

Arbres de décision

F. De Comité

Licence-Master Informatique

15 janvier 2009

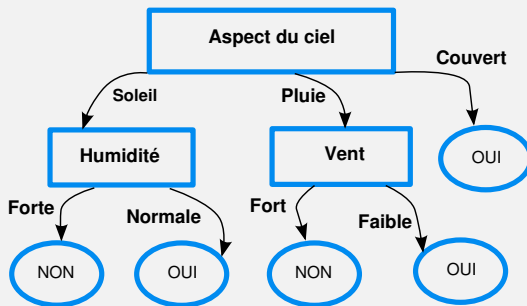
Notes

Caractéristiques

- Algorithme de classification.
- Prédiction d'une classe **discrète**.
- Probabilité d'appartenance à chaque classe.

Notes

Un exemple



Notes

L'échantillon

Ciel, Température, Humidité, vent, Jouer ?

Soleil, Chaud, Forte, faible, Non
Soleil, Chaud, Forte, Fort, Non
Couvert, Chaud, Forte, faible, Oui
Pluie, Doux, Forte, faible, Oui
Pluie, Frais, Normale, faible, Oui
Pluie, Frais, Normale, Fort, Non
Couvert, Frais, Normale, Fort, Oui
Soleil, Doux, Forte, faible, Non
Soleil, Frais, Normale, faible, Oui
Pluie, Doux, Normale, faible, Oui
Soleil, Doux, Normale, Fort, Oui
Couvert, Doux, Forte, Fort, Oui
Couvert, Chaud, Normale, faible, Oui
Pluie, Doux, Forte, Fort, Non

Notes

Comprendre l'arbre

- Rectangles : tests sur un attribut.
- Arcs : valeur de l'exemple pour le test.
- Feuilles (ellipses) : classe.

Notes

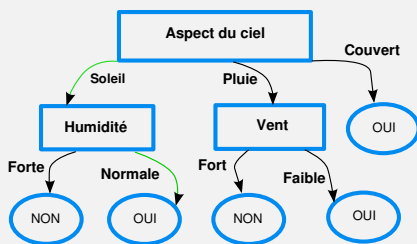
Utiliser l'arbre pour classer un exemple

- Partir de la racine.
- Répondre aux questions en suivant la branche correspondante.
- Lire la valeur de la feuille atteinte.

Notes

Exemple

- Aspect du ciel : Soleil
- Humidité : Normale
- Vent : Faible
- Température : 30



Notes

Exemple

Remarques

- Pour un exemple, tous les attributs ne servent pas.
- Selon l'exemple, des attributs différents sont utilisés.
- La température n'intervient pas.

Notes

Construire des arbres

- Algorithme construisant un arbre **à partir d'exemples** ?
- Trouver le **meilleur** arbre ?
 - Pour classer exactement l'échantillon ?
 - Pour classer correctement un nouvel exemple ?

Notes

Construction des arbres

Remarque

Si deux descriptions identiques ont des classes identiques, alors il existe un arbre classant parfaitement l'échantillon.

Remarque

Un tel arbre n'est pas forcément intéressant : il colle trop aux données : mauvais **pouvoir de généralisation**.

Notes

Trouver le meilleur arbre

- Si la population est infinie : sans espoir.
- Population finie : engendrer et tester tous les arbres ? Impraticable.

Notes

Construction de l'arbre

Puisqu'on ne peut espérer trouver le meilleur arbre, on se contentera d'un **bon** arbre :

- Arbre compact : éviter les **peignes** (listes de décisions).
- Arbre d'erreur faible.
- Arbre lisible, compréhensible par un non-expert.
- Complexité algorithmique (temps espace) acceptable.

Notes

Algorithme : ID3 C4.5

Fonction **CréerNoeud**(S, L)

Entrées

- S : ensemble d'exemples.
- L : liste des attributs non encore utilisés.

Sortie : Un arbre de décision.

- 1 Si tous les exemples sont de la même classe, créer et retourner une feuille étiquetée par cette classe.
- 2 Sinon :
 - 1 Trouver le meilleur attribut dans $L : A$
 - 2 Répartir les exemples de S en n sous-ensembles, selon leur valeur pour cet attribut.
 - 3 Allouer un nœud étiqueté par A
 - 4 Pour toutes les valeurs de A : appeler **CréerNoeud**($S_i, L - A$) et faire de ces nœuds les fils du nœud étiqueté par A

Notes

Algorithme

- **But** : Obtenir des feuilles où tous les exemples sont de la même classe.
- **Moyen** :
 - Trouver le **meilleur** attribut en chaque nœud.
 - Pouvoir mesurer le **degré de mélange** d'un ensemble d'exemples.

Notes

Mesure du degré de mélange

- n exemples, deux classes : $\frac{n}{2}$ exemples de classe $c_1 \rightarrow$ mélange maximal.
- n exemples, deux classes : $n - 1$ exemples de classe c_1 : ensemble presque pur.
- n exemples, deux classes : 1 exemple de classe c_1 : idem.

Notes

Mesurer le mélange

- Fonction de p_1, \dots, p_n (les proportions d'exemples pour chaque classe).
- Maximale quand tous les p_i sont égaux.
- Minimale quand tous les exemples sont de la même classe.

Deux classes

- Fonction de p (l'autre classe : $1 - p$)
- Maximale pour $p = \frac{1}{2}$
- Minimale quand $p = 0$ ou $p = 1$

Notes

Mesurer le mélange

- Polynômes : paraboles, quadriques.

- Fonction de Gini :

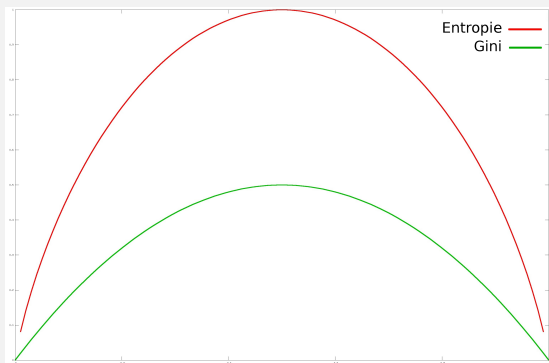
$$2p(1-p)$$
$$1 - \sum_{i=1}^n p_i^2$$

- Entropie :

$$-p \log_2(p) - (1-p) \log_2(1-p)$$
$$-\sum_i p_i \log_2(p_i)$$

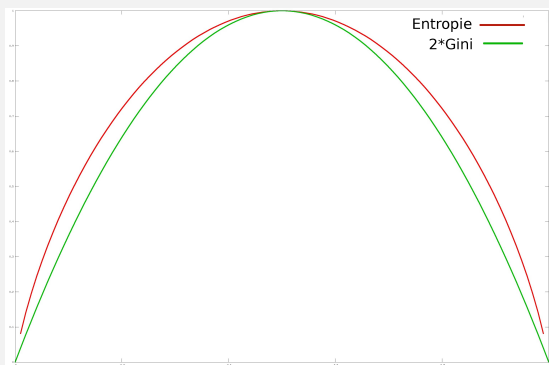
Notes

Mesurer le mélange



Notes

Mesurer le mélange



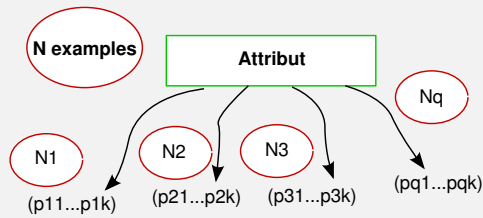
Notes

Entropie

- Thermodynamique ...
- Théorie de l'information : nombre moyen de bits nécessaires pour coder la classe d'un exemple ...

Notes

Utiliser l'entropie



Notes

Utiliser l'entropie

- Au départ : N exemples, k classes, entropie E .
- Pour l'attribut testé : q branches.
- N_i exemples suivent la branche i .
- Dans la branche i , la répartition des classes est donnée par $p_{i1} \dots p_{ik}$

Questions

- Que nous apprend de plus, pour la classe, le fait de connaître la valeur de l'exemple pour l'attribut testé ?
- Quel est le **gain** associé à cet attribut ?

Notes

Gain

Définition

Le gain d'un ensemble d'exemples S pour un attribut A est défini par :

$$\text{Gain}(S, A) = \text{Entropie}(S) - \sum_{i=1}^q \frac{|S_i|}{|S|} \times \text{Entropie}(S_i)$$

- S_i : sous-ensemble des exemples suivant la branche i
- $|S_i|$: cardinal de S_i .
- q : nombre de branches du test.

Définition

L'attribut choisi par l'algorithme est celui qui maximise le gain.

Notes

Gain : exemple de calcul

- 10 exemples, 2 classes (C_1 et C_2) (5 exemples de chaque classe).
- deux attributs à tester : A (deux valeurs A_1 et A_2), et B (deux valeurs B_1 et B_2).

	A_1	A_2
C_1	4	1
C_2	1	4

	B_1	B_2
C_1	3	2
C_2	2	3

Notes

Gain : exemple de calcul

$$E(S) = -\frac{5}{10} \times \log_2\left(\frac{5}{10}\right) - \frac{5}{10} \times \log_2\left(\frac{5}{10}\right) = 1$$

$$\text{Gain}(S, A) = 1 - \frac{5}{10} \times \text{Entropie}(4, 1) - \frac{5}{10} \times \text{Entropie}(1, 4)$$

$$\text{Gain}(S, A) = 1 - \frac{5}{10} \times \text{Entropie}(3, 2) - \frac{5}{10} \times \text{Entropie}(2, 3)$$

Notes

Gain : exemple de calcul

$$\text{Entropie}(4, 1) = -\frac{1}{5} \times \log_2\left(\frac{1}{5}\right) - \frac{4}{5} \times \log_2\left(\frac{4}{5}\right) = 0.72$$

$$\text{Entropie}(2, 3) = -\frac{2}{5} \times \log_2\left(\frac{2}{5}\right) - \frac{3}{5} \times \log_2\left(\frac{3}{5}\right) = 0.97$$

$$\text{Gain}(S, A) = 1 - 0.5 \times 0.72 - 0.5 \times 0.72 = 0.28$$

$$\text{Gain}(S, B) = 1 - 0.5 \times 0.97 - 0.5 \times 0.97 = 0.03$$

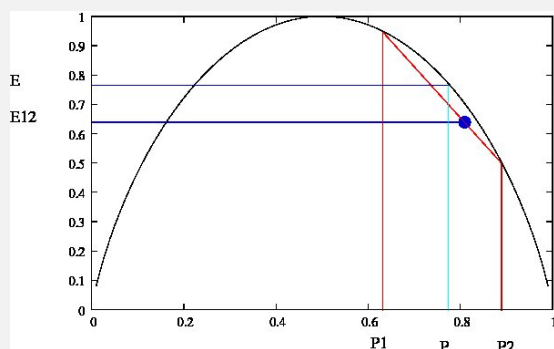
Notes

Gain : exemple de calcul

- L'attribut A conduit au meilleur gain : les deux fils du n ud sont **plus purs**
- Conna tre la valeur d'un exemple pour l'attribut A m'apporte en moyenne plus d'information sur sa classe que sa valeur pour l'attribut B
- Cette connaissance me permet de diminuer l'erreur de classification.

Notes

Concavit 



Notes

Aspect du ciel

$$\text{Gain}(\text{Aspect}, S) = E(S) - \frac{5}{14} \times E(3, 2) - \frac{4}{14} \times E(4, 0) - \frac{5}{14} \times E(2, 3)$$

$$E(S) = -\frac{9}{14} \times \log_2\left(\frac{9}{14}\right) - \frac{5}{14} \times \log_2\left(\frac{5}{14}\right)$$

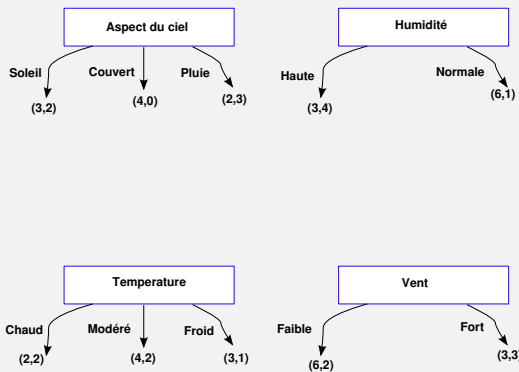
$$E(4, 0) = 0$$

$$E(3, 2) = E(2, 3) = -\frac{2}{5} \times \log_2\left(\frac{2}{5}\right) - \frac{3}{5} \times \log_2\left(\frac{3}{5}\right)$$

$$\text{Gain}(\text{Aspect}, S) = 0.246$$

Notes

Gain : exemple



Notes

Gain deuxième exemple

- Aspect : 0.246
- Humidité : 0.151
- Vent : 0.048
- Température : 0.029

C'est Aspect qui est choisi à la racine.

Notes

Gain : deuxième exemple

- Couvert : tous les exemples sont de la même classe : fini.
- Soleil :
 - Humidité : 0.970 → choisi.
 - Température : 0.570
 - Vent : 0.019
- Pluie : Vent est choisi.

Notes

Quelques remarques

Remarque

Lecture aisée, possibilité d'extraire des règles :
si (Aspect=Soleil) et (Humidité=Normale) alors oui

Remarque

Permet de repérer les attributs les plus déterminants pour la classification.

Remarque

Complexité : A chaque niveau de l'arbre, pour chaque attribut possible, parcourir l'ensemble complet des exemples :
 $O(n \times k \times p) \simeq O(n)$ (n : nombre d'exemples, p : nombre d'attributs, k : profondeur de l'arbre).

Notes

Biais

- On ne parcourt pas l'ensemble de tous les arbres possibles.
- Au départ, tous les arbres sont possibles
- Quand un attribut est choisi : on restreint l'espace de recherche.
- On ne remet jamais en cause les choix précédents.
- En sélectionnant les attributs les plus discriminants d'abord, on engendre des arbres courts et compacts (par observation ...).

Notes

Limitations

- 3 attributs binaires (0,1), 8 exemples (de 0 à 7 écrit en binaire), 2 classes (parité du nombre de '1' dans la description).
- Aucun des attributs n'apporte de gain \rightarrow arbre réduit à sa racine.
- Un arbre parfait existe pourtant !

Notes

Attributs continus

A partir de cet attribut continu, obtenir un test binaire (n -aire).
Couper l'intervalle des valeurs de l'attribut en ce nœud en n sous-intervalles : discrétiser.

- Combien d'intervalles ?
- Où effectuer les coupures ?
- Pas de solution trop coûteuse ! (ne pas explorer toutes les coupures en n intervalles, n variant).

Notes

Attributs continus

Exemples

Exemple	E1	E2	E3	E4	E5	E6
valeur	2	3	1	0	0	5
classe	1	2	2	2	1	1

Ordonner selon les valeurs de l'attribut :

Exemple	E4	E5	E3	E1	E2	E6
valeur	0	0	1	2	3	5
classe	2	1	2	1	2	1

Notes

Attributs continus

7 coupures binaires possibles : laquelle apporte le meilleur gain ?

Remarque

L'entropie initiale étant toujours la même, on se contente de **minimiser** l'entropie moyenne des fils.

- Coupure 1 : un seul fils (3 c1, 3 c2) : entropie moyenne nulle, pas de gain.
- Coupure 2 : $\frac{1}{6} \times E(0, 1) + \frac{5}{6} \times E(3, 2) = 0.8$ Coupure choisie.
- Coupure 3 : $\frac{2}{6} \times E(1, 1) + \frac{4}{6} \times E(2, 2)$ pas de gain.
- Coupure 4 : $\frac{3}{6} \times E(1, 2) + \frac{3}{6} \times E(2, 1) = 0.92$
- Coupure 5 : $\frac{4}{6} \times E(2, 2) + \frac{1}{6} \times E(1, 1)$: comme la coupure 3 (pas de gain).
- Coupure 6 : comme la coupure 2.
- Coupure 7 : voir coupure 1.

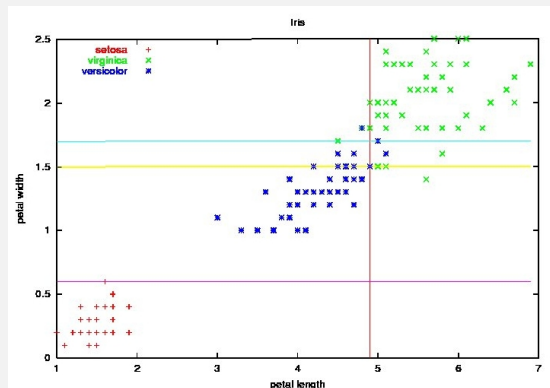
Notes

Attributs continus

- Complexité : coût d'un tri (la recherche de la coupure peut se faire en temps linéaire).
- Complexité de l'algorithme complet : $O(n \log n)$.
- Seuil choisi = information supplémentaire sur les données (discretisation ...).

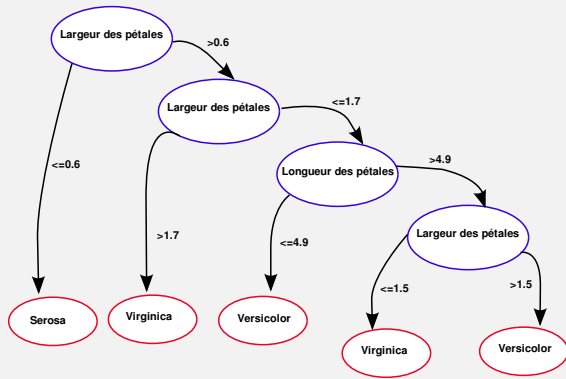
Notes

Attributs continus



Notes

Attributs continus



F. De Comité Arbres de décision

Notes

Valeurs manquantes

Différentes stratégies

- Ignorer l'exemple : **modifie l'échantillon ...**
- Créer une nouvelle valeur 'manquant' : **être manquant devient un critère apparaissant dans le modèle ...**
- Remplacer la valeur :
 - Au hasard
 - Majoritaire (en général, par rapport à la classe, par rapport à des exemples similaires : cf clustering ...)

F. De Comité Arbres de décision

Notes

Valeurs manquantes

Une autre stratégie

- Soit un exemple ayant une valeur manquante pour l'attribut en cours d'évaluation.
- Puisqu'on ne connaît pas la valeur de l'attribut pour cet exemple, on va lui affecter une probabilité d'avoir chaque valeur de l'attribut.
- On estimera cette probabilité à partir des exemples ayant une valeur pour cet attribut (proportions d'exemples sur chaque branche)
- On affectera à chaque branche **une partie** de l'exemple.

F. De Comité Arbres de décision

Notes

Valeurs manquantes : exemple

Exemple	E1	E2	E3	E4	E5	E6
valeur	A1	A2	A2	?	A2	A1
classe	C1	C1	C2	C2	C1	C2

F. De Comité Arbres de décision

Notes

Valeurs manquantes

- Gain calculé sur les exemples sans valeur manquante.
- Gain coefficienté par la proportion d'exemples sans valeur manquante.
- **Ensuite** : $\frac{3}{5}$ de D4 suivent la branche A1, $\frac{2}{5}$ suivent la branche A2.

Notes

Valeurs manquantes

Remarque

A chaque exemple présent en un nœud, est associé un **poids** : c'est ce poids qui est pris en compte lors des calculs de gain.

Remarque

Le calcul du gain s'étend sans problème aux poids fractionnaires, de même que la définition de la classe majoritaire en une feuille.

Remarque

Une fraction d'exemple peut encore ensuite être fractionnée à nouveau.

Notes

Valeurs manquantes

Classification

- Si l'exemple arrive en un nœud correspondant à un attribut pour lequel sa valeur est manquante, **il suit toutes les branches** issues de ce test, au prorata du nombre d'exemples ayant suivi ces branches lors de la construction de l'arbre.
- On calcule alors la **probabilité** qu'il a d'appartenir à chaque classe.
- La classe de plus forte probabilité est alors affectée à cet exemple.

Remarque

La valeur de retour du classifieur pourrait être justement ce vecteur de probabilités.

Notes

Valeur manquantes : Exemple

On reprend l'exemple du golf (1er exemple modifié : humidity=high changé en ?) :

sunny	hot	?	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	no 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

Notes

```

outlook = overcast : yes      outlook = overcast : yes
(4.0)                         (4.0)
outlook = sunny :           outlook = sunny :
| humidity = high : no      | humidity = high : no
(3.0)                         (2.5)
| humidity = normal : yes   | humidity = normal :
(2.0)                         | | temperature = hot : no
outlook = rainy :           (0.5)
| windy = TRUE : no (2.0)    | | temperature = mild :
| windy = FALSE : yes (3.0)  yes (1.0)
                              | | temperature = cool :
                              yes (1.0)
                              outlook = rainy :
                              | windy = TRUE : no (2.0)
                              | windy = FALSE : yes (3.0)

```

Notes

Valeurs manquantes : exemple

- Le premier attribut est toujours **outlook** : son gain n'est pas modifié, celui de **humidity** diminue.
- Pour **outlook=sunny** : 5 exemples, dont un avec valeur manquante.

$$\text{Gain}(\text{Hum}, S) = \frac{4}{5} \times (E(3, 2) - \frac{1}{2} \times E(0, 2) - \frac{1}{2} \times E(0, 2))$$

$$\text{Gain}(\text{Hum}, S) = \frac{4}{5} \times (E(3, 2) - 0)$$

- Gain maximum!
- L'exemple à valeur manquante descend dans les deux branches (d'où les 2.5 et 0.5).

Notes

Valeurs manquantes : exemple

Utilisation en classification

A partir de l'arbre construit sans valeurs manquantes

```

outlook = overcast : yes (4.0)
outlook = sunny :
| humidity = high : no (3.0) | humidity = normal :
yes (2.0)
outlook = rainy :
| windy = TRUE : no (2.0)
| windy = FALSE : yes (3.0)

```

Notes

Valeurs manquantes : exemple

Soit l'exemple (?, hot, high, FALSE).

- il descend dans les 3 branches de **outlook** :
- $\frac{4}{14}$ dans overcast : classe yes.
- $\frac{5}{14}$ dans sunny : puis dans **humidity=high** : classe no.
- $\frac{5}{14}$ dans rainy : puis dans **windy=false** : classe yes.
- Proba(yes) = $\frac{9}{14}$ Proba(no) = $\frac{5}{14}$: on lui attribue la classe yes.

Notes

Gain ratio

Le critère du gain avantage les attributs discrets ayant beaucoup de valeurs distinctes :
Soit A un attribut tel que les n exemples aient une valeur différente pour A

$$\text{Gain}(A, S) = \text{Entropie}(S) - \sum_{i=1}^n \frac{1}{n} \times \text{Entropie}(0, 1)$$

$$\text{Gain}(A, S) = \text{Entropie}(S) \text{ maximum !}$$

Notes

Gain ratio

Pénaliser les attributs :

- Prenant beaucoup de valeurs différentes.
- Répartissant équitablement les exemples dans toutes les branches.

Notes

Gain ratio

Definition

$$\text{SplitInfo}(S, A) = \sum_{i=1}^k -\frac{|S_i|}{|S|} \times \log_2 \frac{|S_i|}{|S|}$$

avec :

- S : ensemble d'exemples. S_i : nombre d'exemples valant A_i pour l'attribut A .
- k : nombre de valeurs de l'attribut A

Notes

Gain Ratio

Definition

$$\text{GainRatio}(S, A) = \frac{\text{Gain}(S, A)}{\text{SplitInfo}(S, A)}$$

Notes

Sur-spécialisation

C4.5 continue à faire croître l'arbre tant que :

- Il reste un nœud contenant des exemples de plusieurs classes.
- Il reste au moins un test utilisable.
- Il reste un gain intéressant.

Notes

La sur-spécialisation

L'arbre obtenu alors :

- Est grand.
- Compliqué (difficilement lisible).
- Trop proche des données de l'ensemble d'apprentissage.

surspécialisation = overfitting

Notes

Sur-spécialisation

Exemple

- Deux classes : c_0 et c_1
- c_0 est la classe majoritaire, probabilité $p > 0.5$
- Les descriptions ne sont pas corrélées avec les classes.

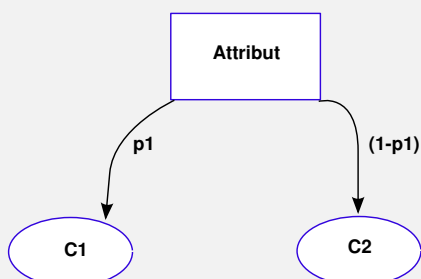
Premier classifieur : règle majoritaire : erreur $1 - p$

Notes

Sur-Spécialisation

Deuxième classifieur

Un arbre à un seul test (binaire) :



Notes

Sur-spécialisation

Erreur du deuxième classifieur

- Probabilité pour un exemple de descendre dans la première branche : p_1 .
- Probabilité pour cet exemple d'être mal classé : $(1 - p)$
- Probabilité pour un exemple de descendre dans la deuxième branche : $(1 - p_1)$
- Probabilité pour cet exemple d'être mal classé : p .
- Probabilité d'erreur du classifieur :

$$p_1(1 - p) + (1 - p_1)p$$

$$p_1(1 - p) + (1 - p_1)p \geq p_1(1 - p) + (1 - p_1)(1 - p)$$

$$p_1(1 - p) + (1 - p_1)p \geq (1 - p)$$

- L'inégalité est stricte dès que $p > 0.5$

Notes

Sur-spécialisation

- Remplacer l'arbre par un nœud donne un meilleur arbre !
- Des cas similaires se rencontrent souvent à partir d'une certaine profondeur dans l'arbre.
- Les exemples sont moins nombreux : les régularités sont plus probablement dûes au hasard.
- Créer de nouveaux nœuds : n'apporte rien, ou pire, sème la confusion !

Notes

Sur-spécialisation

Remèdes

- S'arrêter avant d'engendrer un nœud inutile (pre-élagage).
- Construire l'arbre, puis supprimer les nœuds "trop spécialisants" (post-élagage).

Notes

Pré-élagage

- Selon quel critère renoncera-t-on à développer un nœud ?
- Gain pas assez important :
 - On risque d'ignorer un test qui "prépare le terrain" pour un ou plusieurs tests plus discriminants.
 - Comment fixer le seuil des gains intéressants ?
- **Interêt** : diminue le temps de calcul de l'arbre ...

Notes

Principe

- Construire l'arbre complet.
- Pour chaque nœud interne, regarder s'il ne serait pas meilleur de le remplacer :
 - Par une feuille.
 - Par un de ses fils (son fils le plus fréquent).

```

physician fee freeze = n :
| adoption of the budget resolution = y : democrat (151.0)
| adoption of the budget resolution = u : democrat (1.0)
| adoption of the budget resolution = n :
| | education spending = n : democrat (6.0)
| | education spending = y : democrat (9.0)
| | education spending = u : republican (1.0)
physician fee freeze = y :
| synfuels corporation cutback = n : republican (97.0/3.0)
| synfuels corporation cutback = u : republican (4.0)
| synfuels corporation cutback = y :
| | duty free exports = y : democrat (2.0)
| | duty free exports = u : republican (1.0)
| | duty free exports = n :
| | .....
physician fee freeze = u :
| water project cost sharing = n : democrat (0.0)
| water project cost sharing = y : democrat (4.0)
| water project cost sharing = u :
| | mx missile = n : republican (0.0)
| | mx missile = y : democrat (3.0/1.0)
| | mx missile = u : republican (2.0)

```

```

physician fee freeze = n : democrat (168.0/2.6)
physician fee freeze = y : republican (123.0/13.9)
physician fee freeze = u :
| mx missile = n : democrat (3.0/1.1)
| mx missile = y : democrat (4.0/2.2)
| mx missile = u : republican (2.0/1.0)

```

- Deux nœuds ont été remplacés par des feuilles.
- Un autre nœud a été remplacé par un de ses fils.
- Le nombre de mal classés pour chaque feuille est une *estimation pessimiste*.

Post-élagage

Comment faire ?

- Si on connaissait exactement les taux d'erreur en chaque nœud :
 - On calculerait l'erreur en chaque nœud (i.e. la somme des erreurs de ses fils).
 - On comparerait à l'erreur obtenue si ce nœud n'était pas développé (i.e. restait une feuille).
 - Si cette deuxième erreur est plus faible : on remplace le nœud par la feuille.
 - (idem pour comparer un nœud avec son fils le plus fréquent).
- Peut-on estimer ces taux d'erreur ?

Notes

Estimer les erreurs

- Utilisation d'un ensemble de validation :
 - Il faut réserver beaucoup d'exemples (descente dans l'arbre)
 - L'ensemble d'apprentissage en est réduit d'autant.
 - L'arbre obtenu est moins bon.
- Utiliser l'ensemble d'apprentissage ... attention au biais !

Notes

Post-élagage

- Pour un nœud donné, soit N le nombre d'exemples arrivant en ce nœud.
- Soit, parmi ces N exemples, E le nombre d'exemples mal classés (dans la suite de l'arbre).
- Le taux d'erreur $\frac{E}{N}$ est manifestement **optimiste**.
- Trouver une valeur plus pessimiste de la probabilité d'erreur en ce nœud.

Notes

Estimation pessimiste de l'erreur

- Soit p la "vraie" probabilité d'erreur en ce nœud.
- Soit $\frac{E}{N}$ la probabilité optimiste constatée en ce nœud.
- Quelle est la valeur maximale de p qui fait qu'une erreur de $\frac{E}{N}$ ne soit pas rare ?
- Quelle est la valeur maximale de p telle que obtenir $\frac{E}{N}$ erreurs s'observe dans au moins 25% des cas ?

Notes

Estimation pessimiste

- Soit p la probabilité de mal classer un exemple.
- Probabilité de rencontrer E erreurs pour N exemples :

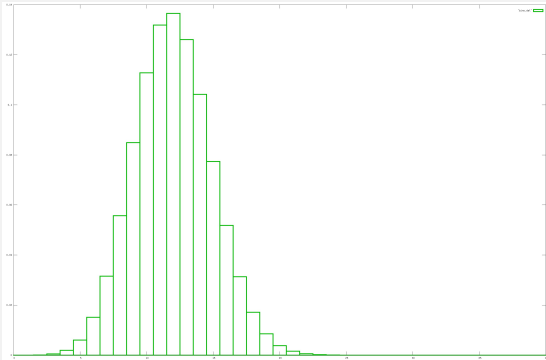
$$C_N^E p^E (1-p)^{N-E}$$

(loi binomiale).

- Calculer l'intervalle de confiance (binomiale \rightarrow loi normale \rightarrow tables).
- p est l'estimation pessimiste de l'erreur.

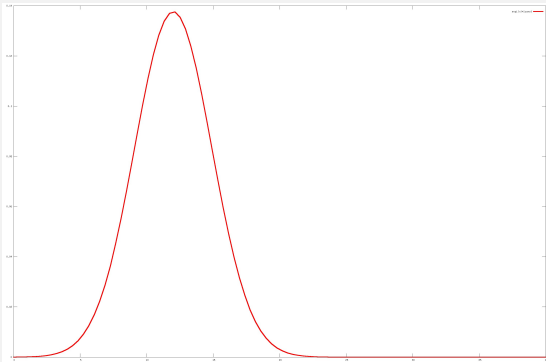
Notes

Loi binomiale ($n = 40, p = 0.3$)



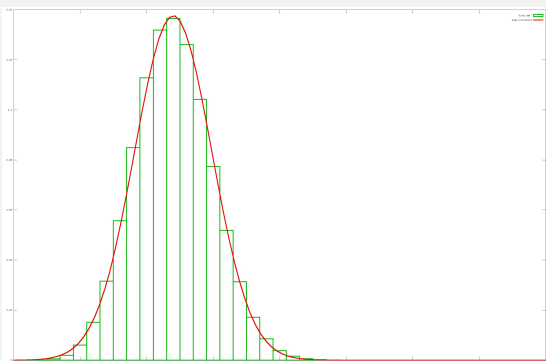
Notes

Loi normale approchant



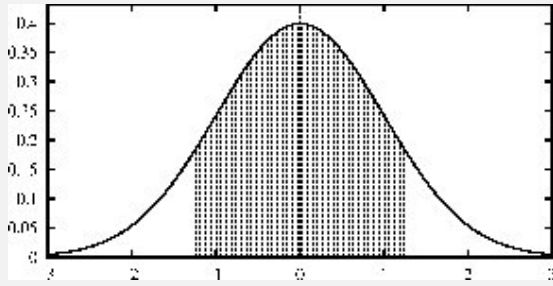
Notes

Vérification



Notes

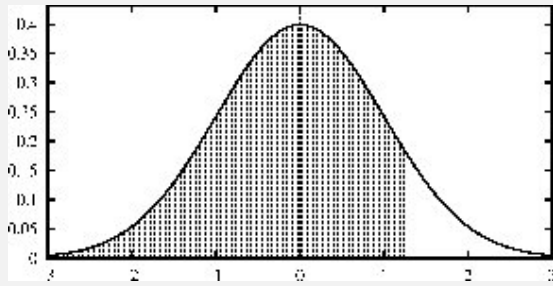
Intervalle de confiance



Dans 80% des cas, la valeur de la variable aléatoire est dans la zone hachurée.

Notes

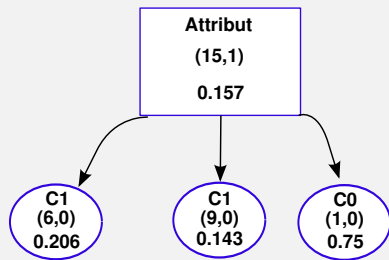
Intervalle de confiance



Dans 90% des cas, la valeur de la variable aléatoire est dans la zone hachurée.

Notes

Exemple



Notes

Exemple

Evaluation on training data (300 items) :
Before Pruning After Pruning

Size Errors Size Errors Estimate

25 8(2.7%) 7 13(4.3%) (6.9%) <<

Evaluation on test data (135 items) :
Before Pruning After Pruning

Size Errors Size Errors Estimate

25 7(5.2%) 7 4(3.0%) (6.9%) <<

Notes
