yes	4	0.1429	1	0.2222	0.3636	1.5556	1.5	1.5
32	Humidity=Normal & Outlook=Overcast	PlayTennis=Yes	3	0.1429	1	0.2222	0.3636	1.5556	1.5	1.5
44	Outlook=Overcast & Temperature=Hot & Wind=Weak	PlayTennis=Yes	4	0.1429	1	0.2222	0.3636	1.5556	1.5	1.5
39	Temperature=Cool & Wind=Weak	PlayTennis=Yes	3	0.1429	1	0.2222	0.3636	1.5556	1.5	1.5
43	Humidity=Normal & Temperature=Cool & Wind=Weak	PlayTennis=Yes	4	0.1429	1	0.2222	0.3636	1.5556	1.5	1.5
31	Humidity=High & Wind=Weak	PlayTennis=Yes	3	0.2857	0.8	0.4444	0.5714	1.2444	1.3125	1.3125
28	Wind=Weak	PlayTennis=Yes	2	0.5	0.7778	0.7778	0.7778	1.2099	1.375	1.375
40	Temperature=Mild & Wind=Weak	PlayTennis=Yes	3	0.2143	0.75	0.3333	0.4615	1.1667	1.125	1.125
34	Humidity=Normal & Temperature=Cool	PlayTennis=Yes	3	0.2143	0.75	0.3333	0.4615	1.1667	1.125	1.125
35	Humidity=Normal & Wind=Weak	PlayTennis=Yes	3	0.2143	0.75	0.3333	0.4615	1.1667	1.125	1.125
27	Temperature=Cool	PlayTennis=Yes	2	0.2143	0.75	0.3333	0.4615	1.1667	1.125	1.125
30	Humidity=High & Temperature=Mild	PlayTennis=Yes	3	0.2143	0.75	0.3333	0.4615	1.1667	1.125	1.125



Enjeux des règles d'association prédictives

Coûts des calculs

- Ne produire que les itemsets fréquents intégrant la variable cible
- Ne produire que les règles avec la variable cible dans le conséquent
- ➔ On doit dépasser le simple (et prohibitif) « construire toutes les règles d'association »
- + « supprimer les règles ne comportant pas la cible dans le conséquent »

Comportement en classement

- Certaines règles sont redondantes, il faut pouvoir les simplifier
 - Les règles sont de qualité inégale, il faut en sélectionner les meilleures
 - Adopter une stratégie lorsque plusieurs règles sont déclenchées
- (Prendre les « k » les plus précises ? Ou vote ? Ou vote pondéré ?)



- Adopter une stratégie lorsqu'aucune règle n'est déclenchée

Ex. ces règles sont redondantes par rapport à la n°26

Id	Antecedent	Consequent	Suppo	Confidence
26	Outlook=Overcast	PlayTennis=Yes	0.2857	1
29	Humidity=High & Outlook=Overcast	PlayTennis=Yes	0.1429	1
32	Humidity=Normal & Outlook=Overcast	PlayTennis=Yes	0.1429	1
36	Outlook=Overcast & Temperature=Hot	PlayTennis=Yes	0.1429	1
37	Outlook=Overcast & Wind=Weak	PlayTennis=Yes	0.2143	1
41	Humidity=High & Outlook=Overcast & Wind=Weak	PlayTennis=Yes	0.1429	1
44	Outlook=Overcast & Temperature=Hot & Wind=Weak	PlayTennis=Yes	0.1429	1

Remarque : ici, les règles ont toutes la même confiance (conf. = 1). Est-ce que la simplification est licite si elles n'ont pas la même importance ?



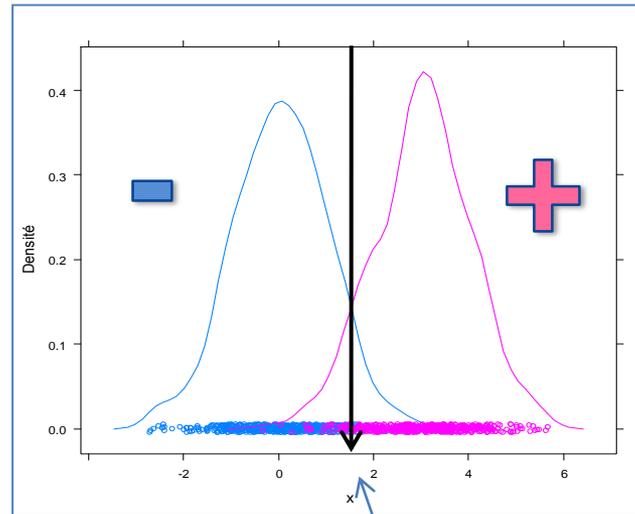
Règles prédictives floues



Règles floues – Problème de la discrétisation « nette » (crisp)

Idée : lors de la discrétisation des variables prédictives quantitatives, les seuils obtenus ne tiennent pas compte de l'imprécision ...

Découpage de X en fonction de Y (à 2 modalités {+, -})



```
#générer les valeurs
v1 <- rnorm(500,mean=0,sd=1)
v2 <- rnorm(500,mean=3,sd=1)
x <- c(v1,v2)
#générer la variable cible
f <- factor(c(rep(1,500),rep(2,500)))
#graphique
library(lattice)
densityplot(x,groups=f)
```

Si $X \leq 1.5$ Alors $Y = -$ Sinon $Y = +$

Cette borne de découpage (1.5) occulte les informations suivantes

- elle est calculée sur un échantillon, donc imprécise par nature ;
- des observations sont du « mauvais côté » dans le voisinage immédiat du seuil.

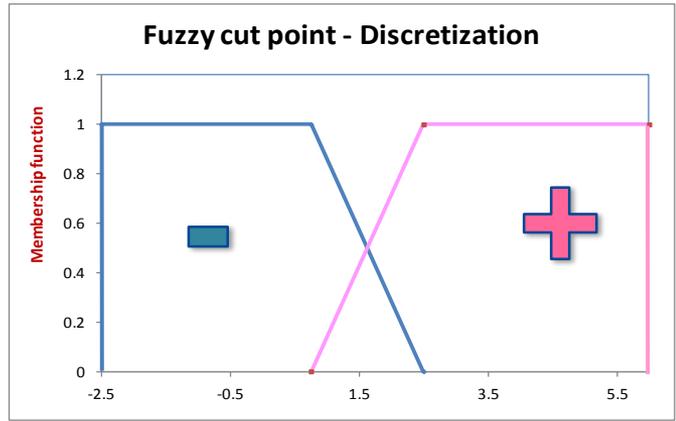


Règles floues – Discrétisation « floue » (fuzzy)

Idée : instiller une gradation dans l'appartenance à une région... utiliser une « fonction d'appartenance ».

Découpage de X en fonction de Y (à 2 modalités)

La règle est plus complexe mais rend mieux compte de la réalité.



Si $X \in < -2.5, -2.5, 0.75, 2.5 >$ Alors $Y = -$
Si $X \in < 0.75, 2.5, 6.0, 6.0 >$ Alors $Y = +$

Toutes les règles sont évaluées en classement (μ est la fonction d'appartenance)



$X = -1 : \mu(-) = 1$ et $\mu(+)= 0 \rightarrow$ Prédiction = -
 $X = 2 : \mu(-) = 0.3$ et $\mu(+)= 0.7 \rightarrow$ Prédiction = +

Mais la décision peut être nuancée

Valeurs calculées par interpolation

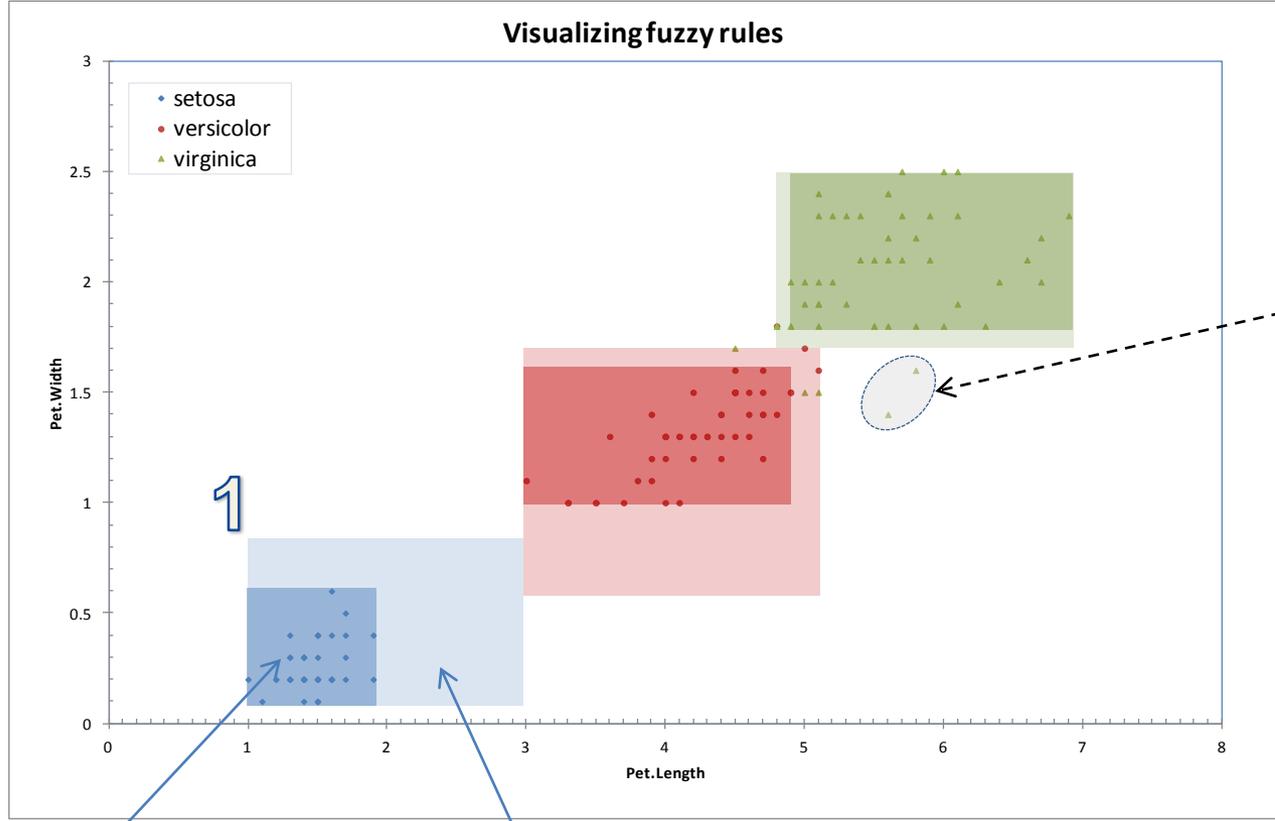


Règles floues avec plusieurs variables

Visualisation des règles dans le plan – 2 variables prédictives

3 règles sont
générées
(principalement)

- (1) Si $\text{pet.Length} \in < 1.0, 1.0, 1.9, 3.0 >$ et $\text{pet.width} \in < 0.1, 0.1, 0.6, 0.8 >$ Alors Iris = setosa
- (2) Si $\text{pet.Length} \in < 3.0, 3.0, 4.9, 5.1 >$ et $\text{pet.width} \in < 0.6, 1.0, 1.6, 1.7 >$ Alors Iris = versicolor
- (3) Si $\text{pet.Length} \in < 4.8, 4.9, 6.9, 6.9 >$ et $\text{pet.width} \in < 1.7, 1.8, 2.5, 2.5 >$ Alors Iris = virginica



Non représentée ici, une autre règle a été générée pour cette partie résiduelle

! Ce n'est pas le cas ici, mais les zones d'appartenance faible peuvent se recouvrir.

Zone d'appartenance forte ($\mu = 1$)

Zone d'appartenance faible (décroissante, μ décroît jusqu'à 0)



Règles prédictives floues – Avantages et inconvénients

(par rapport aux approches « crisp » notamment)

Avantages

L'approche est plus subtile, tient mieux compte de la réalité des données

Moindre « variance » (dépendance aux données), donc a priori plus performante

Applicable aux variables prédictives qualitatives (pour la gestion de valeurs manquantes notamment)

Inconvénients

Lecture difficile des règles, surtout lorsque beaucoup de variables en jeu

Implantation dans les systèmes d'information, nécessite des calculs supplémentaires en classement



Bilan



L'induction de règles consiste à construire une base de règles pour l'analyse prédictive.

Les méthodes separate-and-conquer constituent une alternative peu connue des arbres de décision pour produire des règles. Les bases induites sont généralement plus compactes. En revanche, des problèmes apparaissent (nécessité de discrétisation préalable des variables quantitatives, s'assurer de l'exhaustivité des règles avec une règle par défaut, etc.)

Nous avons mis l'accent sur les approches top-down. D'autres stratégies separate-and-conquer existent (bottom-up, algorithmes génétiques). Elles souffrent généralement d'un temps de calcul prohibitif. Elles sont impraticables dès qu'il s'agit d'appréhender des bases de données volumineuses.

Les règles d'association prédictives sont une alternative intéressante. La profusion des règles enrichit l'interprétation des relations entre les variables. Mais cette même profusion devient contraignante dans le processus de décision (gérer le déclenchement de plusieurs règles).

Les règles de prédictions floues permettent de prendre en compte l'imprécision des bornes de discrétisation, inhérente au découpage en intervalles des variables quantitatives à partir d'un échantillon de données.

Par rapport aux arbres de décision, toutes ces techniques sont moins rapides sur les grandes bases...



Références



« [Induction de règles prédictives](#) », novembre 2009.

Présentation de l'approche separate-and-conquer, comparaison de logiciels (Tanagra, Weka, R avec le package Rweka, RapidMiner, Orange)

« [Listes de décision \(vs. arbres de décision\)](#) », mars 2008.

Comparaison des performances avec CART.

« [Règles d'association prédictives](#) », février 2009.

Montre comment préciser la variable cible (et modalité cible) pour les règles d'association sous Tanagra. Le cadre est plutôt descriptif que prédictif.

« [Induction de règles floues avec Knime](#) », février 2010.

Knime propose l'induction de règles floues. Un petit topo introductif sur la spécificité des règles floues précède la partie démo.

Remarque : des packages existent également pour le logiciel R (ex. [frbs](#), ...)



Ouvrages et articles de référence

R. Quinlan, « C4.5: Programs for Machine Learning », 1993.

Présente en détail la méthode C4.5 Rules et son intérêt (chapitre 5: From Trees to Rules)

J. Furnkranz, « [Separate-and-Conquer Rule Learning](#) », in Artificial Intelligence Review, 13: 3-54, 1999.

Un survey qui fait autorité concernant les méthodes separate-and-conquer.

P. Clark, T. Nibblet, « [The CN2 Induction Algorithm](#) », in Machine Learning, 3(4): 261-283, 1989.

Une des premières méthodes efficaces pour produire les listes de décision.

P. Clark, R. Boswell, « [CN2: Some recent improvements](#) », in 5th European Conf. Machine Learning, 151-163, 1991.

Une variante de CN2 permettant de produire des règles indépendantes.

T. Mitchell, « Machine Learning », McGraw Hill, 1997 ; chapitre 9 : « [Genetic Algorithms](#) ».

Donne des détails sur les variantes de l'algorithme (différents croisements, paramétrage, etc.)

W. Li, J. Han, J. Pei, « [CMAR](#): accurate and efficient classification based on multiple class association rules », in ICDM'01, 369-376, 2001.

M.R. Berthold, « [Mixed fuzzy rule formation](#) », in Int. Jour. of Approximate Reasoning, 32: 67-84, 2003.

Présente la méthode qui est implémentée dans le logiciel Knime (cf. [tutoriel](#)).

