

Decision Trees with Arbitrary Losses

Yann Chevaleyre

(source: David S. Rosenberg)

November 3, 2022

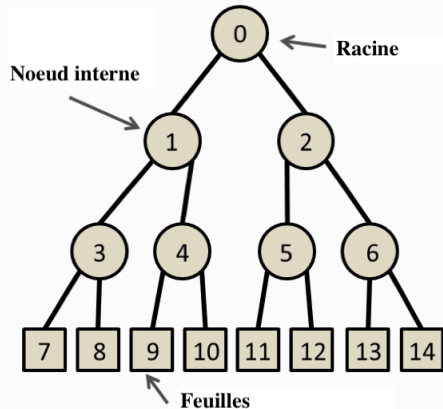
Contents

- 1 Arbres
- 2 Arbres de Régression
- 3 Arbres de décision pour la Classification

Arbres

Terminologie

Structure d'un Arbre de Décision



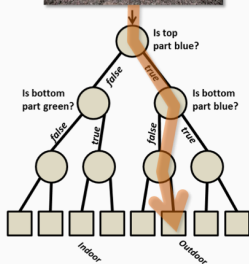
From Criminisi et al. MSR-TR-2011-114, 28 October 2011.

Arbre de décision binaire

arbre de décision binaire: chaque noeud a 2 ou 0 fils



A decision tree

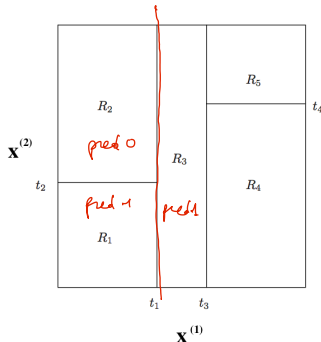
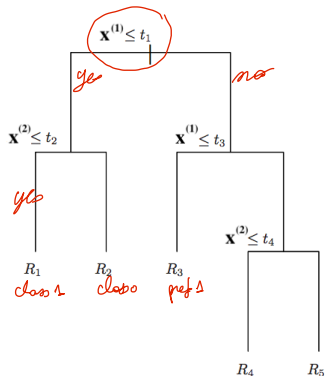


From Criminisi et al. MSR-TR-2011-114, 28 October 2011.

Arbre de décision binaire sur \mathbb{R}^2

- Soit un arbre binaire sur $\mathcal{X} = \mathbb{R}^2$

$$\mathcal{X} = (x^{(1)}, x^{(2)})$$



- A la racine et à chaque noeud interne, une variable $x^{(j)}$ est appelée **variable de test** ou **variable de décision** du noeud, et t est appelé le **seuil**

From *An Introduction to Statistical Learning, with applications in R* (Springer, 2013) G. James, D. Witten, T. Hastie and R. Tibshirani.

Types d'arbres de décision

- Nous nous intéressons aux:

- **arbres binaires** (vs arbres avec plus de 2 fils)
- les décisions à chaque noeuds portent sur une seule variable
- les décisions sur les variables continues sont de la forme

$$x^{(j)} \leq t$$

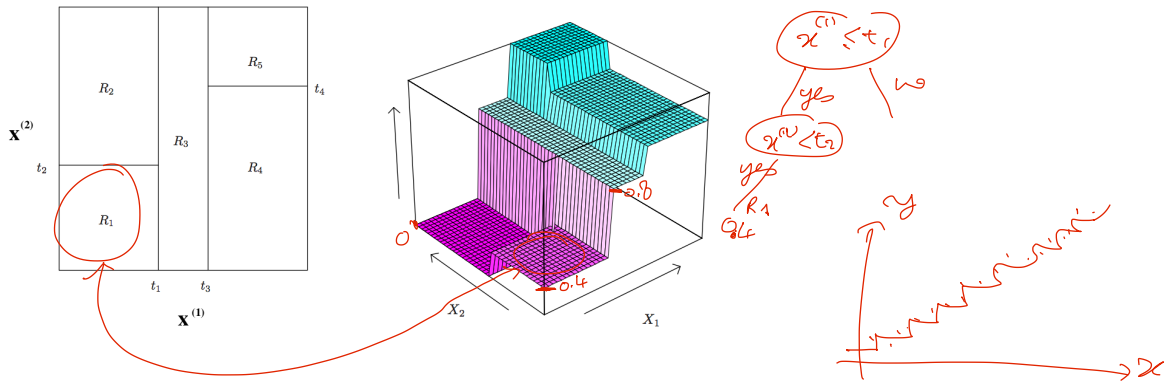
- pour les variables discrètes, les décision partitionnent les valeurs en 2 groupes
- Autres types d'arbres:
 - **oblique decision trees** or **binary space partition trees** (BSP trees)
 - **sphere trees**



Arbres de Régression

Arbres de régression sur \mathbb{R}^2

- Soit un arbre binaire sur $\mathcal{X} = \mathbb{R}^2$. Un exemple est $x = (x^{(1)}, x^{(2)})$



From *An Introduction to Statistical Learning, with applications in R* (Springer, 2013) G. James, D. Witten, T. Hastie and R. Tibshirani.

Apprendre un arbre de régression

- Un arbre de décision partitionne \mathcal{X} en regions:

$$\{R_1, \dots, R_M\}.$$

- Rappel:

$$\mathcal{X} = R_1 \cup R_2 \cup \dots \cup R_M$$

et

$$R_i \cap R_j = \emptyset \quad \forall i \neq j$$

- Soit $N_m = |\{i : x_i \in R_m\}|$

Apprendre un arbre de régression

	total_bill	sex	smoker	day	time	size	tip
0	16.99	Female	No	Sun	Dinner	2	1.01
1	10.34	Male	No	Sun	Dinner	3	1.66
2	21.01	Male	No	Sun	Dinner	3	3.50
3	23.68	Male	No	Sun	Dinner	2	3.31
4	24.59	Female	No	Sun	Dinner	4	3.61

Apprendre un arbre de régression

- Avec la partition $\{R_1, \dots, R_M\}$, la prédiction finale est:

$$f(x) = \sum_{m=1}^M c_m 1(x \in R_m)$$

dataset $S = \{(x_i, y_i)\}_{i=1}^N$
 $x_i \in \mathbb{R}^d$
 $y_i \in \mathbb{R}$

- Supposons qu'on ait déjà la partition, comment choisir c_1, \dots, c_M ?
- Pour la fonction de perte $\ell(\hat{y}, y) = (\hat{y} - y)^2$, comment appliquer l'ERM ?

$$\begin{aligned} c_1, \dots, c_M &= \argmin \sum_{i=1}^N (f(x_i) - y_i)^2 \\ &= \argmin \sum_{i=1}^N \left(\sum_{m=1}^M c_m 1(x_i \in R_m) - y_i \right)^2 \\ &= \argmin \sum_{m=1}^M \sum_{i \in R_m} (c_m - y_i)^2 \end{aligned}$$

$$\forall m, c_m = \argmin_{c_m} \sum_{i \in R_m} (c_m - y_i)^2 = \text{average}(y_i \mid i \in R_m)$$

Apprendre un arbre de régression

- Avec la partition $\{R_1, \dots, R_M\}$, la prédiction finale est:

$$f(x) = \sum_{m=1}^M c_m 1(x \in R_m)$$

- Supposons qu'on ait déjà la partition, comment choisir c_1, \dots, c_M ?
- Pour la fonction de perte $\ell(\hat{y}, y) = (\hat{y} - y)^2$, comment appliquer l'ERM ?

$$\hat{c}_m = \text{average}(y_i \mid x_i \in R_m) = \frac{1}{\{i : x_i \in R_m\}} \sum_{\{i : x_i \in R_m\}} y_i$$

- car

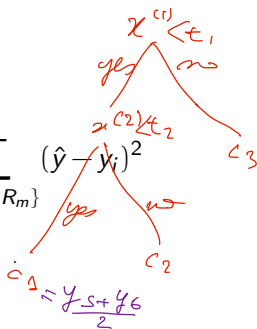
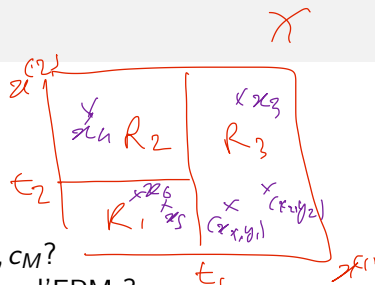
$$\text{average}(y_i \mid x_i \in R_m) = \arg \min_{\hat{y}} \sum_{\{i : x_i \in R_m\}} \ell(\hat{y}, y_i) = \arg \min_{\hat{y}} \sum_{\{i : x_i \in R_m\}} (\hat{y} - y_i)^2$$

- Quelle est la perte associée à ce noeud ?

for c_m ,

$$\sum_{i \in R_m} \ell(\hat{c}_m, y_i) = \sum_{i \in R_m} (y_i - \text{average}(y_i \mid i \in R_m))^2$$

$$= N_m \times \hat{\text{Var}}(\{y_i : i \in R_m\})$$



Apprendre un arbre de régression

- Avec la partition $\{R_1, \dots, R_M\}$, la prédiction finale est:

$$f(x) = \sum_{m=1}^M c_m 1(x \in R_m)$$

- Supposons qu'on ait déjà la partition, comment choisir c_1, \dots, c_M ?
- Pour la fonction de perte $\ell(\hat{y}, y) = (\hat{y} - y)^2$, comment appliquer l'ERM ?

$$\hat{c}_m = \text{average}(y_i \mid x_i \in R_m) = \frac{1}{\{i : x_i \in R_m\}} \sum_{\{i : x_i \in R_m\}} y_i$$

- car

$$\text{average}(y_i \mid x_i \in R_m) = \arg \min_{\hat{y}} \sum_{\{i : x_i \in R_m\}} \ell(\hat{y}, y_i) = \arg \min_{\hat{y}} \sum_{\{i : x_i \in R_m\}} (\hat{y} - y_i)^2$$

- Quelle est la perte associée à ce noeud ?

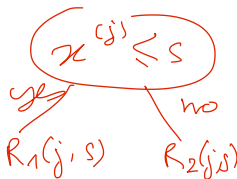
$$\sum_{\{i : x_i \in R_m\}} (\hat{c}_m - y_i)^2 = \sum_{\{i : x_i \in R_m\}} (\text{average}(y_j \mid x_j \in R_m) - y_i)^2 = N_m \cdot \hat{Var}(\{y_i : x_i \in R_m\})$$

Noeud racine, Variables réelles

- Soit $x = (x^{(1)}, \dots, x^{(d)}) \in \mathbb{R}^d$. (d variables)
- **variable de test** $x^{(j)}$.
- **seuil** $s \in \mathbb{R}$.
- Partition basée sur $x^{(j)}$ et s :

$$R_1(j, s) = \{x \mid x^{(j)} \leq s\}$$

$$R_2(j, s) = \{x \mid x^{(j)} > s\}$$

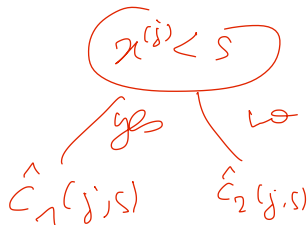


Noeud Racine, variables continues

- Pour chaque variable $x^{(j)}$ et seuil s ,

$$\hat{c}_1(j, s) = \text{average}(y_i \mid x_i \in R_1(j, s))$$

$$\hat{c}_2(j, s) = \text{average}(y_i \mid x_i \in R_2(j, s))$$



Noeud Racine, variables continues

- Pour chaque variable $x^{(j)}$ et seuil s ,

$$\hat{c}_1(j, s) = \text{average}(y_i \mid x_i \in R_1(j, s))$$

$$\hat{c}_2(j, s) = \text{average}(y_i \mid x_i \in R_2(j, s))$$

- Trouver j, s qui minimisent

$$\begin{aligned} L(j, s) &= \sum_{i: x_i \in R_1(j, s)} (y_i - \hat{c}_1(j, s))^2 + \sum_{i: x_i \in R_2(j, s)} (y_i - \hat{c}_2(j, s))^2 \\ &= N_1 \cdot \hat{Var}(\{y_i : x_i \in R_1(j, s)\}) + N_2 \cdot \hat{Var}(\{y_i : x_i \in R_2(j, s)\}) \end{aligned}$$

- Comment ?

Trouver le seuil

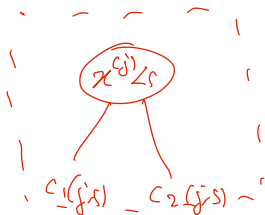
- Supposons qu'on choisisse la variable de test $x^{(j)}$.
- Supposons que $x_1^{(j)} \dots x_N^{(j)}$ soient triées en ordre croissant
 - traditionnellement, choisir le seuil entre deux valeurs consécutives:

$$s_j \in \left\{ \frac{1}{2} \left(x_i^{(j)} + x_{i+1}^{(j)} \right) \mid i = 1, \dots, \underline{N-1} \right\}.$$

- Donc, on teste $N-1$ seuils
- complexité pour trouver le noeud et le seuil..?

Naïve: for $j \in 1 \dots d$, for s_j , for $i = 1 \dots N \Rightarrow O(N^2 d)$

Improved: $O(d N \log N)$



Trouver le seuil


- Supposons qu'on choisisse la variable de test $x^{(j)}$.
- Supposons que $x_1^{(j)} \dots x_N^{(j)}$ soient triées en ordre croissant
 - traditionnellement, choisir le seuil entre deux valeurs consécutives:

$$s_j \in \left\{ \frac{1}{2} \left(x_i^{(j)} + x_{i+1}^{(j)} \right) \mid i = 1, \dots, N-1 \right\}.$$

- Donc, on teste $N-1$ seuils
- complexité pour trouver le noeud et le seuil..?
- $O(dN^2)$

Continuer l'apprentissage de l'arbre récursivement

1) for all j, s , eval $\pi^{(j,s)}$
2) keep the best j, s



- 1 On a déterminé R_1 et R_2 , on a donc un arbre avec une racine et 2 feuilles. On continue
 - 2 Choisir le meilleur split (j, s) dans R_1
 - 3 Choisir le meilleur split (j', s') dans R_2
 - 4 Continuer...
- Quand s'arrête-t-on ?



Controler la complexité de l'arbre

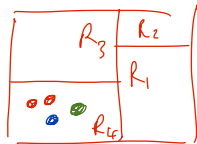
- Si l'arbre est trop grand \Rightarrow chaque exemple x_i possèdera sa propre partition \Rightarrow Surapprentissage.
- Si trop petit, underfitting
- On peut limiter la profondeur de l'arbre
- On peut ne poursuivre les divisions que sur les noeuds contenant un nombre minimum d'exemples.
- On peut faire de l'élagage à postériori (ex: **CART**, Breiman et al 1984):
 - ① Construire un arbre très grand et profond (ex. jusqu'à ce que chaque région ait ≤ 5 points).
 - ② **Elaguer** l'arbre de façon gloutonne, du bas vers le haut, tant que la performance de l'arbre estimée sur un ensemble de test ne décroît pas.

Arbres de décision pour la Classification

Arbres pour la Classification (avec perte 0/1)

- Soit $\mathcal{Y} = \{1 \dots K\}$. Même raisonnement qu'avant: on suppose qu'on a déjà les régions R_i
- Le noeud m représente la région R_m , avec N_m exemples
- La proportion d'exemple de classe $k \in \mathcal{Y}$ dans R_m est

$$\hat{\eta}_{m,k} = \hat{P}(Y = k | X \in R_m) = \frac{1}{N_m} \sum_{\{i: x_i \in R_m\}} 1(y_i = k).$$



$$\hat{\eta}_{4, \text{red}} = \frac{1}{2}$$

- Si on prédit au noeud m la classe k , alors le taux d'erreur sur les exemples d'apprentissage de R_m sera (à tracer au tableau)

$$1 - \hat{\eta}_{m,k}$$

$$\hat{\eta}_{4, \text{blue}} = \frac{1}{4}$$

$C_4 = \text{red} \Rightarrow \text{error rate in } R_4$
is $1 - \frac{1}{2}$

- Donc pour minimiser le taux d'erreur (perte 0/1), la classe prédite pour le noeud m sera

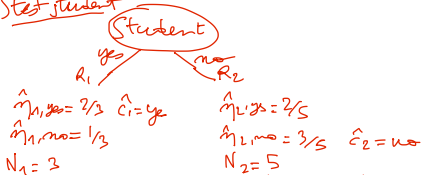
$$\hat{y}(m) = \arg \min_k 1 - \hat{\eta}_{m,k} = \arg \max_k \hat{\eta}_{m,k}$$

predict the majority class

Exemple (sous forme d'exercice)

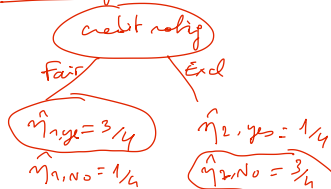
Student	Credit Rating	Class: Buy PDA
No	Fair	No
No	Excellent	No
No	Fair	Yes
No	Fair	Yes
Yes	Fair	Yes
Yes	Excellent	No
Yes	Excellent	Yes
No	Excellent	No

1) test student



$$\text{Error rate} = \frac{N_1}{N} \times \frac{1}{3} + \frac{N_2}{N} \times \frac{2}{5} = \frac{3}{8}$$

2) test rating



\Rightarrow Test Rating is the Best

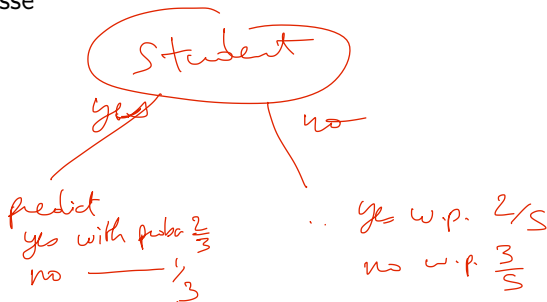
$$\Rightarrow \text{error rate} = \frac{1}{4}$$

Exemple (sous forme d'exercice)

Arbres de décision pour la prédiction de densité de classes (CP loss)

- **But:** apprendre un “soft”-classifieur, prédisant des probabilités de classe: $f : \mathcal{X} \rightarrow \Delta_K$, où Δ_K est le simplexe de probabilités de dimension K .
- **Exercice:** dessinez un arbre de décision comportant un seul noeud, proposant des prédictions sur les probabilités de classe

Student	Credit Rating	Class: Buy PDA
No	Fair	No
No	Excellent	No
No	Fair	Yes
No	Fair	Yes
Yes	Fair	Yes
Yes	Excellent	No
Yes	Excellent	Yes
No	Excellent	No



Rappel: En régression, on avait:

- Avec la partition $\{R_1, \dots, R_M\}$, la prédiction finale est:

$$f(x) = \sum_{m=1}^M c_m 1(x \in R_m)$$

- Supposons qu'on ait déjà la partition, comment choisir c_1, \dots, c_M ?
- Pour la fonction de perte $\ell(\hat{y}, y) = (\hat{y} - y)^2$, l'ERM propose:

$$\hat{c}_m = \arg \min_{\hat{y}} \sum_{\{i: x_i \in R_m\}} \ell(\hat{y}, y_i) = \arg \min_{\hat{y}} \sum_{\{i: x_i \in R_m\}} (\hat{y} - y_i)^2 = \text{average}(y_i \mid x_i \in R_m).$$

- et la perte associée à ce noeud est $\sum (\hat{c}_m - y_i)^2$

Avec une perte à probabilité de classe, on a...

- Plaçons nous dans le cas $\mathcal{Y} = \{0, 1\}$. Le noeud m représente la region R_m , avec N_m exemples
- La proportion d'exemple de classe ~~$k \in \mathcal{Y}$~~ dans R_m est

$$\hat{\eta}_m = \hat{P}(Y = 1 | X \in R_m) = \frac{1}{N_m} \sum_{\{i: x_i \in R_m\}} 1(y_i = 1).$$

- l'ERM nous propose la prédiction suivante pour la classe 1 :
 $\hat{c}_m = \arg \min_{\hat{y}} \sum_{\{i: x_i \in R_m\}} \ell(\hat{y}, y_i) = \arg \min_{\hat{y}} N_m \times (\hat{\eta}_m \ell(\hat{y}, 1) + (1 - \hat{\eta}_m) \ell(\hat{y}, 0))$
- Si la fonction de perte est une CP-loss (estimation de probabilités conditionnelles) propre (ex: cross-entropie) alors..
If the loss is proper then, $\hat{c}_m = \hat{\eta}_m$

Avec une perte à probabilité de classe, on a...

- Plaçons nous dans le cas $\mathcal{Y} = \{0, 1\}$. Le noeud m représente la region R_m , avec N_m exemples
- La proportion d'exemple de classe $k \in \mathcal{Y}$ dans R_m est

$$\hat{\eta}_m = \hat{P}(Y = 1 \mid X \in R_m) = \frac{1}{N_m} \sum_{\{i: x_i \in R_m\}} 1(y_i = 1).$$

- l'ERM nous propose la prédiction suivante pour la classe 1 :
 $\hat{c}_m = \arg \min_{\hat{y}} \sum_{\{i: x_i \in R_m\}} \ell(\hat{y}, y_i)$
- Si la fonction de perte est une CP-loss (estimation de probabilités conditionnelles) *propre* (ex: cross-entropie) alors..

$$\hat{c}_m = \hat{\eta}_m$$

- La valeur de la perte en R_m sera alors...?

Avec une perte à probabilité de classe, on a...


- Plaçons nous dans le cas $\mathcal{Y} = \{0, 1\}$. Le noeud m représente la region R_m , avec N_m exemples
- La proportion d'exemple de classe $k \in \mathcal{Y}$ dans R_m est

$$\hat{\eta}_m = \hat{P}(Y = 1 \mid X \in R_m) = \frac{1}{N_m} \sum_{\{i: x_i \in R_m\}} 1(y_i = 1).$$

- l'ERM nous propose la prédiction suivante pour la classe 1 :
 $\hat{c}_m = \arg \min_{\hat{y}} \sum_{\{i: x_i \in R_m\}} \ell(\hat{y}, y_i)$
- Si la fonction de perte est une CP-loss (estimation de probabilités conditionnelles) *propre* (ex: cross-entropie) alors..

$$\hat{c}_m = \hat{\eta}_m$$

- La valeur de la perte en R_m sera alors...?

$$\sum_{\{i: x_i \in R_m\}} \ell(\hat{\eta}_m, y_i)$$


Avec une perte à probabilité de classe, on a...

- Plaçons nous dans le cas $\mathcal{Y} = \{0, 1\}$. Le noeud m représente la region R_m , avec N_m exemples
- La valeur de la perte en R_m sera alors

$$\sum_{\{i: x_i \in R_m\}} \ell(\hat{\eta}_m, y_i)$$

- Pour la cross-entropy $\ell(\hat{\eta}, y) = -y \log \hat{\eta} - (1 - y) \log (1 - \hat{\eta})$, cette valeur sera ...?

Avec une perte à probabilité de classe, on a...

- Plaçons nous dans le cas $\mathcal{Y} = \{0, 1\}$. Le noeud m représente la region R_m , avec N_m exemples
- La valeur de la perte en R_m sera alors

$$\sum_{\{i: x_i \in R_m\}} \ell(\hat{\eta}_m, y_i) = N_m \times \hat{\eta}_m \times \ell(\hat{\eta}_m, 1) + N_m \times (1 - \hat{\eta}_m) \times \ell(\hat{\eta}_m, 0)$$

Handwritten notes: $-\log \hat{\eta}_m$ (above the first term), $-\log(1 - \hat{\eta}_m)$ (above the second term)

- Pour la cross-entropy $\ell(\hat{\eta}, y) = -y \log \hat{\eta} - (1 - y) \log(1 - \hat{\eta})$, cette valeur sera ...? *Handwritten note: $-\log(1 - \hat{\eta}_m)$*
- $N_m \cdot H_\ell = -N_m \cdot (\hat{\eta}_m \log \hat{\eta}_m + (1 - \hat{\eta}_m) \log(1 - \hat{\eta}_m)) = \text{Entropie de Shannon.}$
- Si on cherche à prédire une classe plutôt qu'une probabilité, on prendra $\hat{y}(R_m) = 1$ si $\hat{\eta}_m > \frac{1}{2}$, 0 sinon
- Remarque: pour toute perte PC ℓ , la valeur de cette perte dans la région R_m est l'entropie généralisée $H_\ell(\hat{\eta}_m)$.

Two-Class Node Impurity Measures

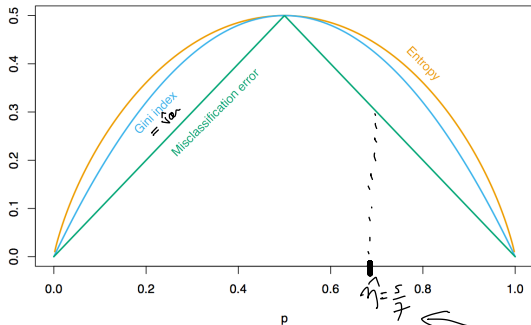
- Consider binary classification

Two-Class Node Impurity Measures

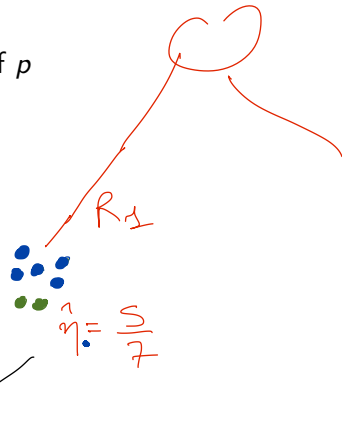
- Consider binary classification
- Let p be the relative frequency of class 1.

Two-Class Node Impurity Measures

- Consider binary classification
- Let p be the relative frequency of class 1.
- Here are three node impurity measures as a function of p



HTF Figure 9.3



Classification Trees: Node Impurity Measures

- Consider leaf node m representing region R_m , with N_m observations

Classification Trees: Node Impurity Measures

- Consider leaf node m representing region R_m , with N_m observations
- Three measures $Q_m(T)$ of **node impurity** for leaf node m :
 - Misclassification error:

$$H_{0/1}(\hat{\eta}) = \min_k 1 - \hat{\eta}_{m,k}.$$

Classification Trees: Node Impurity Measures

- Consider leaf node m representing region R_m , with N_m observations
- Three measures $Q_m(T)$ of **node impurity** for leaf node m :
 - Misclassification error:

$$H_{0/1}(\hat{\eta}) = \min_k 1 - \hat{\eta}_{m,k}.$$

- Gini index:

$$H_{Gini}(\hat{\eta}) = \sum_{k=1}^K \hat{\eta}_{m,k}(1 - \hat{\eta}_{m,k})$$

Classification Trees: Node Impurity Measures

- Consider leaf node m representing region R_m , with N_m observations
- Three measures $Q_m(T)$ of **node impurity** for leaf node m :
 - Misclassification error:

$$H_{0/1}(\hat{\eta}) = \min_k 1 - \hat{\eta}_{m,k}.$$

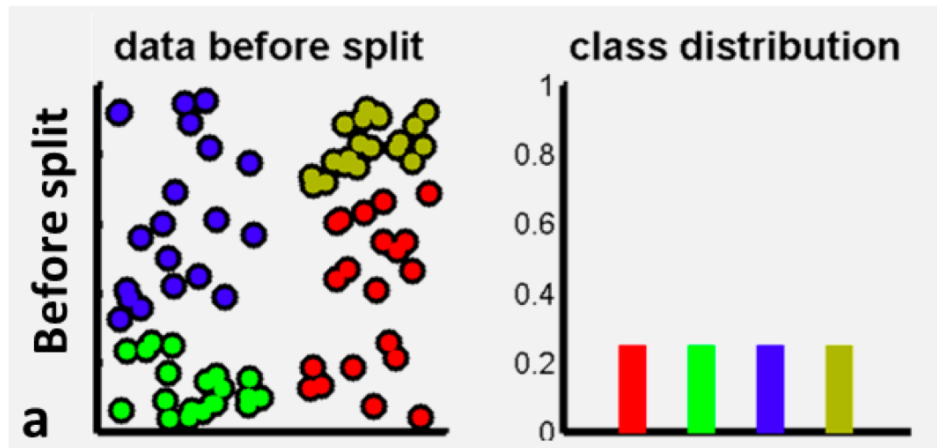
- Gini index:

$$H_{Gini}(\hat{\eta}) = \sum_{k=1}^K \hat{\eta}_{m,k}(1 - \hat{\eta}_{m,k})$$

- Entropy or deviance (equivalent to using information gain):

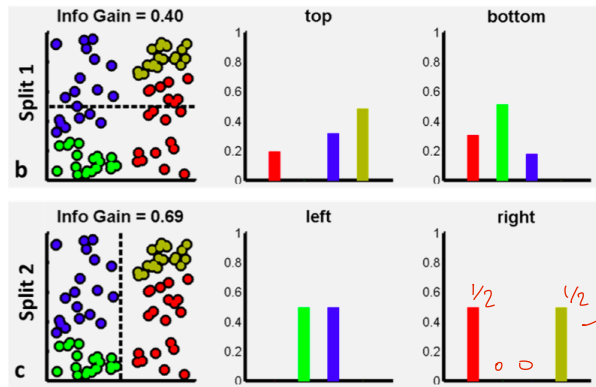
$$H_{CE}(\hat{\eta}) = - \sum_{k=1}^K \hat{\eta}_{m,k} \log \hat{\eta}_{m,k}.$$

Class Distributions: Pre-split



From Criminisi et al. MSR-TR-2011-114, 28 October 2011.

Class Distributions: Split Search



for k classes:

$$H(\hat{\pi}_1, \dots, \hat{\pi}_k) = -\sum_{k=1}^K \hat{\pi}_k \log_2 \hat{\pi}_k$$

initial class distribution = $(\frac{1}{4}, \dots, \frac{1}{4})$

$$\text{entropy} = -\log_2 \frac{1}{4} = 2$$

(impurity)

$$\text{entropy} = \left(-\frac{1}{2} \log_2 \frac{1}{2}\right) \times 2 + 0 \times 2$$
$$= \log_2 2 = 1$$

(Maximizing information gain is equivalent to minimizing entropy.)

arg entropy of this tree $\frac{1}{2} \times 1 + \frac{1}{2} \times 1 = 1$ is

Splitting nodes: How exactly do we do this?

- Let R_1 and R_2 be regions corresponding to a potential node split.

Splitting nodes: How exactly do we do this?

- Let R_1 and R_2 be regions corresponding to a potential node split.
- Suppose we have N_1 points in R_1 and N_2 points in R_2 .

Splitting nodes: How exactly do we do this?

- Let R_1 and R_2 be regions corresponding to a potential node split.
- Suppose we have N_1 points in R_1 and N_2 points in R_2 .
- Let $H_\ell(R_1)$ and $H_\ell(R_2)$ be generalized entropy (the node impurity measures)

Splitting nodes: How exactly do we do this?

- Let R_1 and R_2 be regions corresponding to a potential node split.
- Suppose we have N_1 points in R_1 and N_2 points in R_2 .
- Let $H_\ell(R_1)$ and $H_\ell(R_2)$ be generalized entropy (the node impurity measures)
- Then find split that minimizes the **weighted average of node impurities**:

$$\text{avg entropy} = \frac{N_1}{N} H(R_1) + \frac{N_2}{N} H(R_2)$$



Classification Trees: Node Impurity Measures

- For building the tree, Gini and Entropy seem to be more effective.
- They push for more pure nodes, not just misclassification rate
- A good split may not change misclassification rate at all!

Classification Trees: Node Impurity Measures

- For building the tree, Gini and Entropy seem to be more effective.
- They push for more pure nodes, not just misclassification rate
- A good split may not change misclassification rate at all!
- Two class problem: 4 observations in each class.
- Split 1: (3,1) and (1,3) [each region has 3 of one class and 1 of other]
- Split 2: (2,4) and (2,0) [one region has 2 of one class and 4 of other, other region pure]
- Misclassification rate for two splits are same.
- **Gini and entropy split prefer Split 2.**