Departement d'informatique

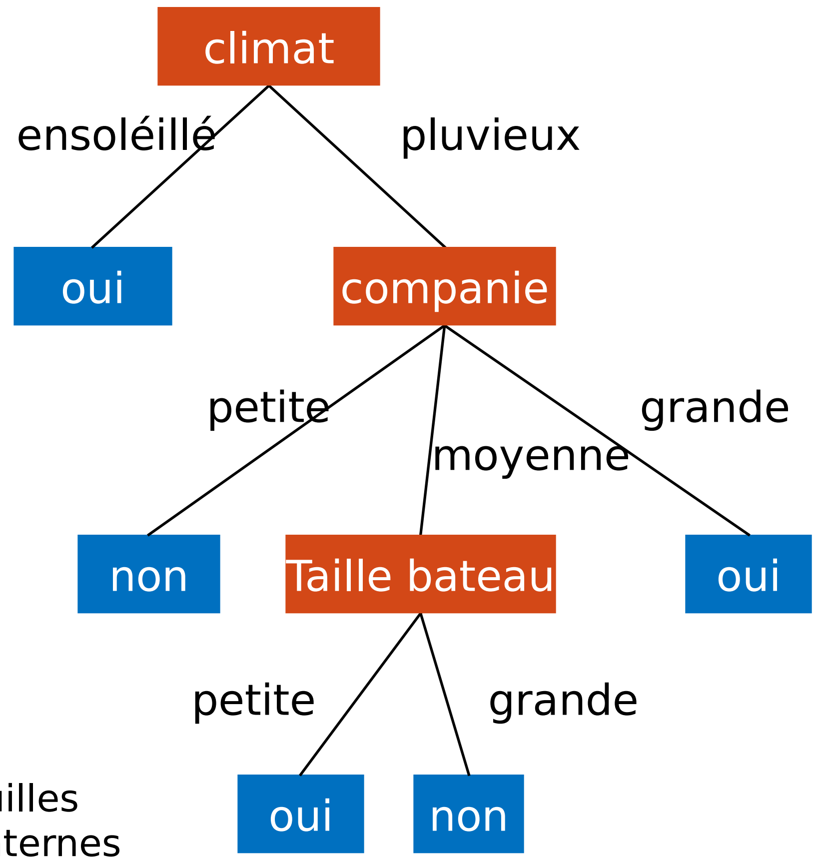
Les Arbres de Decision (Partie 1)

Master 1 MID S2

Enseignant:
Hadjila Fethallah

Exemple d’arbre de décison

#	*Attribut*			*Classe*
	climat	Companie	taille-bateau	naviguer?
1	ensoléillé	grande	petite	oui
2	ensoléillé	moyenne	petite	oui
3	ensoléillé	moyenne	grande	oui
4	ensoléillé	petite	petite	oui
5	ensoléillé	grande	grande	oui
6	pluvieux	petite	petite	non
7	pluvieux	moyenne	petite	oui
8	pluvieux	grande	grande	oui
9	pluvieux	petite	grande	non
10	pluvieux	moyenne	grande	non



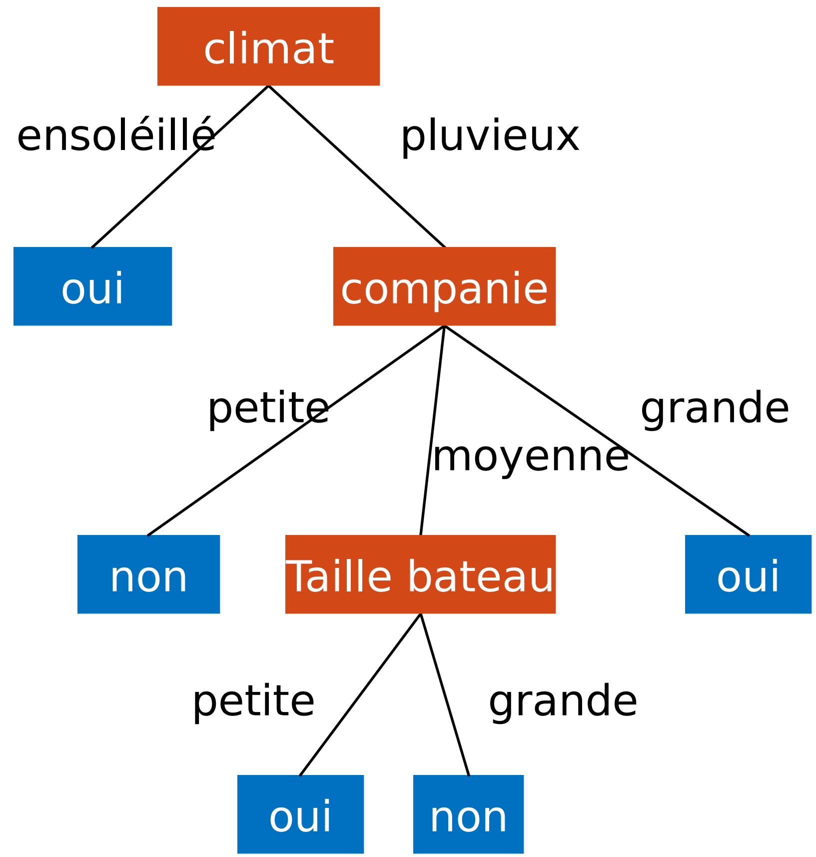
Definition:
A.D est un classifieur en forme d’arbre , ses feuilles représentent les classes de sorties, les nœuds internes représentent les tests à exécuter sur un attribut, et les branches sont étiquetées avec les valeurs du test

Motivations

- Traitement de données nominales
  - Ex: Variable Métier ={footballeur, enseignant, étudiant, recteur...}
- Traitement de données discrètes et continues
- L'utilisation de plusieurs arbres de décision simples peut améliorer les performance de généralisation (réduction de la variance)
- La connaissance induite par l'arbre de décision peut être représentée avec des règles de production .
- Disjonction de règles (une règle est une conjonction de conditions).
- Grande capacité d'interprétation
- Classifieur de nature supervisée

test

#	Attribut			Classe
	climat	Companie	taille bateau	naviguer?
1	ensolleillé	petite	grande	?
2	pluvieux	grande	petite	?



semantique de l'arbre de decision

- ✓ chaque noeud interne représente
  - o Une partie de la base d'apprentissage
  - o L' attribut de division courant
- ✓ Chaque arc est etiqueté avec la valeur de l'attribut de division

Difficultés

- Le choix de la variable de division (quelle heuristique)
- Le traitement des variables continues.
- Le traitement des valeurs manquantes.
- Le traitement des variables (attributs) ayant des couts différents.
- La Presence de données bruitées ou conflictuelles
- La determination du nombre optimal de noeuds
  - (methodes d’élagage de l’arbre)

l'apprentissage

- Base d'apprentissage etiquetée
  - Chaque example est caractérisé par ses attributs + sa Classe
- Principe d'induction des arbres de decision
  - Minimiser l'impurté des noeuds.
  - Partionnement recursive

Formalisation de l'apprentissage

- Algorithme d'apprentissage general
- Entrée: base etiquetée
- Sortie: arbre de decision (AD)

1. noeud-courant= racine de l'arbre (tous les exemples)
2. Initialiser (AD, noeud-courant)
3. Repeter
    1. Decider si Noeud-courant est un noeud terminal
        a) Si oui affecter cette feuille à une classe donnée
        b) Sinon : choisir un attribut de division selon la mesure d'impurté et creer les enfants de Noeud-courant et mettre à jour AD
    2. MAJ du noeud courant (ie passer aux enfants non explorés)
4. Jusqu'à ce que (**plus d'enfants à diviser** ou **plus d'attributs de division)**
5. Retourner AD

Algorithmes d'apprentissage

- ID3 (Quinlan 79)
- CHAID (Kass 80)
- CART (Brieman et al. 84)
- Assistant (Cestnik et al. 87)
- C4.5 (Quinlan 93)
- C 5.0, See5 (Quinlan 97)
- …

Critère de Selection d' Attribut

- principe
  - Selectionner l'attribut qui maximise la purté des enfants
- Plusieurs mesures d'impurté
  - Entropie (adoptée par ID3)
  - Index de Gini (adopté par CART)
  - X^2 (adopté par de CHAID)
  - ReliefF
  - ...
- Plusieurs ameliorations possibles

Algorithme ID3 (Quinlan 79)

- Signifie 3eme serie de “Iterative dichotomizer”
- Utilise uniquement les variables discretes
- Les variables continues doivent etre discretisées avec des intervalles
- La mesure d’impurté employée est celle de l’entropie
- Pas de procedure d’élagage (possibilité d’overfiting)

Algorithme ID3 (Quinlan 79)

- Entrée :base d'apprentissage S
- Sortie: arbre de decision noté AD

1. Si S est vide alors retourner un noeud nommé "echec"

   Sinon

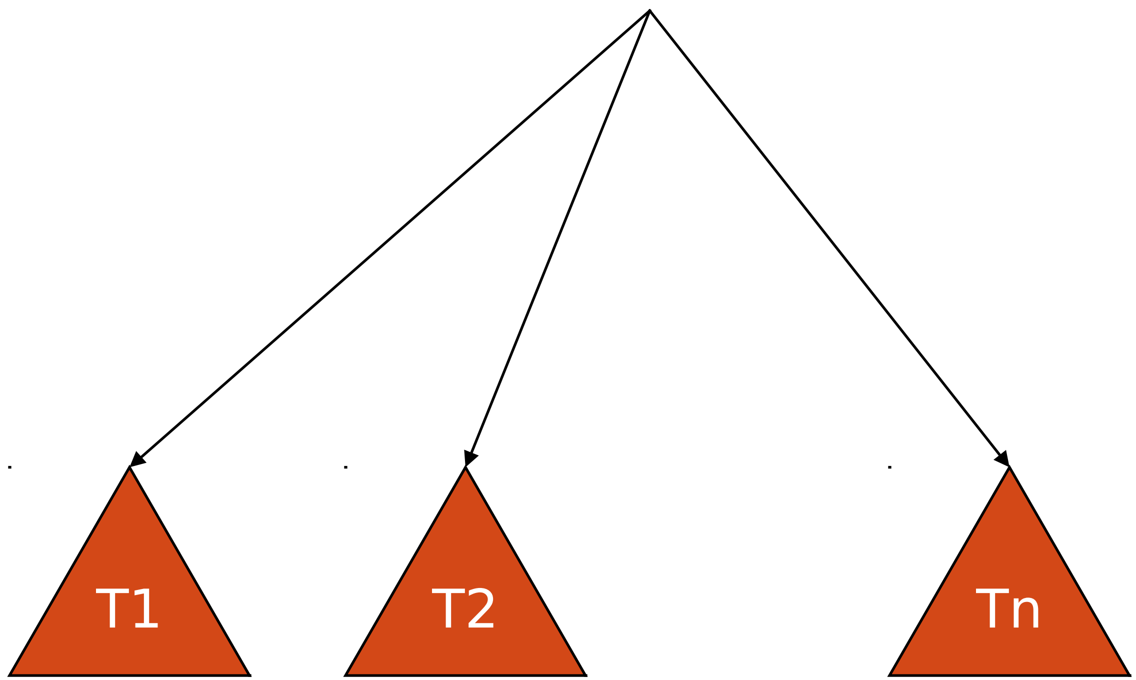
   1. Si S contient des exemples de meme classe alors retourner un seule feuille contenant la valeur de cette classe
   2. Si l'ens des attributs est vide
      1. alors retourner un seule feuille etiquetée par la classe majoritaire

      Sinon

      1. Choisir un attribut qui maximise le gain d'information
      2. Pour chaque enfant i du test precedant : SADi=ID3(i)
      3. Mettre à jour AD (affecter le test au parent et ajouter les sous arbres SADi à AD)

Retourner (AD)

Algorithme ID3



Approche à base de théorie d’information

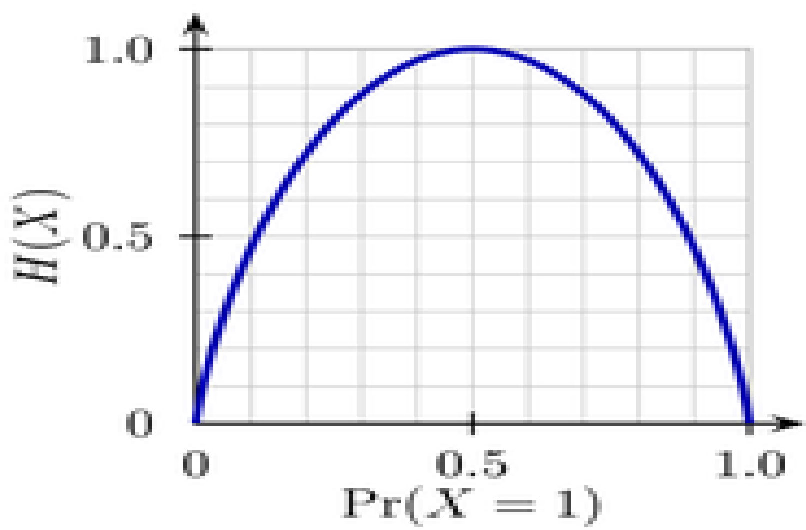
- Pour classer un objet on aura besoin d’une certaine quantité d’information (nombre de bits)
  - I represente cette quantité d’information
  - Apres l’utilisation de l’attribut A, on a besoin seulement d’une quantité d’information restante (ou résiduelle) pour classer l’objet
  - Ires, represente la quantité d’information résiduelle
- Gain
  - Gain(A) = I – Ires(A)
- L’attribut le plus ‘informatif’ est celui qui minimise Ires, *i.e.*, (ou maximise le Gain)

Entropie

- C'est le degré de désordre d'un ensemble
- C'est la quantité d'information moyenne que l'on a besoin pour coder les elements d'un ensemble

$$I = -\sum_{c} p(c) \log_2 p(c)$$

- Plus l'evenement est sur, plus sa quantité d'information est faible et vice versa.
- Pour un problème binaire (l'entropie de l'ensemb



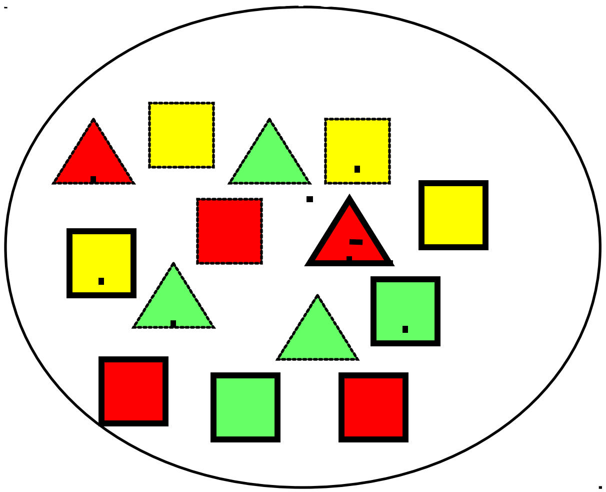
Information Residuelle

- Aprés l’application de *A*, *S* est partitionné en *v* sous ensembles (*v* est le nombre de valeurs de *A)*
- *Ires* =la moyenne des quantités d’information des enfants

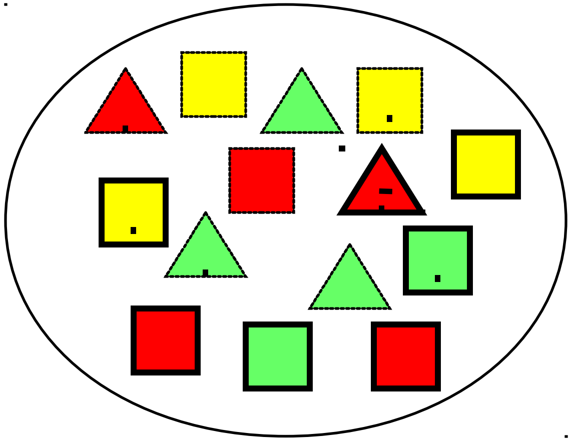
$$I_{res} = -\sum_{v} p(v) \sum_{c} p(c|v) \log_2 p(c|v)$$

Triangles et carrés

#	Attribut			forme
	Couleur	trait	point	
1	vert	pointillé	non	triangle
2	vert	pointillé	oui	triangle
3	yellow	pointillé	non	carré
4	rouge	pointillé	non	carré
5	rouge	plein	non	carré
6	rouge	plein	oui	triangle
7	vert	plein	non	carré
8	vert	pointillé	non	triangle
9	yellow	plein	oui	carré
10	rouge	plein	non	carré
11	vert	plein	oui	carré
12	yellow	pointillé	oui	carré
13	yellow	plein	non	carré
14	rouge	pointillé	oui	triangle



Entropie



- 5 triangles
- 9 carrés
- Probabilités des classes

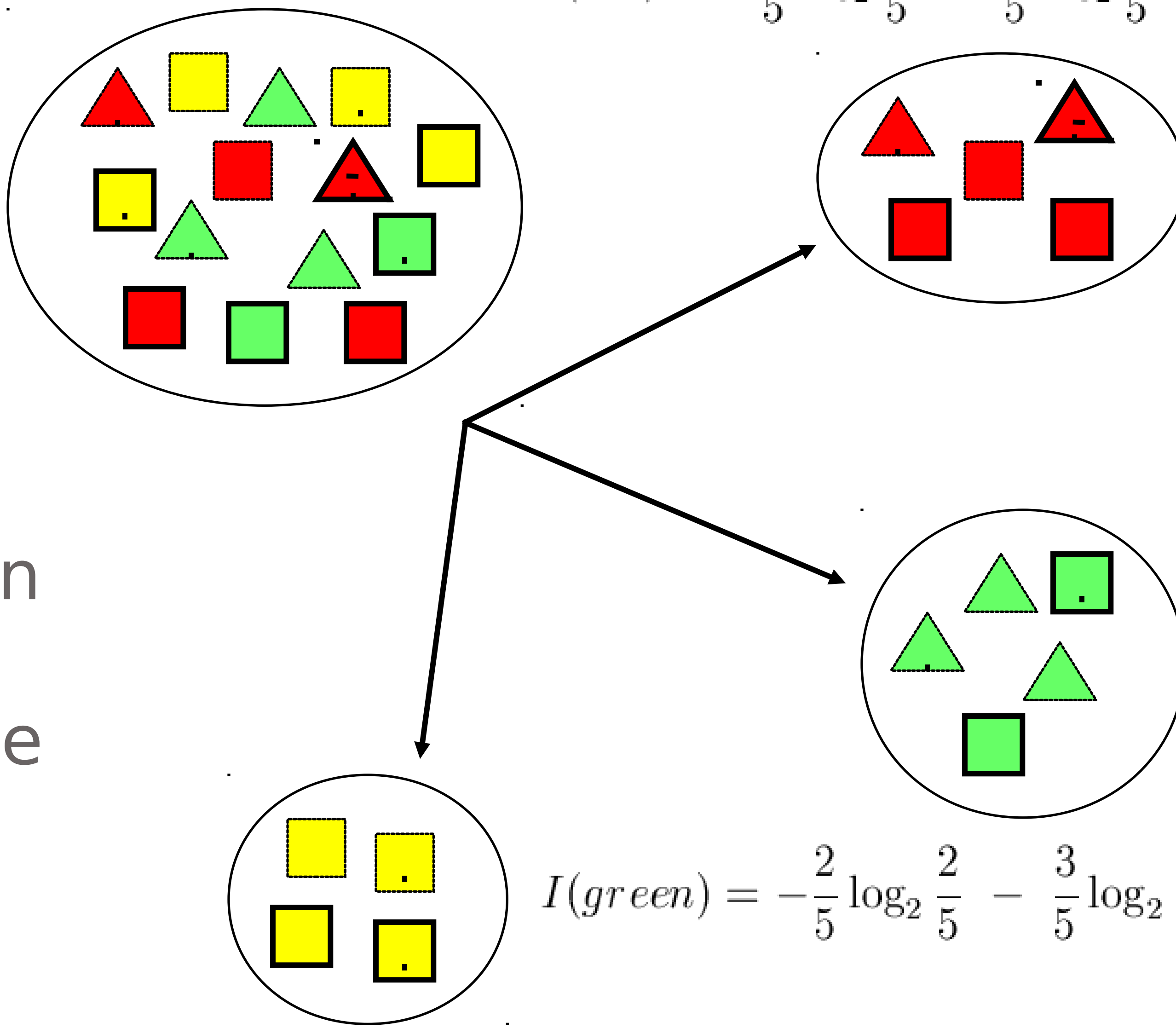
$$p(\square) = \frac{9}{14}$$

$$p(\triangle) = \frac{5}{14}$$

- entropie

$$I = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940 \text{ bits}$$

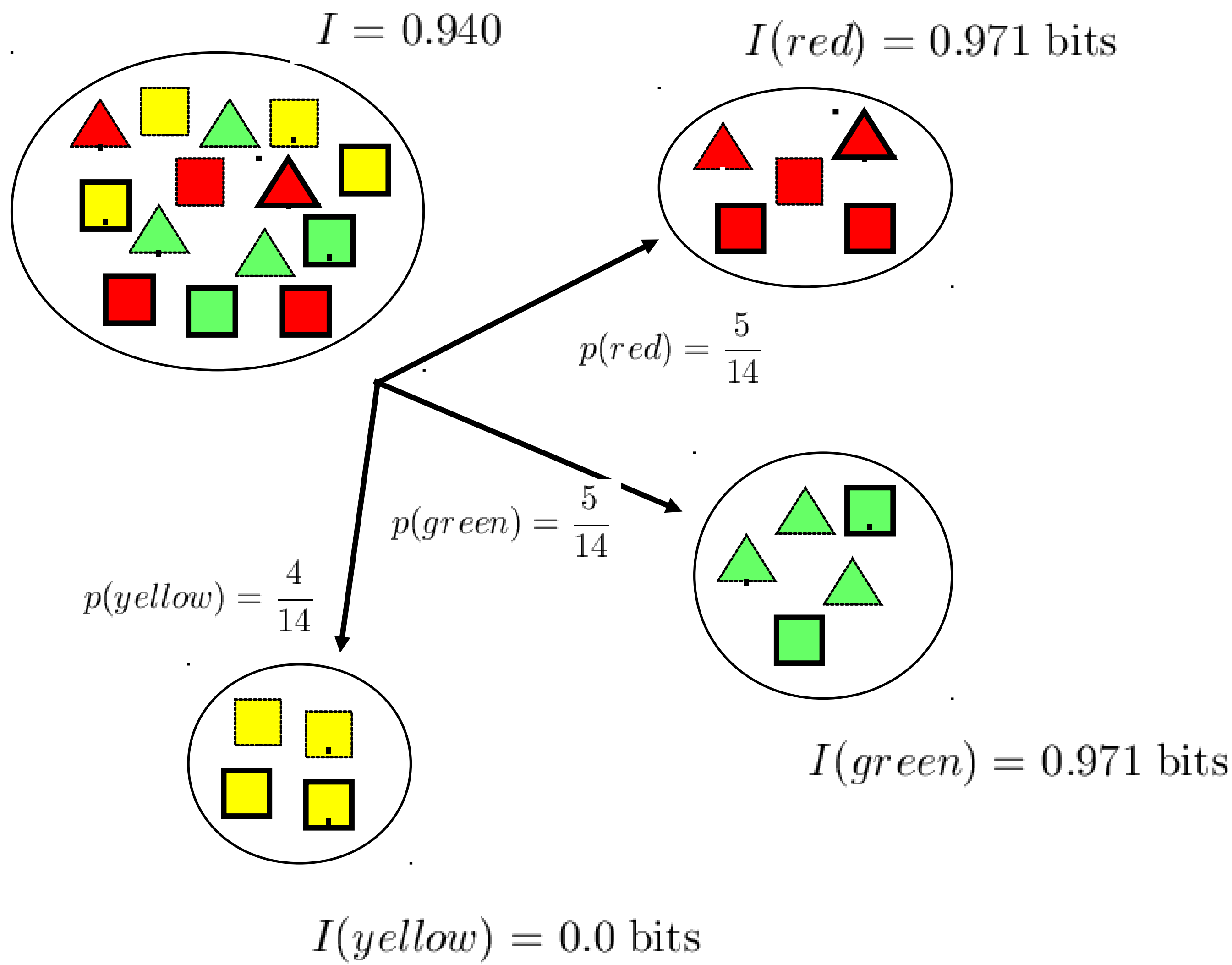
$$I(red) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971 \text{ bits}$$



reduction de l’entropie

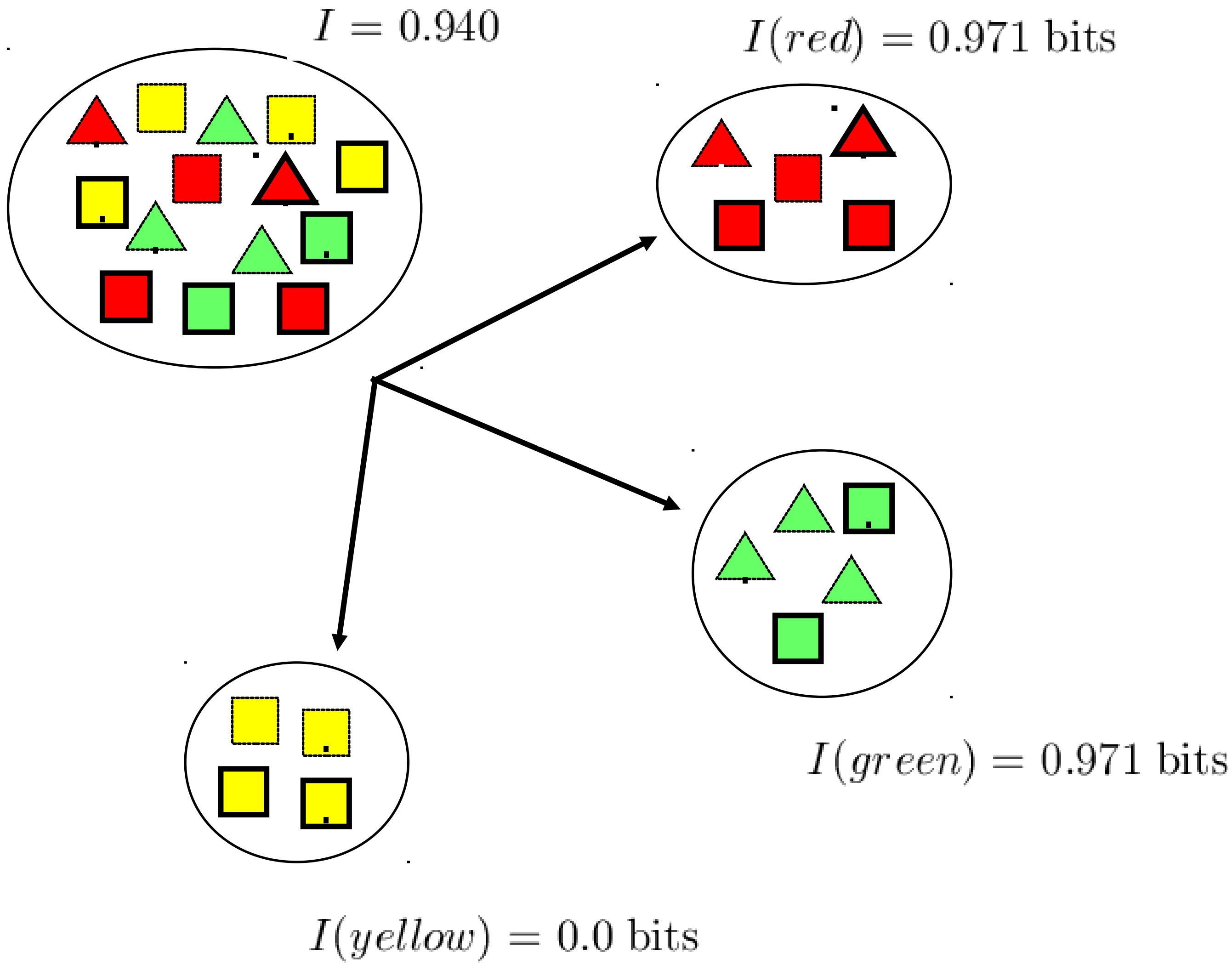
$$I(green) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971 \text{ bits}$$

$$I(yellow) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0.0 \text{ bits}$$



$$I_{res}(\text{Color}) = \sum p(v) I(v) = \frac{5}{14} 0.971 + \frac{5}{14} 0.971 + \frac{4}{14} 0.0 = 0.694\ bits$$

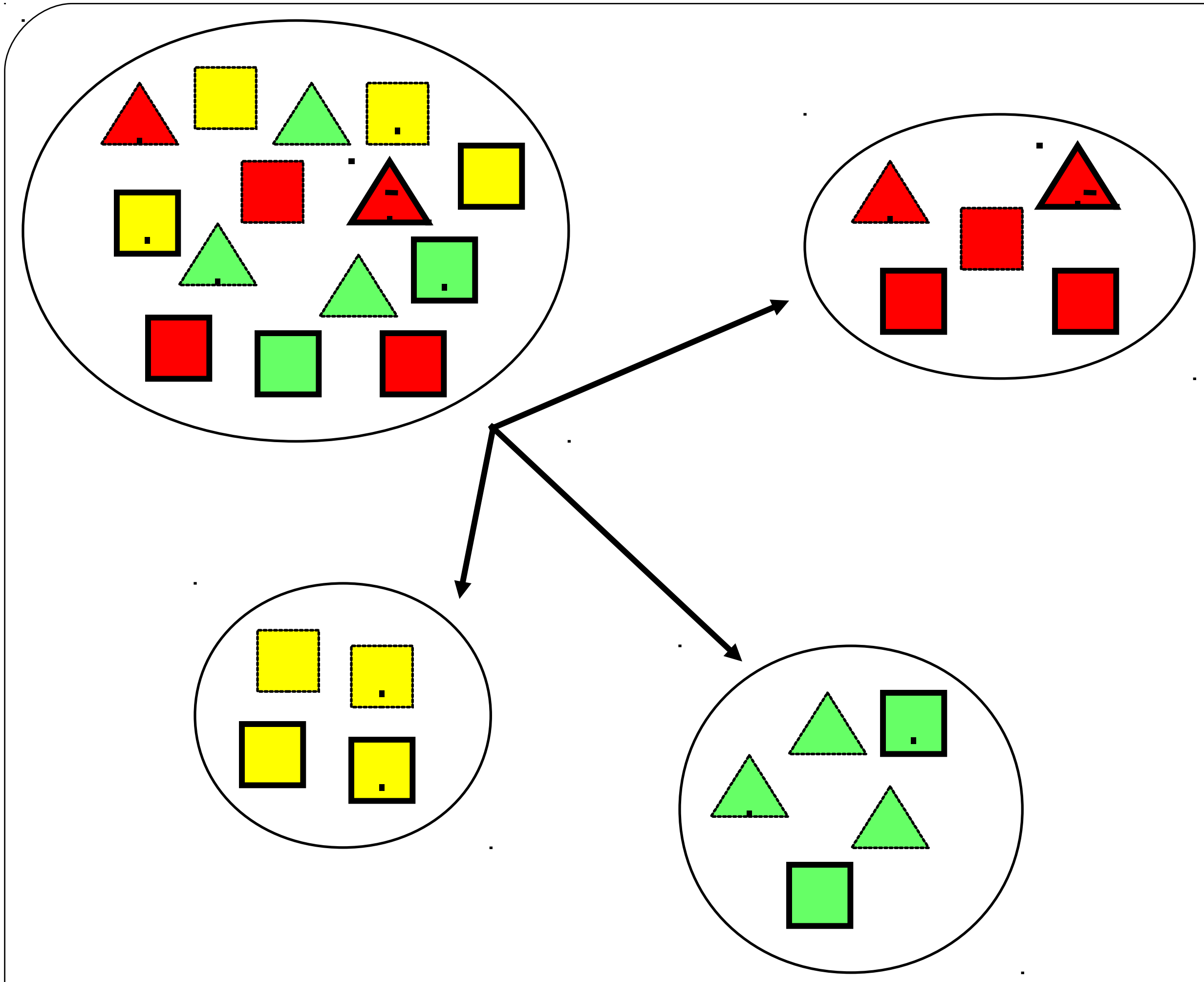
Gain d'information



$$\boxed{Gain(\text{Color}) = I - I_{res}(\text{Color}) = 0.940 - 0.694 = 0.246\ bits}$$

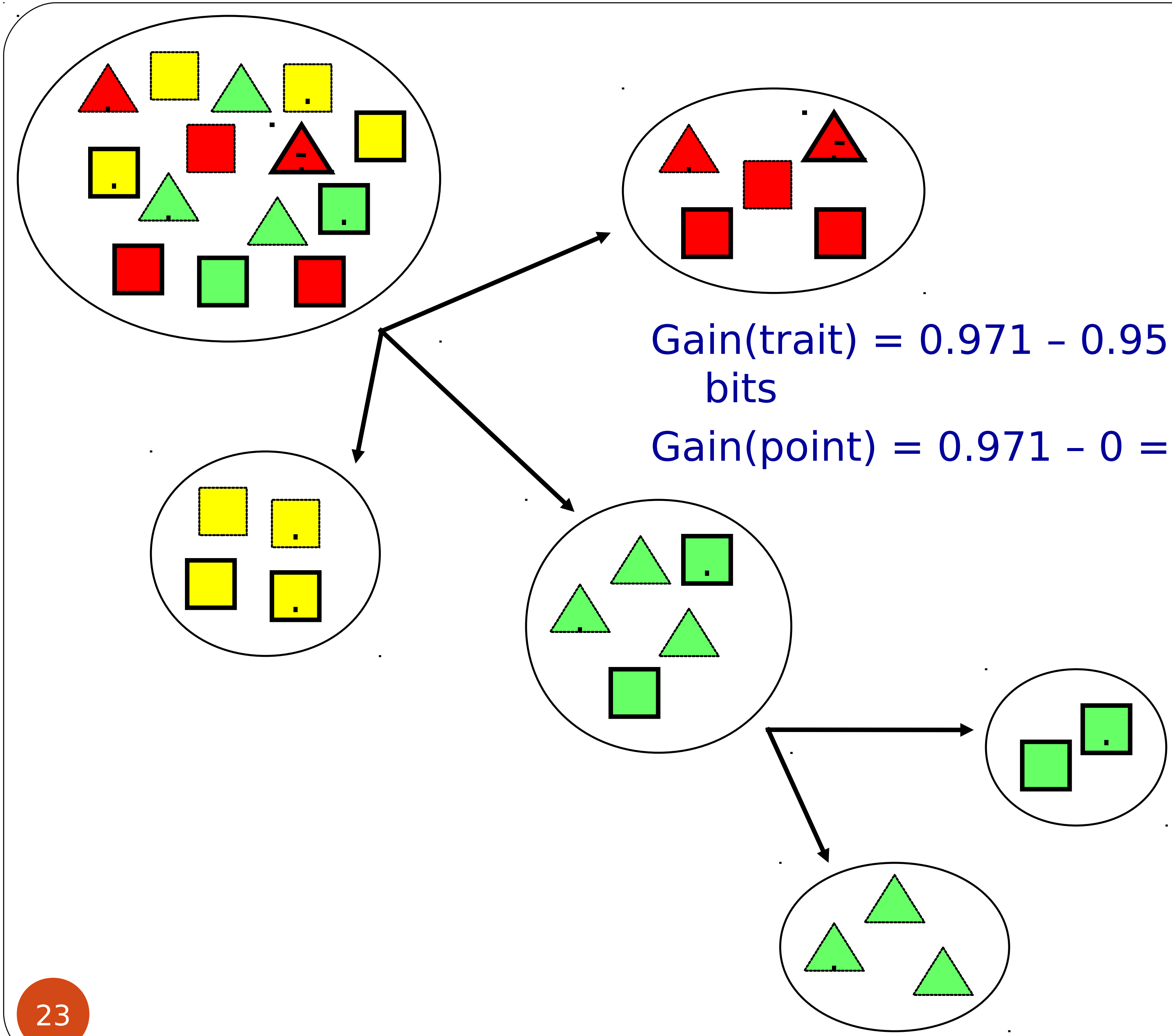
Information Gain of The Attribute

- Attributs
  - Gain(Couleur) = 0.246
  - Gain(trait) = 0.151
  - Gain(point) = 0.048
- Heuristique: on choisit l’attribut ayant le plus grand gain
- cette heuristique est locale (minimisation locale de l’ impurté)



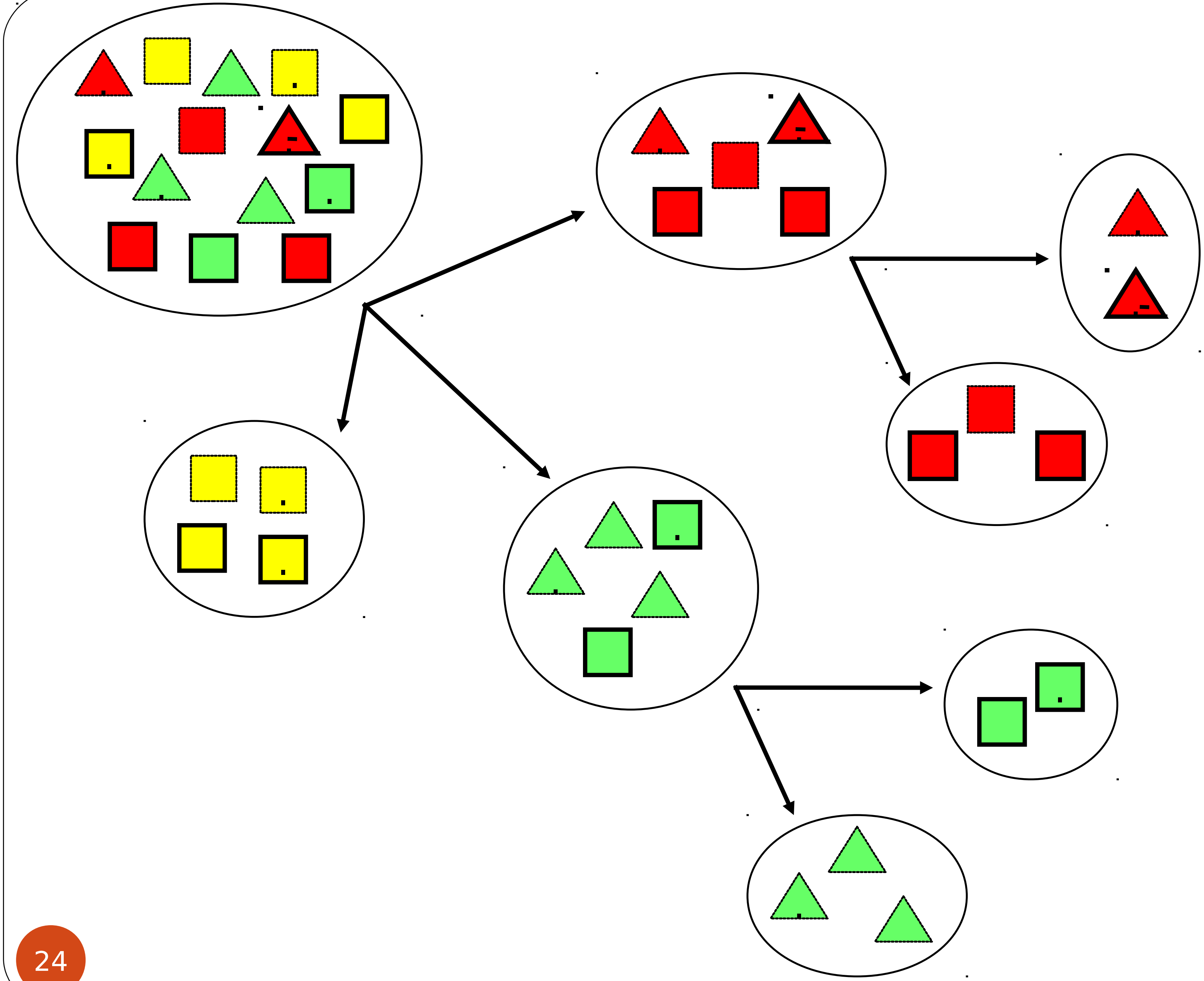
Gain(trait) = 0.971 – 0 = 0.971 bits

Gain(point) = 0.971 – 0.951 = 0.020 bits

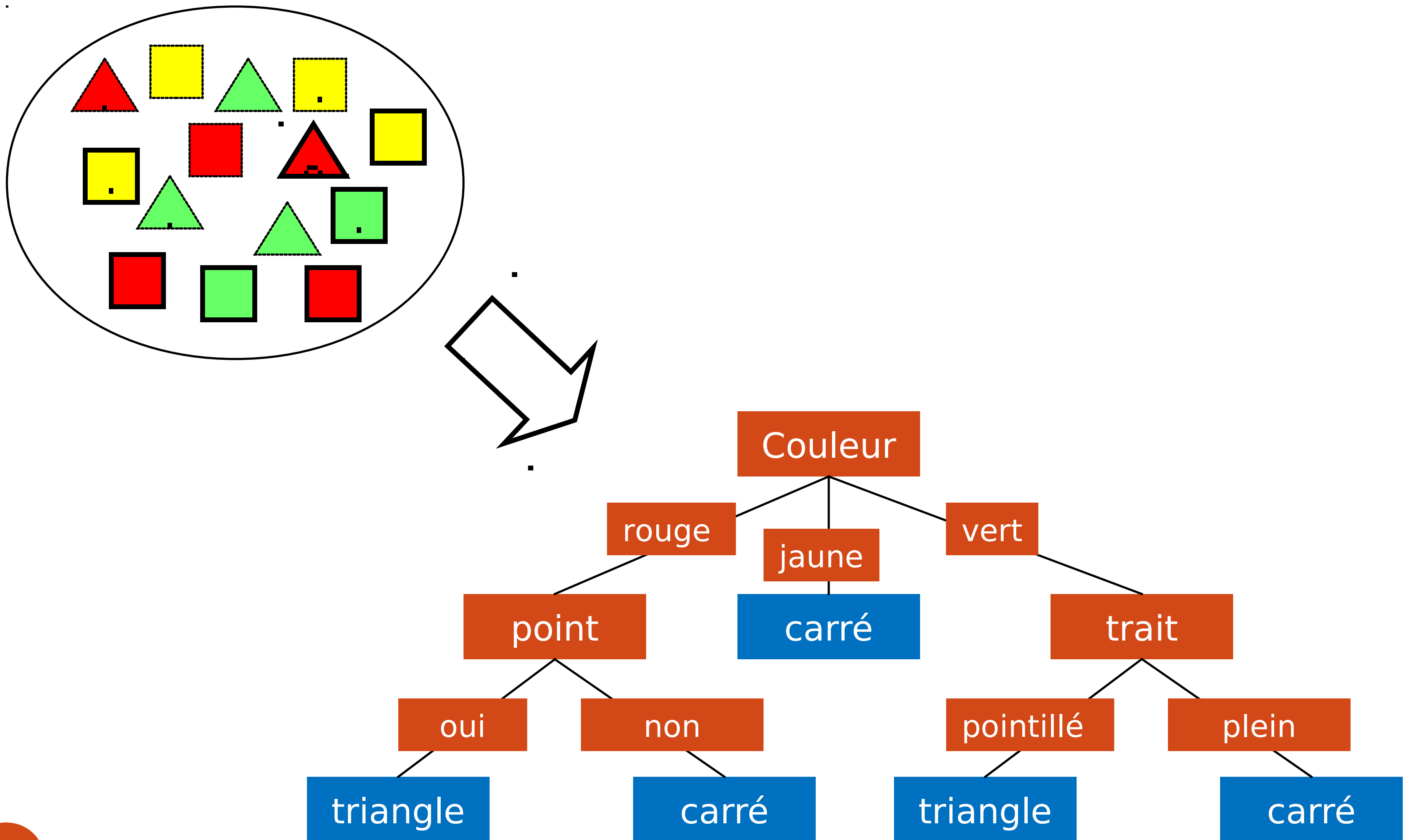


Gain(trait) = 0.971 – 0.951 = 0.020 bits

Gain(point) = 0.971 – 0 = 0.971 bits



L’arbre de Decision



Les lacunes de l'entropie

- *Le gain d'information* favorise les attributes ayant plusieurs valeurs
- Un attribut (ayant plusieurs valeurs) divise S en plusieurs sous ensembles, et si ces derniers sont petits, il auront une tendance à etre purs.
- Le **ratio de gain d' information** est considéré comme un moyen de correction de cette lacune.

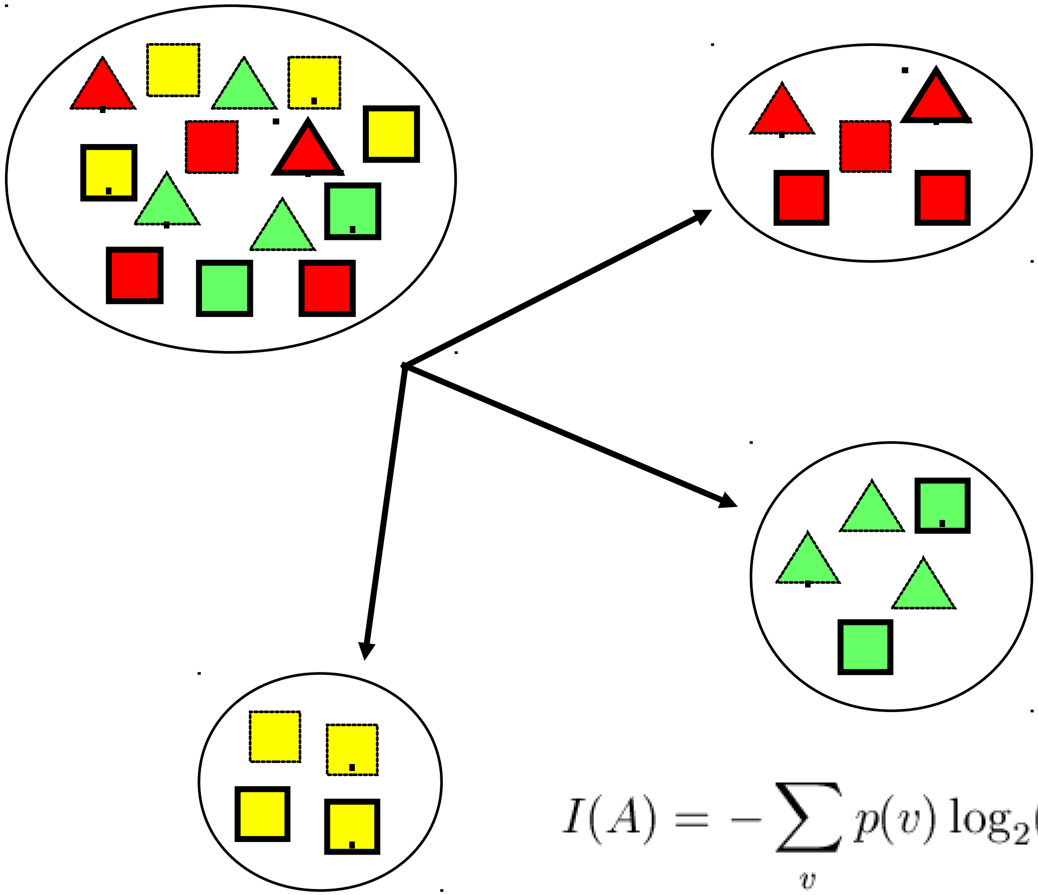
Ratio de Gain d'Information

- I(A) is quantité d'information necessaire pour coder A

$$I(A) = -\sum_{v} p(v) \log_2(p(v))$$

- Ratio de gain d'Information

$$GainRatio(A) = \frac{Gain(A)}{I(A)} = \frac{I - I_{res}(A)}{I(A)}$$



$$I(A) = -\sum_{v} p(v)\log_2(p(v))$$

$$I(\text{Color}) = -\frac{5}{14}\log_2\frac{5}{14} - \frac{5}{14}\log_2\frac{5}{14} - \frac{4}{14}\log_2\frac{4}{14} = 1.58 \text{ bits}$$

$$GainRatio(\text{Color}) = \frac{Gain(\text{Color})}{I(\text{Color})} = \frac{0.940 - 0.694}{1.58} = 0.156$$

Le gain d'information et le Ratio de gain d'information

A	|v(A)|	Gain(A)	GainRatio(A)
Couleur	3	0.247	0.156
trait	2	0.152	0.152
point	2	0.048	0.049

L’Index de Gini

- Constitue une autre mesure d’impurté (i et j sont des classes)

$$Gini = \sum_{i \neq j} p(i)p(j)$$

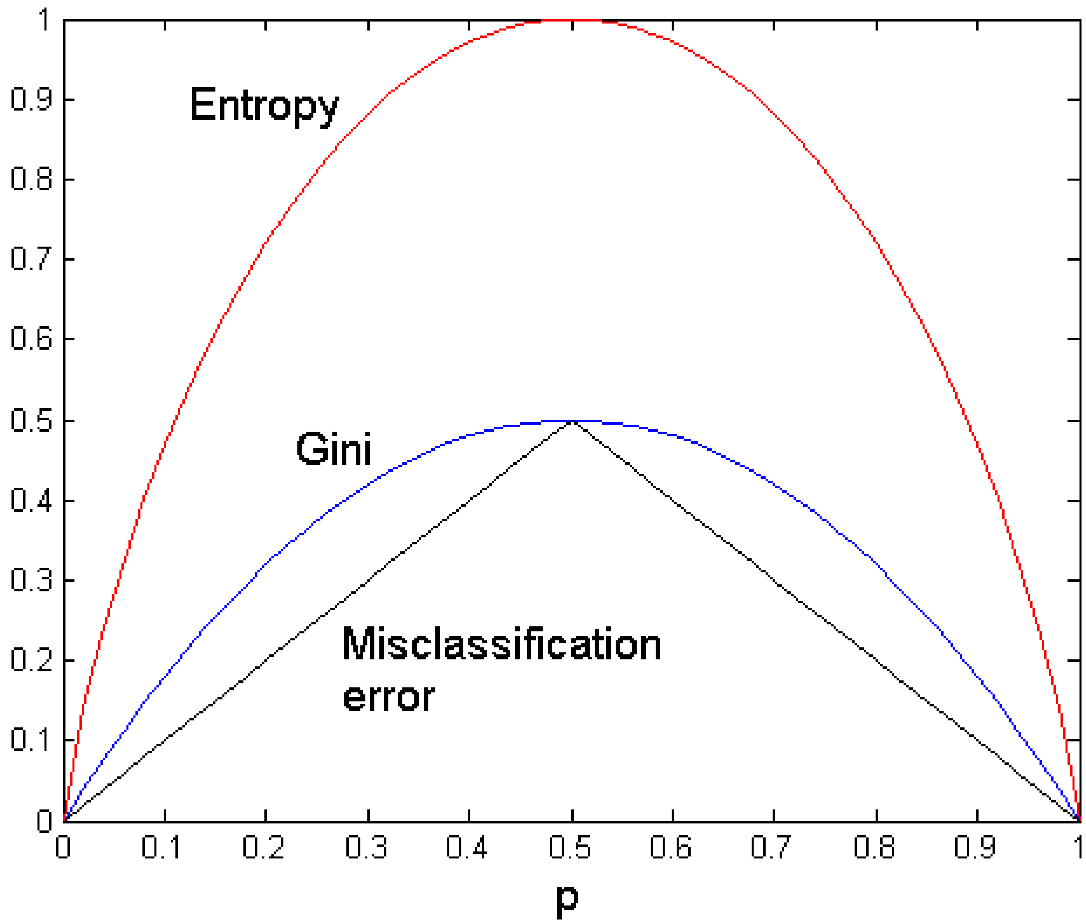
- Aprés l’application de A, l’index Gini du resultat sera:

$$Gini(A) = \sum_{v} p(v) \sum_{i \neq j} p(i|v)p(j|v)$$

- Gini signifie aussi l’esperance du taux d’erreur

L’Index de Gini vs l’entropie

- Classification binaire



Mesure d’Impurté: GINI

- L’Index Gini pour un noeud t :

$$GINI(t) = 1 - \sum_{j} [p(j|t)]^2$$

(NOTE: $p(j \mid t)$ la frequence de la classe j pour le noeud t).

- Le maximum de gini est atteint lorsque les classes sont équitablement réparties
- Le Minimum est atteint lorsque tous les exemples appartiennent à la meme classe

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

Exemples de calcul de GINI

$$GINI(t) = 1 - \sum_j [p(j|t)]^2$$

C1	**0**
C2	**6**

$P(C1) = 0/6 = 0$ $P(C2) = 6/6 = 1$

$Gini = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$

C1	**1**
C2	**5**

$P(C1) = 1/6$ $P(C2) = 5/6$

$Gini = 1 - (1/6)^2 - (5/6)^2 = 0.278$

C1	**2**
C2	**4**

$P(C1) = 2/6$ $P(C2) = 4/6$

$Gini = 1 - (2/6)^2 - (4/6)^2 = 0.444$

Division à base de GINI

- Principe utilisé dans CART, SLIQ, SPRINT.

$$GainGini = GINI(parent) - \left(\sum_{i=1}^{k} \frac{n_i}{n} GINI(i)\right)$$

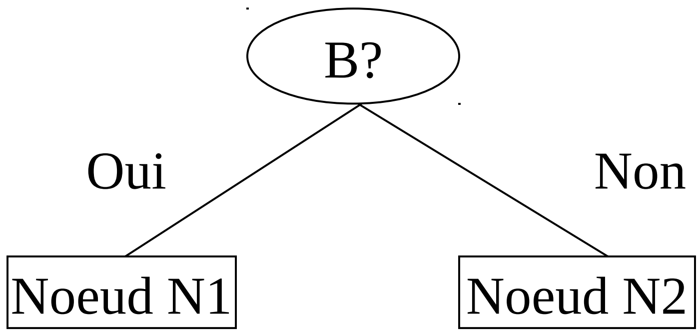
avec, n_i = la taille de l'enfant i,

n = la taille du parent.

Attributs binaires

On calcule le Gini du père
On calcule la moyenne des Gini des enfants
On fait la différence pour avoir le gain

	Parent
C1	**6**
C2	**6**
Gini = 0.500	



Gini(N1)=
1 – $(5/6)^2$ – $(2/6)^2$
= 0.194

Gini(N2)=
1 – $(1/6)^2$ – $(4/6)^2$
= 0.528

	N1	**N2**
C1	**5**	**1**
C2	**2**	**4**
Gini=0.333		

Gini(Enfant)
= 7/12 * 0.194
+
5/12 * 0.528
= 0.333

Attributs symboliques

- Theoriquement on peut faire une division binaire ou multi-label mais dans la pratique on utilise toujours une division binaire
- Gini(père)= 0.48

division interdite

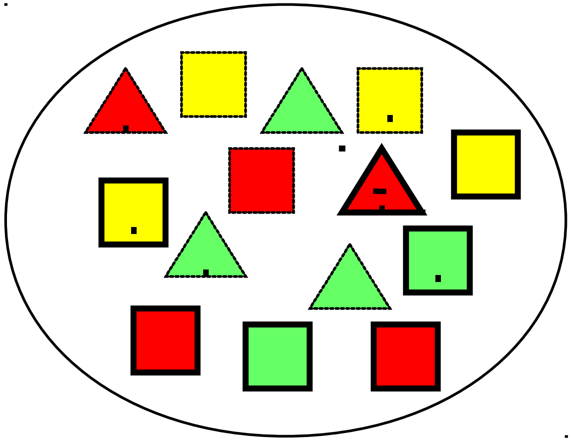
	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

	CarType	
	{Sports, Family }	{Luxury }
C1	3	1
C2	5	1
Gini	0.43	

	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

L’Index de variance (pour le cas binaire)



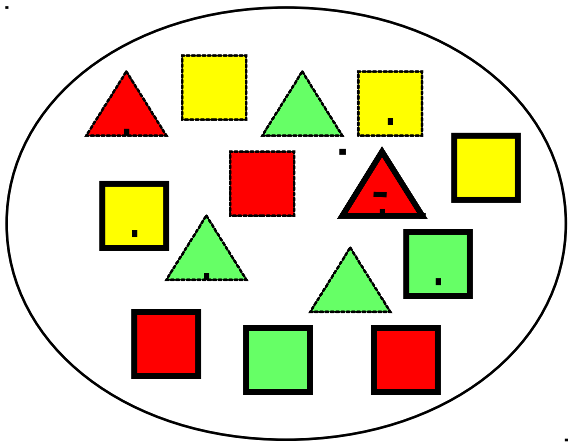
$$p(\square) = \frac{9}{14}$$

$$p(\Delta) = \frac{5}{14}$$

Index de variance = p(c1).p(c2)

Index de variance = 9/14 * 5/14

L'index de classification erronée



$$p(\square) = \frac{9}{14}$$

$$p(\Delta) = \frac{5}{14}$$

$$P(i \mid t)$$

MiscalssificationIndex(t)=1- 9/14=5/14

conclusion

- L’arbre de décision constitue un classifieur pouvant traiter des données continues, discrètes et symboliques
- L’apprentissage suit une approche gloutonne pour choisir les variables de division
- La réduction du sur-apprentissage constitue un problème majeur pour les arbres de décisions
- Les performances peuvent être boostées si on utilise des approches aléatoires et coopératives (Bagging)