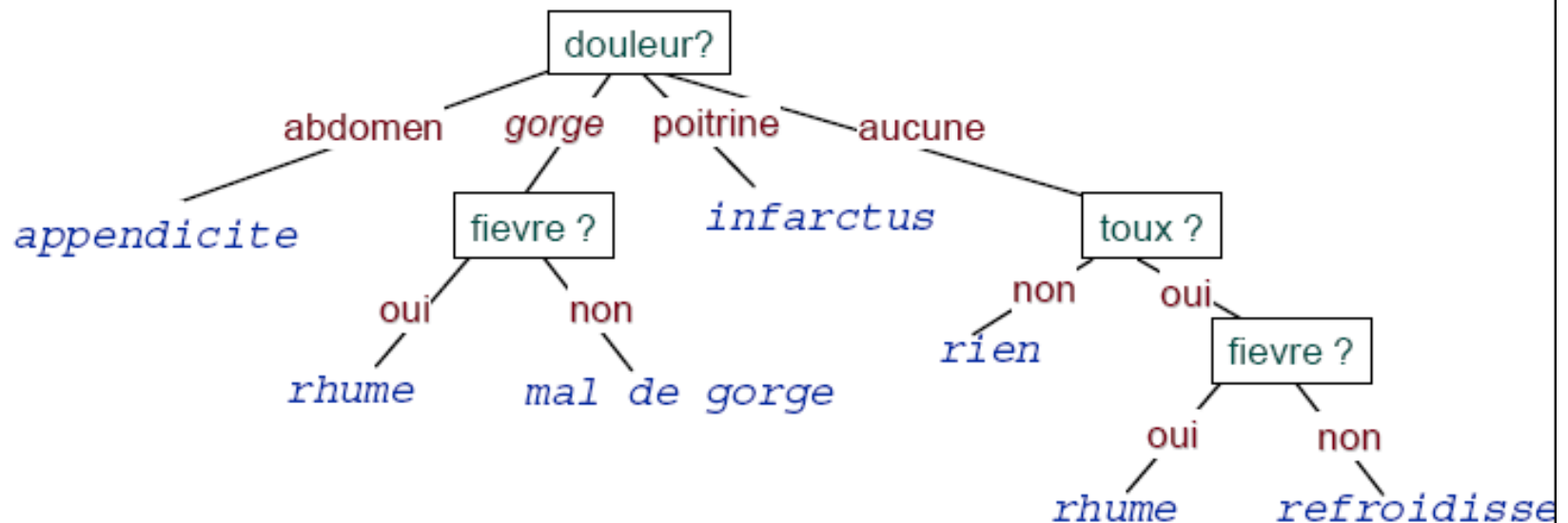


Arbres de décision

Arbres de décision

(Antoine Cornuéjols)

- Les arbres de décision sont des classifieurs pour des instances représentées dans un formalisme attribut/valeur
 - Les **noeuds** de l'arbre testent les attributs
 - Il y a une **branche** pour chaque valeur de l'attribut testé
 - Les **feuilles** spécifient les catégories (deux ou plus)




Arbres de décision : construction

- Chaque **instance** est décrite par un vecteur d'attributs/valeurs

	<u>Toux</u>	<u>Fièvre</u>	<u>Poids</u>	<u>Douleur</u>
Marie	non	oui	normal	gorge
Fred	non	oui	normal	abdomen
Julie	oui	oui	maigre	aucune
Elvis	oui	non	obese	poitrine

- En entrée** : un ensemble d'instances et leur classe (correctement associées par un “professeur” ou “expert”)

	<u>Toux</u>	<u>Fièvre</u>	<u>Poids</u>	<u>Douleur</u>
Marie	non	oui	normal	gorge
Fred	non	oui	normal	abdomen
.....				



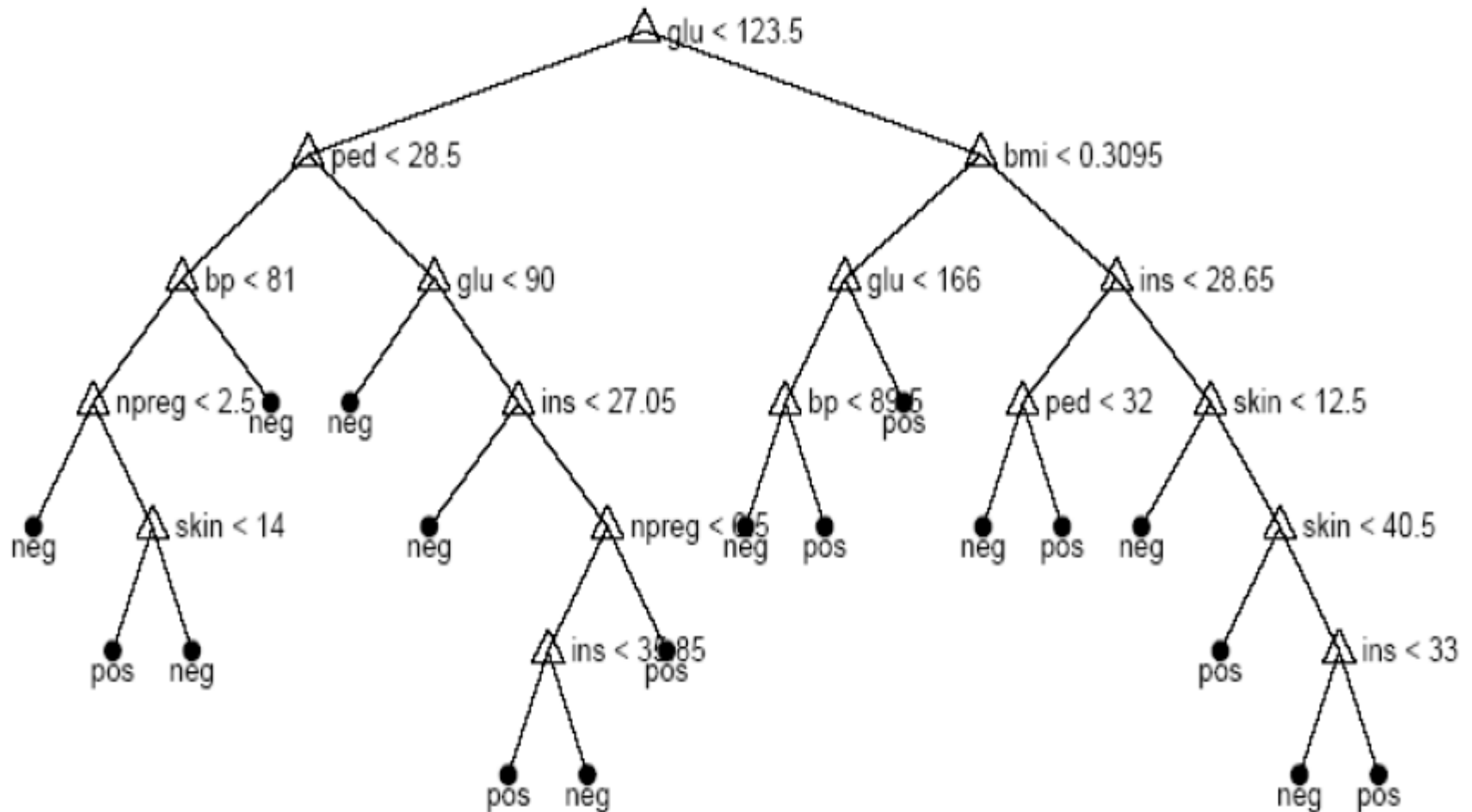
Diagnostic
rhume
appendicite

- L'algorithme d'apprentissage doit construire un **arbre de décision**

Un arbre de décision pour le diagnostic

C'est une des principales applications de l'apprentissage !

Attributs quantitatifs



Arbres de décision : choix

- Si le langage est adéquat,
il est toujours possible de construire un arbre de décision qui classe correctement les exemples d'apprentissage.
- Il y a le plus souvent de nombreux arbres de décision possibles corrects.

↪ Valeur d'un arbre ?

- Impossibilité de procéder par énumération / évaluation (NP-complet)

□ $\prod_{i=1}^A i^{V^{A-i}}$ 4 attributs & 3 valeurs / attribut : 55296 arbres

↪ *Nécessité d'une démarche constructive incrémentale*

Arbres de décision : impureté

- **Comment obtenir un arbre “simple” ?**

- ❑ **Arbre simple** : Minimise l’espérance du nombre de tests pour classer un nouvel objet
- ❑ *Comment traduire ce critère global en une procédure de choix locale ?*

- **Critères de choix de chaque noeud**

- ❑ On ne sait pas associer un critère local au critère global objectif

➡ Recours à des heuristiques

- ❑ La notion de **mesure d’“impureté”**

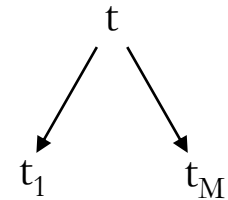
- Index Gini

- Critère entropique (ID3, C4.5, C5.0)

- ...

Construction de l'arbre

- On choisit le partitionnement s qui maximise le gain d'impureté :
- $\Delta i(s) = i(t) - \sum_{m=1}^M p_m i(t_m)$



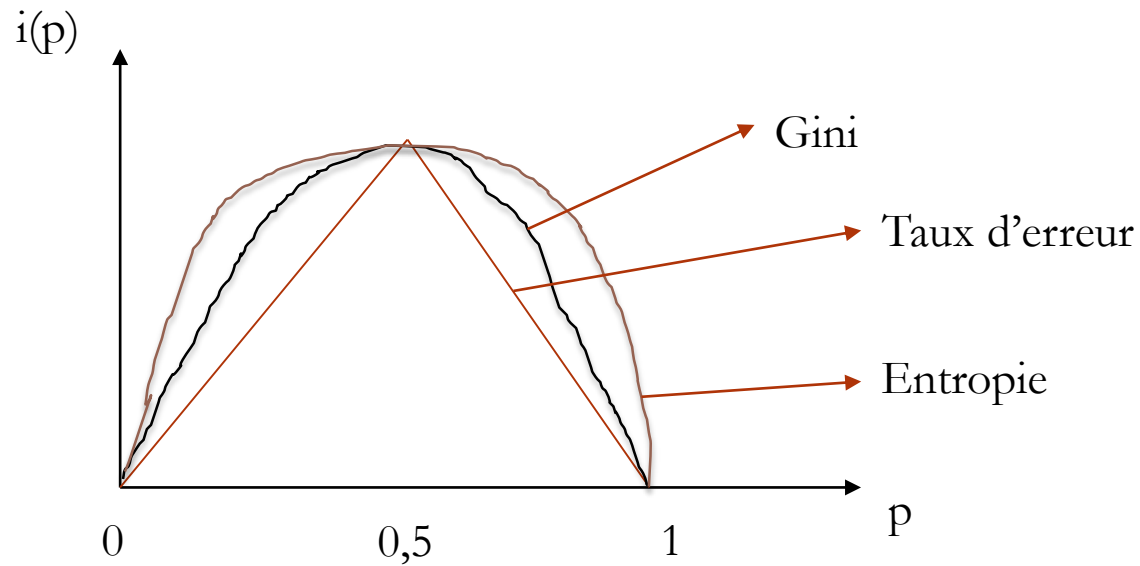
avec M : nb de modalités (2 si binaire)

p_m ou p_k = proportion d'exemples de la modalité ou classe

- Critères :
 - Taux de mal classé : $i(t) = 1 - \max_k p_k(t)$
 - Entropie : $i(t) = - \sum_{k=1}^K p_k(t) \cdot \log_2 p_k(t)$
 - Critère de Gini : $i(t) = 1 - \sum_{k=1}^K p_k^2(t)$

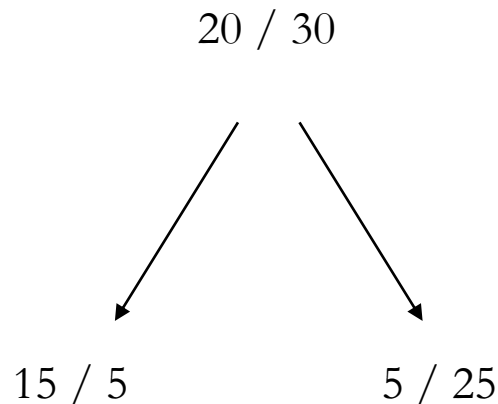
Courbe du critère

- Si $K = 2$ (2 classes)



Exemple de calcul de gain

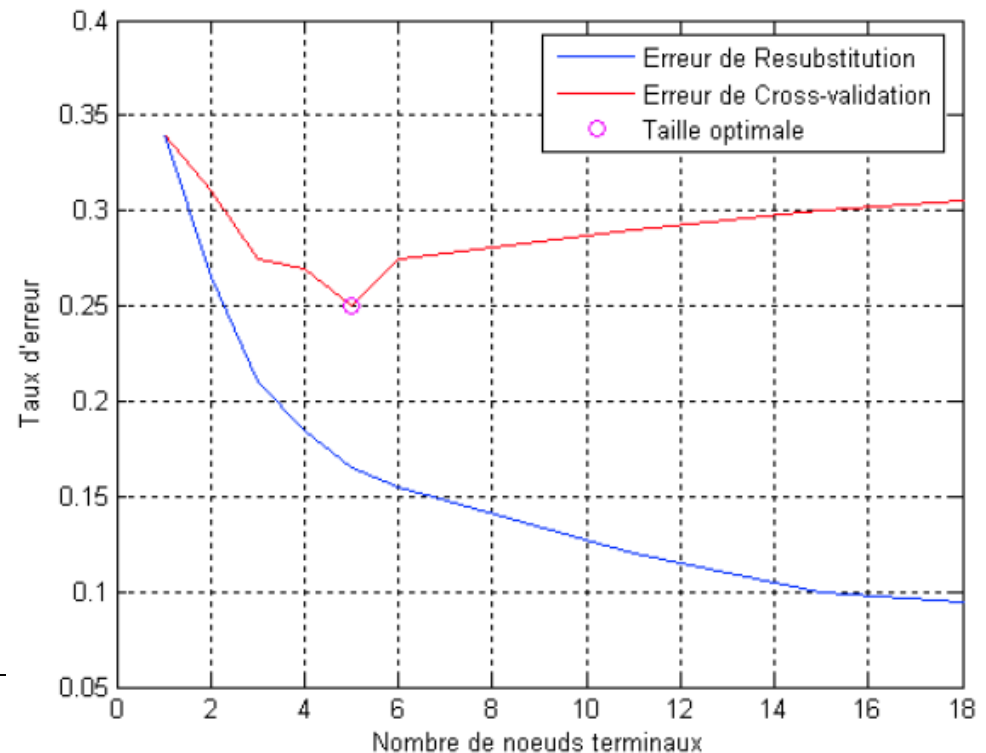
- Calculer le gain en entropie (quantité d'information)



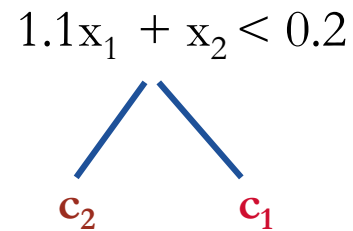
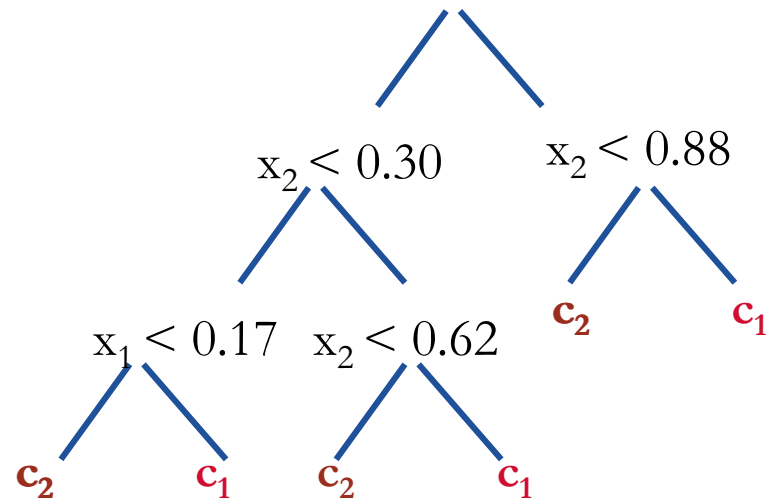
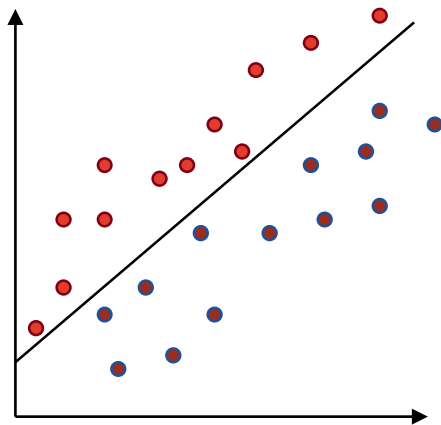
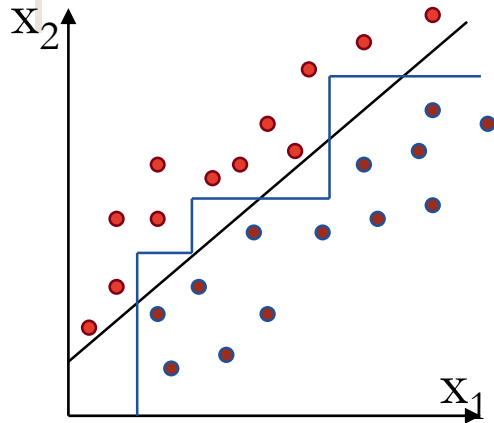
- $i(t) = -20/50 \log_2 (20/50) - 30/50 \log_2 (30/50) = 0,971$
- $i(t_1) = -15/20 \log_2 (15/20) - 5/20 \log_2 (5/20) = 0,811$
- $i(t_2) = ?$
- $\Delta i(s) = ?$

Arrêt et élagage

- Pre-pruning : on s'arrête de développer l'arbre avant d'arriver à des feuilles pures
- Post-pruning : on enlève les branches qui sont les moins utiles



Arbres multivariés



Random forests



Principes Random Forests

- On va construire un grand nombre d'arbres de décision pour un même problème
 - On prend la décision finale par un vote majoritaire
 - Le résultat final sera bon si les arbres sont différents (complémentaires) et efficaces
-



Arbres différents

- On ajoute de l'aléatoire dans la construction des arbres
 - ▶ Bagging (Adaboost) : A partir d'une base d'apprentissage, on tire aléatoirement des sous-bases d'exemples
 - ▶ A partir de toutes les M variables décrivant les exemples, on tire aléatoirement à chaque nœud de l'arbre les K qui vont servir
 - ▶ ...
-



Construction des arbres

- Les arbres de décision sont alors construits automatiquement sans pre- ou post-pruning
 - C'est très rapide car simple et peu de caractéristiques
 - Choix de K : essais avec 1 ou \sqrt{M} ou $\log M + 1$
On peut prendre K aléatoire entre 1 et M pour éviter un paramètre
-



Performances (Breiman 2001)

Data set	Adaboost	Selection	Forest-RF single input	One tree
Glass	22.0	20.6	21.2	36.9
Breast cancer	3.2	2.9	2.7	6.3
Diabetes	26.6	24.2	24.3	33.1
Sonar	15.6	15.9	18.0	31.7
Vowel	4.1	3.4	3.3	30.4
Ionosphere	6.4	7.1	7.5	12.7
Vehicle	23.2	25.8	26.4	33.1
German credit	23.5	24.4	26.2	33.3
Image	1.6	2.1	2.7	6.4
Ecoli	14.8	12.8	13.0	24.5
Votes	4.8	4.1	4.6	7.4
Liver	30.7	25.1	24.7	40.6
Letters	3.4	3.5	4.7	19.8
Sat-images	8.8	8.6	10.5	17.2
Zip-code	6.2	6.3	7.8	20.6
Waveform	17.8	17.2	17.3	34.0
Twonorm	4.9	3.9	3.9	24.7
Threenorm	18.8	17.5	17.5	38.4
Ringnorm	6.9	4.9	4.9	25.7



Performances au bruit

- Random Forests moins sensibles au bruit sur les classes de sorties que Adaboost
 - Pas très sensibles non plus aux caractéristiques peu informantes
-



Sélection de variables

- A partir des arbres construits, on peut en déduire une hiérarchisation des variables
 - D'où une possibilité de sélection de variables
 - Amélioration des performances et des temps de traitement d'autres classificateurs comme les kPPV ou SVM
-

Méthodes structurelles

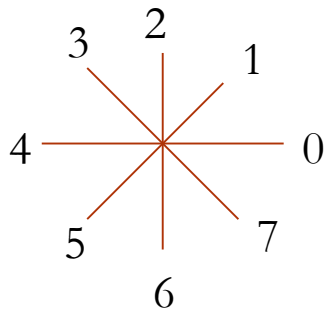


Méthodes structurelles

- Description des formes à l'aide de primitives et de liens entre primitives
- Outils utilisés : chaînes, **graphes** et grammaires

Chaînes

- Longueur fixe : la position de la primitive dans la chaîne a une signification
- Longueur variable : chaîne d'un seul type de primitive
- Exemple : code de Freeman



Distance élastique

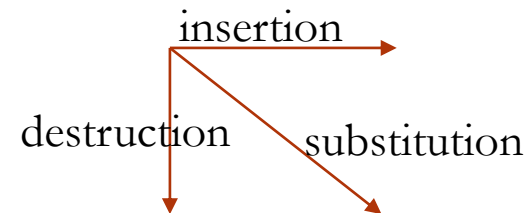
- A et B deux chaînes, $A = x_1x_2\dots x_n$, $B = y_1y_2\dots y_m$
- $d(A, B) = d(n, m)$, $d(2, 3) = d(x_1x_2, y_1y_2y_3)$
- $d(x, y) = 1$ si $x \neq y$
 $= 0$ si $x = y$
- $d(1, 1) = d(x_1, y_1)$
 $d(i, j) = \min [d(i-1, j), d(i, j-1), d(i-1, j-1)] + d(x_i, y_j)$
pour $2 \leq i \leq n$, $2 \leq j \leq m$

- Exemple :

A / B	a	b	b	b	c	c
a	0	1	2	3	4	5
a	0	1	2	3	4	5
b	1	0	0	0	1	2
c	2	1	1	1	0	0
b	3	1	1	1	1	1
c	4	2	2	2	1	1

Distance d'édition

- On cherche la transformation de coût minimum qui permette de passer de la chaîne A à la chaîne B
- On a un ensemble de transformation avec le coût associé. Par exemple :
 - Insertion : $\varepsilon \rightarrow a$. Coût w_i
 - Destruction : $a \rightarrow \varepsilon$. Coût w_d
 - Substitution : $a \rightarrow b$. Coût w_s
- Algo
 - $d(0, j) = j \cdot w_i$
 - $d(i, 0) = i \cdot w_d$
 - $d(i, j) = \min[d(i-1, j) + w_d, d(i, j-1) + w_i, d(i-1, j-1) + w_s]$

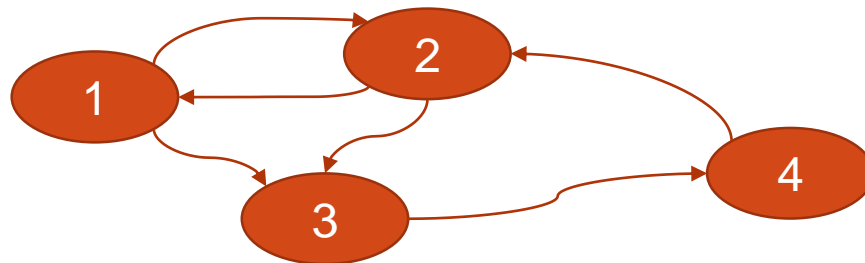


Exemple distance d'édition

A / B	a	b	b	b	c	c
a	0	1	2	3	4	5
a	0	1	2	3	4	5
b	1	0	0	0	1	2
c	2	1	1	1	0	0
b	3	1	1	1	1	1
c	4	2	2	2	1	1

Graphes

- $G(X, U)$ avec X : nœuds et U : arêtes ou arcs
- En RF, nœuds : primitives et arcs : liens entre primitives
- $G(X, U)$: structure associée à une forme
- Graphe de formes : graphe attribué
- Exemple : $X = \{ 1, 2, 3, 4 \}$
 $U = \{ (1,2), (2,1), (1,3), (2,3), (3,4), (4,2) \}$





Classification avec graphes

- Apprentissage : choisir des graphes prototypes (plusieurs graphes de formes représentatifs des classes)
- Reconnaissance : Quantifier la distance / ressemblance entre les graphes à reconnaître et les graphes prototypes. Puis kPPV avec cette distance.



Isomorphisme

- Isomorphisme de graphe :
 G_1 et G_2 sont isomorphes s'il existe une bijection f entre X_1 et X_2 qui vérifie :
 $\forall x_1 \in X_1 \text{ et } \forall x_2 \in X_2, (x_1, x_2) \in U_1 \Leftrightarrow (f(x_1), f(x_2)) \in U_2$
C'est un isomorphisme entre structures
 G_1 et G_2 doivent avoir le même ordre (nb nœuds).
- Pour des graphes de formes, il faut en plus que les valeurs des nœuds et des arcs se correspondent.
- Cette notion est peu utilisable en RF

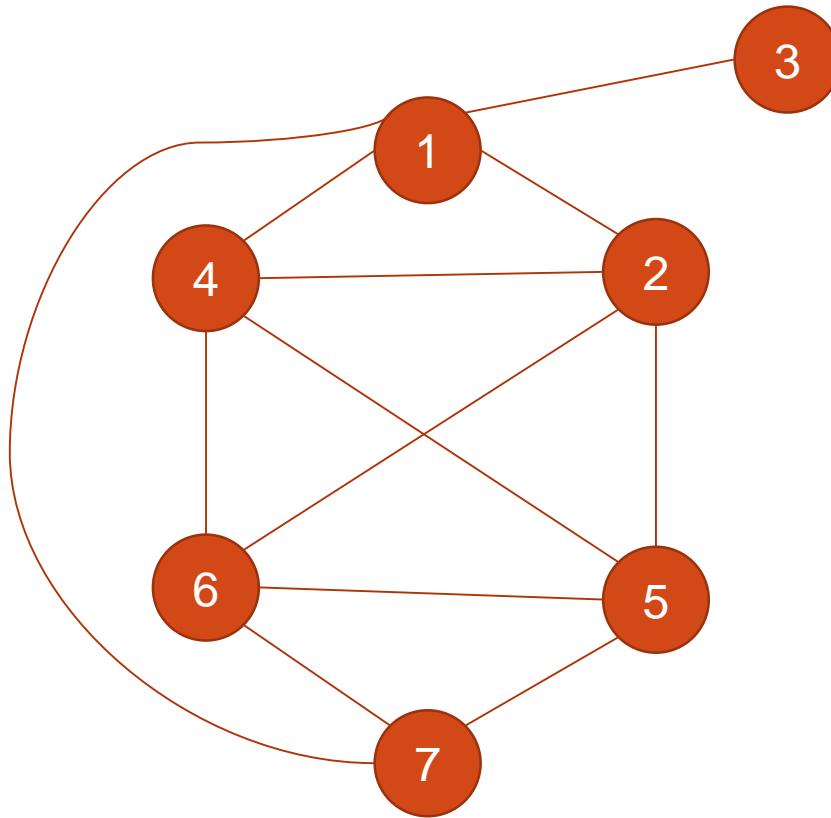


Distance entre 2 graphes

- Méthodes basées Noyau
 - Apprenables
 - Sensibles à de petites modifications
- GED : Graph Edit Distance
 - Pas simple à calculer
 - Résultats satisfaisants

Clique

- Sous-graphes en relation complète





Méthodes grammaticales

- Grammaire
 - Alphabet (A)
 - symboles Internes (I)
 - symboles de départ ou Start (S)
 - règles de Production (P)
- Le langage est généré par une grammaire.
 $L (G (A, I, S, P))$ est l'ensemble des chaînes générables par G.

Grammaires pour la RF

- Apprentissage : inférence grammaticale
Exemples d'une classe $\xrightarrow{\text{Apprentissage}}$ Grammaire
Grammaire non unique
Utilisation d'exemples positifs et négatifs D+ et D-
Les exemples négatifs ne sont pas dérivables
- Reconnaissance :
soit une forme x , chercher G parmi $G_1 \dots G_K$, telle
que $L(G) = x$ (parsing)
 $S \rightarrow P \rightarrow \dots \rightarrow x$
 $x \rightarrow P^{-1} \rightarrow \dots \rightarrow S$