Introduction aux arbres de décision

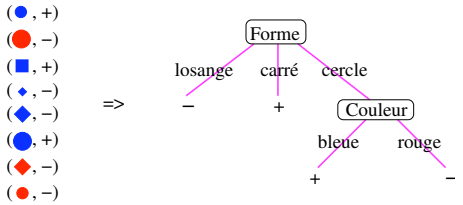
Liva Ralaivola

LIF, UMR 6166 CNRS
Université de Provence
liva.ralaivola@lif.univ-mrs.fr

14 janvier 2008

Aperçu

Exemple



- Lecture d'un arbre ?
- Construction de l'arbre ?
- Régularisation/sur-apprentissage ?

Plan

Plan

Contexte

Utilisation

- Classification supervisée (*pattern recognition*)
  - $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)\}$ ensemble d'apprentissage
  - $\mathbf{x}_i \in \mathcal{X},\ y_i \in \mathcal{Y}$
- Utilisation
  - apprentissage (plutôt) rapide
  - interprétabilité du modèle
  - possible bruit sur les données
- Exemples dans ce cours
  - $\mathcal{X}$ espace de vecteurs d'attributs discrets
  - classification binaire

Vocabulaire

Définitions

nœuds *chaque nœud correspond à une question sur un attribut et à un ensemble d'exemples*

branches *chaque branche part d'un nœud et correspond à une réponse possible à la question posée en ce nœud*

- *CART [Breiman et al., 1984] : 2 branches par nœud*
- *ID3 [Quinlan, 1986], C4.5 [Quinlan, 1993] : autant de branches que de valeurs possibles pour l'attribut étudié*

feuilles *nœuds d'où ne part aucune branche ; correspond à une classe*

Utilisation d'un arbre de décision

Arbre $\mathcal{T}$ et **x** instance à classifier

La classification d'une instance se fait de la racine de $\mathcal{T}$ vers les feuilles :

- $n \leftarrow$racine de l'arbre
- Tant que **x** n'atteint pas une feuille
  - poser la question associée à n sur **x** (par exemple : "le i-ème attribut de **x** est-il 1 ou 0 ?")
  - $n \leftarrow$nœud vers lequel oriente la réponse à la question précédente
- fin tant que
- renvoyer la classe associée à la feuille identifiée

Méthode TDIDT (1)

Apprentissage

TDIDT : *Top Down Induction of Decision Tree*

- Induction : arbre de décision est un modèle induit à partir d'exemples d'apprentissage (comme pour les réseaux de neurones)
- Top-Down : l'algorithme d'apprentissage est dit Top-Down car il part d'un modèle (vide) qui est ajusté pour correspondre aux données (notion inverse : Bottom-up)
- Partitionnement récursif de l'espace $\mathcal{X}$
- Pour un nœud donné, une question ne peut porter sur un attribut qui a déjà servi dans un chemin menant à ce nœud

Méthode TDIDT (2)

Algorithme (Description Haut-niveau)

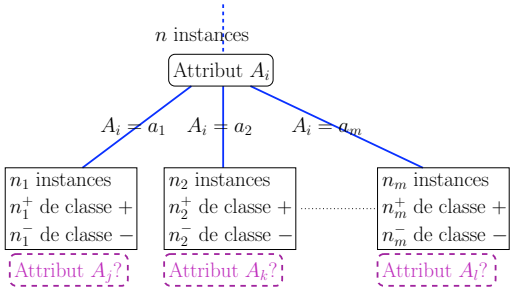
méthode *construit_arbre($\mathcal{S}$)*

- **si** *tous les exemples de $\mathcal{S}$ sont de la même classe ou bien il n'existe plus de question possible* **alors**
  - *créer une feuille de la classe majoritaire de ce nœud*
- **sinon**
  - *choisir la meilleure question pour créer un nœud : $\mathcal{S}$ est partitionné en $\mathcal{S}_1, \ldots, \mathcal{S}_m$ (e.g. m est le nombre de modalités que peut prendre l'attribut sur lequel porte la question)*
    - **pour** *i allant de* 1 *à m* **faire**
      construit_arbre($\mathcal{S}_i$)

Problématiques

Question

Comment choisir à chaque étape de la construction la meilleure question (i.e. le meilleur attribut) à poser ?

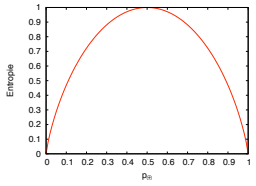


Entropie (1/3)

Définition ([Shannon, 1948])

Soit $C \in \mathcal{C}$ une v.a. discrète, $\mathcal{C} = \{c_1, \ldots, c_m\}$

- $p_i = P(C = c_i)$, $p_i \geq 0$ *et* $\sum_{i=1}^{m} p_i = 1$
- *Entropie de $p_1, \ldots, p_m$:* $\boxed{I(p_1, \ldots, p_m) = -\sum_{i=1}^{m} p_i \log_2 p_i}$



Cas binaire $c_1 = +$, $c_2 = -$

- $p_\oplus = P(C = +)$
- $p_\ominus = P(C = -) = 1 - p_\oplus$
- $I(p_\oplus, p_\ominus) =$
  $-p_\oplus \log_2 p_\oplus - (1 - p_\oplus) \log_2 (1 - p_\oplus)$

Entropie (2/3)

Interprétations

- Entropie élevée ⇔ désordre
- Entropie faible ⇔ ordre
- Nombre minimum de bits pour coder la classe d'un exemple tiré au hasard dans $\mathcal{S}$
- Fournit une mesure de l'impureté d'un nœud/d'une feuille pour les arbres de décision

Entropie (3/3)

Intérêt pour l'induction d'arbres de décision

- Nœud contenant 9 exemples $+$ et 5 exemples $-$:

$$\begin{aligned} I([9+,5-]) &= I(9/14, 5/14) \\ &= -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) \\ &= 0.940 \end{aligned}$$

- Nœud contenant 14 exemples $+$ et 0 exemple $-$

$$\begin{aligned} I([14+,0-]) &= -(14/14)\log_2(14/14) - (0/14)\log_2(0/14) \\ &= 0 \end{aligned}$$

degré d'impureté 0 (i.e. feuille)

Maximisation du gain d'information (1/2)

Choix d'un attribut A_i

Lors du développement de chaque nœud, choisir l'attribut A_i permettant le gain d'information le plus important avec

$$Gain(A_i) = I_0 - I(A_i)$$

où

- I_0 correspond à l'entropie de l'ensemble d'exemples correspondant au nœud étudié
- $I(A_i)$ correspond à l'entropie 'pondérée' du sous-arbre résultant du développement selon l'attribut A_i

Maximisation du gain d'information (2/2)

Entropie pondérée selon l'attribut A_i

- si le nœud étudié contient n exemples et que A_i permet d'obtenir m nœuds, $[n_1^+ +, n_1^- -], \ldots, [n_m^+ +, n_m^- -]$ alors l'entropie pondérée du sous arbre obtenu en développant le nœud selon A_i est

$$I(A_i) = \sum_{j=1}^{m} \frac{n_j}{n} I([n_j^+ +, n_j^- -])$$

avec $n_j = n_j^+ + n_j^-$

Construction d'un arbre de décision

Exercice

Taille	Forme	Couleur	Classe
petit	cercle	bleu	+
grand	cercle	rouge	-
grand	carré	bleu	+
petit	losange	bleu	-
grand	losange	bleu	-
grand	cercle	bleu	+
grand	losange	rouge	-
petit	cercle	rouge	-

Montrer que

- $gain(taille) = 0.003$
- $gain(forme) = 0.454$
- $gain(couleur) = 0.347$

Construire l'arbre de décision

Index de Gini G

Critère utilisable à la place de l'entropie : $G = 2p_\oplus(1 - p_\oplus)$

Plan

Sur-apprentissage (1/2)

Constats

- Si l'ensemble des exemples d'apprentissage est consistant, c'est-à-dire si on n'a pas un même exemple etiqueté de deux façons différentes, alors l'apprentissage par arbre de décision permet d'obtenir une représentation avec des feuilles pures uniquement, i.e. il est possible de ne faire aucune erreur sur l'ensemble d'apprentissage
- Par ailleurs, le critère usuel d'arrêt d'apprentissage par arbre de décision correspond à l'obtention de feuilles pures uniquement ou bien l'impossibilité de développer l'arbre

Sur-apprentissage (2/2)

Conséquences des constats précédents

- L'apprentissage par arbre de décision peut conduire au phénomène de sur-apprentissage
- Les arbres obtenus peuvent être très grands et les feuilles ne contenir que peu d'instances

Solution : élagage

- pré-élagage : un critère permet d'arrêter la construction de l'arbre avant l'obtention de l'arbre complet
- post-élagage : l'arbre complet est appris puis des branches de l'arbre sont coupées en fonction d'un critère donné

Pré-élagage

Critères d'arrêt du développement de l'arbre

- Nombre faible d'instances dans un nœud
- Gain d'information faible
- Test du χ^2 permettant de mesurer l'indépendance statistique de la population d'un nœud par rapport à une classe
- ...

Post-élagage

Critères guidant l'élagage de l'arbre

- Mesure sur un échantillon indépendant de l'erreur de classification : élaguer l'arbre tant que cette mesure ne croît pas
- Critère *ad hoc* type C4.5 de Quinlan

Prise en compte de données numériques

Problème

Tel que présenté, l'algorithme d'induction d'arbre de décision proposé ne permet pas de gérer des attributs numériques

Exercice

Proposer une méthode introduisant des seuils permettant de classifier des instances contenant des attributs numériques.

Bagging

Objectif

- réduire la variance de l'arbre appris par rapport à l'échantillon d'apprentissage
- pour améliorer la généralisation

Algorithme [Breiman 94]

$\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)\}$

1. Tirer de $\mathcal{S}$ avec replacement B échantillons $\mathcal{S}_1, \ldots, \mathcal{S}_B$ de taille ℓ
2. Apprendre B arbres de décision h_i sur $\mathcal{S}_1, \ldots, \mathcal{S}_B$
3. Prédire avec $h(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} h_i(\mathbf{x})$

Bornes d'erreur dépendant des données

Theorem (Bartlett and Mendelson, 2002)

Theorem 17 *Let P be a probability distribution on $\mathcal{X} \times \{-1, 1\}$, and let H be a set of binary-valued functions defined on $\mathcal{X}$. Let T be the class of decision trees of depth no more than d, with decision functions from H. For a training sample $(X_1, Y_1, \ldots, X_n, Y_n)$ drawn from P^n and a decision tree from T, let $\tilde{P}_n(l)$ denote the proportion of all training examples which reach leaf l and are correctly classified. Then with probability at least $1 - \delta$, every decision tree t from T with L leaves has $\Pr(y \neq t(x))$ no more than*

$$\hat{P}_n(y \neq t(x)) + \sum_l \min(\tilde{P}_n(l), cdG_n(H)) + \sqrt{\frac{c\ln(L/\delta)}{2n}}\,.$$

... avec

Quantité importante, $G_n(F)$, complexité Gaussienne (cf. également complexité de Rademacher) :

$$\hat{G}_n(F) = \mathbf{E}\left[\sup_{f \in F}\left|\frac{2}{n}\sum_{i=1}^{n} g_i f(X_i)\right| \middle| X_1, \ldots, X_n\right]$$

avec $G_n(F) = \mathbb{E}[\hat{G}_n(F)]$

Plan

Résumé

A retenir

- Interprétabilité du modèle par arbre de décision
- Méthode d'apprentissage TDIDT
  - entropie (Shannon)
  - élagage

Non couvert

- arbres de régression
- apprentissage incrémental
- forêt d'arbres
- ...

Weka

- Port de lentilles de contact (plusieurs classes)
- Reconnaissance de chiffres manuscrits (plusieurs classes + valeurs numériques)

WEKA

Exercice

Construire l'arbre associé aux données [Mitchell, 1997]

Exemple	Prévision	Température	Humidité	Vent	Tennis
1	soleil	élevée	haute	faible	non
2	soleil	élevée	haute	fort	non
3	nuage	élevée	haute	faible	oui
4	pluie	moyenne	haute	faible	oui
5	pluie	basse	normale	faible	oui
6	pluie	basse	normale	fort	non
7	nuage	basse	normale	fort	oui
8	soleil	moyenne	haute	faible	non
9	soleil	basse	normale	faible	oui
10	pluie	moyenne	normale	faible	oui
11	soleil	moyenne	normale	fort	oui
12	nuage	moyenne	haute	fort	oui
13	nuage	élevée	normale	faible	oui
14	pluie	moyenne	haute	fort	non

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984).
Classification and Regression Trees.
Wadsworth and Brooks, Monterey, CA.

Mitchell, T. (1997).
Machine Learning.
McGraw Hill.

Quinlan, J. R. (1986).
Induction of decision trees.
Machine Learning, 1 :81–106.

Quinlan, J. R. (1993).
C4.5 : Programs for Machine Learning.
Morgan Kaufmann.

Shannon, C. E. (1948).
A Mathematical Theory of Communication.
The Bell System Techincal Journal, 27 :379–423,623–656.