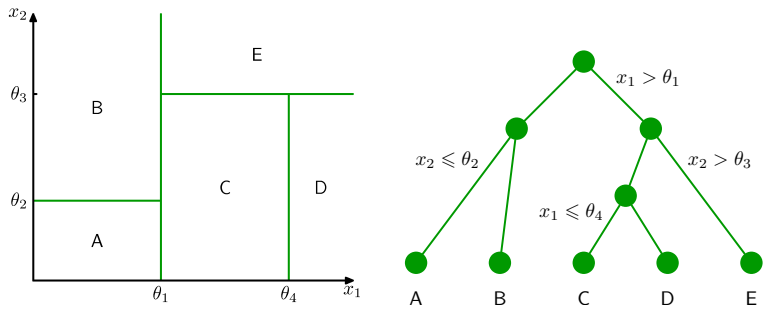
Arbres de décision

Benjamin Monmege
benjamin.monmege@lsv.ens-cachan.fr

1er mars 2012

On considère un problème de régression ou de classification dans lequel le but est de prédire une variable t (données continues ou classes) à partir d'un vecteur $\mathbf{x} = (x_1, \dots, x_D)$ de dimension D en entrée. Les données d'apprentissage consistent en un ensemble de vecteurs d'entrée $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ et leurs étiquettes respectives $\{t_1, \dots, t_N\}$. On cherche à construire un arbre de décision pour ces données. La figure ci-dessous montre un exemple où l'espace de dimension $D = 2$ a été partitionné en 5 régions dont les frontières sont alignés avec les axes. La première étape divise le plan en deux régions $\{x_1 \leq \theta_1\}$ et $\{x_1 > \theta_1\}$. La région $\{x_1 \leq \theta_1\}$ peut ensuite être subdivisée en deux régions $\{x_2 \leq \theta_2\}$ et $\{x_2 > \theta_2\}$, notées respectivement A et B. Ces régions, tout comme les régions C, D et E, sont les régions terminales de la partition. Dans chacune des régions terminales, il faut ensuite choisir la meilleure prédiction possible : dans un problème de régression, on choisira souvent une valeur constante commune à toute la région, alors que dans un problème de classification, on associera chaque région à une classe.



Exercice 1. On se place ici dans le cas d'un problème de classification en K classes $\mathcal{C}_1, \dots, \mathcal{C}_K$. Pour un arbre de décision T fixé, on note $|T|$ le nombre de feuilles de l'arbre T, et on numérote les feuilles $\tau = 1, \dots, |T|$: la feuille d'indice τ dénote la région de $\mathcal{R}_\tau \subseteq \mathbb{R}^D$. Notons N_τ le nombre de points dans la région $\mathcal{R}_\tau$. On définit $p_{\tau,k}$ comme la fréquence des points de la région $\mathcal{R}_\tau$ qui appartient à la classe $\mathcal{C}_k$:

$$p_{\tau,k} = \frac{1}{N_\tau} \sum_{\mathbf{x}_n \in \mathcal{R}_\tau} \delta_{t_n,k}$$

où $\delta_{i,j}$ est la fonction de Kronecker. On définit deux fonctions d'erreur :

- Indice de Gini : $\mathcal{E}_\tau(T) = \sum_{k=1}^K p_{\tau,k}(1 - p_{\tau,k})$;
- Cross-entropie ou déviance : $\mathcal{E}_\tau(T) = -\sum_{k=1}^K p_{\tau,k} \ln p_{\tau,k}$.

1. En supposant que la partition de l'espace d'entrée est donnée (c'est-à-dire l'arbre tel qu'il est représenté ci-dessus), l'algorithme doit prédire la classe c_τ à associer aux éléments de la région $\mathcal{R}_\tau$. Comment prédire la meilleure classe c_τ à associer à chaque région τ ?
2. On cherche désormais à trouver le meilleur arbre de décision. Peut-on simplement énumérer l'ensemble des arbres et trouver le meilleur ?

À la place, on procède à la construction de l'arbre à l'aide d'un algorithme glouton, qui construit l'arbre nœud après nœud en commençant par la racine, sans être assuré de trouver le meilleur arbre.

3. À chaque étape, une feuille doit être choisie, ainsi qu'une des dimensions x_d et un seuil θ, pour poursuivre la partition. Il est utile de choisir la meilleure règle de partition possible. Quelle méthode proposez-vous ?
4. Il est finalement utile de connaître un critère d'arrêt décidant s'il faut, ou pas, continuer à partitionner les feuilles d'un arbre. Dans un premier temps, on choisit d'arrêter la partition lorsque chaque région contient un petit nombre, fixé, de données (par exemple 5). Vous pouvez essayer d'implémenter cette méthode... Cependant, vous pouvez également utiliser la classe `classregtree` disponible dans la Toolbox stats de MATLAB [1] ? Tester sur l'exemple en 2 dimensions que nous avons utilisé dans le TP Plus proches voisins : que vaut le nombre de régions et la profondeur de l'arbre ?
5. Afin de simplifier l'arbre de décision ainsi obtenu, on coupe des branches (*pruning*) en se basant sur un critère de complexité de l'arbre. Étant donné un arbre T_0, on cherche donc à construire un sous-arbre T de T_0 en combinant certaines régions de l'arbre T_0. La fonction de complexité est du type
$$C_\lambda(T) = \sum_{\tau=1}^{|T|} \mathcal{E}_\tau(T) + \lambda|T|$$
où λ est le paramètre de régularisation, déterminant le compromis entre l'erreur globale et la complexité de l'arbre T (mesurée comme son nombre de feuilles). À quoi correspond la choix $\lambda = 0$? Comment déterminer le meilleur paramètre λ ?
6. En MATLAB, et pour les deux fonctions d'erreur possibles, construire l'arbre T optimal et comparer l'arbre obtenu avec celui que vous aviez à la fin de la question 4. Tester ensuite votre méthode avec l'exemple des iris [2]. Essayer d'utiliser une méthode de validation, et étudier la variabilité du résultat selon la découpe de l'ensemble de données en ensemble d'apprentissage et ensemble de test.
7. On considère un ensemble de données comprenant 400 points dans une classe $\mathcal{C}_1$ et 400 points dans une classe $\mathcal{C}_2$. Supposons qu'un arbre de décision A divise les données en $(300, 100)$ dans la première région et $(100, 300)$ dans la seconde région (où (n, m) dénote que n points sont associés à la classe $\mathcal{C}_1$ et m points sont associés à la classe $\mathcal{C}_2$). De manière similaire, supposons qu'un autre arbre de décision B divise les données en $(200, 400)$ dans la première région et $(200, 0)$ dans la seconde. Évaluer les *taux d'erreur de classification* pour les deux arbres et montrer qu'ils sont identiques. De même, évaluer les critères associés à l'indice de Gini et la déviance pour les deux arbres et montrer qu'ils sont tous les deux inférieurs pour l'arbre B que l'arbre A.

Exercice 2. On considère désormais le cas d'un problème de régression. On réutilise les notations de l'exercice précédent. Le même procédé s'applique mis à part le fait qu'il faut modifier la fonction d'erreur pour correspondre à un problème de régression : on utilise ainsi une fonction $\mathcal{E}_\tau(T)$ de somme des erreurs au carré.

1. En supposant que la partition de l'espace d'entrée est donnée (c'est-à-dire l'arbre tel qu'il est représenté ci-dessus), et qu'on cherche à minimiser la fonction d'erreur des moindres carrés, montrer que la valeur de prédiction optimale pour chaque région est la moyenne des valeurs t_n pour les points $\mathbf{x}_n$ qui sont dans cette région.
2. Comment choisir la meilleure règle de partition possible ?
3. Tester en MATLAB.

1. Cette classe n'est pas disponible en Octave... J'en profite pour vous rappeler qu'une version de MATLAB vous est mise à disposition par l'école à l'adresse suivante `http://intranet.ens-cachan.fr/version-francaise/ressources-numeriques/informatique-et-reseaux/produits-logiciels/`.

2. `http://archive.ics.uci.edu/ml/datasets/Iris`