INF8225

Leçon 8

Christopher Pal
École Polytechnique de Montréal

Plan du cours

Discussion de l'intra et questions préparatoires sur le tableau

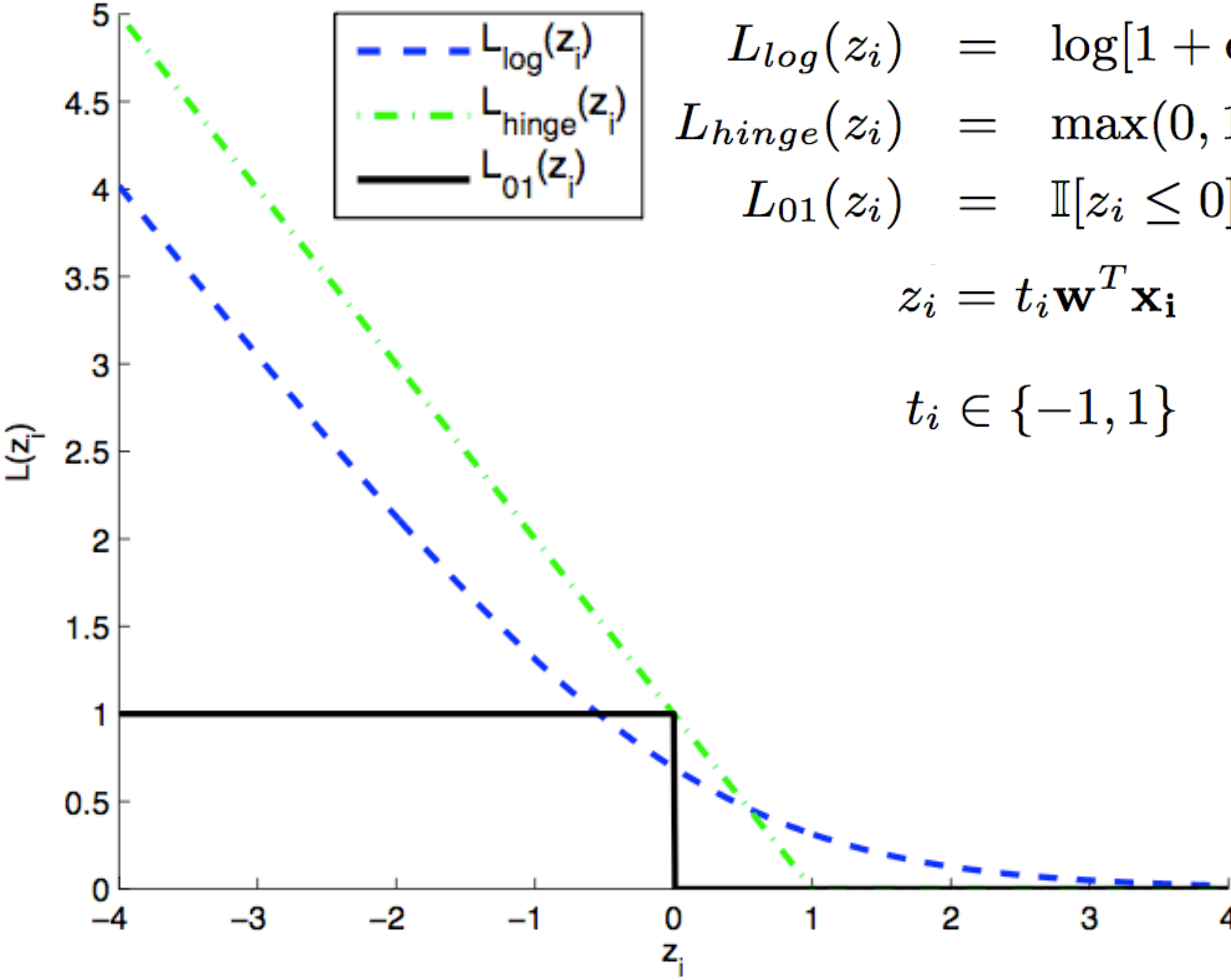
Deux approches considérées standard et utilisées beaucoup pour l'apprentissage automatique

1. Les machines à vecteurs de support ou les séparateurs à vaste marge
2. Les arbres de décision

- Discussion des sujets individuellement pour le projet (pendant les pauses)

Machines à vecteurs de support
« Support Vector Machines »

Comparez les « loss functions »



$$
\begin{aligned}
L_{log}(z_i) &= \log[1 + \exp(-z_i)] \\
L_{hinge}(z_i) &= \max(0, 1 - z_i) \\
L_{01}(z_i) &= \mathbb{I}[z_i \leq 0]
\end{aligned}
$$

$$z_i = t_i \mathbf{w}^T \mathbf{x_i}$$

$$t_i \in \{-1, 1\}$$

Machines à vecteurs de support

- Les SVMs sont probablement encore l'approche la plus populaire pour l'apprentissage supervisé pour plusieurs problèmes à cause de leur performance supérieur
- Il y a trois propriétés qui les rendent séduisantes

Propriétés séduisantes

- Les SVMs construisent un **séparateur à marge maximale**, un frontière de décision avec la plus grande distance possible entre les points d'apprentissage.
- Cela leur permet de bien généraliser
- C'est un problème convexe – **avec un seule minima globale**

Propriétés séduisantes

- Les SVMs créent un hyperplan (linéaire séparateur), mais sont capables de projeter les données dans un espace de dimension plus grande en utilisant ce qu'on appelle **l'astuce du noyau**
- Il arrive souvent que des données qui ne sont pas linéairement séparables dans l'espace d'entrée d'origine le soient dans un espace de plus grande dimension
- Le séparateur linéaire de l'espace de plus grande dimension est en fait **non linéaire** *dans l'espace d'origine*

Propriétés séduisantes

- Les SVMs sont une méthode non paramétrique : elles mémorisent tous les exemples d'apprentissage et peuvent avoir besoin de tous les stocker
- Toutefois, souvent en pratique après l'optimisation de notre fonction d'objective, notre SVM va utiliser seulement un petit nombre d'exemples – <u>les vecteurs de support</u> – pour créer notre classificateur
- Donc, ils nous permet de représenter des fonctions complexes, mais ils sont résistantes au surapprentissage

Formulation des SVMs

- Nous utilisons les valeurs de +1 et -1 pour les étiquettes y_i
- Nous utilisons un vecteur de poids **w** pour les valeurs des paramètres
- Le séparateur correspondre avec un hyperplan, défini par $\{ \mathbf{x} : f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = 0\}$
- La règle de classification : $G(\mathbf{x}) = \text{signe}(\mathbf{w}^T\mathbf{x} + b)$
- Pour les données séparable $y_i f(\mathbf{x}_i) > 0 \quad \forall i$

Le séparateur à marge maximale

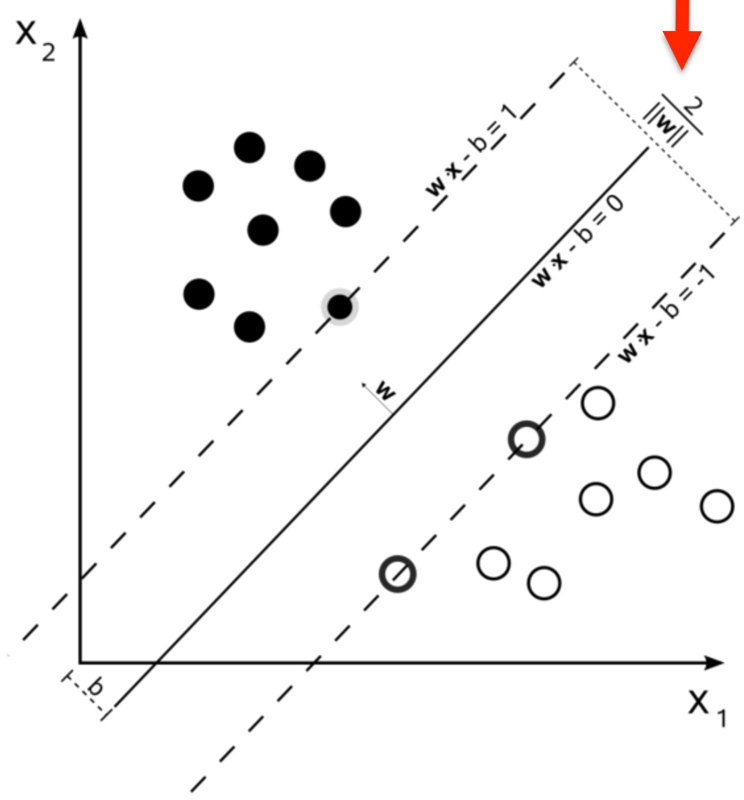
- Nous cherchons l'hyperplan ou **le séparateur à marge maximale –** ou la largeur de la zone délimitée dans la figure, c.-à-d. la distance de la droite séparatrice au point exemple le plus proche

Avec $C = \frac{1}{\|\mathbf{w}\|}$

Nous avons un problème d'optimisation de

$\max_{\mathbf{w}, b, \|\mathbf{w}\|=1} C$ tels que

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq C, \quad i = 1, \ldots, N$$



Formulation d'une Lagrangienne

- On peut rééecrire le problème comme

$$\min_{\mathbf{w},b} \|\mathbf{w}\| \quad \text{tels que}$$

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad i = 1,\ldots,N$$

- Nous utilisions les multiplicateurs de Lagrange, $\alpha_n \geq 0$, un pour chaque contraint, qui nous donne une fonction Lagrangienne de

$$L(\mathbf{w},b,\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i \left\{ y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \right\}$$

$$\left\{ \frac{\partial L}{\partial \mathbf{w}} = 0, \frac{\partial L}{\partial a} = 0 \right\} \rightarrow w = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i, \sum_{i=1}^{N} \alpha_i y_i = 0$$

La « représentation duale »

- L'élimination de w et b et avec ces conditions nous donne

$$\arg\max_{\alpha} \sum_{\mathrm{i}=1}^{\mathrm{N}} \alpha_i - \frac{1}{2} \sum_{j=1}^{N} \sum_{k=1}^{N} \alpha_j \alpha_k y_j y_k \mathbf{x}_j^T \mathbf{x}_k$$

$$\alpha_i, \xi_{\mathrm{i}} \geq 0 \quad \forall i, \ \sum\nolimits_{\mathrm{j}} \alpha_j y_j = 0 \quad \forall j$$

- Ce qui donne les multiplicateurs de Lagrange optimaux α_j^* et nous avons $\mathbf{w} = \sum\nolimits_{\mathrm{j}} \alpha_{\mathrm{j}} \mathbf{x}_j$
- C'est un problème d'optimisation de <u>programmation quadratique</u>

Les vecteurs de support

- Afin d'obtenir l'hyperplan solution, on remplace **w** par sa valeur optimale **w***

$$h(\mathbf{x})=\text{signe}\left(\sum_{\mathrm{j}} \alpha_j^* y_j (\mathbf{x}^T \mathbf{x}_j) - b\right)$$

- Et après l'optimisation nous avons une autre propriété importante associé avec les MVS – les poids α_j associé à chaque point sont <u>nuls</u> sauf pour les vecteurs de support, c.-à-d. pour les points les plus proches du séparateur
- Les vecteurs de support sont généralement beaucoup moins nombreux que les exemples

La formulation standard

- Rappel : on peut écrire le problème comme

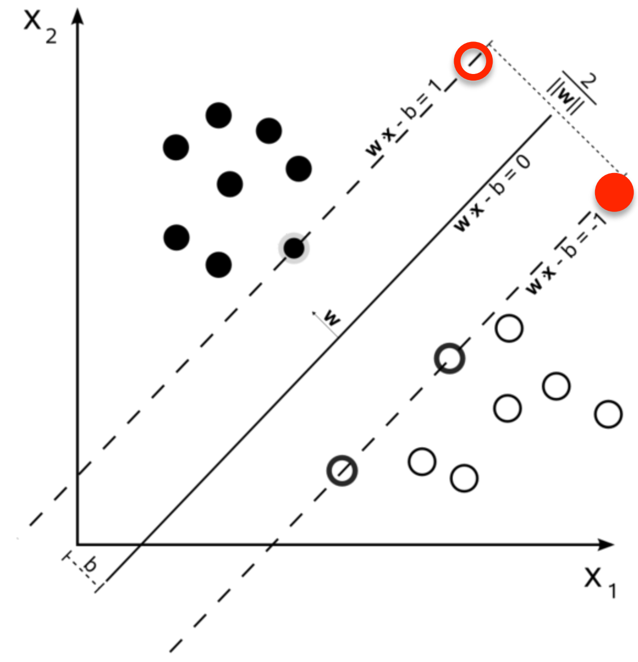
$$\min_{\mathbf{w},b} \|\mathbf{w}\| \quad \text{tels que}$$

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1,\ldots,N$$

- Que faire si nos données ne sont pas linéairement séparable?
- On peut modifier les contraintes comme

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq C(1 - \xi_i),$$

Les marges MVS vs. RL (la régression logistique)



$$\max_{\mathbf{w},b} \frac{1}{\|\mathbf{w}\|}, \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq C(1-\xi_i), \quad \forall \mathrm{i}$$

Comparez avec la même idée dans le contexte de la régression logistique

$$\max\left[\sum_{i=1}^{N}\left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - \log Z(\mathbf{w},x_i)\right)\right]$$

La formulation du cas inséparable

- On peut formuler le problème comme

$$\min_{\mathbf{w},b,\xi} \|\mathbf{w}\| \qquad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq C(1-\xi_i), \quad \forall \mathrm{i},$$
$$\xi_\mathrm{i} \geq 0, \sum \xi_\mathrm{i} \leq const.$$

- Il correspondre (comme le cas séparable) avec un problème d'optimisation quadratique et <u>convexe</u>
- Depuis $\xi_\mathrm{i} \geq 1$ implique que point i est en erreur, on pourrait interpréter $\sum_i \xi_\mathrm{i}$ comme une limite supérieure sur les erreurs
- Paramètre C donc control le nombre des erreurs
- Donc il est typique de sélectionner la valeur de C pendant une phase de validation croisée

La régression logistique et l'astuce de noyau (RBF)

$$p(\mathbf{y}\,|\,\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\exp\left(\mathbf{y}^T\theta\mathbf{x}\right) \rightarrow$$

$$p(\mathbf{y}\,|\,\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\exp\left(\mathbf{y}^T\theta\mathbf{f}(\mathbf{x})\right), f_j(\mathbf{x}_i) = K(\mathbf{x}_i, \mathbf{x}_j)$$

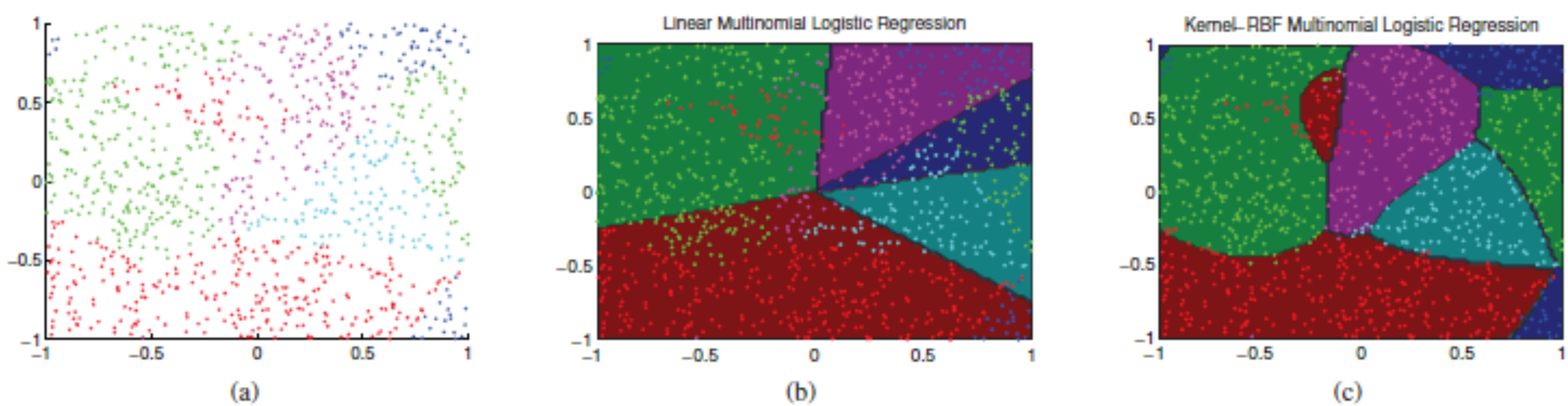


Figure 1.10: (a) Some 5 class data in 2d. (b) Multinomial logistic regression in the original feature space. (c) RBF basis functions with bandwidth of 1. We use all the data points as centers. Figure generated by `logregMultinomKernelDemo`.

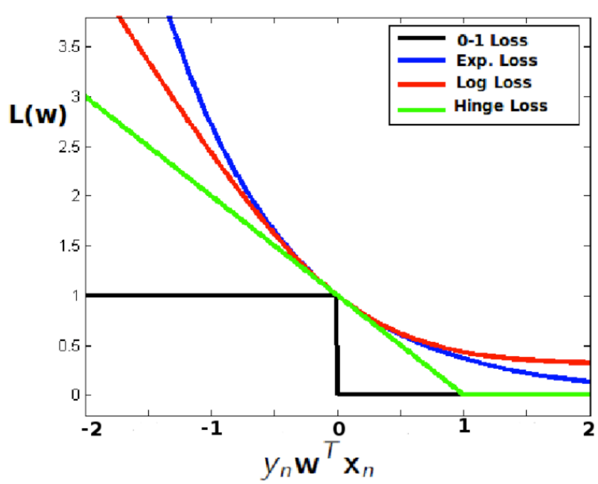
L'astuce de noyau (RBF) et la régression logistique

$$\max\left[\sum_{i=1}^{N}\left(\mathbf{y}_i^T\theta\mathbf{f}(\mathbf{x}_i)-\log\left\{\sum_i \exp\left(\mathbf{y}_i^T\theta\mathbf{f}(\mathbf{x}_i)\right)\right\}\right)\right]$$

Comparez avec la même idée appliqué aux MVS

$$\min_{\mathbf{w},b}\ \|\mathbf{w}\|\ ,\quad y_i(\mathbf{w}^T\mathbf{x}_i+b)\geq C(1-\xi_i),\quad \forall \mathrm{i}$$

Comparez des « fonctions de perte » :



0 - 1	$1 - \mathrm{H}(0)$, (1 - Heaviside)
hinge	$1 - y_i\mathbf{w}^T\mathbf{x}_i,\ \max\{0, 1 - y_i\mathbf{w}^T\mathbf{x}_i\}$
log	$\log(1+\exp(y_i\mathbf{w}^T\mathbf{x}_i))$
exp	$\exp(-y_i\mathbf{w}^T\mathbf{x}_i)$

L'astuce du noyau dans le contexte d'un MVS

$$h(\mathbf{x})=\text{signe}\left(\sum_{\mathrm{j}} \alpha_j^* y_j (\mathbf{x}^T \mathbf{x}_j) - b\right)$$

- Exemple: étant donné notre modèle linéaire, on peut remplacer $\mathbf{x}^T_i\mathbf{x}_j$ par $\mathbf{f}(\mathbf{x}_i)^T\mathbf{f}(\mathbf{x}_i)$, ex.

$$f_1 = x_1^2, f_2 = x_2^2, f_3 = \sqrt{2}x_1x_2$$

- Nous avons un « noyau » de $K(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i \, . \mathbf{x}_j)^2$
- Généralement: On peut trouver de façon efficace des séparateur linéaires optimaux dans des espaces des caractéristiques de dimension de l'ordre du milliard, ou dans certains cas infinie
- Important : nous avons un séparateur non-linéaire dans l'espace original (c.-à-d. $\mathbf{x}$)

Minimisation de la « hinge loss »

$$\min_{\mathbf{w},w_0} \frac{1}{2}\|\mathbf{w}\| + C\sum_{i=1}^{N} \max(0, 1 - t_i(\mathbf{w}^T\mathbf{x}_i + w_0))$$

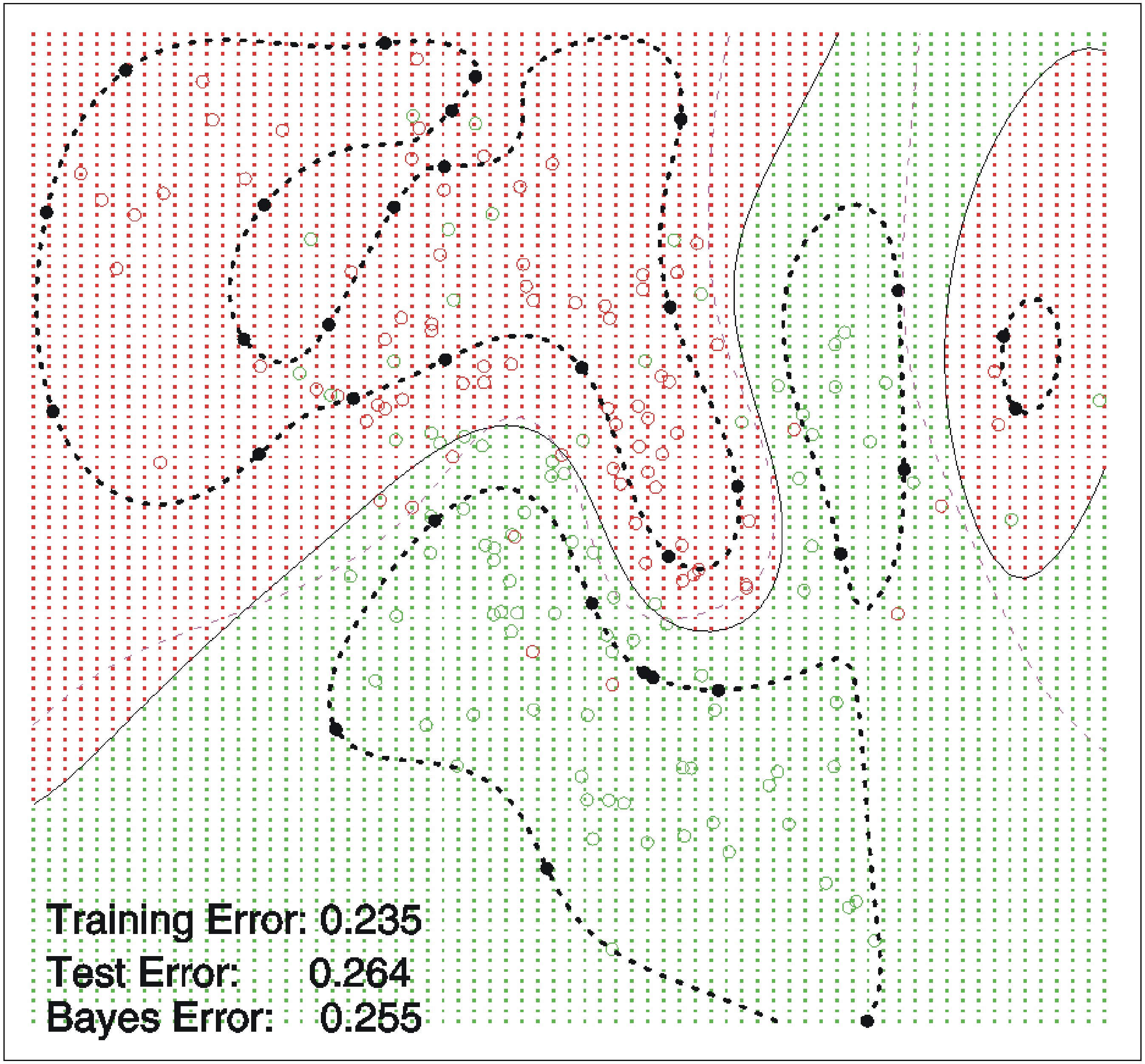
En ajoutant un « slack variable », ξ,

$$\min_{\mathbf{w},w_0,\xi} \frac{1}{2}\|\mathbf{w}\| + C\sum_{i=1}^{N} \xi_i \quad \text{(t.q/s.t.)}$$

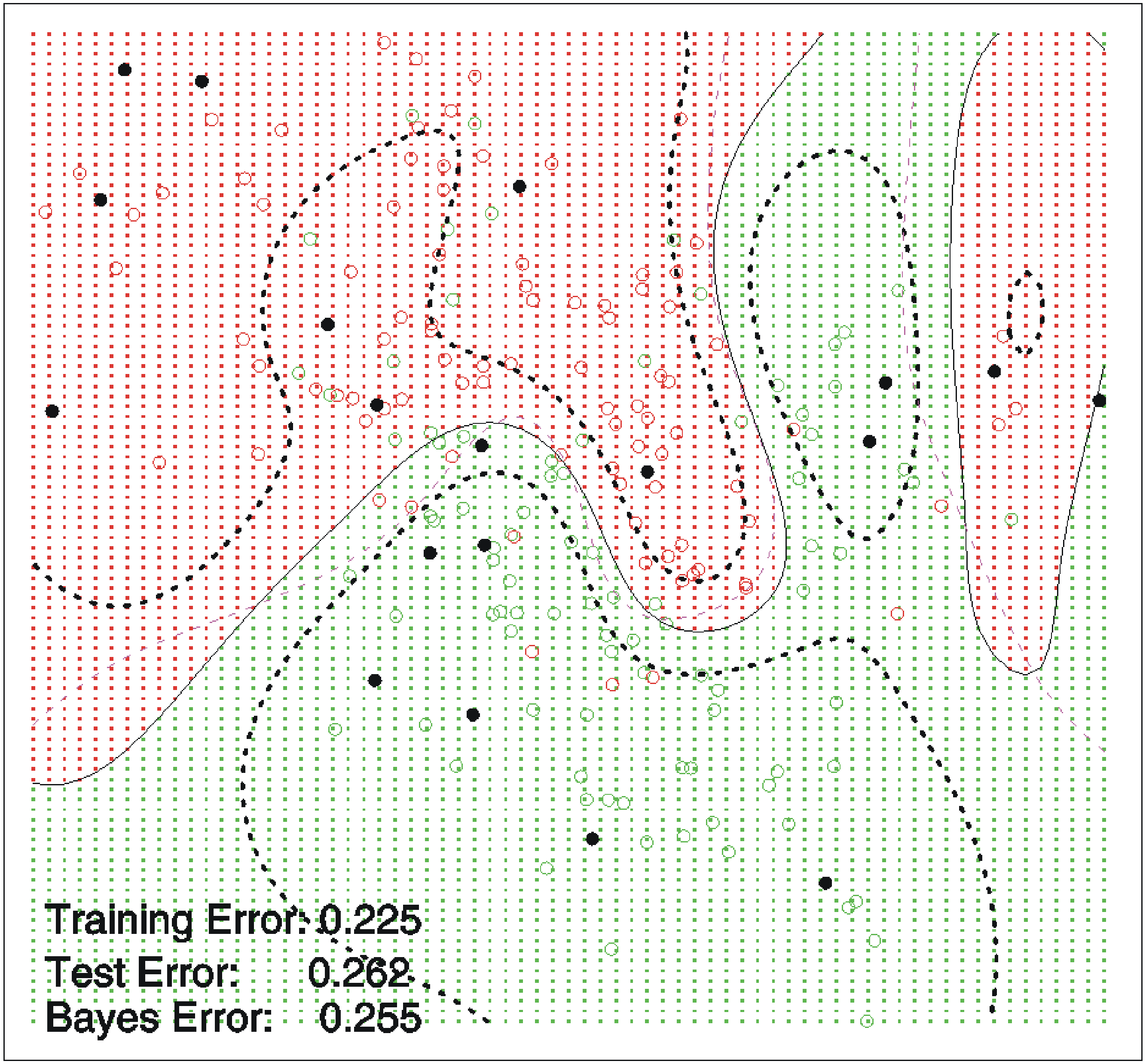
$$\xi_i \geq 0, \quad \mathrm{t}_\mathrm{i}(\mathbf{w}^T\mathbf{x}_i + w_0) \geq 1 - \xi_i, \quad i = 1:N$$

Puis, on a un programme quadratique avec N+D+1 variables sujet à O(N) contraintes.

SVM - 130 Support Points



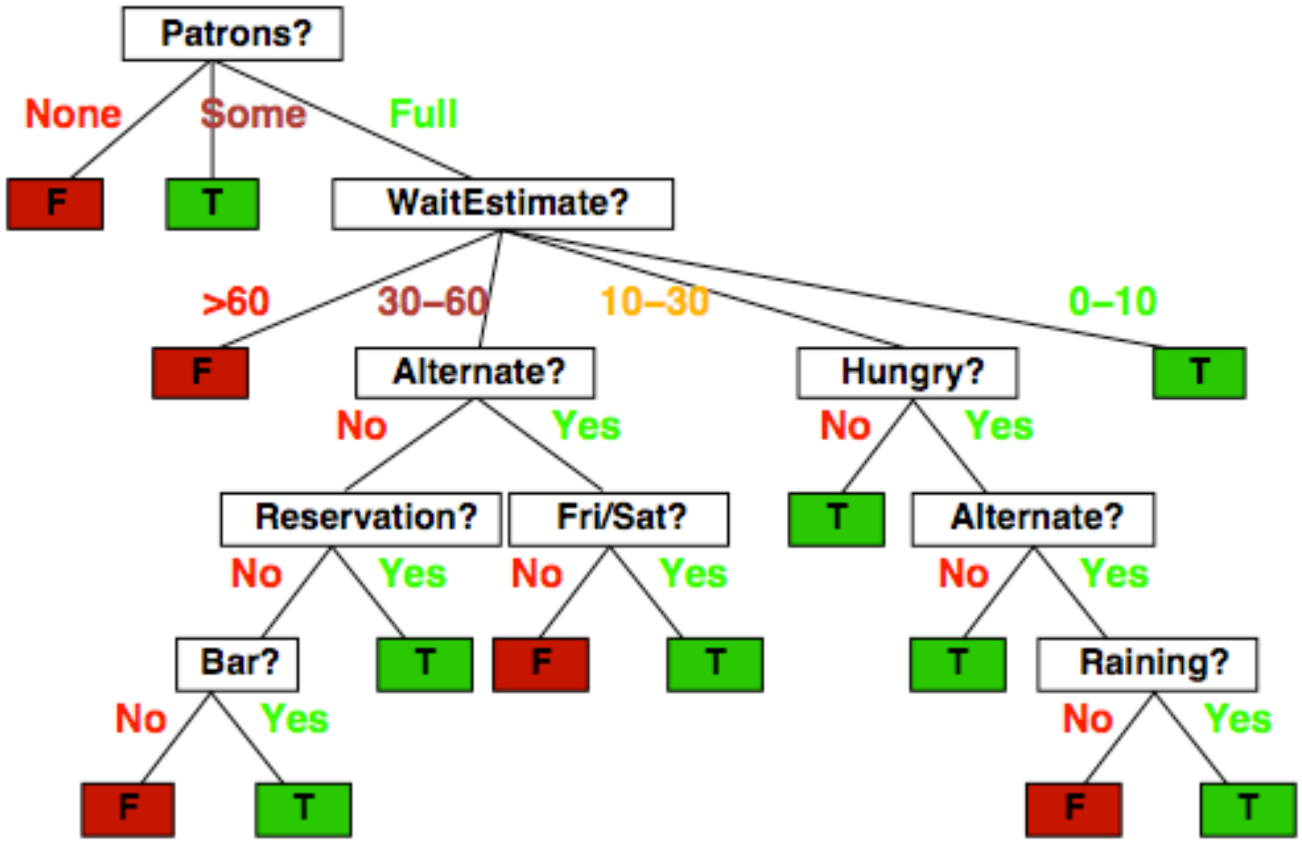
IVM - 19 Import Points



Les arbres de décision

et apprentissage

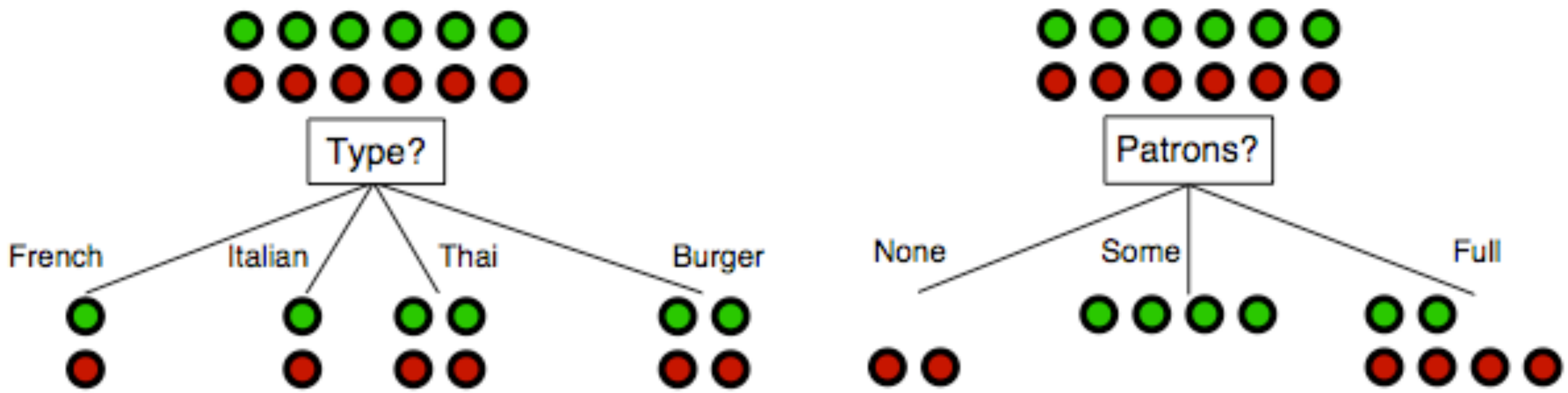
Arbre de décision pour décider d'attendre ou non une table



Exemple: Ensemble d'exemples

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*10–30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*F*	*T*	*French*	*>60*	*F*
X_6	*F*	*T*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Italian*	*0–10*	*T*
X_7	*F*	*T*	*F*	*F*	*None*	*$*	*T*	*F*	*Burger*	*0–10*	*F*
X_8	*F*	*F*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Thai*	*0–10*	*T*
X_9	*F*	*T*	*T*	*F*	*Full*	*$*	*T*	*F*	*Burger*	*>60*	*F*
X_{10}	*T*	*T*	*T*	*T*	*Full*	*$$$*	*F*	*T*	*Italian*	*10–30*	*F*
X_{11}	*F*	*F*	*F*	*F*	*None*	*$*	*F*	*F*	*Thai*	*0–10*	*F*
X_{12}	*T*	*T*	*T*	*T*	*Full*	*$*	*F*	*F*	*Burger*	*30–60*	*T*

Répartition des exemples en testant les attributs



- Le choix de type n'aide pas à différentier les exemples négatifs des positifs
- Le choix de Clients parvient bien à séparer les exemples

L’entropie

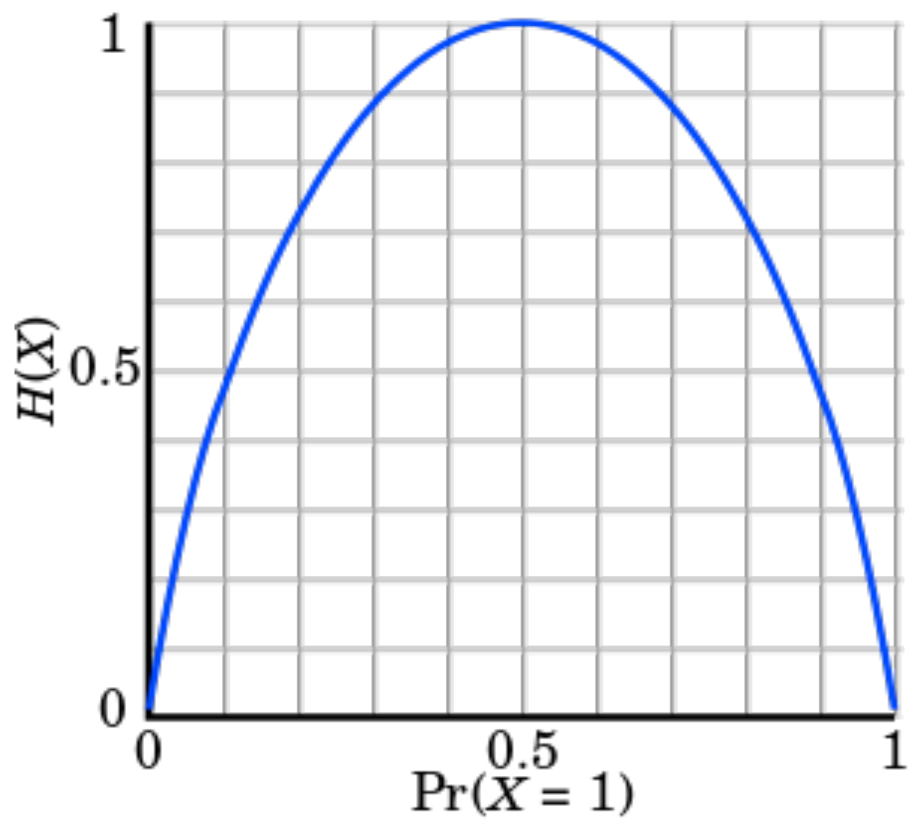
- Dans la théorie de l’information il existe le concept d’entropie
- L’entropie mesure le montant d’incertitude

$$H(P(v_1),\ldots P(v_n)) = \sum_{i=1}^{n} -P(v_i)\log_2 P(v_i)$$

- si les réponses possible v_i ont les probabilités $P(v_i)$, exemple: une pièce de monnaie

$$H(P(0),P(1)) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

L’entropie d’une variable binaire



Comment choisir l'attribut

- En utilisant la théorie de l'information
- Soit *N* le nombre d'exemples
- Soit une variable de décision d ayant les valeurs possibles $d_1, d_2 \ldots d_k$
- Supposons que l'ensemble d'exemples contient n_i items pour chaque valeur possible d_i
- Initialement la valeur de l'entropie est

$$Entropie_I = \sum_{i=1}^{k} -\left(\frac{n_i}{N}\right)\log_n\left(\frac{n_i}{N}\right)$$

Comment choisir l'attribut (suite)

- Supposons maintenant que l'on choisisse l'attribut A, qui a les valeurs possibles $a_1, a_2 \ldots a_j$
- Si on choisit la valeur a_i, on se retrouve avec un sous-ensemble de M_i exemples tel que on a m_v items pour chaque valeur de décision $v = d_1 \ldots d_k$
- L'entropie dans ce cas est donc $$E(a_i) = \sum_{v=1}^{k} -\left(\frac{m_v}{M_i}\right)\log_2\left(\frac{m_v}{M_i}\right)$$
- On fait la moyenne sur toutes les valeurs de A : $$\text{Reste}(A) = \sum_{i=1}^{j}\left(\frac{M_i}{N}\right)E(a_i)$$
- On choisit alors l'attribut A qui maximise le gain $$\text{Gain}(A) = \text{Entropie}_I - \text{Reste}(A)$$

Apprentissage d'arbre de décision

fonction CREER-ARBRE-DECISION(*exemples, attributs, défaut*)
  si *exemples* = $\emptyset$ **retourner** *défaut*
  sinon si tous les exemples ont la même classification **alors**
    retourner cette classification
  sinon si attributs = $\emptyset$ **alors**
    retourner la valeur majoritaire parmi les exemples
  sinon
    meilleur **<-** CHOISIR-ATTRIBUT*(attributs, exemples*)
    arbre **<-** nouvel arbre avec attribut *meilleur* comme racine
    m **<-** la valeur majoritaire parmi les exemples
    pour chaque valeur v_i de *meilleur* **faire**
      $exemples_i$ **<-** { $e \in$ *exemples* | valeur de *meilleur* pour $e = v_i$ }
      attributs' = *attributs* – {*meilleur* }
      sous-arbre = CREER-ARBRE-DECISION($exemples_i$,*attributs', m*)
      ajouter une branche à *arbre* avec attribut v_i et *sous-arbre*
    retourner *arbre*

Arbre de décision - exemple

A	*B*	*C*	*Décision*
0	2	1	1
1	1	0	1
2	2	1	1
2	0	2	1
0	0	0	1
1	2	2	0
0	2	2	0
1	0	0	0
2	1	2	0
2	2	2	0

Entropie = - 0,5*log(0,5) – 0,5*log(0,5)
= 1

Arbre de décision – exemple (suite)

A	***B***	***C***	***Décision***
0	2	1	1
1	1	0	1
2	2	1	1
2	0	2	1
0	0	0	1
1	2	2	0
0	2	2	0
1	0	0	0
2	1	2	0
2	2	2	0

Attribut A:
Entropie si A = 0: - 0,66 log 0,66 – 0,33 log 0,33

Arbre de décision – exemple (suite)

A	*B*	*C*	*Décision*
0	2	1	1
1	1	0	1
2	2	1	1
2	0	2	1
0	0	0	1
1	2	2	0
0	2	2	0
1	0	0	0
2	1	2	0
2	2	2	0

Attribut A:
Entropie si A = 0: 0,92

Arbre de décision – exemple (suite)

A	*B*	*C*	*Décision*
0	2	1	1
1	1	0	1
2	2	1	1
2	0	2	1
0	0	0	1
1	2	2	0
0	2	2	0
1	0	0	0
2	1	2	0
2	2	2	0

Attribut A:
Entropie si A = 0: 0,92
Entropie si A = 1: - 0,33 log 0,33 – 0,66 log 0,66

Arbre de décision – exemple (suite)

A	*B*	*C*	*Décision*
0	2	1	1
1	1	0	1
2	2	1	1
2	0	2	1
0	0	0	1
1	2	2	0
0	2	2	0
1	0	0	0
2	1	2	0
2	2	2	0

Attribut A:

Entropie si A = 0: 0,92

Entropie si A = 1: 0,92

Arbre de décision – exemple (suite)

A	*B*	*C*	*Décision*
0	2	1	1
1	1	0	1
2	2	1	1
2	0	2	1
0	0	0	1
1	2	2	0
0	2	2	0
1	0	0	0
2	1	2	0
2	2	2	0

Attribut A:
Entropie si A = 0: 0,92
Entropie si A = 1: 0,92
Entropie si A = 2:

Arbre de décision – exemple (suite)

A	*B*	*C*	*Décision*
0	2	1	1
1	1	0	1
2	2	1	1
2	0	2	1
0	0	0	1
1	2	2	0
0	2	2	0
1	0	0	0
2	1	2	0
2	2	2	0

Attribut A:
Entropie si A = 0: 0,92
Entropie si A = 1: 0,92
Entropie si A = 2: - 0,5 log 0,5 – 0,5 log 0,5

Arbre de décision – exemple (suite)

A	*B*	*C*	*Décision*
0	2	1	1
1	1	0	1
2	2	1	1
2	0	2	1
0	0	0	1
1	2	2	0
0	2	2	0
1	0	0	0
2	1	2	0
2	2	2	0

Attribut A:
Entropie si A = 0: 0,92
Entropie si A = 1: 0,92
Entropie si A = 2: 1

Arbre de décision – exemple (suite)

A	*B*	*C*	*Décision*
0	2	1	1
1	1	0	1
2	2	1	1
2	0	2	1
0	0	0	1
1	2	2	0
0	2	2	0
1	0	0	0
2	1	2	0
2	2	2	0

Attribut A:
Entropie si A = 0: 0,92
Entropie si A = 1: 0,92
Entropie si A = 2: 1
Moyenne = 0,3*0,92 + 0,3*0,92 + 0,4*1
= 0,95
Gain = 1 – 0,95 = 0,05

Arbre de décision – exemple (suite)

A	*B*	*C*	*Décision*
0	2	1	1
1	1	0	1
2	2	1	1
2	0	2	1
0	0	0	1
1	2	2	0
0	2	2	0
1	0	0	0
2	1	2	0
2	2	2	0

Attribut B:
Entropie si B = 0: 0,92
Entropie si B = 1: 1
Entropie si B = 2: 0,97
Moyenne = 0,3*0,92 + 0,2*1 + 0,5*0,97
= 0,96
Gain = 1 – 0,96 = 0,04

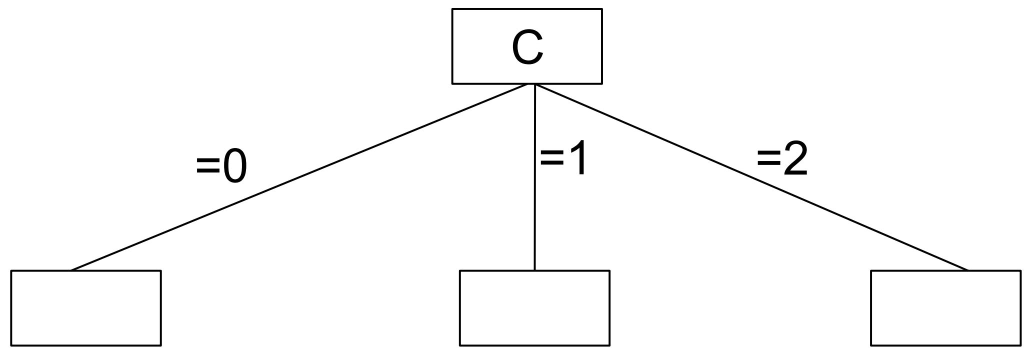
Arbre de décision – exemple (suite)

A	*B*	*C*	*Décision*
0	2	1	1
1	1	0	1
2	2	1	1
2	0	2	1
0	0	0	1
1	2	2	0
0	2	2	0
1	0	0	0
2	1	2	0
2	2	2	0

Attribut C:
Entropie si C = 0: 0,92
Entropie si C = 1: 0
Entropie si C = 2: 0,72
Moyenne = 0,3*0,92 + 0,2*0 + 0,5*0,72
= 0,64
Gain = 1 – 0,64 = 0,36

On choisit donc l'attribut C comme racine de l'arbre

Arbre de décision (partiel)



A	B	C	Décision
1	1	0	1
0	0	0	1
1	0	0	0

A	B	C	Décision
0	2	1	1
2	2	1	1

A	B	C	Décision
2	0	2	1
1	2	2	0
0	2	2	0
2	1	2	0
2	2	2	0

Arbre de décision – exemple (suite)

Cas C = 0

A	*B*	*C*	*Décision*
1	1	0	1
0	0	0	1
1	0	0	0

Entropie initiale : 0,92

Arbre de décision – exemple (suite)

Cas C = 0

A	*B*	*C*	*Décision*
1	1	0	1
0	0	0	1
1	0	0	0

Attribut A:
Entropie si A = 0: 0
Entropie si A = 1: 1
Moyenne = 0,33*0 + 0,66*1
= 0,66
Gain = 0,92 – 0,66 = 0,26

Arbre de décision – exemple (suite)

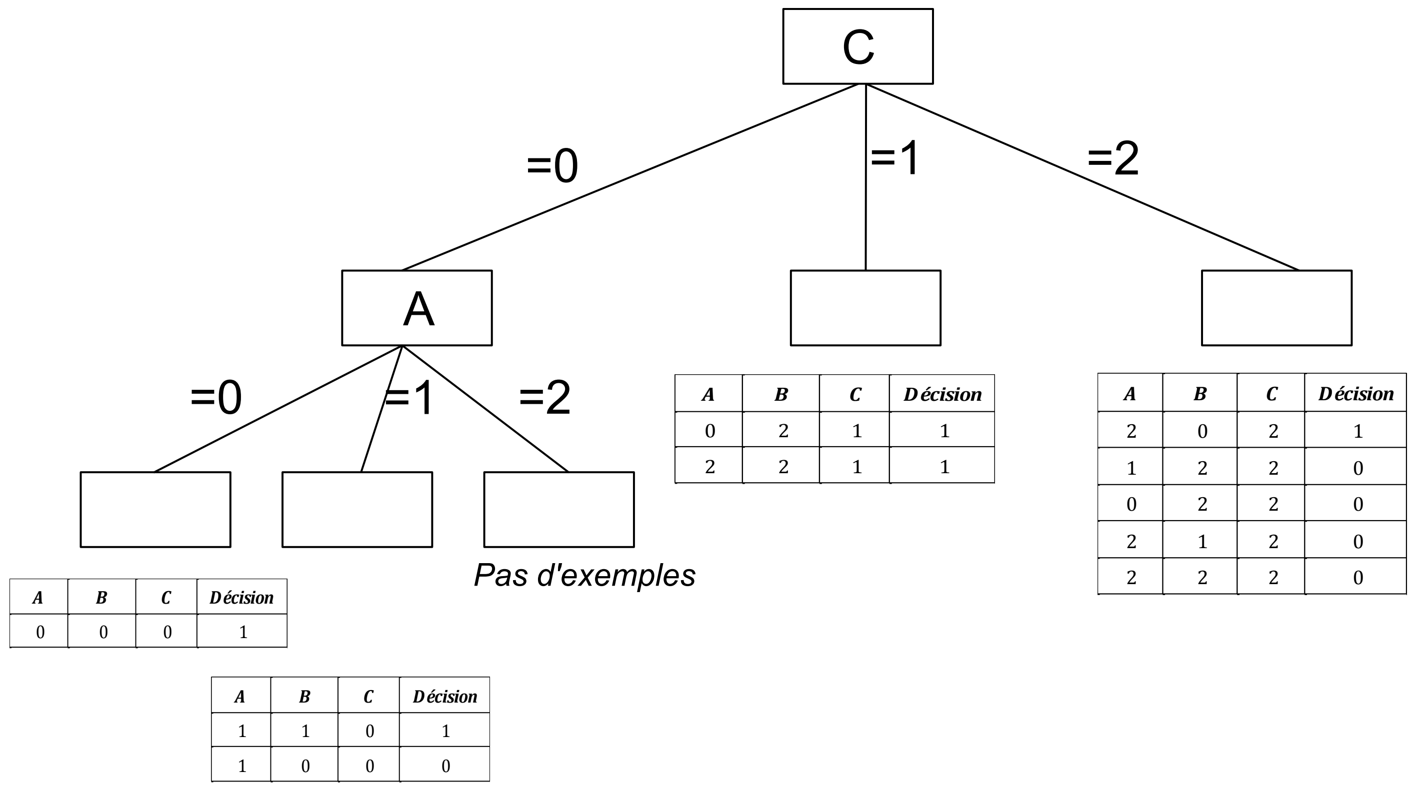
Cas C = 0

A	*B*	*C*	*Décision*
1	1	0	1
0	0	0	1
1	0	0	0

Attribut B:
Entropie si B = 0: 1
Entropie si B = 1: 0
Moyenne = 0,66*1 + 0,33*0
= 0,66
Gain = 0,92 – 0,66 = 0,26

Les deux attributs sont équivalents. Pas d'importance que l'on choisisse A ou B.

Arbre de décision (partiel)



A	B	C	Décision
0	0	0	1

A	B	C	Décision
1	1	0	1
1	0	0	0

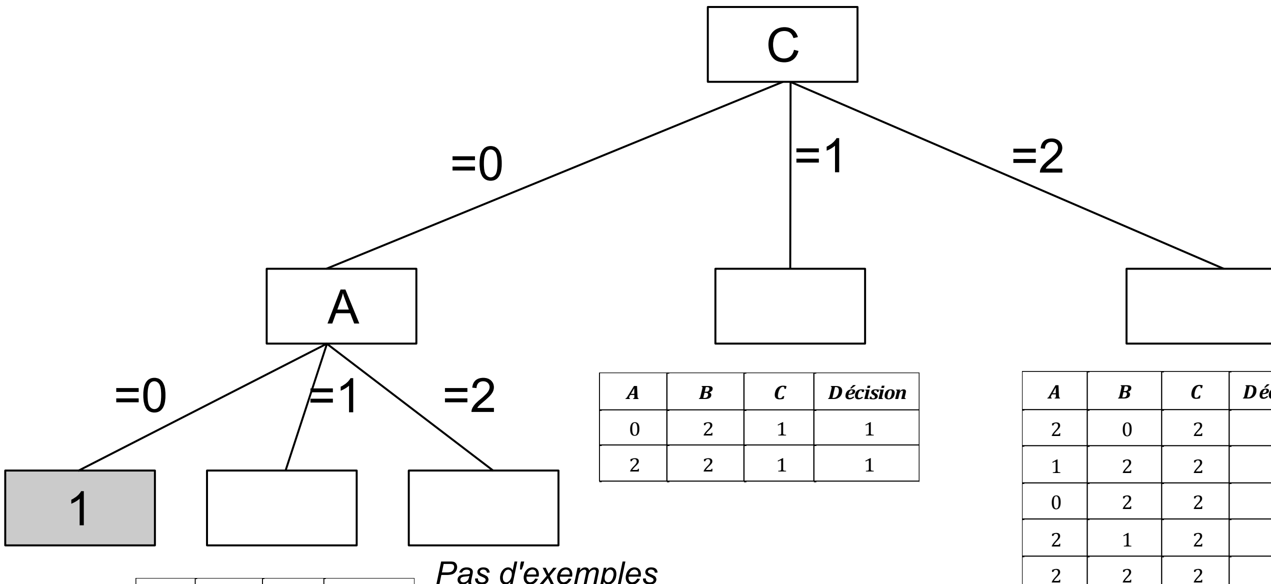
A	B	C	Décision
0	2	1	1
2	2	1	1

A	B	C	Décision
2	0	2	1
1	2	2	0
0	2	2	0
2	1	2	0
2	2	2	0

Arbre de décision – exemple (suite)

Cas C = 0 et A = 0
Tous les exemples ont la même valeur.
On retourne donc cette valeur.

Arbre de décision (partiel)



A	*B*	*C*	*Décision*
1	1	0	1
1	0	0	0

A	*B*	*C*	*Décision*
0	2	1	1
2	2	1	1

A	*B*	*C*	*Décision*
2	0	2	1
1	2	2	0
0	2	2	0
2	1	2	0
2	2	2	0

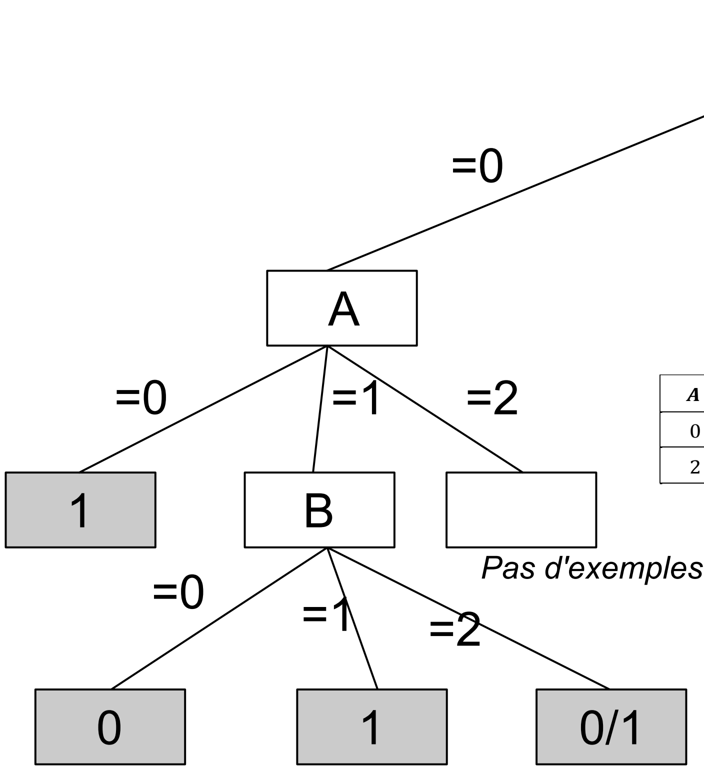
Arbre de décision – exemple (suite)

Cas C = 0 et A = 1

Pour B = 0 et B = 1, tous les exemples sont de la même classe (un seul exemple pour chaque cas)

Pour B = 2, on a aucun exemple. On doit donc choisir la valeur majoritaire au noeud parent. Comme il n'y a que deux exemples et qu'on a 1 dans un cas et 0 dans l'autre, on peut choisir indifféremment l'une ou l'autre.

Arbre de décision (partiel)



A	*B*	*C*	*Décision*
0	2	1	1
2	2	1	1

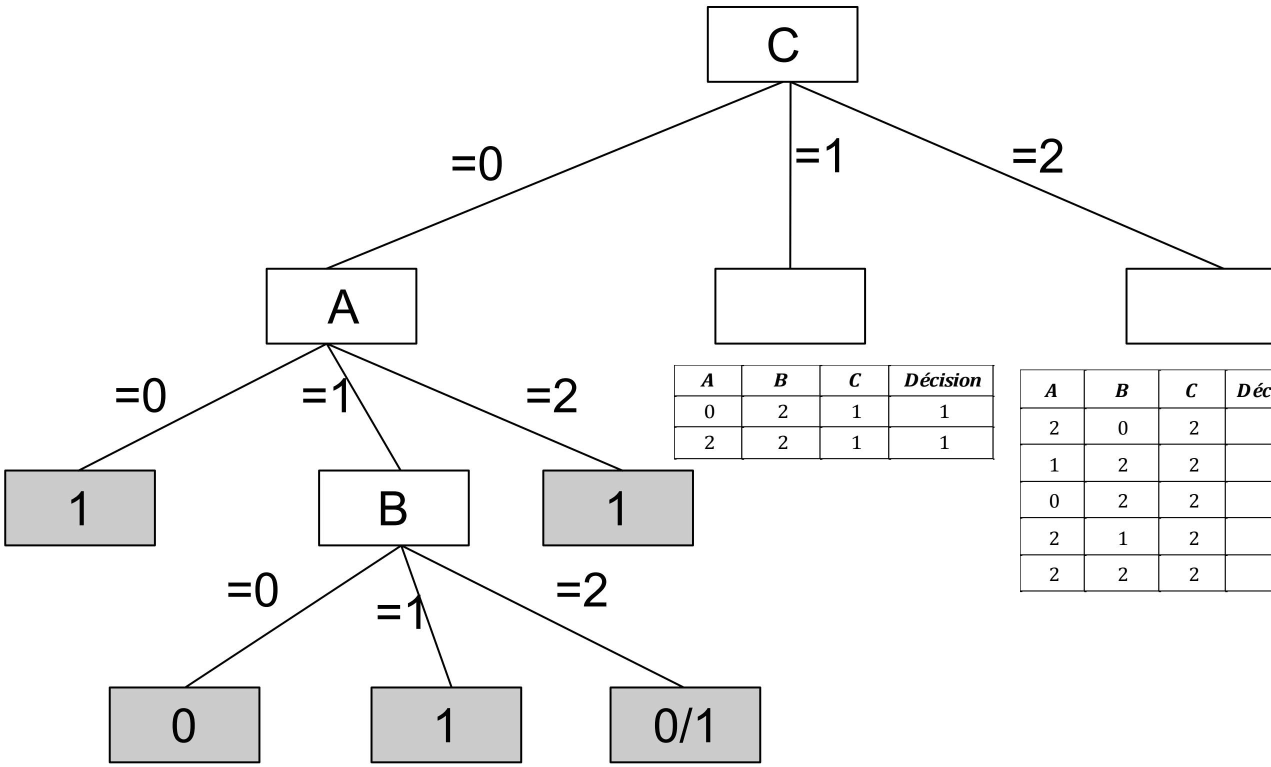
A	*B*	*C*	*Décision*
2	0	2	1
1	2	2	0
0	2	2	0
2	1	2	0
2	2	2	0

On doit choisir une valeur par défaut. Comme les deux ont les mêmes proportions, on peut prendre l'une ou l'autre

Arbre de décision – exemple (suite)

Cas C = 0 et A = 2
Aucun exemple.
Pour C = 0 la valeur majoritaire est 1
On retourne donc cette valeur par défaut:

Arbre de décision (partiel)



A	*B*	*C*	*Décision*
0	2	1	1
2	2	1	1

A	*B*	*C*	*Décision*
2	0	2	1
1	2	2	0
0	2	2	0
2	1	2	0
2	2	2	0

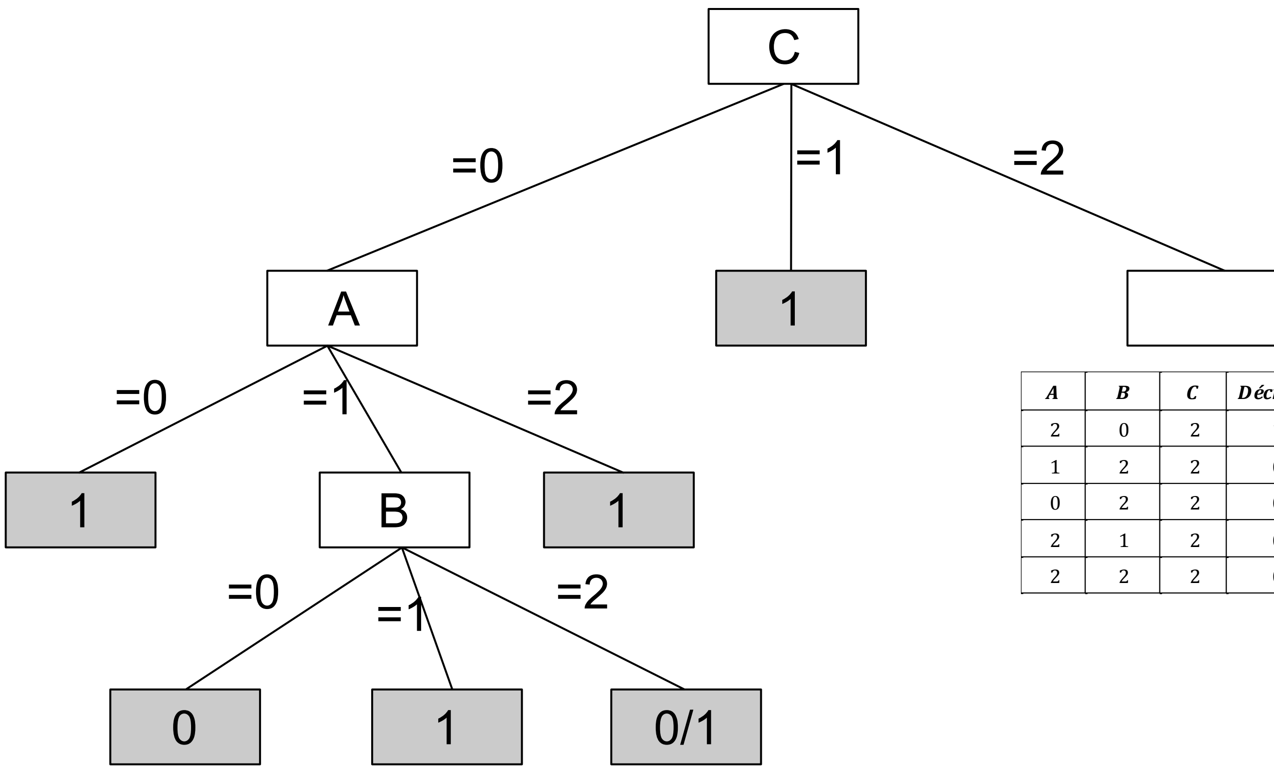
Arbre de décision – exemple (suite)

Cas C = 1

A	B	C	$Décision$
0	2	1	1
2	2	1	1

Tous les exemples sont de la même catégorie.

Arbre de décision (partiel)



A	*B*	*C*	*Décision*
2	0	2	1
1	2	2	0
0	2	2	0
2	1	2	0
2	2	2	0

Arbre de décision – exemple (suite)

Cas C = 2

A	*B*	*C*	*Décision*
2	0	2	1
1	2	2	0
0	2	2	0
2	1	2	0
2	2	2	0

Entropie initiale : 0,72

Arbre de décision – exemple (suite)

Cas C = 2

A	*B*	*C*	*Décision*
2	0	2	1
1	2	2	0
0	2	2	0
2	1	2	0
2	2	2	0

Attribut A:
Entropie si A = 0: 0
Entropie si A = 1: 0
Entropie si A = 2: 0,92
Moyenne = 0,2*0 + 0,2*0 + 0,6*0,92
= 0,55
Gain = 0,72 – 0,55 = 0,17

Arbre de décision – exemple (suite)

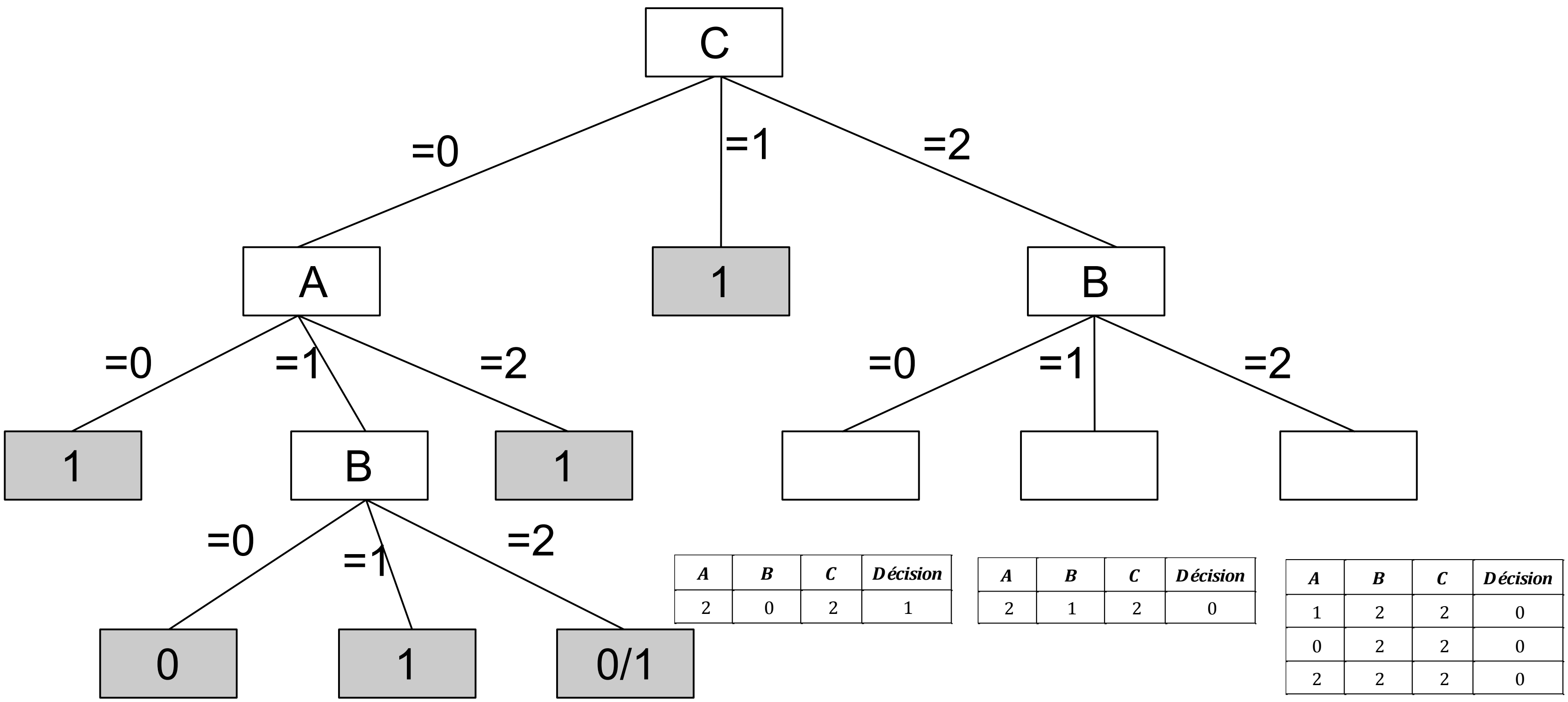
Cas C = 2

A	*B*	*C*	*Décision*
2	0	2	1
1	2	2	0
0	2	2	0
2	1	2	0
2	2	2	0

Attribut B:
Entropie si B = 0: 0
Entropie si B = 1: 0
Entropie si B = 2: 0
Moyenne = 0,2*0 + 0,2*0 + 0,6*0
= 0
Gain = 0,72 – 0 = 0,72

On choisit l'attribut B

Arbre de décision (partiel)

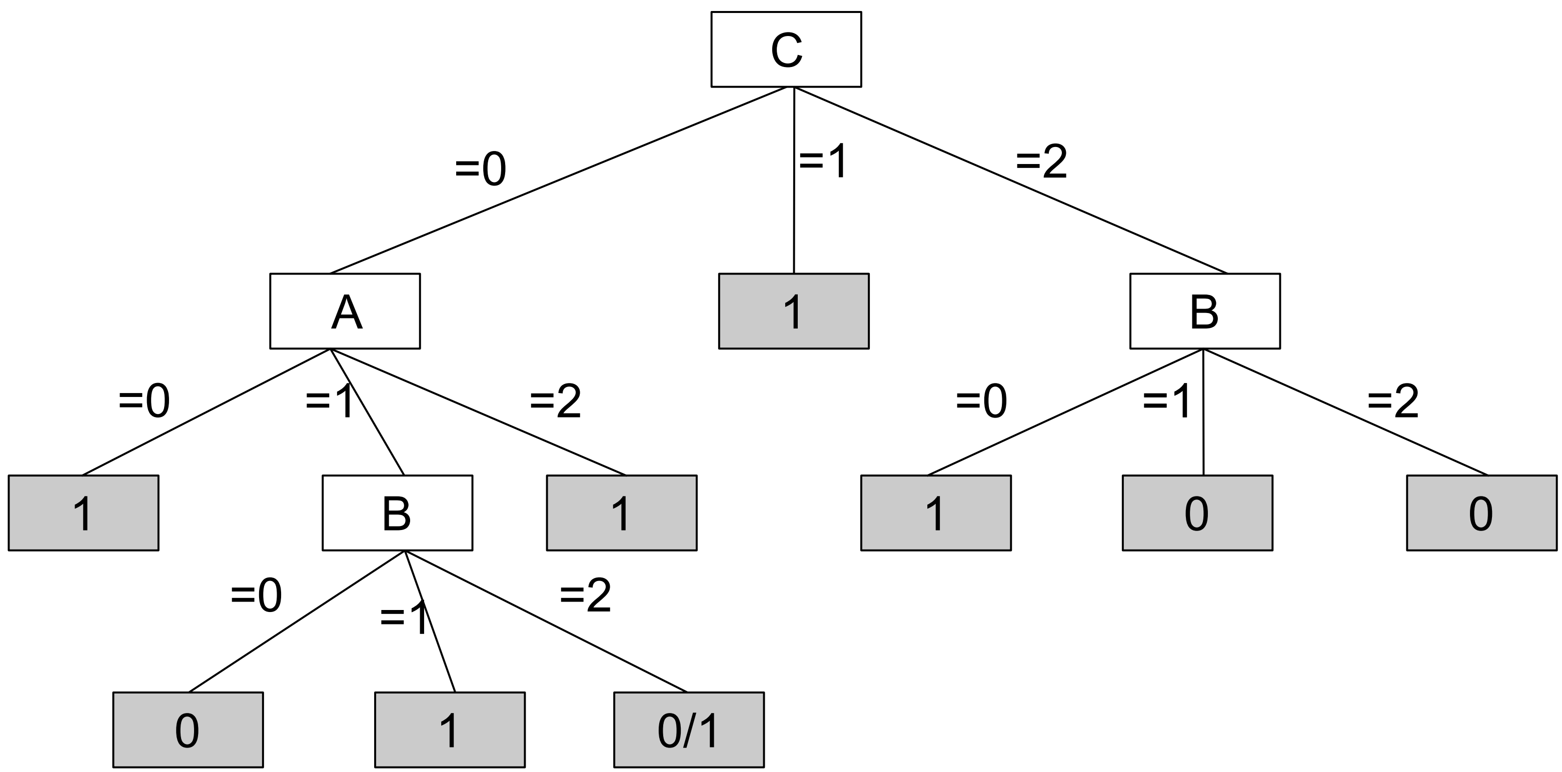


A	*B*	*C*	*Décision*
2	0	2	1

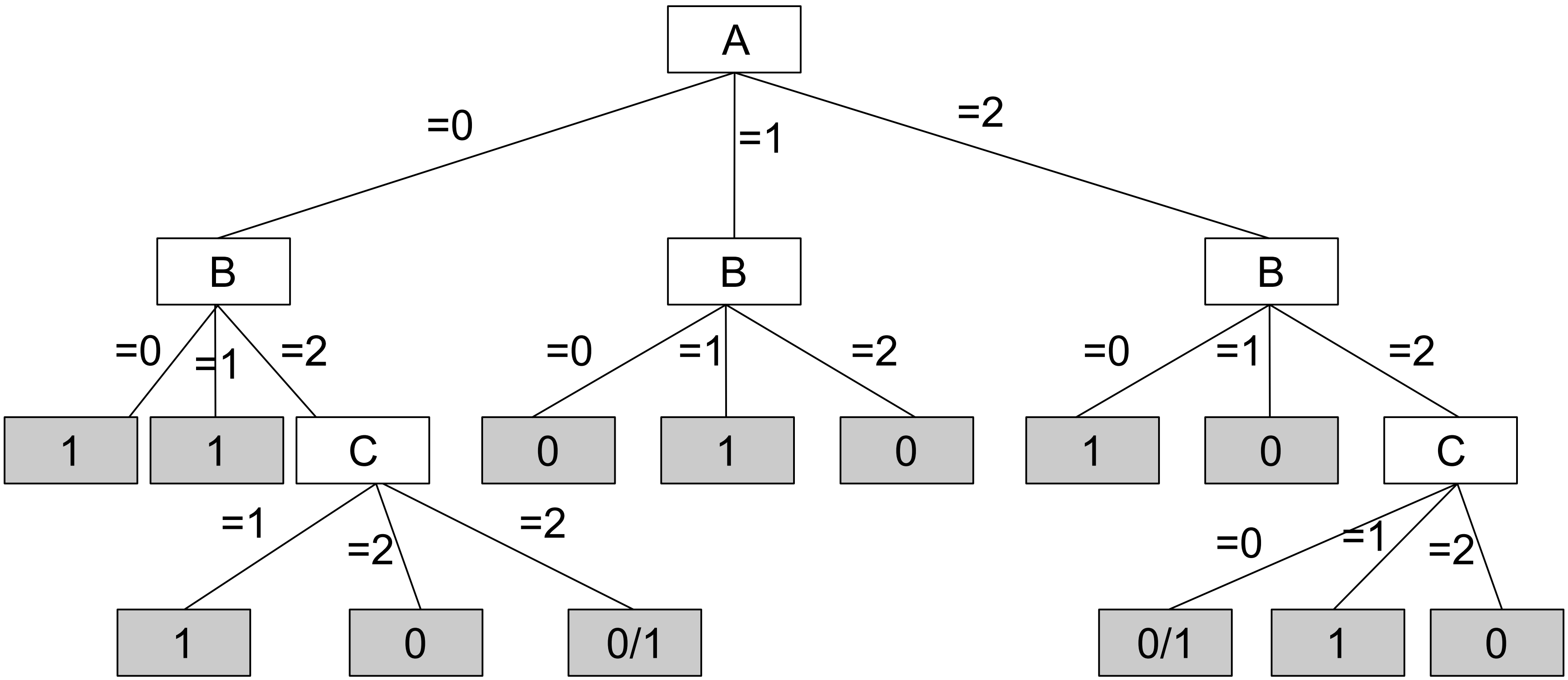
A	*B*	*C*	*Décision*
2	1	2	0

A	*B*	*C*	*Décision*
1	2	2	0
0	2	2	0
2	2	2	0

Arbre de décision final



Arbre de décision avec ordre A,B,C



On obtient un arbre plus complexe

Ratio de gain

- Problème: un attribut à plusieurs valeurs peut présenter un meilleur gain, par le simple fait de la tendance à répartir les exemples dans de petits ensembles
- Solution: diviser le gain par la valeur d'entropie de cet attribut:

$$\sum_{i=1}^{k} -(n_i / N) \, log_2(n_i / N)$$

où k est le nombre de valeurs pour l'attribut en question, N le nombre total d'exemples pour ce noeud, et n_i le nombre d'occurrences de la $i$ème valeur de l'attribut

Méthodologie d’évaluation d'apprentissage

- Collecter un grand ensemble d'exemples
- Diviser en un ensemble d'entraînement et un ensemble de **validation**
- Appliquer l'algorithme sur l'ensemble d'entraînement afin de générer l'hypothèse *h*
- Mesurer la proportion d'exemples de l'ensemble test qui sont correctement classés par l'hypothèse *h*
- Répéter avec d'autre partitions de l'ensemble d'exemples
- Important : n’ajustez pas les hyperparamètres, et ne faites pas des choix concernant quand d’arrêter la croissance d’arbre basé sur l'ensemble de **validation** !,
- Cette recette est juste pour calculer la performance moyenne sur des partitions de données

Quand stopper la subdivision?

- Première solution:
  - À chaque niveau de subdivision dans l'arbre, on teste avec l'ensemble de validation
  - Lorsque l'erreur atteint un minimum, on arrête
- Deuxième solution:
  - On fixe un seuil minimal de gain
  - Difficile de fixer cette valeur
- Troisième solution:
  - On arrête la subdivision d'un noeud lorsque le nombre d'exemples tombe en dessous d'un certain seuil (nombre absolu ou % de l'ensemble initial)

Quand stopper la subdivision? (suite)

- Quatrième solution:
  - Condition d'arrêt basée sur un critère global:

$$\alpha \cdot \text{taille} + \sum_{N \,\epsilon\, \text{feuilles}} \text{Entropie}(N)$$

  - Ici la taille peut être le nombre total de noeuds ou de branches
  - Difficile de fixer α

Quand stopper la subdivision? (suite)

- Cinquième solution:
  - Utilisation d'un test d'hypothèse pour vérifier si le gain est significatif
  - Test de chi-carré
- Sixième solution:
  - Élagage
  - On produit tout l'arbre
  - Toutes les feuilles qui sont enfants d'un même noeud sont éliminées si elles apportent peu de gain