

ARBRES DE DÉCISION

Les arbres de décision

2

□ Technique :

- intuitive ++

- populaire +

 - Utilisation dans meta-classifieurs

□ Avantages

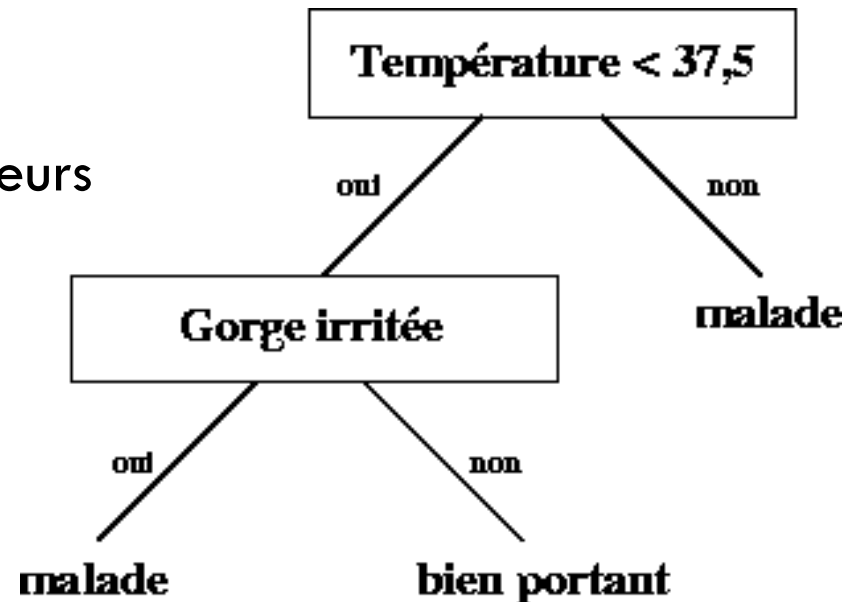
- Modèle explicite

- Traitement des données

 - hétérogènes

 - manquantes








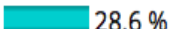

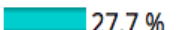
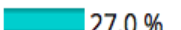
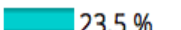
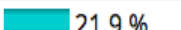
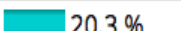
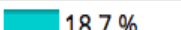

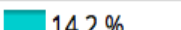
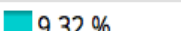
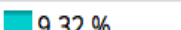
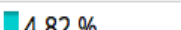
□ *Classification hiérarchique descendante supervisée*



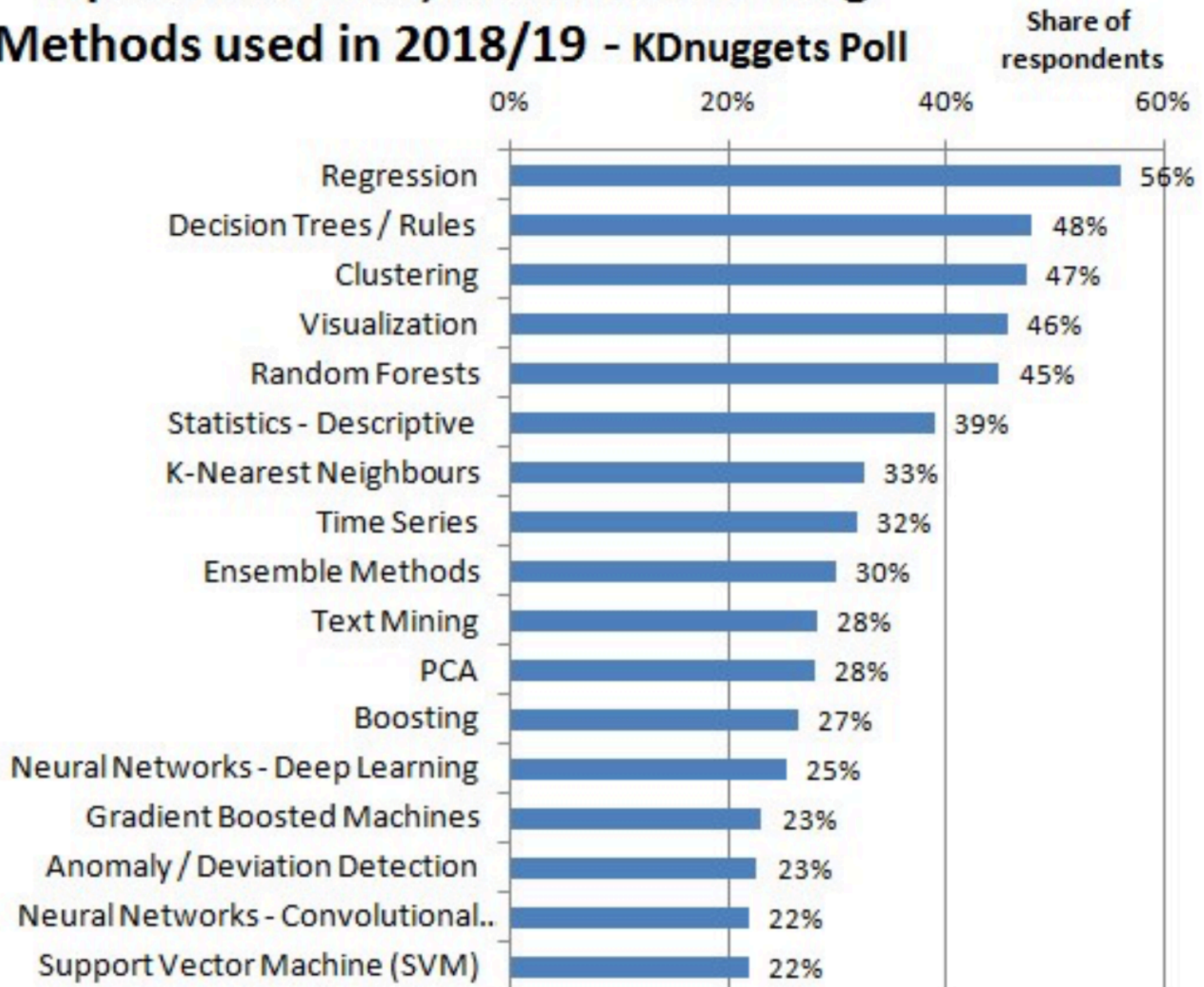
Popularité

3

Which methods/algorithms did you use for data analysis in 2011? [311 voters]

Decision Trees/Rules (186)	 59.8 %
Regression (180)	 57.9 %
Clustering (163)	 52.4 %
Statistics (descriptive) (149)	 47.9 %
Visualization (119)	 38.3 %
Time series/Sequence analysis (92)	 29.6 %
Support Vector (SVM) (89)	 28.6 %
Association rules (89)	 28.6 %
Ensemble methods (88)	 28.3 %
Text Mining (86)	 27.7 %
Neural Nets (84)	 27.0 %
Boosting (73)	 23.5 %
Bayesian (68)	 21.9 %
Bagging (63)	 20.3 %
Factor Analysis (58)	 18.7 %
Anomaly/Deviation detection (51)	 16.4 %
Social Network Analysis (44)	 14.2 %
Survival Analysis (29)	 9.32 %
Genetic algorithms (29)	 9.32 %
Uplift modeling (15)	 4.82 %

Top Data Science, Machine Learning Methods used in 2018/19 - KDnuggets Poll



Popularité

7

Algorithm usage bias by Employment

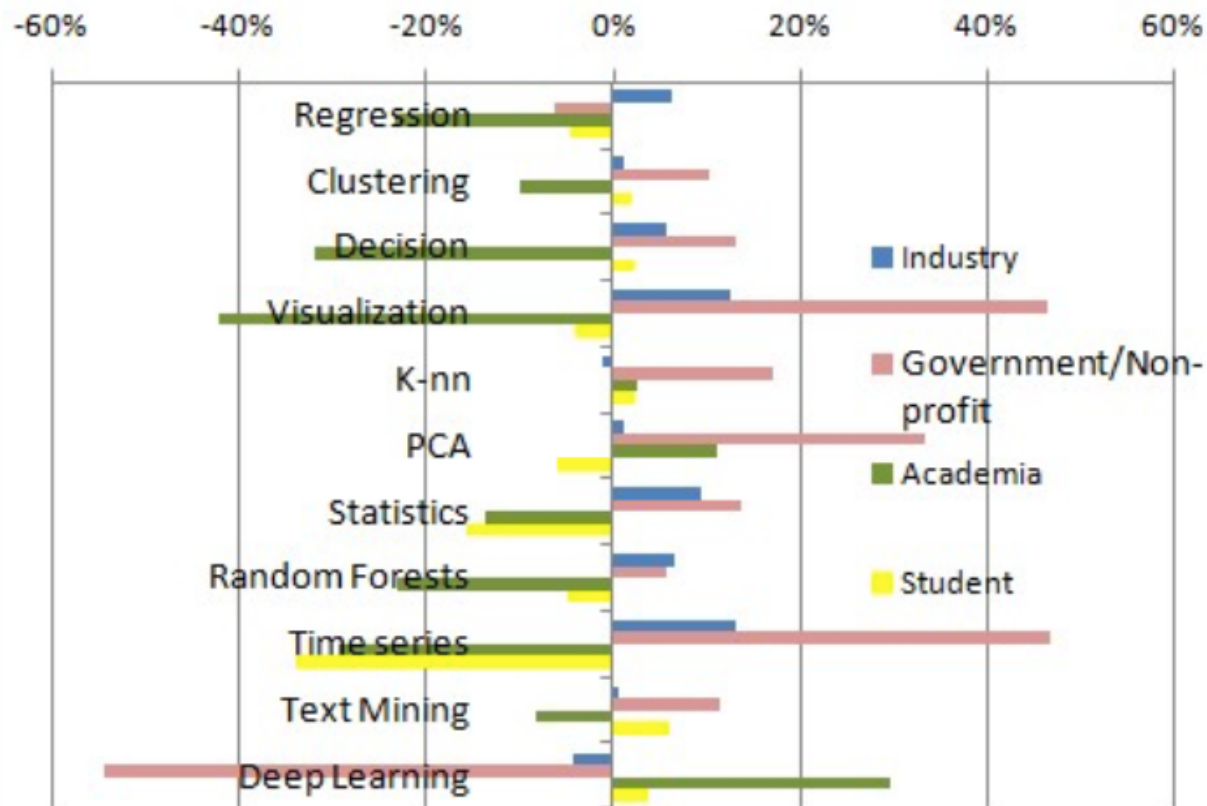


Fig. 2: Algorithm usage bias by Employment.

Principe de l'arbre de classification

8

- Répartir des individus d'une population **X** dans **Y** constitué de p classes prédéfinies
 - ▣ Enchaînement de règles construites automatiquement à partir de **X**
 - ▣ Subdiviser **X** en sous-ensembles de plus en plus homogènes par rapport à **Y**, ensemble des classes à prédire
- Construction :
 - ▣ Choisir le descripteur qui sépare **le mieux** les individus de **X** dans chaque classe
 - ▣ Processus top-down et récursif

Un peu plus formellement...

9

- Soit un ensemble X de n données x_i
 - ▣ Définies par un ensemble de descripteurs d_i
 - ▣ Étiquetées y_k , avec $y_k \in Y$
- Un arbre de décision construit sur X est constitué de:
 - ▣ Nœuds : test sur un d_i
 - Racine : premier nœud de l'arbre
 - ▣ Branches : différentes valeurs possibles du test
 - ▣ Feuilles : associées à un y_k

Remarques

10

- Par construction : feuilles constituées en majorité d'individus d'une seule classe
- Couverture d'une donnée par un nœud
 - ▣ Sa description amène la donnée de la racine jusqu'au nœud
- Affectation de chaque donnée à une feuille
 - Forte probabilité d'appartenir à la classe
- Modèle de classification = ensemble des règles de chaque feuille

Construction d'un arbre de décision

11

□ Algorithme générique

Soit : un corpus X

Initialisation

Arbre vide, racine=nœud courant

Répéter

Nœud courant terminal?

Si nœud terminal

Alors affecter une classe

Sinon

Sélectionner un test

Créer TOUS les nœuds fils

Fin

Passer au prochain nœud non exploré (if exists)

Tant que nœud non exploré existe

Retour : arbre de décision

Construction d'un arbre de décision

12

- Problème complexe dont la solution optimale ne peut être trouvée efficacement (NP-complet)
 - Recherche d'une bonne solution
- 3 choix majeurs
 - ▣ Quel test associer à un nœud?
 - Critère de séparation
 - ▣ Un nœud est-il terminal?
 - Critère d'arrêt
 - ▣ Quelle classe affecter à la feuille?
 - Classe majoritaire
 - Si égalité : classe la plus représentée

Construction d'un arbre de décision

13

- Algorithmes classiques
 - ▣ CART [Breiman et al., 1984]
 - Nœuds binaires uniquement
 - ▣ ID3 [Quinlan, 1986]
 - Test d'un attribut qualitatif par nœud, pas de traitement des valeurs manquantes
 - ▣ C4.5 [Quinlan, 1993]
 - Amélioration de ID3 : prise en compte des attributs quantitatifs, des valeurs manquantes, ...

Critères de séparation

14

- Intuition : les sous-ensembles des nœuds fils doivent être plus homogènes que le nœud courant
- Evaluation de la meilleure séparation?
 - ▣ Critère du χ^2
 - ▣ Calcul de l'entropie [Shannon, 1948]
 - Utilisé dans ID3 et C4.5
 - ▣ Indice de Gini
 - Utilisé dans CART

Calcul de l'entropie

15

- Estimation de la quantité d'information contenue dans une source d'information
 - ▣ Entropie ++ → Hétérogénéité ++
- Calcul
 - ▣ Soit un ensemble X dont chacune des instances est associée à une classe de Y (1 à n valeurs)
 - ▣ On note $p_{k \in Y}$ la proportion d'exemples de X dont la classe associée est k .
- Mesure de l'entropie de X :

$$H(X) = - \sum_{k \in Y} p_k \cdot \log p_k$$

Remarques sur l'entropie

16

- Comprise entre 0 et 1
- Exemple d'une classification à 2 classes :
 - ▣ Entropie nulle :
 - Si $p_+=0$ ou $p_-=0$ alors $H(X)=0$
 - ▣ Entropie maximale :
 - Si $p_+=0.5$ et $p_-=0.5$ alors $H(X)=1$ (si $\log 2$, 0.7 sinon)

Critère de Gini

17

- Estimation de la dispersion d'une distribution (fonction de pureté)
 - ▣ Indice de Gini ++ → Hétérogénéité ++
- Calcul
 - ▣ Soit un ensemble X dont chacune des instances est associée à une classe de Y (1 à n valeurs)
 - ▣ On note $p_{k \in Y}$ la proportion d'exemples de X dont la classe associée est k .
- Mesure de l'indice de Gini de X :

$$Gini(X) = 1 - \sum_{k \in Y} p_k^2$$

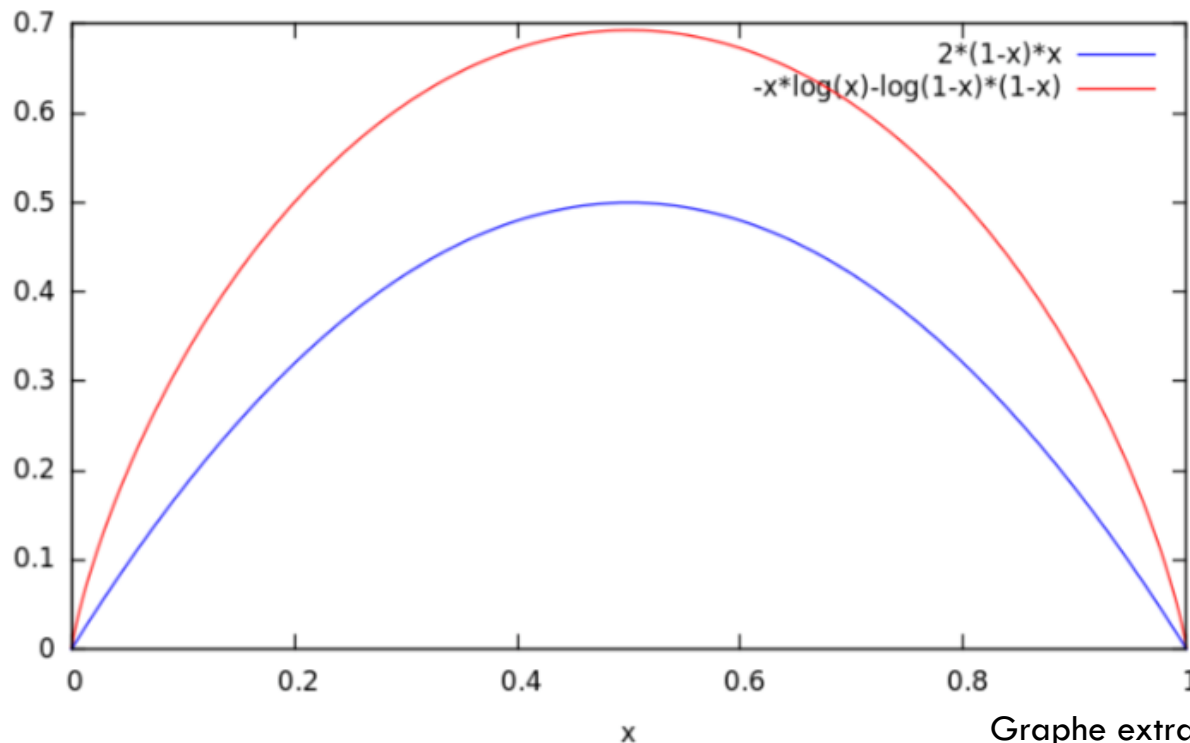
Comparaison Gini/Entropie

18

□ Exemple pour k=2 classes

$$Gini(X) = 1 - (x^2 + (1 - x)^2) = 2x * (1 - x)$$

$$H(X) = -x \log x - (1 - x) \log(1 - x)$$



Graphé extrait du cours de C. Capponi

Choix du test de séparation

19

- Test ou critère de séparation : permet d'obtenir les sous-ensembles les plus homogènes
- Calcul du **gain d'information** selon *test*
 - ▣ Soit *test*, qui implique p partitions de X en X_j
 - ▣ Variation du critère de séparation f causée par la partition de X selon *test* :

$$Gain(X, test) = f(X) - \sum_{j=1}^{j=p} \frac{\#X_j}{\#X} f(X_j)$$

- ▣ Avec $f \in \{H, Gini, \dots\}$
- ▣ Choix du test qui maximise le gain

Types de test

22

- Descripteurs qualitatifs
 - ▣ Binaires
 - ▣ n-aires
 - Pbm privilège des descripteurs de grande arité
 - Ratio de gain C4.5
- Descripteurs quantitatifs
 - ▣ Discrétisation
 - ▣ Test binaire $a < v$ selon les valeurs de a dans S
- Descripteurs textuels
 - ▣ en tant que tel ?
 - Extension des arbres de décision aux arbres de décision sémantique

Avantages des arbres de décision

31

- Interprétation humaine facile
 - ▣ Ensemble de règles intelligibles : boîte blanche!
 - ▣ Sélection automatique de descripteurs pertinents
- Rapidité
 - ▣ Construction : parallélisation possible
 - ▣ Classification : parcours d'un seul chemin
- Robustesse
 - ▣ Influence faible des exemples erronés
 - ▣ Possible solution pour les données manquantes

Inconvénients des arbres de décision

32

- Attention au sur-apprentissage
 - ▣ Arbre trop complexe
 - ▣ Peu de généralisation
- Obtention d'un arbre optimal NP-complet
 - ▣ Algorithme glouton
 - ▣ Optimisation locale, pas de backtracking
- Instabilité de l'arbre
 - ▣ Performances fortement dépendantes de la taille de S
 - Léger changement dans les données
 - Arbres résultants différents surtout si changement racine
 - → arbres très différents

Extensions des arbres de décision

33

- Arbre de décision sémantique
 - ▣ Prise en compte de la notion de texte
- Bagging d'arbres de décision
- Forêts aléatoires (Random forest)
- Boosting d'arbres de décision « faible »
 - ▣ Et aussi boosting d'arbres de décision sémantique

Arbre de décision sémantique

34

- Proposé par R. Kuhn et R. De Mori en 1995
- Application des arbres de décisions au langage naturel pour la compréhension de la parole (**SLU** – Spoken Language Understanding)
 - ▣ Système de compréhension = Arbre de décision sémantique (**SCT** – Semantic Classification Tree)
 - Formuler la tâche de SLU comme une tâche de classification
 - ▣ 2 problématiques
 - Comment extraire le sens?
 - Comment faire face aux erreurs de transcriptions?

Propriétés des SCT

35

- Apprentissage de règles pour classer des nouvelles chaînes de mots à partir d'un ensemble de chaînes de mots classées.
- Tests = expressions régulières composées de 2 types de symboles : « vocab » et « + »
- Génération et sélection des tests complètement automatique, chaque symbole susceptible d'apparaître dans un test
- Deux types de SCT : *single-symbol* ou *set-membership*
 - ▣ *Set-membership* : création d'ensemble de mots

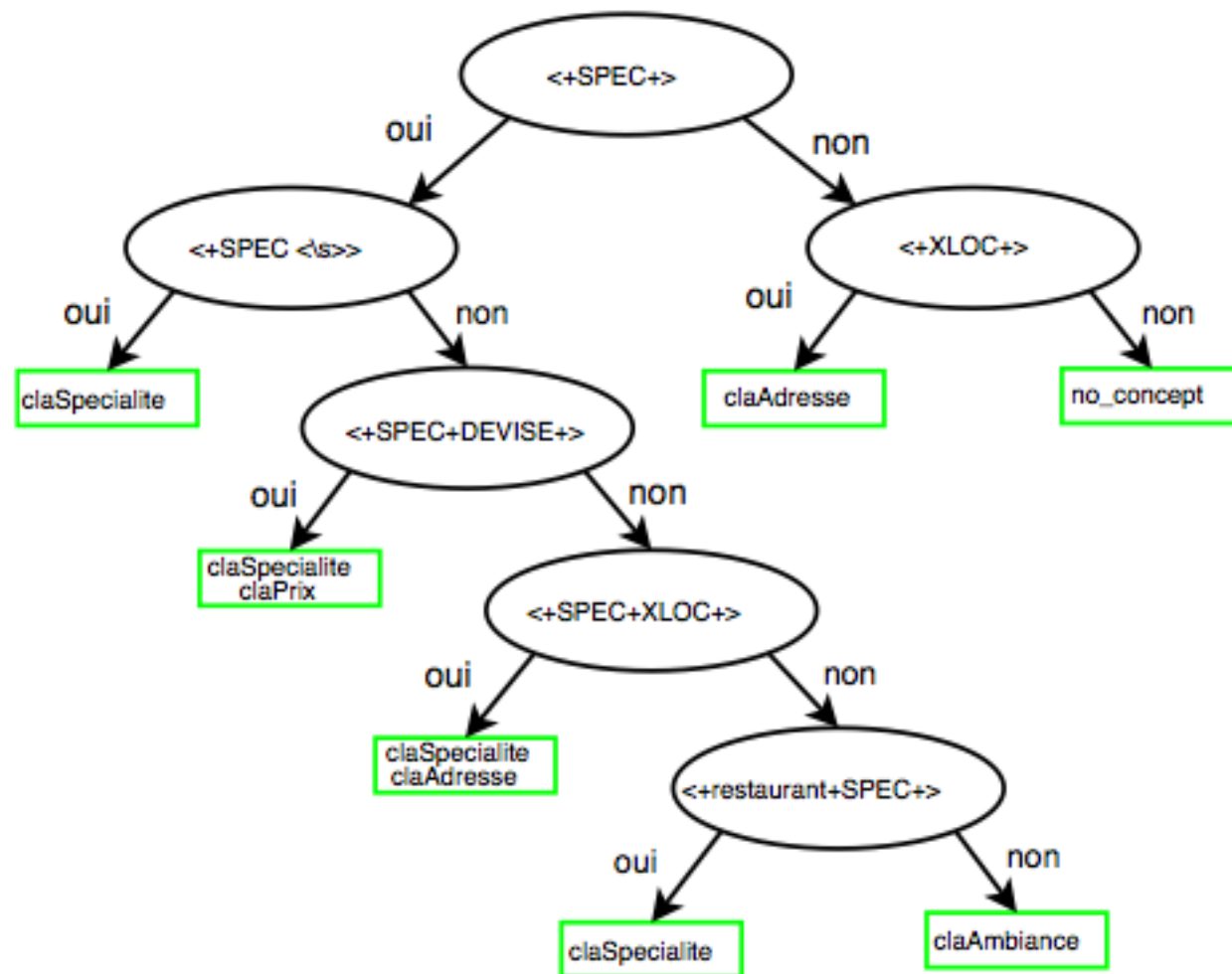


FIG. 2.3 – Schéma simplifié d'un arbre de classification sémantique

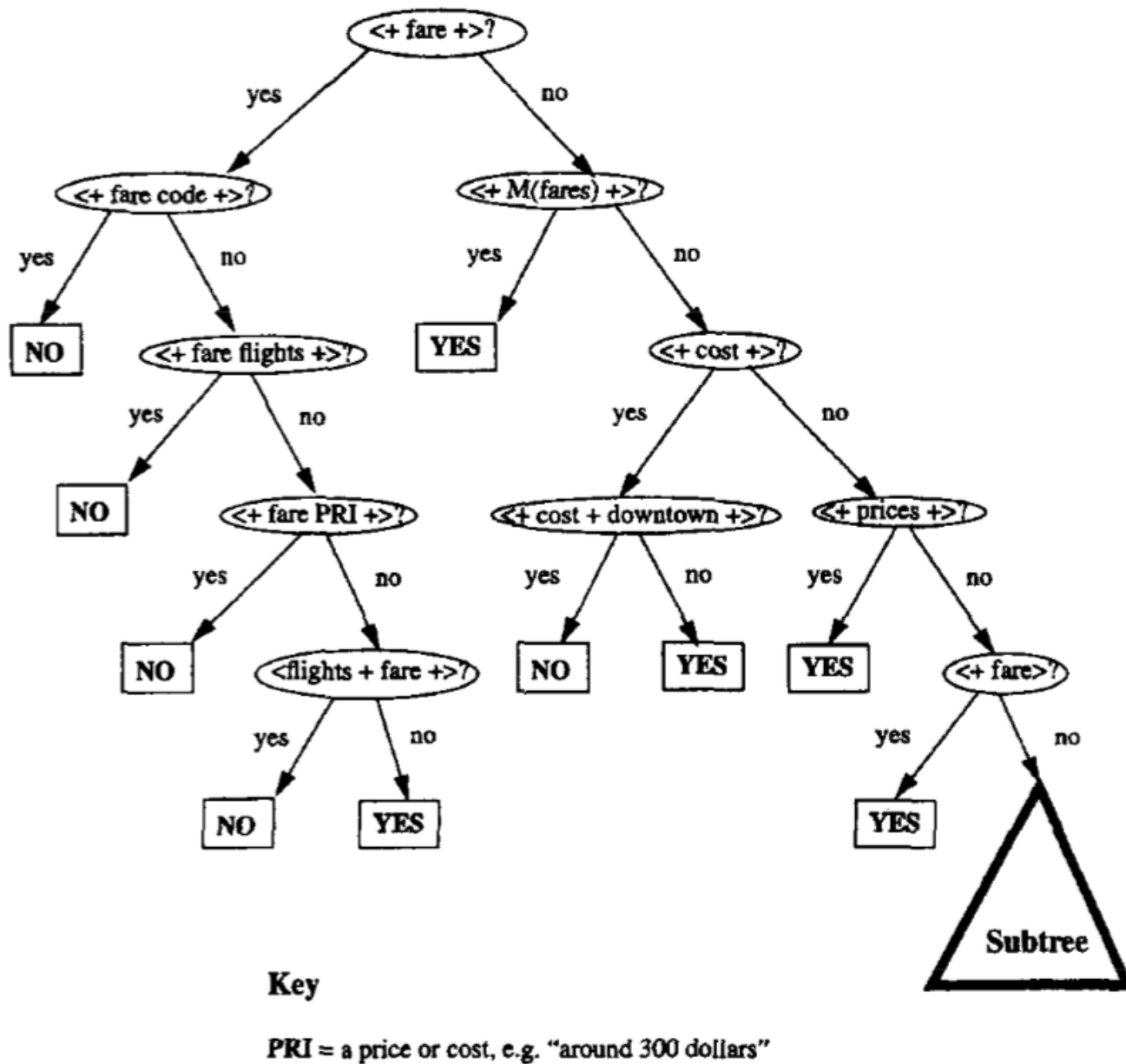
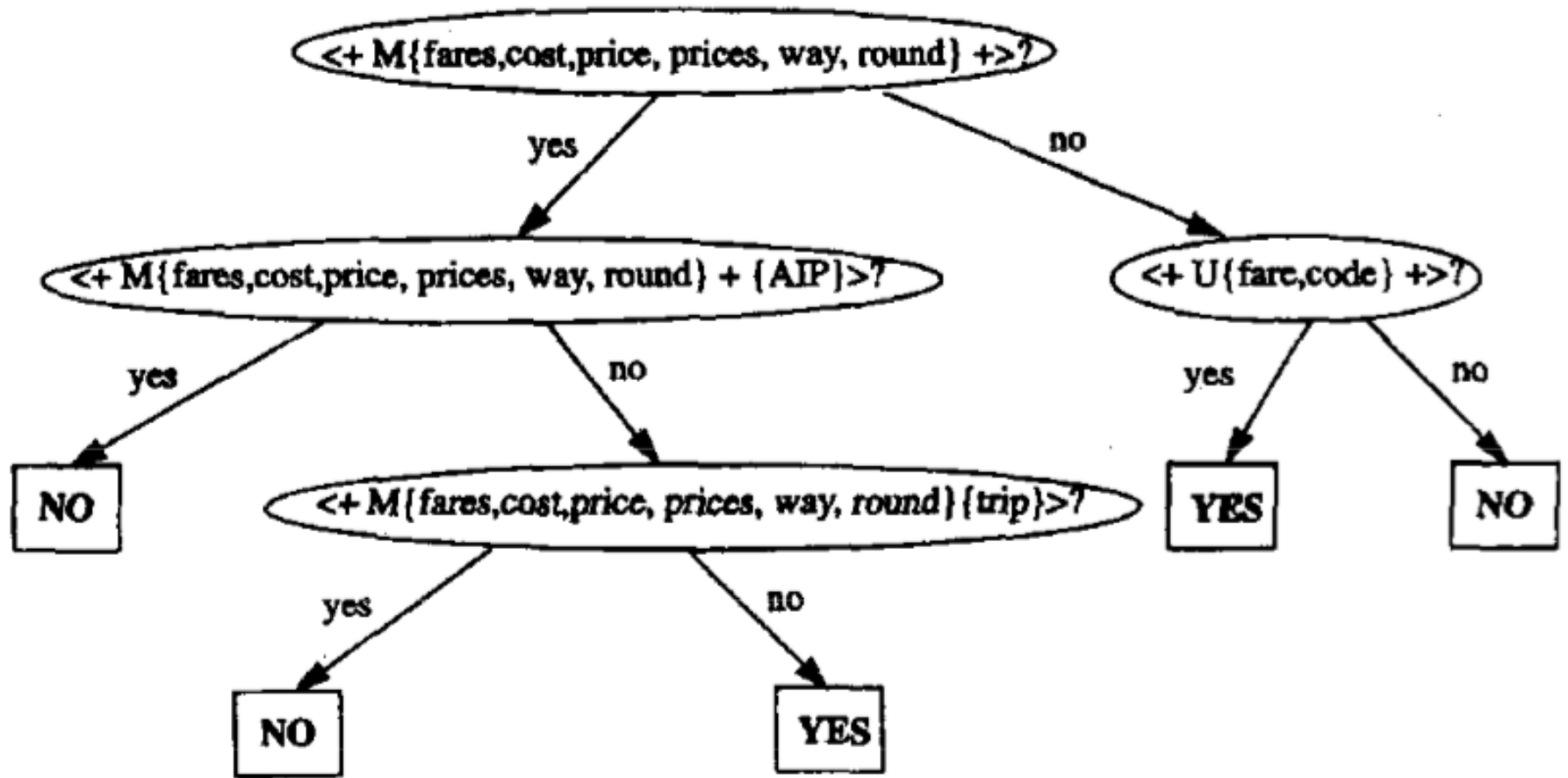


Fig. 1. Single-symbol SCT for fare.fare_id (37 nodes).

[Kuhn and De Mori, 1995]



Key

AIP = airport name, e.g. "Logan airport"

Fig. 3. Set-membership SCT for *fare.fare_id* (9 nodes).

Construction des tests du SCT

39

- Prise en compte des descripteurs textuels en tant qu'élément d'un langage naturel!
 - ▣ Ordre des mots
 - ▣ Différents niveaux de représentation d'un mot

MOT	je	cherche	un	restaurant	chinois	dans	le	3e	arrondissement
POS	PPER1S	V1S	DETMS	NMS	AMS	PREP	DETM S	MOTINC	NMS
LEMME	il	chercher	un	restaurant	chinois	dans	le	3e	arrondissement
SEM	XX	XX	X_NB	X_LIEU	X_SPEC	XX	XX	X_ORD	X_LOC

Construction d'un SCT

40

- Ensemble d'expressions régulières
 - ▣ Soit $w \in V$, avec V le vocabulaire de l'ensemble des descripteurs textuels de S
 - ▣ Soit $+$ symbole indiquant une séquence de mot non vides
 - ▣ Soit $\pi_0 = \{w, +w, w+, +w+\}$, l'ensemble des expressions régulières possibles
 - Test de présence de $4^{|V|}$ expressions régulières **à la racine**
- Choix du test j qui maximise la variation selon l'indice de Gini

$$\Delta Gini^j = Gini(S) - p_{oui}^j Gini(S_{oui}^j) - p_{non}^j Gini(S_{non}^j)$$

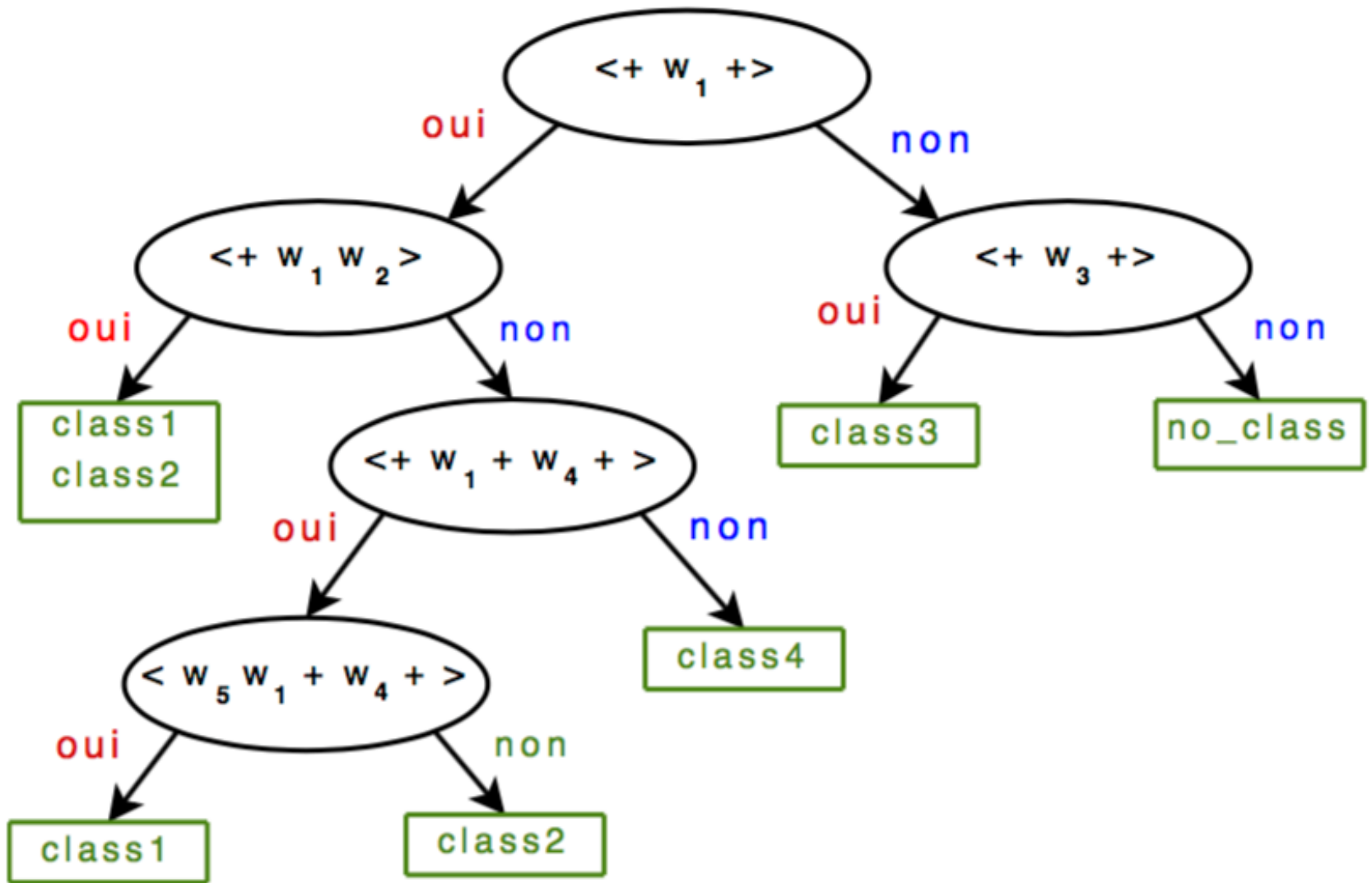
Test des fils « oui »?

41

- Substitution du symbole $+$ par π_0
- Soit le test du nœud issu d'une branche oui :
 $+W_1 + W_2 + W_3 + \dots + W_i + \dots$, avec W_i fixé
- Ensemble des questions possibles :

$$\begin{aligned} &\pi_0 W_1 + W_2 + W_3 + \dots + W_i + \dots \\ &+ W_1 \pi_0 W_2 + W_3 + \dots + W_i + \dots \\ &+ W_1 + W_2 \pi_0 W_3 + \dots + W_i + \dots \\ &\dots \end{aligned}$$

Soit $4 * | + | * | \vee_{\{\text{oui}\}} |$ tests possibles



Avantages des SCT

43

- Avantage des arbres de décisions
 - ▣ Modèle lisible (intelligible ++)
 - ▣ Rapidité de construction
- Robustesse des règles de décisions
 - ▣ Possibilité des + génériques
 - ▣ Possibilité de classes de mots
 - ▣ Prise en compte des différents niveaux du mot
- Résistance aux différentes formulations
- Lissage sur les erreurs de reconnaissance
 - sauf si le mot impacté fait partie de la règle...

Bagging d'arbres de décision

44

- Notion de Bagging introduite par Breiman en 1996
 - ▣ Tree Bagging si arbres de décision
- Idée :
 - ▣ Entraîner des modèles sur des sous-ensembles différents d'observations puis les agréger
 - Réduire la variance
- Principe du Bagging : Bootstrap AGGREGatING
 - ▣ Méta-algorithme qui combine
 - Booststrapping
 - Tirage au hasard de n individus parmi n avec remise dans S
 - Aggrégation
 - Calculer un arbre de décision par échantillon
 - Classification par vote majoritaire

Random forest

45

- Proposé par Breiman, 2001
- Amélioration du Bagging
 - ▣ Objectif : Décorrélér au maximum les différents arbres de décision
 - ▣ Double échantillonnage
 - Bootstrap : sélection d'un sous-ensemble d'individus
 - Feature sampling : sélection d'un sous-ensemble de descripteurs, par défaut $\sqrt{\#descripteurs}$

Boosting d'arbre de décision

46

- Algorithme Adaboost (Adaptative Boosting)
proposé par Y. Freund et R. Schapire en 1996
- Principe du boosting :
 - ▣ Faire coopérer plusieurs modèles « faibles »
 - ▣ Diriger l'apprentissage sur les données difficiles à classer
 - ▣ Apprentissage séquentiel
 - ▣ Vote pondéré

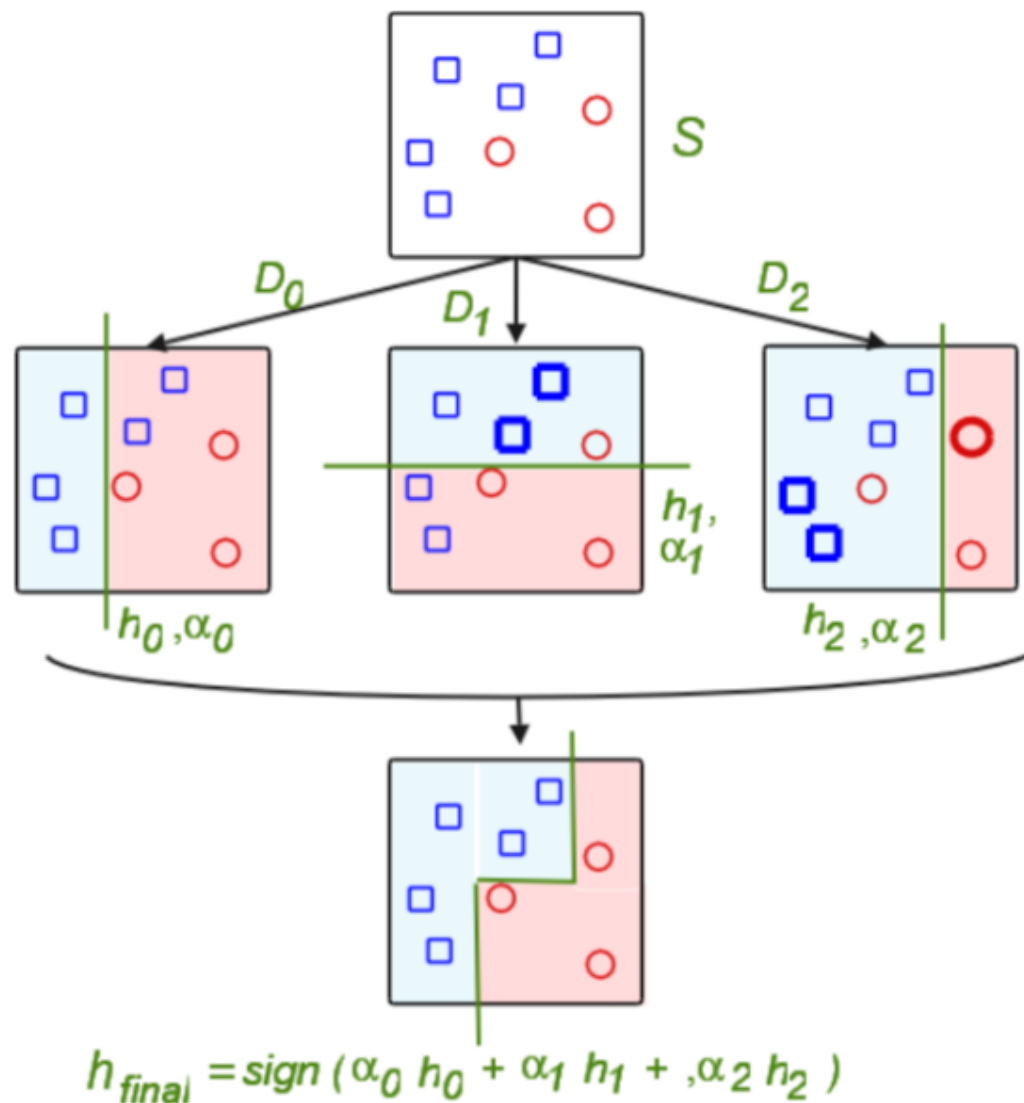


FIG. 2.4 – Schéma simplifié de l'algorithme de boosting. À chaque tour de l'algorithme, une hypothèse faible permettant de séparer les données est faite. Les exemples mal classés voient leur poids augmenter à la distribution suivante. L'hypothèse finale est une combinaison des hypothèses faibles.

Adaboost : Adaptive Boosting

48

Étant donné :

- Un jeu de données $S : (x_1, y_1), \dots, (x_m, y_m)$ où, à chaque exemple $x_i \in X$, on associe une étiquette $y_i \in Y = \{-1, +1\}$;
- Une distribution initiale des poids $D_1(i) = 1/m$ uniforme sur ces données ;
- Un apprenant faible (*weak learner*).

Alors pour chaque tour $t = 1, \dots, T$:

- Entraîner l'apprenant faible sur le jeu de données S avec la distribution D_t ;
- Obtenir l'hypothèse faible $h_t : X \rightarrow \{-1, +1\}$ ainsi que l'erreur $\epsilon_t = \sum_{i: h_t(x_i) \neq y_i} D_t(i)$
- Calculer la *pondération du tour* t : $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$
- Mettre à jour la distribution : $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
avec Z_t un facteur de normalisation permettant à D_{t+1} d'être une distribution.

En sortie, on obtient une hypothèse finale combinée qui est un vote pondéré de toutes les hypothèses faibles : $h_{final} = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$

FIG. 2.3 – Algorithme général d'AdaBoost.MH et AdaBoost.MR.

Implémentations

49

- LIA-SCT : Arbre de décision sémantique
 - ▣ http://lia.univ-avignon.fr/chercheurs/bechet/download_fred.html
- Boosting d'arbres de décision pour le texte
 - ▣ BoosTexter
 - Schapire, R.E. & Singer, Y., « BoosTexter: A Boosting-based System for Text Categorization », Machine Learning (2000) 39: 135.
 - ▣ lcsiboost
 - <https://github.com/benob/lcsiboost>
 - ▣ Bonzaiboost
 - <http://bonzaiboost.gforge.inria.fr/>

Bibliographie

50

□ Livres

- ▣ « Data Mining et Statistique décisionnelle », S. Tufféry, Ed. Technip, 2009.
- ▣ « Data Science : fondamentaux et études de cas », E. Biernat et Michel Lutz, Ed. Eyrolles, 2017 (5^e éd.).

□ Cours en ligne

- ▣ https://eric.univ-lyon2.fr/~ricco/cours/slides/Arbres_de_decision_Introduction.pdf
- ▣ <http://pageperso.lif.univ-mrs.fr/~cecile.capponi/lib/exe/fetch.php?media=cours-arbres.pdf>
- ▣ <http://www.grappa.univ-lille3.fr/~ppreux/Documents/notes-de-cours-de-fouille-de-donnees.pdf>
- ▣ https://eric.univ-lyon2.fr/~ricco/cours/slides/bagging_boosting.pdf

- Thèse N. Camelin http://lia.univ-avignon.fr/fileadmin/documents/Users/Intranet/fich_art/TheseCamelin2007.pdf

Bibliographie de référence

51

□ Référence

- J. Ross Quinlan, « Induction of decision trees », *Machine Learning*, p. 81-106, 1986.
- J. R. Quinlan, « *C4.5: Programs for Machine Learning* », Morgan Kaufmann, San Mateo, CA, 1993.
- R. Kuhn et R. De Mori, « The application of semantic classification trees to spoken language understanding », *IEEE Transactions on pattern analysis and machine intelligence*, Vol 17 N°4, Avril 1995.
- L. Breiman, « Bagging predictors », *Machine Learning*, Springer, 1996
- Y. Freund et R.E. Schapire, « Experiments with a new boosting algorithm », *Machine Learning*, 1996.
- R.E. Schapire et Y. Singer, « BoosTexter : A boosting-based system for text categorization », *Machine Learning*, 39:135– 168, 2000.
- L. Breiman, « Random forests », *Machine Learning*, Springer, 2001.