

Data Mining

Classification

Nicolas Pasquier

<http://www.i3s.unice.fr/~pasquier>

1

Classification supervisée

- Tâche de prédiction de valeurs de variables
- Deux types de classification
 - Variables catégorielles (nominales, symboliques) : classement
 - Variables numériques : régression
 - Méthodes de calcul différentes
- Par abus de langage (anglicisme) le classement est souvent appelé « classification »
- Apprentissage : un modèle est appris des données

2

Classement

- Objectifs :
 - Construire un modèle décrivant chaque classe d'objets
 - Ex : patients sains et patients atteints d'un cancer
 - Appliquer ce modèle pour de nouveaux objets
 - Ex : prédire le risque de développer un cancer pour de nouveaux patients
- Différent du clustering (classification non-supervisée)
 - Clustering : classes inconnues à l'avance
 - Classification : classes connues à l'avance
- Le modèle appris est appelé classifieur

3

Classement

- Le modèle prédit la valeur d'une variable catégorielle (nominale)
- Attribut cible : attribut de classe
- Tous types d'attributs en entrée (prédicatifs)
- Exemple : demande de crédit
 - Classe : *risquée, non-risquée*
 - Attributs prédicatifs : revenus, statut marital, endettement, etc.

Modèle
(règles de
classification)

si *revenus* < 1200 et *maridé* = non alors *risqué*
si *revenus* < 1200 et *maridé* = oui alors *non-risqué*
si *dette* = 0.0 et *revenus* > 2000 alors *non-risqué*
...

4

Régression

- Le modèle prédit la valeur d'une variable numérique continue (*prediction*)
- Attribut cible : attribut de score
- Attributs prédicatifs : éléments de la fonction
 - Attributs numériques uniquement
- Exemple : score de risque de *churning* (départ) en téléphonie mobile
 - Score : 0.0% à 100% probabilité de *churn*
 - Attributs prédicatifs : facture, ancienneté, minutes, revenus, etc.

Modèle

$score = facture * 1,567 + minutes * 0,854 - ancienneté * 0,498 \dots$

5

Classement

- Les instances (lignes) du jeu de données sont parfois appelées « exemples »
- Méthode
 1. Construire un modèle de classement des instances en se basant sur un ensemble appelé ensemble d'apprentissage (EA) (*training set*)
 2. Tester sa précision sur un ensemble de test (ET) (*test set*)
 3. Utiliser le modèle pour classifier de nouvelles instances
- Le modèle résultant peut prendre différentes formes

6

Classement : exemple

- Jeu de données D
- Clients ayant ou non acheté un PC (classe)

	Attributs prédictifs				Attributs cible
	Age	Revenus	Étudiant	Crédit	Achète_PC
	<=30	élevés	non	moyen	non
	<=30	élevés	non	excellent	non
	31...40	élevés	non	moyen	oui
	>40	moyens	non	moyen	oui
	>40	faibles	oui	moyen	oui
	>40	faibles	oui	excellent	non
	31...40	faibles	oui	excellent	oui
	<=30	moyens	non	moyen	non
	<=30	faibles	oui	moyen	oui
	>40	moyens	oui	moyen	oui
	<=30	moyens	oui	excellent	oui
	31...40	moyens	non	excellent	oui
	31...40	élevés	oui	moyen	oui
	>40	moyens	non	excellent	non

7

Forme des classifieurs

- Arbres de décision
 - Les feuilles sont les classes et les chemins les critères de classification
- Règles de classification
 - Représentation simplifiée de l'arbre de décision sous forme de règles d'implication
- Classifieurs bayésiens
 - Modèles probabilistes d'appartenance des objets à chaque classe
- Réseaux de neurones
 - Tentent de reproduire le raisonnement humain

8

Classifieurs bayésiens

- Classifieur statistique qui prédit la probabilité pour un exemple d'appartenir à une classe
- Basé sur la Théorie de Bayes (probabilités conditionnelles)
- Approche la plus utilisée pour certains types d'apprentissage, car aussi performante que arbres de décision et réseaux de neurones
- Deux formes :
 - Classement « naïf »
 - Classement par réseaux bayésiens

9

Classifieurs bayésiens

- Modèles probabiliste d'appartenance aux classes
- Probabilité a posteriori
 - $P(C | X)$ = probabilité que l'objet $X = \{x_1, \dots, x_n\}$ appartienne à la classe C
 - Classement : assigner X à la classe C telle que $P(C | X)$ soit maximal
 - Exemple :
 - $X = \{\text{âge} \leq 30, \text{étudiant} = \text{non}, \text{revenus} = \text{élevé}, \text{crédit} = \text{moyen}\}$
 - Déterminer la probabilité maximale :
 - $P(\text{achète_PC} = \text{vrai} | X)$?
 - $P(\text{achète_PC} = \text{faux} | X)$?

10

Classifieurs bayésiens

- Théorème de Bayes

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

- Maximiser $P(C | X)$ revient à maximiser $P(X | C) \cdot P(C)$
- Probabilité a priori
 - $P(C)$ = probabilité que X appartienne à la classe C sans tenir compte des valeurs de X (calculée par comptage sur l'EA)
- Problème : calcul de $P(X | C)$ infaisable (trop coûteux)

11

Classifieurs bayésiens naïfs

- Assomption naïve : indépendance des attributs
 - $P(X | C) = P(x_1, \dots, x_n | C) = P(x_1 | C) \cdot \dots \cdot P(x_n | C)$
 - On calcule toutes les probabilités $P(x_i | C_i)$ pour les exemples de l'EA
- Algorithme
 - Calcul des $P(C_i)$ et $P(x_i | C_i)$ à partir des fréquences de ces éléments dans l'EA pour chaque classe C_i
 - Ensuite un nouvel exemple Y est classifié dans la classe maximisant $P(Y | C) \cdot P(C)$
- Nombre de calculs à effectuer
 - Nombre de valeurs d'attributs prédictifs * Nombre de valeurs d'attribut-cible

12

Classifieurs bayésiens : exemple

- $P(C_i)$
 - $P(\text{achète_PC} = \text{oui}) = 9/14 = 0.643$
 - $P(\text{achète_PC} = \text{non}) = 5/14 = 0.357$
- Calculer $P(X|C_i)$ pour chaque classe
 - $P(\text{age} = \leq 30 | \text{achète_PC} = \text{oui}) = 2/9 = 0.222$
 - $P(\text{age} = \leq 30 | \text{achète_PC} = \text{non}) = 3/5 = 0.6$
 - $P(\text{revenus} = \text{moyens} | \text{achète_PC} = \text{oui}) = 4/9 = 0.444$
 - $P(\text{revenus} = \text{moyens} | \text{achète_PC} = \text{non}) = 2/5 = 0.4$
 - $P(\text{étudiant} = \text{oui} | \text{achète_PC} = \text{oui}) = 6/9 = 0.667$
 - $P(\text{étudiant} = \text{oui} | \text{achète_PC} = \text{non}) = 1/5 = 0.2$
 - $P(\text{crédit} = \text{moyen} | \text{achète_PC} = \text{oui}) = 6/9 = 0.667$
 - $P(\text{crédit} = \text{moyen} | \text{achète_PC} = \text{non}) = 2/5 = 0.4$
 - ...

13

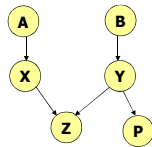
Classifieurs bayésiens : exemple

- $X = (\text{age} \leq 30, \text{revenus} = \text{moyens}, \text{étudiant} = \text{oui}, \text{crédit} = \text{moyen})$
- $P(X | C_i)$:
 - $P(X | \text{achète_PC} = \text{oui}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$
 - $P(X | \text{achète_PC} = \text{non}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$
- $P(X | C_i) \cdot P(C_i)$:
 - $P(X | \text{achète_PC} = \text{oui}) \cdot P(\text{achète_PC} = \text{oui}) = 0.028$
 - $P(X | \text{achète_PC} = \text{non}) \cdot P(\text{achète_PC} = \text{non}) = 0.007$
- X est placé dans la classe ("achète_PC = oui")

14

Réseaux bayésiens

- Modèle graphique des relations causales entre les valeurs des variables
 - Représente les dépendances entre variables
 - Fournit une spécification de la distribution des probabilités jointes
- Graphe orienté acyclique
 - Nœud = variable
 - Arc = dépendance entre variables
- Exemple
 - A parent de X, B de Y, Y de P
 - X et Y sont les parents de Z
 - Pas de dépendance entre Z et P



15

Réseaux bayésiens : exemple

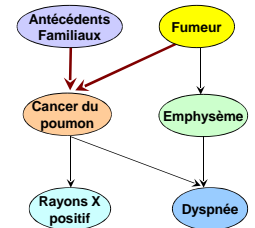
Table des probabilités conditionnelles pour « Cancer du poumon »

	(AF,F)	(AF,-F)	(¬AF,F)	(¬AF,-F)
CP	0.8	0.5	0.7	0.1
-CP	0.2	0.5	0.3	0.9

Probabilité conditionnelle pour chaque combinaison des parents

Dérivation de la probabilité pour une combinaison particulière de valeurs de X :

$$EX : P(\text{Dyspnée} = \text{vrai}) = P(\text{Dyspnée} = \text{vrai} | \text{parents}(\text{Dyspnée})) = P(\text{Dyspnée} = \text{vrai} | \text{Cancer du poumon}, \text{Emphysème})$$



$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{Parents}(Y_i))$$

16

Approches bayésiennes : intérêt

- Basées sur le calcul des probabilités a posteriori
- Permettent de calculer une probabilité a posteriori pour une hypothèse candidate en se basant sur la probabilité a priori et les données observées
- Donnent de bons résultats, même lorsque l'assomption d'indépendance est violée
 - Le classement ne requiert pas le calcul exact des probabilités tant que la probabilité maximale est assignée à la classe correcte
- Mais la présence d'attributs redondants peut poser des problèmes

17

Arbres de décision

- Méthode d'apprentissage inductif largement utilisée pour variables à valeurs discrètes
- Structure arborescente de type organigramme
 - Un nœud représente un test sur un attribut
 - Une branche correspond à un résultat de test
 - Les feuilles représentent les classes ou les distributions de classes
- Utilisation : classement d'une nouvelle instance
 - Nouvelle instance à classer : combinaison de valeurs des attributs prédictifs
 - Par comparaison de la valeur de ses attributs avec les nœuds de l'arbre
 - Chemin de la racine vers une feuille

18

Arbres de décision

- Algorithme basique ID3 [Quinlan1986]
 - Arbre construit de la racine vers les feuilles selon la stratégie diviser-pour-régner
 - Au début, tous les exemples sont à la racine
 - Les variables (attributs) sont catégorielles, les variables continues sont discrétisées
 - Les exemples sont ensuite répartis sur des branches pour chaque valeur de l'attribut choisi comme test
 - Le processus est répété sur l'EA correspondant à chaque nœud descendant

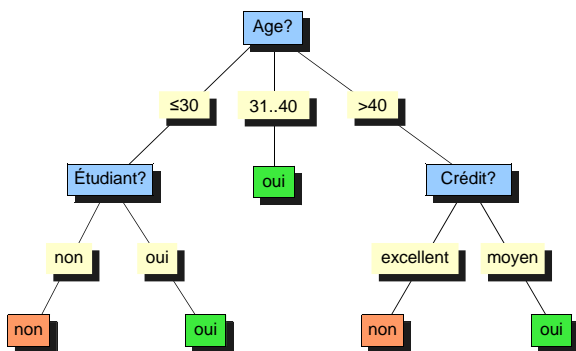
19

Arbres de décision

- Algorithme basique
 - Les attributs de test sont choisis selon un critère heuristique (ex : information gain) ou un critère statistique (ex : gini index)
- Le processus de partitionnement (split) s'arrête lorsque
 - Tous les exemples d'un nœud appartiennent à la même classe
 - Il ne reste plus d'attribut pour base de partitionnement (le scrutin majoritaire est utilisé pour classifier le nœud)
 - Il ne reste aucun exemple à classifier

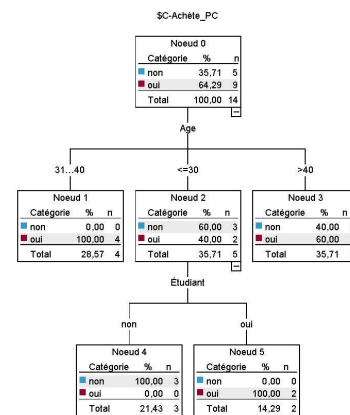
20

Arbres de décision : exemple



21

Arbres de décision : exemple



22

Règles de classification

- L'arbre peut être représenté sous forme de règles de classification
- Une règle pour chaque branche allant de la racine à une feuille
- Chaque terme attribut-valeur constitue un opérande de la conjonction en partie gauche
- Chaque feuille correspond à une classe à prédire

si age ≤ 30 et étudiant = non	alors achete_PC = faux
si age ≤ 30 et étudiant = oui	alors achete_PC = vrai
si age = 31..40	alors achete_PC = vrai
si age > 40 et crédit = excellent	alors achete_PC = faux
si age > 40 et crédit = moyen	alors achete_PC = vrai

23

Élagage

- Pré-élagage
 - Limiter la profondeur de l'arbre en stoppant son développement vertical
- Solution : appliquer des règles qui limitent la profondeur des branches
- Par exemple
 - Fixer un seuil limite du nombre de nœuds au dessus duquel un chemin ne peut plus être développé
 - Fixer un seuil limite du nombre d'enregistrements en dessous duquel un nœud ne peut plus être éclaté

24

Élagage

- Post-élagage
 - Développer l'arbre à son maximum puis, élaguer des branches jusqu'à leur taille minimum pour ne pas compromettre leur valeur
- Solution : utiliser une heuristique ou l'intervention de l'utilisateur
- Par exemple
 - Utiliser un ensemble de données différent de l'EA pour tester si un sous-arbre améliore suffisamment l'exactitude entière (estimer le taux d'erreur)

25

Arbres de décision : avantages

- Peu coûteux à construire
 - Font peu de parcours des données
 - Supportent de nombreuses variables prédictives
- Facile à interpréter
- Efficaces dans le cas d'une majorité de variables nominales
- Ils ont une valeur prédictive comparable aux autres méthodes dans la plupart des applications

26

Arbres de décision : inconvénient

- Ils utilisent un critère naïf pour le choix de l'attribut de partitionnement :
 - Ce critère ne tient pas compte des incidences produites sur les partitionnements ultérieurs
 - Le choix est fait « sur le moment » et n'est pas remis en question
 - Le processus est
 - Séquentiel donc un partitionnement dépend toujours du précédent
 - Univarié (il ne s'intéresse qu'à une variable à chaque nœud)
 - donc limitation du nombre de règles explorées
 - et détection difficile des relations entre attributs

27

Sélection des attributs de test

- Le meilleur attribut
- Intuitivement : celui qui partitionne le mieux les instances en classes, celui qui maximise la distance entre les groupes obtenus après partitionnement
- Celui qui minimise l'information (le nombre de tests) nécessaire pour classifier les exemples selon la partition résultante et qui reflète le désordre minimum dans cette partition ce qui garanti que l'arbre résultant sera simple
- Plus formellement : déterminer une mesure de séparabilité

28

Sélection des attributs de test

- *Attribute selection* ou *Feature selection*
- Quantification du poids des attributs dans la distinction des classes
- Gain d'information / Gain Ratio
 - Critères heuristiques
 - Mesures de réduction de l'entropie
 - Algorithmes ID3, C4.5, C5.0
- Index Gini
 - Critère statistique
 - Mesure l'impureté d'un nœud
 - Algorithmes C&RT, SLIT, SPRINT

29

Mesure d'Entropie

- Étant donnée une distribution de probabilités, l'information requise pour prédire un événement est l'entropie de la distribution
- Soit S l'ensemble d'exemples
- Supposons que l'attribut à prédire prenne M valeurs distinctes définissant M classes C_1, \dots, C_M
- L'entropie $E(S)$ est définie par

$$E(S) = - \sum_{i=1}^{i=M} p_i \cdot \log_2(p_i)$$

où p_i désigne la proportion d'exemples de S appartenant à C_i

30

Gain d'information

- Sélectionner l'attribut avec le plus fort gain d'information
 - Soit p_i la probabilité qu'une instance arbitraire de D appartienne à la classe C_i , estimée par $|C_{iD}| / |D|$
 - Information attendue** (entropie) nécessaire pour classer une instance de D :
$$Info(D) = -\sum_{i=1}^m p_i \log_2(p_i)$$
 - Information requise** (après avoir utilisé A pour diviser D en v partitions) pour classer D :
$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times I(D_j)$$
- Information gagnée** en testant l'attribut A :

$$Gain(A) = Info(D) - Info_A(D)$$

31

Gain d'information : exemple

- Classes : achète_PC = oui / achète_PC = non
- Information attendue

$$Info(D) = I(9,5) = -\frac{9}{14} \log_2\left(\frac{9}{14}\right) - \frac{5}{14} \log_2\left(\frac{5}{14}\right) = 0.940$$

- Information requise pour l'attribut Age

$$Info_{age}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

ou $\frac{5}{14} I(2,3)$ signifie que
« age ≤ 30 » concerne

5 des 14 exemples avec

2 achète_PC = oui et

3 achète_PC = non

age	oui		non	
	p_i	n_i	p_i	n_i
≤ 30	2	3	0	0,97
31...40	4	0	0	
> 40	3	2	0,97	

32

Gain d'information : exemple

- Gain d'information pour Age
 $Gain(age) = Info(D) - Info_{age}(D) = 0.246$
- Calcul pour les autres attributs :
 $Gain(revenus) = 0.029$
 $Gain(étudiant) = 0.151$
 $Gain(crédit) = 0.048$
- Age est donc l'attribut partitionnant le plus efficacement les instances
- Le premier attribut testé est Age

33

Gain d'information : inconvénient

- Favorise les éclatements en un grand nombre de partitions, chacune étant pure
- Exemple :
 - Introduire un attribut *date* dans les données
 - Valeur distincte pour chaque instance
 - $Gain(date)$ est maximum puisque date suffit à prédire la classe
- On introduit le Gain Ratio (C4.5, C5.0)

34

Gain Ratio

- Introduit une information de partitionnement

$$SplitInfo_A(D) = -\sum_{j=1}^v \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

- Ajuste le Gain d'information avec l'entropie du partitionnement
- Pénalise un éclatement dans un grand nombre de petites partitions

35

Gain Ratio : exemple

- Attribut *Revenus*
- Split Info :

$$SplitInfo_A(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 0.926$$

- Gain Ratio :

$$GainRatio(revenus) = 0.029 / 0.926 = 0.031$$

- On choisit l'attribut dont le Gain Ratio est maximal

36

Index Gini

- Si un ensemble d'exemples D contient n classes, l'index Gini de D est défini par :

$$Gini(D) = 1 - \sum_{j=1}^n p_j^2$$

où p_j est la fréquence relative de la classe j dans S

- Si l'ensemble D est partitionné selon l'attribut A en deux partitions D_1 et D_2 l'index Gini est défini par :

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

- On choisit l'attribut dont l'index Gini est minimal

37

Index Gini : exemple

- Index Gini de D (9 instance *achète_PC = oui* et 5 *non*)

$$Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- Partition selon l'attribut *Revenus* en {faibles, moyens} (10 instances) et {élevés} (4 instances)

$$\begin{aligned} Gini_{\text{revenus} \in \{\text{faibles, moyens}\}}(D) &= \left(\frac{10}{14}\right) Gini(D_1) + \left(\frac{4}{14}\right) Gini(D_2) \\ &= \frac{10}{14} \left(1 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2\right) + \frac{4}{14} \left(1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2\right) \\ &= 0.450 \\ &= Gini_{\text{revenus} \in \{\text{élevés}\}}(D) \end{aligned}$$

- Mais $Gini_{\text{revenus} \in \{\text{moyens, élevés}\}}(D) = 0.30$ ce qui est mieux

38

Sélection d'attributs

- Les 3 renvoient de bons résultats mais
- Information Gain
 - Biaisé vers les attributs à nombreuses valeurs
- Gain Ratio
 - Tend à favoriser les divisions déséquilibrées dans lesquelles une partition est beaucoup plus petites que les autres
- Gini Index
 - Tend à favoriser les divisions en de nombreuses partitions de tailles égales ayant une grande pureté
 - Rencontre des difficultés si le nombre de classes est grand
- Aucune n'est significativement supérieure à l'autre

39

Sélection d'attributs

- Autres méthodes
- Basées sur le test d'indépendance du χ^2
 - CHAID, G-statistics
- Basées sur des conditions multi-attributs
 - CART
- Minimiser l'information nécessaire pour coder l'arbre
 - Minimal Description Length
- Toutes produisent d'assez bons résultats
- Pas de méthode universellement meilleure que les autres

40

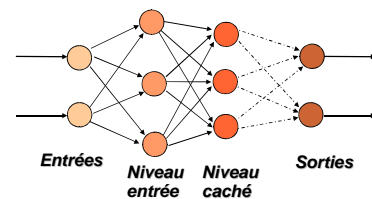
Réseaux de neurones

- Définition
 - Ensemble d'unités d'entrée/sortie (neurones) connectées avec un poids associé à chaque connexion (dirigée)
- Phase d'apprentissage
 - On applique le réseaux au jeu de données exemples
 - Le réseau « apprend » en ajustant les poids des connexions et le biais
 - Rétro-propagation (*back-propagation*) : ajustement des poids en remontant dans les connexions du réseau
- Arrêt
 - Lorsque les poids et biais permettent de classifier « correctement » les exemples

41

Réseaux de neurones

- Les entrées correspondent aux attributs des exemples
- Elles sont placées sur une couche (input layer) et alimentées simultanément

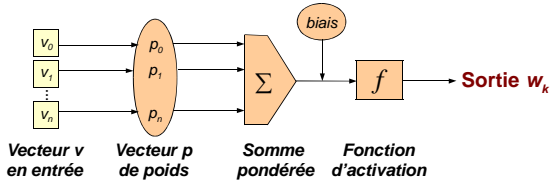


- Les sorties du niveau entrée alimentent un niveau caché

42

Réseaux de neurones

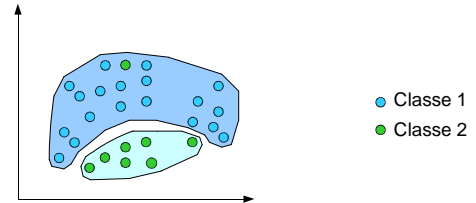
- Entrées : sorties des neurones du niveaux précédent
- Entrées sont multipliées par le poids correspondant, la somme est ajoutée à la valeur biais associée au neurone
- Fonction non-linéaire d'activation f : valeur transformée dans l'intervalle $[0,1]$



43

Réseaux de neurones

- Avantages :
- Peuvent identifier des modèles plus complexes
- Plus précis dans certains cas



44

Réseaux de neurones

- Inconvénients :
- Peuvent outrepasser les données : identifier des modèles dans des données non significatives
- Ne traitent que des valeurs numériques
- Difficulté à fixer les paramètres
- Interprétation difficile du modèle généré (boite noire)

45

Évaluation du modèle

- Évaluation : calcul du taux d'erreur (proportion d'exemples mal classés) sur l'ensemble de test
- La valeur de l'attribut-cible est connue pour chaque exemple de l'ET
- Cette valeur est comparée à la classe prédite par le modèle
- L'ET doit être indépendant de l'EA sinon il y a risque de sur-évaluation (*over-fitting*)
 - Taux d'erreur biaisé car l'algo est devenu spécialiste de l'ensemble d'entraînement
 - N'indique rien des capacités du modèle pour de nouvelles données

46

Ensemble de test

- Diviser l'ensemble des données en ensemble d'apprentissage (2/3) et ensemble de test (1/3)
- Utiliser la validation croisée (*cross-validation*)
- Utiliser toutes les données dans l'ensemble d'apprentissage
mais appliquer un test statistique (ex : χ^2) pour estimer si le développement ou l'élagage d'un nœud peut améliorer l'exactitude (C4.5)

47

Validation croisée

- L'ensemble des exemples est divisé en K partitions d'effectifs égaux
- Apprentissage et test en K étapes
- A chaque étape :
 - Utiliser $K-1$ partitions comme EA et 1 comme ET (*K-fold cross-validation*)
 - Calculer le taux d'erreur e_k
- Taux d'erreur estimé : moyenne des e_k
- On prend souvent $K=10$

48

Erreurs de classement

- Différentes erreurs
 - Prédire un exemple dans C alors qu'il appartient à non_C
 - Prédire un exemple dans non_C alors qu'il appartient à C
- Calcul du taux d'erreur
 - Vrai Positif (*True Positive*) : exemple prédit dans C et appartenant à C
 - Vrai Négatif (*True Negative*) : exemple prédit dans non_C et appartenant à non_C
 - Faux Positif (*False Positive*) : exemple prédit dans C et appartenant à non_C
 - Faux Négatif (*False Negative*) : exemple prédit dans non_C et appartenant à C

49

Matrice de confusion

- « Matrice de coïncidences »
- Indique le nombre de classements corrects et incorrects pour chaque classe (i.e. chaque valeur de l'attribut-cible)

		Classe prédite	
		C	\bar{C}
Classe réelle	C	VP	FN
	\bar{C}	FP	VN

- Taux d'erreur : $FP + FN / VP + FP + VN + FN$
- Recall : $VP / VP + FN$
- Précision : $VP / VP + FP$

50

Pondération des erreurs

- Différents coûts des erreurs
- Exemple
 - Accord de crédit : FP plus coûteux que FN
 - Diagnostic médical : FN plus coûteux que FP
- Pondérations différentes des erreurs FP et FN

		Classe prédite	
		C	\bar{C}
Classe réelle	C		coût1
	\bar{C}	coût2	

51

Matrice de confusion

- Valeurs à prédire pour l'attribut cible : C_1, C_2, C_3

		Classe prédite			S	Nombre de succès
		C_1	C_2	C_3		
Classe réelle	C_1	S_{11}	E_{12}	E_{13}	E	Nombre d'échecs
	C_2	E_{21}	S_{22}	E_{23}		
	C_3	E_{31}	E_{32}	S_{33}		

- Taux d'erreurs : $\sum(E_{ij}) / \sum(S_{ij}) + \sum(E_{ij})$

52

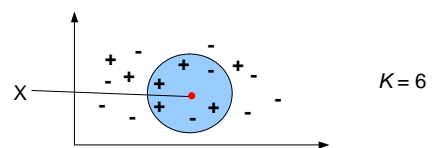
Autres méthodes

- Classement basées sur les instances
 - On cherche quelles sont les instances de l'EA qui ressemblent le plus à une nouvelle instance pour la classifier
 - Les exemples de l'EA sont stockés et le processus d'évaluation est retardé au moment où un nouvel exemple doit être classifié
 - Instance-based learning / Lazy evaluation methods*
- Méthodes :
 - K-Nearest Neighbors (KNN)
 - Raisonnement à base de cas

53

K-Nearest Neighbors

- Apprentissage par analogie
- Chaque exemple est représenté par un point dans l'espace de dimensions n
- K-Nearest Neighbor: pour un nouvel exemple X , l'algorithme cherche les K exemples de l'EA les plus proches (distance euclidienne)
- X est classifié dans la classe la plus représentée parmi les K voisins



54

Résumé

- Méthodes simples donnent le plus souvent de bons résultats
 - Résistantes aux bruit et erreurs
- Méthodes plus évoluées peuvent améliorer les résultats si elles sont bien paramétrées
- Aucune méthode n'est universellement meilleure que les autres