Module 22 Juin - Arbres et méthodes d'ensemble Partie I

Florence d'Alché-Buc,
florence.dalche@telecom-paristech.fr

Telecom Evolution, Paris, France

Outline

Classification supervisée et régression

Cadre probabiliste et statistique

Soit X un vecteur aléatoire de $\mathcal{X} = \mathbb{R}^p$
Y une variable aléatoire dans $\mathcal{Y} = \{1, \ldots, C\}$ (classification) ou $\mathcal{Y} = \mathbb{R}$
Soit P la loi de probabilité jointe de (X,Y), loi fixée mais inconnue
Supposons que $\mathcal{S} = \{(x_i, y_i), i = 1, \ldots, n\}$ soit un échantillon i.i.d. tiré de la loi P

- A partir de $\mathcal{S}$, déterminer la fonction $h \in \mathcal{H}$ qui minimise $R(h) = \mathbb{E}_P[\ell(X, Y, h(X))]$
- Exemple en classification : $\ell(x, y, h(x)) = 1$ si $h(x) \neq y$, 0 sinon.
- Exemple, en régression: $\ell(x, y, h(x)) = (y - h(x))^2$

Apprendre un classifieur

Approche discriminante

- Définir
  - l'**espace de représentation** des données

Apprendre un classifieur

Approche discriminante

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées

Apprendre un classifieur

Approche discriminante

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe

Apprendre un classifieur

Approche discriminante

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe
  - l'**algorithme de minimisation** de cette fonction de coût

Apprendre un classifieur

Approche discriminante

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe
  - l'**algorithme de minimisation** de cette fonction de coût
  - une **méthode de sélection de modèle** pour définir les hyperparamètres

Apprendre un classifieur

Approche discriminante

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe
  - l'**algorithme de minimisation** de cette fonction de coût
  - une **méthode de sélection de modèle** pour définir les hyperparamètres
  - une méthode d'évaluation des performances

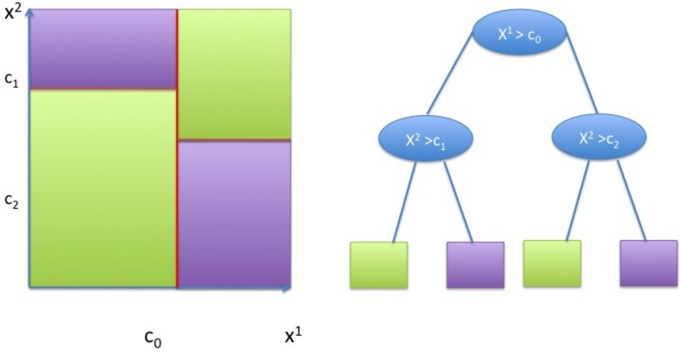
Objectifs

- Arbres de décision et de régression
- Méthodes d'ensemble

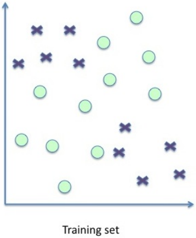
Outline

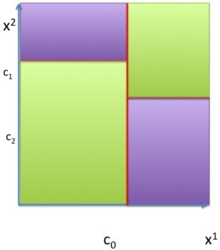
Arbres de décision

Inventés en 1983 en parallèlle par L. Breiman et col. (Berkeley) et R. Quinlan

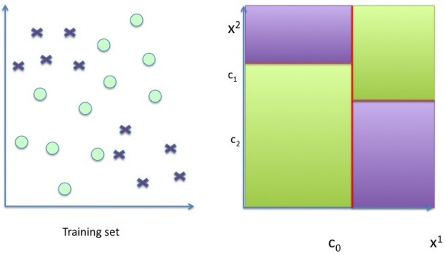


Arbres de décision 1



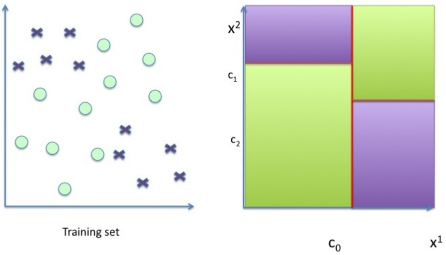


Arbres de décision 1



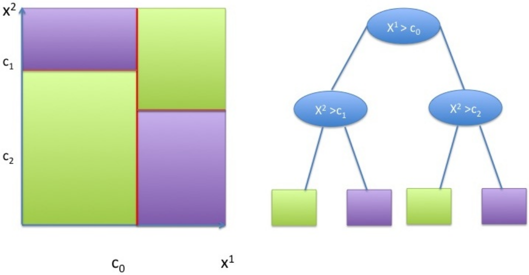
Première idée:
Utiliser non pas 1 mais plusieurs séparateurs linéaires pour construire des frontières de décision non linéaires
Deuxième idée:

Arbres de décision 1



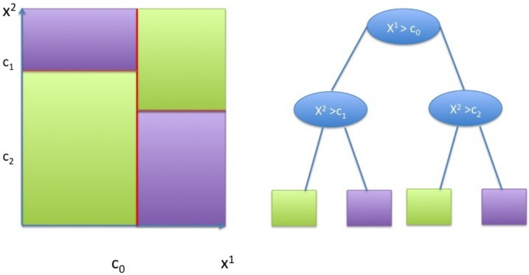
Première idée:
Utiliser non pas 1 mais plusieurs séparateurs linéaires pour construire des frontières de décision non linéaires
Deuxième idée:
Utiliser des séparateurs linéaires orthogonaux chaque vecteur de base, i.e. des hyperplans de la forme $x^j = c$ pour garder une interprétabilité de la fonction construite

Arbres de décision 2



Troisième idée:
La fonction de décision peut être représentée par une structure d'arbre dont chaque noeud intermédiaire est associé à un hyperplan séparateur de la forme $x^j = \theta_j$ et chaque feuille est associée une fonction constante, i.e. une classe.

Arbres de décision 2



A l'issue de la phase d'apprentissage, on connaît les variables explicatives qui interviennent dans la fonction de décision construite

L'arbre code pour un ensemble de règles logiques du type:

si $(x^{j_1} > c_{j_1})$ et $(x^{j_2} \leq c_{j_2})$ et ... alors x est de la classe k

Séparateur linéaire orthogonal à un vecteur de base

Variable x^j continue:

$$t_{j,c}(\mathbf{x}) = \text{signe}(x^j - c) \tag{1}$$

Remarque: on peut aussi traiter une variable x^j catégorielle à K valeurs $\{v_1^j, \ldots, v_K^j\}$:

$$t(x; v_j) = 1(x^j = v_j) \tag{2}$$

Algorithme de construction

1. Soit $\mathcal{S}$ l'ensemble d'apprentissage
2. Construire un noeud racine
3. Chercher la meilleure séparation $t(\mathbf{x})$ à appliquer sur $\mathcal{S}$ telle que le coût local $L(t, \mathcal{S})$ soit minimal
4. Associer le séparateur choisi au noeud courant et sémparer l'ensemble d'apprentissage courant $\mathcal{S}$ en $\mathcal{S}_d$ et $\mathcal{S}_g$ à l'aide de ce séparateur.
5. Construire un noeud fils à droite et un noeud à gauche.
6. Mesurer le critère d'arrêt à droite, s'il est vérifié, le noeud droit devient une feuille sinon aller en 3 avec $\mathcal{S}_d$ comme ensemble courant
7. Mesurer le critère d'arrêt à gauche, s'il est vérifié, le noeud gauche devient une feuille sinon aller en 3 avec $\mathcal{S}_g$ comme ensemble courant.

Fonction de coût locale

Soit un ensemble d'exemples d'apprentissage $\mathcal{S}$ et une fonction de séparation binaire $t_{j,\tau}$. Notons
$\mathcal{D}(\mathcal{S}, j, \tau) = \{(\mathbf{x}, y) \in \mathcal{S}, t_{j,\tau}(\mathbf{x}) > 0\}$ et
$\mathcal{G}(\mathcal{S}, j, \tau) = \{(\mathbf{x}, y) \in \mathcal{S}, t_{j,\tau}(\mathbf{x}) \leq 0\}$.
Parmi tous les paramètres $(j, \tau) \in \{1, \ldots, p\} \times \{\tau_1, \ldots, \tau_m\}$, on cherche $\hat{j}$ et $\hat{\tau}$ qui minimisent :

$$L(t_{j,\tau}, \mathcal{S}) = \frac{n_d}{n} H(\mathcal{D}(\mathcal{S}, j, \tau)) + \frac{n_g}{n} H(\mathcal{G}(\mathcal{S}, j, \tau)) \quad (3)$$

$$n_d = |\mathcal{D}(\mathcal{S}, j, \tau)| \quad (4)$$

$$n_g = |\mathcal{G}(\mathcal{S}, j, \tau)| \quad (5)$$

Fonction de coût locale pour la classification supervisée

On définit pour un ensemble $\mathcal{S}$ de n exemples étiquetés

$$p_C(\mathcal{S}) = \frac{1}{n} \sum_{i=1}^{n} 1(y_i = C)$$

Voici les principaux critères H qui peuvent être utilisés:
Entropie croisée:

$$H(\mathcal{S}) = -\sum_{\ell=1}^{C} p_\ell(\mathcal{S}) \log p_\ell(\mathcal{S})$$

Critères de coût

Entropie croisée:

$$H(\mathcal{S}) = -\sum_{\ell=1}^{C} p_\ell(\mathcal{S}) \log p_\ell(\mathcal{S})$$

Index de Gini

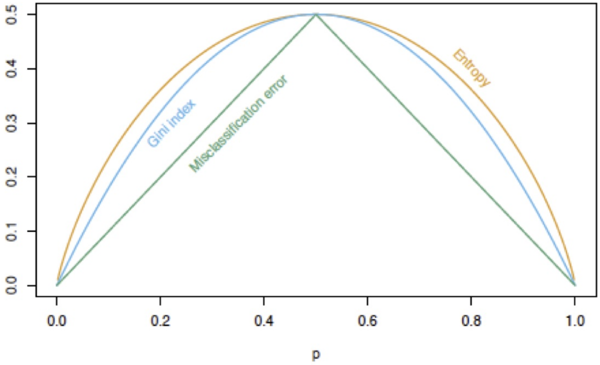
$$H(\mathcal{S}) = \sum_{\ell=1}^{C} p_\ell(\mathcal{S})(1 - p_\ell(\mathcal{S}))$$

Erreur de classification

$$H(\mathcal{S}) = 1 - p_{C(\mathcal{S})},$$

avec $C(\mathcal{S})$: classe majoritaire dans $\mathcal{S}$.

Visualisation des critères de coût



Critères d'arrêt

- La profondeur maximale
- Le nombre maximale de feuilles
- Le nombre minimal d'exemples dans un noeud (pas assez d'exemples)

NB : Si le nombre minimal d'exemples est 1, l'ensemble d'apprentissage est appris jusqu'au bout (dans les limites computationnelles et de mémoire) : risque de sur-apprentissage !

Variables catégorielles, multi-classe

- Pour avoir un arbre binaire : si une variables catégorielle est K valeurs, on la transforme en K variables binaires
- L'algorithme d'apprentissage est approprié pour traiter aussi bien des problèmes biclasse que multi-classe

Arbres de régression

Le critère de coût devient un critère objectif à maximiser:

$$L(t_{j,\tau}, \mathcal{S}) = VAR_{emp}(\mathcal{S}) - \frac{n_d}{n} VAR_{emp}(\mathcal{D}(j,\tau,\mathcal{S})) - \frac{n_g}{n} VAR_{emp}(\mathcal{G}(j,\tau,\mathcal{S}))$$

Soit $\mathcal{S}$.

$$VAR_{emp}(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{(x_i,y_i)\in\mathcal{S}} (y_i - \bar{y})^2$$

On cherche à maximiser l'homogénéité des sorties.
ATTENTION : un arbre de régression est une fonction valeurs constantes par morceaux.

Sélection de modèles

(1) On s'intéressera à déterminer un des hyperparamètres suivants:

- Profondeur maximale
- NB de feuilles maximal
- NB d'exemple minimal dans une feuille/noeud

$\rightarrow$ **par validation croisée**.

Sélection de modèles

(2) **par élagage**
On utilise un ensemble de validation pour re-visiter un arbre appris sans limite sur un ensemble d'apprentissage. On ne garde que les branches qui apportent une amélioration sur l'ensemble de validation.

Avantages et inconvénients des arbres de décision

Avantages

- Construit une fonction de décision non linéaire, interprétable
- Fonctionne pour le multiclasse
- Prise de décision efficace: $O(\log F)$
- Fonctionne pour des variables continues et catégorielles

Inconvénients

- Estimateur à large variance : une petite variation dans l'ensemble d'apprentissage et l'abre est complètement différent
- Pas d'optimisation globale

Exercice

Définir la famille de fonctions de décision induite par un arbre de F feuilles: