

Apprentissage d'Ensembles de Règles de Classification par Analyse de Concepts Formels

Nida MEDDOURI¹

¹École Nationale Supérieure des Mines de Saint-Etienne
29 Rue des Frères Ponchardier 42 023 Saint-Etienne.

Novembre 2020

Plan :

- 1 Introduction
- 2 Analyse de Concepts Formels
 - Pourquoi l'Analyse de Concepts Formels ?
 - Apprentissage à partir d'un treillis de concepts
 - Apprentissage à partir d'un demi/semi treillis de concepts
 - CFC & CNC
- 3 Les méthodes d'ensembles
 - Apprentissage séquentiel
 - Apprentissage parallèle
- 4 Conclusion

Plan :

- 1 Introduction
- 2 Analyse de Concepts Formels
 - Pourquoi l'Analyse de Concepts Formels ?
 - Apprentissage à partir d'un treillis de concepts
 - Apprentissage à partir d'un demi/semi treillis de concepts
 - CFC & CNC
- 3 Les méthodes d'ensembles
 - Apprentissage séquentiel
 - Apprentissage parallèle
- 4 Conclusion

Volume de données numériques créées dans le monde depuis 2010 (en zettaoctets) *

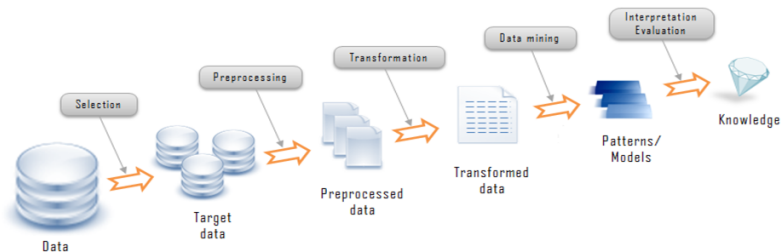


@Statista_FR

* prévisions de 2020 à 2035.

Source : Statista Digital Economy Compass 2019

JDN statista





CLASSICAL MACHINE LEARNING

Data is pre-categorized
or numerical

SUPERVISED

Predict
a category

CLASSIFICATION

«Divide the socks by color»



Predict
a number

REGRESSION

«Divide the ties by length»



Data is not labeled
in any way

UNSUPERVISED

Divide
by similarity

CLUSTERING

«Split up similar clothing
into stacks»



Identify sequences

Find hidden
dependencies

ASSOCIATION

«Find what clothes I often
wear together»

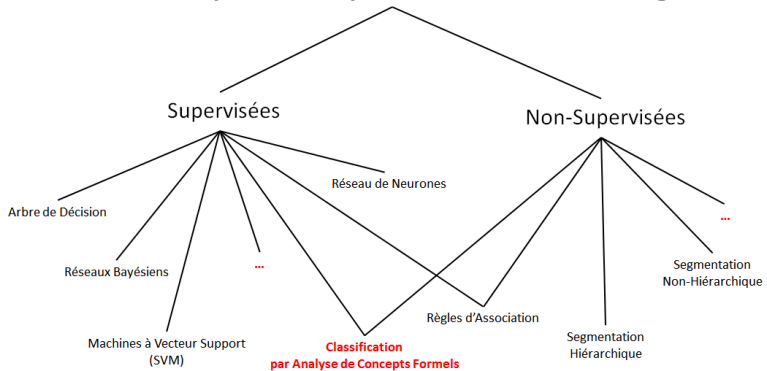


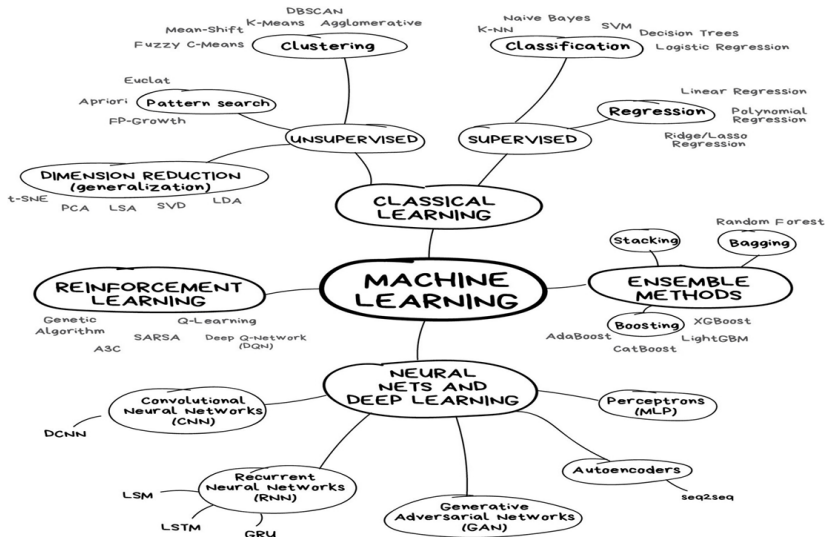
DIMENSION REDUCTION (generalization)

«Make the best outfits from the given clothes»



Techniques Classiques du Machine Learning





Plan :

- 1 Introduction
- 2 **Analyse de Concepts Formels**
 - Pourquoi l'Analyse de Concepts Formels ?
 - Apprentissage à partir d'un treillis de concepts
 - Apprentissage à partir d'un demi/semi treillis de concepts
 - CFC & CNC
- 3 Les méthodes d'ensembles
 - Apprentissage séquentiel
 - Apprentissage parallèle
- 4 Conclusion

Choix de l'Analyse de Concepts Formels

- Traitement de **grandes quantités** de données.
- Extraction de connaissances **cachées**.
- Un concept : **une forme de modélisation** d'un objet avec ses attributs.
- Treillis de concepts formels : **richesse sémantique**.
- **Simplifier la prédiction** des classes.
- Modèle d'apprentissage : **Explicable & Interprétable**.

Les fondements :

Contexte Formel

Un contexte formel est un triplet $\langle O, \mathcal{P}, \mathcal{R} \rangle$:

- O : Ensemble fini d'objets (instances, exemples).
- \mathcal{P} : Ensemble fini de propriétés (attributs).
- \mathcal{R} : relation binaire définie entre O et \mathcal{P} .

O-P	p ₁	p ₂	p ₃	p ₄	p ₅	p ₆	p ₇	p ₈	CLASSE
o ₁	1	1	1	1	1	1	1	0	1
o ₂	1	1	1	1	1	1	0	1	1
o ₃	1	1	1	1	1	0	1	1	1
o ₄	1	1	1	1	0	1	0	0	1
o ₅	1	1	0	1	1	0	1	0	2
o ₆	1	1	1	0	1	0	0	1	2
o ₇	1	0	1	0	0	1	0	0	2

Table – Illustration d'un contexte formel (binaire).

Correspondance & Fermeture de Galois

$X \subseteq O$ et $Y \subseteq P$ deux ensembles finis.

Correspondance/Connexion de Galois :

- $\varphi(X) := \{ p \mid \forall o, o \in X \text{ et } (o,p) \in \mathcal{R} \}.$
- $\delta(Y) := \{ o \mid \forall p, p \in Y \text{ et } (o,p) \in \mathcal{R} \}.$

Opérateurs de Fermeture :

- $X'' = \delta \circ \varphi(X).$
- $Y'' = \varphi \circ \delta(Y).$

On dit qu'un ensemble est fermé s'il est égal à sa fermeture.
Ainsi, X est fermé si $X=X''$ et Y est fermé si $Y=Y''$.

Concept Formel, Sous-concept, Super-concept

Un concept formel (X, Y) :

- X : l'extension du concept formel.
- Y : l'intention du concept formel.
- $\varphi(X) = Y$ et $\delta(Y) = X$.

Sous-concept, Super-concept

- $C_1 = (X_1, Y_1)$
- $C_2 = (X_2, Y_2)$

C_1 est un sous-concept de C_2 et on note $C_1 \leq C_2$ si et seulement si $X_1 \subseteq X_2$. Dans ce cas, C_2 est un super-concept de C_1 .

Cette relation d'ordre sur les concepts peut être définie de manière équivalente à partir de l'inclusion des intensions.

O-P	p₁	p ₂	p ₃	p ₄	p ₅	p ₆	p ₇	p ₈	CLASSE
o₁	1	1	1	1	1	1	1	0	1
o₂	1	1	1	1	1	1	0	1	1
o₃	1	1	1	1	1	0	1	1	1
o₄	1	1	1	1	0	1	0	0	1
o₅	1	1	0	1	1	0	1	0	2
o₆	1	1	1	0	1	0	0	1	2
o₇	1	0	1	0	0	1	0	0	2

Exemple des concepts formels

$$C_1 : (\{o_1, o_2, o_3, o_4, o_5, o_6, o_7\}, \{p_1\})$$

O-P	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	CLASSE
o_1	1	1	1	1	1	1	1	0	1
o_2	1	1	1	1	1	1	0	1	1
o_3	1	1	1	1	1	0	1	1	1
o_4	1	1	1	1	0	1	0	0	1
o_5	1	1	0	1	1	0	1	0	2
o_6	1	1	1	0	1	0	0	1	2
o_7	1	0	1	0	0	1	0	0	2

Exemple des concepts formels

$C_1 : (\{o_1, o_2, o_3, o_4, o_5, o_6, o_7\}, \{p_1\})$

$C_2 : (\{o_1, o_2, o_3, o_4, o_5, o_6\}, \{p_1, p_2\})$

O-P	p ₁	p ₂	p ₃	p ₄	p ₅	p ₆	p ₇	p ₈	CLASSE
o ₁	1	1	1	1	1	1	1	0	1
o ₂	1	1	1	1	1	1	0	1	1
o ₃	1	1	1	1	1	0	1	1	1
o ₄	1	1	1	1	0	1	0	0	1
o ₅	1	1	0	1	1	0	1	0	2
o ₆	1	1	1	0	1	0	0	1	2
o ₇	1	0	1	0	0	1	0	0	2

Exemple des concepts formels

$C_1 : (\{o_1, o_2, o_3, o_4, o_5, o_6, o_7\}, \{p_1\})$

$C_2 : (\{o_1, o_2, o_3, o_4, o_5, o_6\}, \{p_1, p_2\})$

$C_3 : (\{o_1, o_2, o_3, o_4, o_6, o_7\}, \{p_1, p_3\})$

Treillis de Galois I

Ensemble ordonné

Un ordre partiel sur un ensemble E est une relation binaire ' \leq ' sur les éléments de E , tel que pour $(x,y,z) \in E$:

- *Réflexivité* : $x \leq x$
- *Anti-symétrie* : $x \leq y$ et $y \leq x \Rightarrow x = y$
- *Transitivité* : $x \leq y$ et $y \leq z \Rightarrow x \leq z$

Un ensemble E doté d'une relation d'ordre \leq , noté (E, \leq) , est appelé *ensemble partiellement ordonné*.

Majorant, Minorant

Soit (E, \leq) un ensemble ordonné et S une partie de E . On appelle majorant (respectivement minorant) de S tout élément $a \in E$, tel que :

$$\forall x \in S, x \leq a \text{ (respectivement } x \geq a) \quad (1)$$

Treillis de Galois II

Supremum, Infimum

Soit (E, \leq) un ensemble ordonné et S une partie de E . On appelle supremum ou borne supérieure (respectivement infimum ou borne inférieure) de S , s'il existe, et on note $\vee S$ (respectivement $\wedge S$) le plus petit des majorants (respectivement le plus grand des minorants) de S .

Pour une paire (x, y) d'éléments de E , le majorant $\vee \{x, y\}$ (respectivement le minorant $\wedge \{x, y\}$) est noté $x \vee y$ (respectivement $x \wedge y$).

Relation inverse

Soit (E, \leq) un ensemble ordonné. On appelle relation inverse de \leq et on note \geq la relation définie par :

$$\forall (x, y) \in E^2, x \leq y \Leftrightarrow y \geq x. \quad (2)$$

Treillis de Galois III

Principe de dualité

Soit \geq la relation inverse de \leq . Toute assertion sur \leq , \wedge , \vee reste vraie en remplaçant \leq par \geq et en permutant \wedge et \vee .

Si (E, \leq) est un ensemble ordonné, alors (E, \geq) est également un ensemble ordonné. Le principe de dualité permet de déduire les propriétés duales de (E, \geq) à partir des propriétés de (E, \leq)

Successeur direct, Prédécesseur direct

Soit (E, \leq) un ensemble ordonné et a et b deux éléments de E . a est un successeur direct de b et on note $a \prec b$ si $a < b$ et s'il n'existe pas d'élément $c \in E$ tel que $a < c < b$. Dans ce cas, b est le prédécesseur direct de a .

Treillis

Un ensemble ordonné (E, \leq) est un treillis si pour tout x et y de E , $x \vee y$ et $x \wedge y$ existent.

Treillis de Galois IV

Treillis complet

Un ensemble ordonné (E, \leq) est un treillis complet si pour toute partie S de E , $\vee S$ (un supremum de S) et $\wedge S$ (un infimum de S) existent.

Demi-Treillis

Un ensemble ordonné (E, \leq) est un sup-demi-treillis (respectivement inf-demi-treillis) si tout couple d'éléments $(x, y) \in E$ admet un supremum $x \vee y$ (respectivement un infimum $x \wedge y$).

$C_1 : (\{o_1, o_2, o_3, o_4, o_5, o_6, o_7\}, \{p_1\})$

$$C_1 : (\{o_1, o_2, o_3, o_4, o_5, o_6, o_7\}, \{p_1\})$$
$$C_2 : (\{o_1, o_2, o_3, o_4, o_5, o_6\}, \{p_1, p_2\})$$

$C_1 : (\{o_1, o_2, o_3, o_4, o_5, o_6, o_7\}, \{p_1\})$
 $C_2 : (\{o_1, o_2, o_3, o_4, o_5, o_6\}, \{p_1, p_2\})$
 $C_3 : (\{o_1, o_2, o_3, o_4, o_6, o_7\}, \{p_1, p_3\})$

$C_1 : (\{o_1, o_2, o_3, o_4, o_5, o_6, o_7\}, \{p_1\})$

$C_2 : (\{o_1, o_2, o_3, o_4, o_5, o_6\}, \{p_1, p_2\})$

$C_3 : (\{o_1, o_2, o_3, o_4, o_6, o_7\}, \{p_1, p_3\})$

$C_4 :$

$C_5 :$

$C_6 :$

$C_7 :$

...

$C_{16} :$

$C_{17} :$

$C_1 : (\{o_1, o_2, o_3, o_4, o_5, o_6, o_7\}, \{p_1\})$
 $C_2 : (\{o_1, o_2, o_3, o_4, o_5, o_6\}, \{p_1, p_2\})$
 $C_3 : (\{o_1, o_2, o_3, o_4, o_6, o_7\}, \{p_1, p_3\})$
 $C_4 :$
 $C_5 :$
 $C_6 :$
 $C_7 :$
...
 $C_{16} :$
 $C_{17} :$
 $C_{18} : (\{o_1\}, \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\})$

$C_1 : (\{o_1, o_2, o_3, o_4, o_5, o_6, o_7\}, \{p_1\})$
 $C_2 : (\{o_1, o_2, o_3, o_4, o_5, o_6\}, \{p_1, p_2\})$
 $C_3 : (\{o_1, o_2, o_3, o_4, o_6, o_7\}, \{p_1, p_3\})$
 $C_4 :$
 $C_5 :$
 $C_6 :$
 $C_7 :$
...
 $C_{16} :$
 $C_{17} :$
 $C_{18} : (\{o_1\}, \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\})$
 $C_{19} : (\{o_2\}, \{p_1, p_2, p_3, p_4, p_5, p_6, p_8\})$

$C_1 : (\{o_1, o_2, o_3, o_4, o_5, o_6, o_7\}, \{p_1\})$
 $C_2 : (\{o_1, o_2, o_3, o_4, o_5, o_6\}, \{p_1, p_2\})$
 $C_3 : (\{o_1, o_2, o_3, o_4, o_6, o_7\}, \{p_1, p_3\})$
 $C_4 :$
 $C_5 :$
 $C_6 :$
 $C_7 :$
 \dots
 $C_{16} :$
 $C_{17} :$
 $C_{18} : (\{o_1\}, \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\})$
 $C_{19} : (\{o_2\}, \{p_1, p_2, p_3, p_4, p_5, p_6, p_8\})$
 $C_{20} : (\{o_3\}, \{p_1, p_2, p_3, p_4, p_5, p_7, p_8\})$

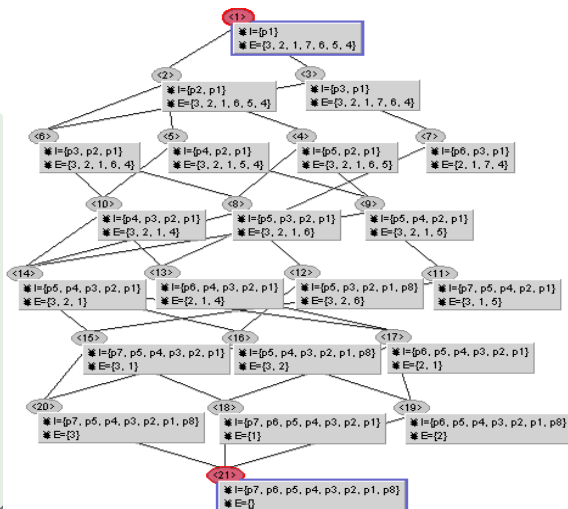


Figure – Le treillis de Galois du contexte initial (Table

Classification à partir d'un treillis de concepts

Déterminer la classe des nouveaux objets.

Le treillis de Galois est vu comme un espace de recherche.

Complexité *exponentielle* (temporelle et spatiale) pour la génération du treillis.

La navigation dans un *large* espace de recherche devient *difficile* et *dure*.

Classification à partir d'un treillis de concepts

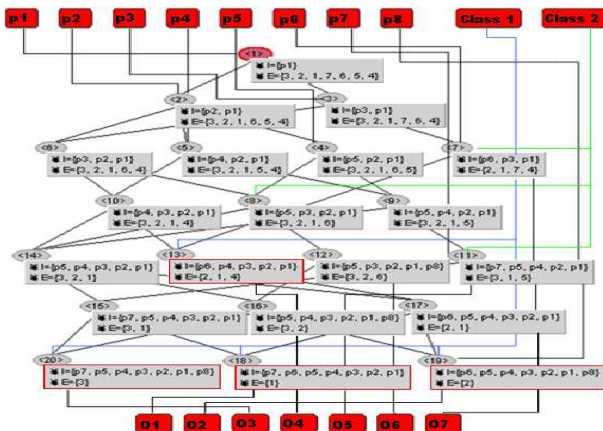


Figure – Le pseudo-treillis des concepts construit par GRAND

Classification à partir d'un demi/semi treillis de concepts

- Construire uniquement une partie du treillis.
- Classification similaire à celle entamée à partir d'un treillis.
- Réduction du complexité théorique et du temps d'exécution.

CITREC construit un treillis complet à partir d'un contexte réduit.

Une instance représentative de chaque classe.

IPR construit une couverture de concepts formels (pertinents).

Construction des concepts redondants.

Classification à partir d'un semi treillis de concepts :

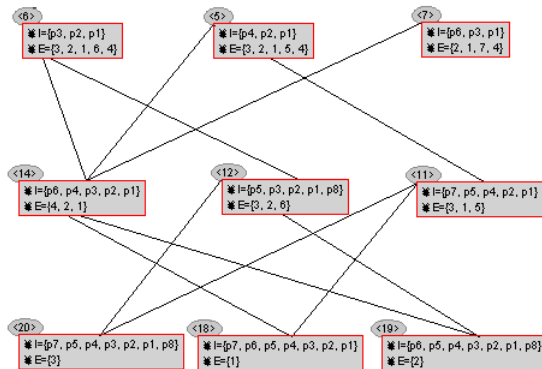


Figure – La couverture des concepts construit par IPR

Problématique

- Nombre **élevé** de connaissances extraites.
- **Pertinence** de connaissances extraites.
- **Perte** d'information.
- **Absence** de l'aspect adaptatif.
- **Sur**-apprentissage.
- **Performances**.

Notez Bien !

L'apprentissage supervisée pas FCA

À partir d'un treillis complet

- ⇒ **Complexité exponentielle.**
- ⇒ **Navigation : Difficile & Dure.**
- ⇒ **Construction exhaustive.**

À partir d'un demi treillis / couverture

- ⇒ **Construction non contextuelle.**

CFC : Classifier Formal Concept

Quel CFC ?

- Variante basée sur la recherche du meilleur pseudo-concept.
- Variante basée sur la fermeture de *Galois*.

Données :

- Soit $O = \{(o_1, y_1), \dots, (o_n, y_n)\}$ un échantillon de données étiquetées tel que $y_i \in Y = \{1, \dots, K\}$.

Résultat :

- h_{CFC} : Une règle de classification.

Début

A partir de O , déterminer l'attribut (binaire) le plus discriminant (Entropie) p^* ;
Calculer la fermeture associée $(\delta(\{p^*\}), \delta \circ \varphi(\{p^*\}))$;
Déterminer la classe majoritaire y^* associée à ce concept pertinent;
Déduire la règle de classification h_{CNC} ;

Fin

Algorithme 1: Algorithme d'apprentissage d'un concept formel basé sur la Fermeture de *Galois*

↑↑ **Données binaires.**

CNC : Classifier Nominal Concept

Quel CNC ?

- Variante basée sur le *Gain Informationnel*.
- Variante basée sur le *H-Ratio (2017)*.

Données :

- Soit $O = \{(o_1, y_1), \dots, (o_n, y_n)\}$ un échantillon de données nominales étiquetées tel que $y_i \in Y = \{1, \dots, K\}$.

Résultat :

- h_{CNC} : Une règle de classification.

Début

A partir de O , déterminer l'attribut (nominal) le plus discriminant (Gain Informationnel) p^* ;
Calculer la fermeture associée $(\delta(\{p^*\}), \delta \circ \varphi(\{p^*\}))$;
Déterminer la classe majoritaire y^* associée à ce concept pertinent;
Déduire la règle de classification h_{CNC} ;

Fin

Algorithme 2: Algorithme d'apprentissage d'un concept nominal basé sur la Fermeture de Galois

Exemple illustratif :

CNC

Rappel : Extension aux données nominales de la méthode *CFC*.

	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Windy</i>	<i>Play</i>
<i>o</i> ₁	Sunny	Hot	High	False	No
<i>o</i> ₂	Sunny	Hot	High	True	No
<i>o</i> ₃	Overcast	Hot	High	False	Yes
<i>o</i> ₄	Rainy	Mild	High	False	Yes
<i>o</i> ₅	Rainy	Cool	Normal	False	Yes
<i>o</i> ₆	Rainy	Cool	Normal	True	No
<i>o</i> ₇	Overcast	Cool	Normal	True	Yes
<i>o</i> ₈	Sunny	Mild	High	False	No
<i>o</i> ₉	Sunny	Cool	Normal	False	Yes
<i>o</i> ₁₀	Rainy	Mild	Normal	False	Yes
<i>o</i> ₁₁	Sunny	Mild	Normal	True	Yes
<i>o</i> ₁₂	Overcast	Mild	High	True	Yes
<i>o</i> ₁₃	Overcast	Hot	Normal	False	Yes
<i>o</i> ₁₄	Rainy	Mild	High	True	No

Table – Spécification de l'échantillon de données *Weather.nominal*.

Exemple illustratif :

Pour simplifier la démonstration :

*Un tirage aléatoire à partir de l'ensemble d'apprentissage (*Weather.nominal*).*

	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Windy</i>	<i>Play</i>
o_2	Sunny	Hot	High	True	No
o_6	Rainy	Cool	Normal	True	No
o_9	Sunny	Cool	Normal	False	Yes
o_{10}	Rainy	Mild	Normal	False	Yes
o_{13}	Overcast	Hot	Normal	False	Yes

Table – Spécification de l'échantillon de données O_{al} .

Exemple illustratif :

CNC

Algorithme d'apprentissage basé sur la *Fermeture de Galois* :

- 1 Déterminer l'attribut le plus discriminant (*Gain Informationel*) : p_* .

	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Windy</i>	<i>Play</i>
o_2	Sunny	Hot	High	True	No
o_6	Rainy	Cool	Normal	True	No
o_9	Sunny	Cool	Normal	False	Yes
o_{10}	Rainy	Mild	Normal	False	Yes
o_{13}	Overcast	Hot	Normal	False	Yes
	0.17	0.17	0.32	0.97	

Exemple illustratif :

Algorithme d'apprentissage basé sur la *Fermeture de Galois* :

- 1 Déterminer l'attribut le plus discriminant (*Gain Informationel*) : p_* .
- 2 Calculer sa fermeture : $\varphi \circ \delta(p_*)$.

	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Windy</i>	<i>Play</i>
o_2	Sunny	Hot	High	True	No
o_6	Rainy	Cool	Normal	True	No
o_9	Sunny	Cool	Normal	False	Yes
o_{10}	Rainy	Mild	Normal	False	Yes
o_{13}	Overcast	Hot	Normal	False	Yes

Exemple illustratif :

CNC

Algorithme d'apprentissage basé sur la *Fermeture de Galois* :

- ❶ Déterminer l'attribut le plus discriminant (*Gain Informationel*) : p_* .
- ❷ Calculer sa fermeture : $\varphi \circ \delta(p_*)$.

	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Windy</i>	<i>Play</i>
\circ_9	Sunny	Cool	Normal	False	Yes
\circ_{10}	Rainy	Mild	Normal	False	Yes
\circ_{13}	Overcast	Hot	Normal	False	Yes

Exemple illustratif :

CNC

Algorithme d'apprentissage basé sur la *Fermeture de Galois* :

- ❶ Déterminer l'attribut le plus discriminant (*Gain Informationel*) : p_* .
- ❷ Calculer sa fermeture : $\varphi \circ \delta(p_*)$.

	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Windy</i>	<i>Play</i>
o ₉	Sunny	Cool	Normal	False	Yes
o ₁₀	Rainy	Mild	Normal	False	Yes
o ₁₃	Overcast	Hot	Normal	False	Yes

Exemple illustratif :

Algorithme d'apprentissage basé sur la *Fermeture de Galois* :

- ❶ Déterminer l'attribut le plus discriminant (*Gain Informationel*) : p_* .
- ❷ Calculer sa fermeture : $\varphi \circ \delta(p_*)$.
- ❸ Déterminer la classe majoritaire associée à ce concept pertinent.

	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Windy</i>	<i>Play</i>
o_9	Sunny	Cool	Normal	False	Yes
o_{10}	Rainy	Mild	Normal	False	Yes
o_{13}	Overcast	Hot	Normal	False	Yes

Exemple illustratif :

Algorithme d'apprentissage basé sur la *Fermeture de Galois* :

- 1 Déterminer l'attribut le plus discriminant (*Gain Informationel*) : p_* .
- 2 Calculer sa fermeture : $\varphi \circ \delta(p_*)$.
- 3 Déterminer la classe majoritaire associée à ce concept pertinent.
- 4 Dédire la règle de classification (classifieur faible).

Classifieur faible (Weak Learner) :

SI Humidity=Normal **ET** Windy=FALSE **ALORS** Play=Yes

Une petite discussion :

L'absence de l'aspect adaptatif :

- Treillis de concepts : **Construction exhaustive.**
- Semi-treillis de concepts : **Construction non contextuelle.**

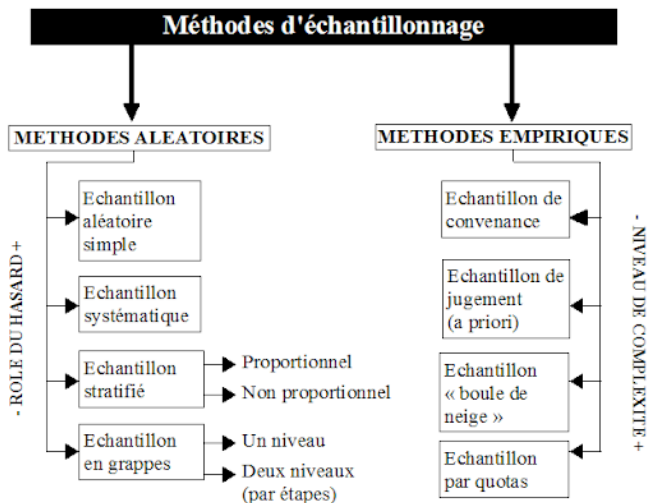
Plan :

- 1 Introduction
- 2 Analyse de Concepts Formels
 - Pourquoi l'Analyse de Concepts Formels ?
 - Apprentissage à partir d'un treillis de concepts
 - Apprentissage à partir d'un demi/semi treillis de concepts
 - CFC & CNC
- 3 Les méthodes d'ensembles
 - Apprentissage séquentiel
 - Apprentissage parallèle
- 4 Conclusion

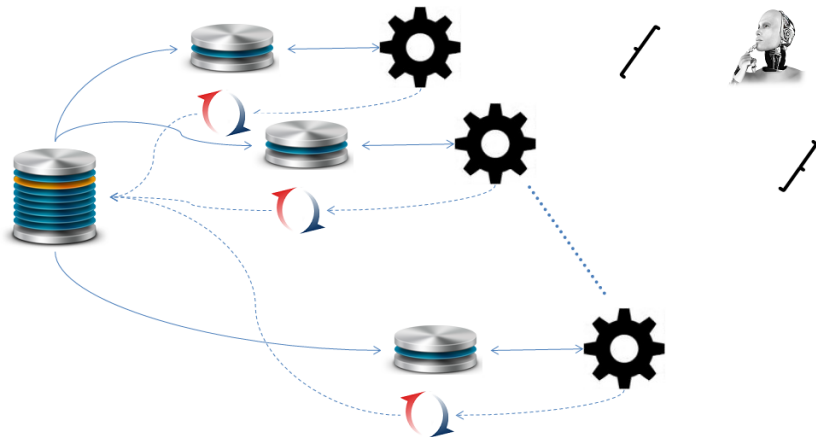
Pourquoi ?

- Combiner des classifieurs.
- Améliorer la précision aussi que l'efficacité.
- Deux stratégies :
 - Séquentiel (Boosting)
 - Parallèle (Bagging)

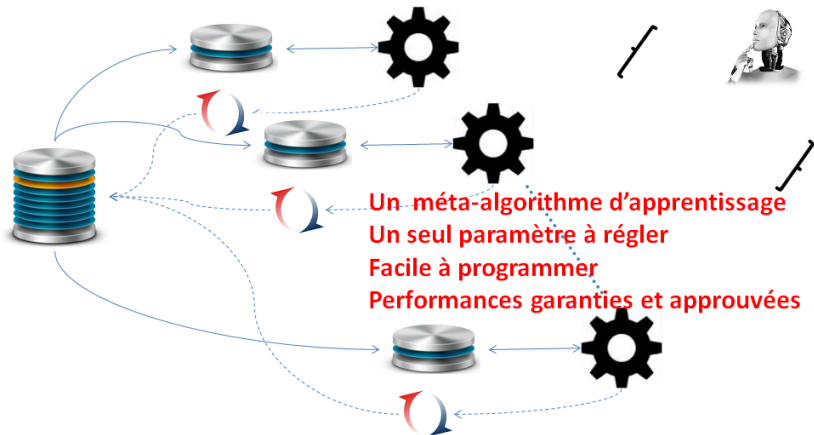
”Une décision plus fiable peut être obtenue en combinant l’avis de plusieurs experts ce qui induit une meilleur précision.”



Pourquoi Boosting ?



Pourquoi Boosting ?



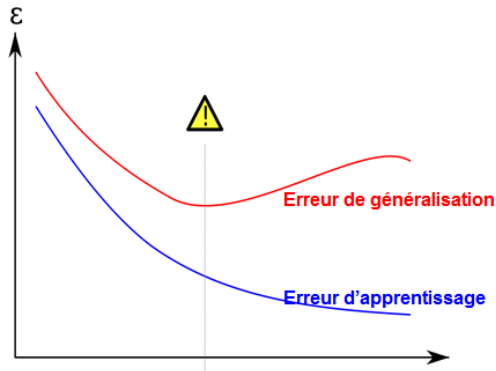
Choix de la méthode d'ensemble Boosting

Boosting de classifieurs :

- Construction séquentielle de plusieurs classifieurs.
- Adaptative.

Intérêts majeurs :

- Moins sensible au bruit.



Sur-apprentissage

L'erreur générée par un classifieur est généralement divisée en deux :

- Une err. d'apprentissage : le nombre de classifications erronées commis sur les instances d'apprentissage.
- Une err. de généralisation : l'erreur commise par le classifieur sur des instances de test.

Un bon classifieur ne doit pas seulement classer correctement les données d'apprentissage, mais il faut aussi classer avec précision les données de test.

Un bon classifieur doit avoir une faible erreur d'apprentissage ainsi qu'une faible erreur de généralisation.

Sur-Apprentissage

Un classifieur qui s'adapte bien aux données d'apprentissage, peut avoir une erreur de généralisation plus faible qu'un classifieur avec une erreur d'apprentissage plus importante.

Le *AdaBoost* (Adaptive Boosting) :

Ne traite que les instances étiquetées par des classes binaires.

Le *AdaBoost.M1* :

- Traite les instances multi-classe.
- Considérer d'avantage la valeur du taux d'erreur d'apprentissage : ϵ_t .
- Dégage peu d'instances mal classées ayant des poids élevés.
- Poids des instances.

Le *AdaBoost.M2* :

Adapter aux classifieurs faibles :

- Considérer *d'avantage* le pseudo-perte du classifieur ($\epsilon_t \geq 0.5$).
- Éviter le sur-apprentissage.
- Une bonne vitesse de convergence.
- Meilleure gestion de bruit.

AdaBoos.M1

Données :

- Soit $O = (o_1, y_1), \dots, (o_n, y_n)$ un échantillon de données étiquetées tel que $y_i \in \mathcal{Y} = \{1, \dots, Y\}$.
- Un classifieur de base h_t .
- Un entier T précisant le nombre maximale d'itérations (nombre de classifieurs).
- Soit o une instance à classer.

Résultat : h_{fin} : Classifieur final.

Début

Initialiser la distribution des poids $D_0 : D_0(i) = \frac{1}{n}$, $i = 1, \dots, n$ à partir de O .

Initialiser le vecteur des poids $w_i^1 = D_0(i)$.

Pour $t = 1$ à T Faire

Calculer la distribution des poids : $D_t = \frac{w_t^i}{\sum_{i=1}^n w_t^i}$.

Appliquer la méthode d'apprentissage sur O muni de la distribution D_t pour obtenir h_t .

Calculer le taux de la pseudo-perte : $\epsilon_t = \sum_{i=1}^n D_t(i)[h_t(o_i) \neq y_i]$.

Si $\epsilon_t \geq 0.5$ Alors

Poser $t = T + 1$

Fin

Sinon

 Calculer la perte : $\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$.

 Définir les nouvelles valeurs du vecteur des poids :

$$w_i^{t+1} = w_i^t \times \beta_t^{1 - [h_t(o_i) \neq y_i]}$$

Fin

Fin

Résultat du vote : $h_{fin}(o) = \arg \max_{y \in \mathcal{Y}} \sum_{t=1}^T (\log(\frac{1}{\beta_t})[h_t(o_i) \neq y_i])$;

AdaBoos.M2

Données :

- Soit $O = (o_1, y_1), \dots, (o_n, y_n)$ un échantillon de données étiquetées tel que $y_i \in \mathcal{Y} = \{1, \dots, Y\}$.
- Un classifieur de base h_t .
- Un entier T précisant le nombre maximale d'itérations (nombre de classifieurs).
- Soit o une instance à classer.

Résultat : h_{fin} : Classifieur final.

Début

Initialiser la distribution des poids $D_0 : D_0(i) = \frac{1}{n}$; $i = 1, \dots, n$ à partir de O .

Initialiser le vecteur des poids $w_{i,y}^1 = \frac{D_0(i)}{Y-1}$; pour $i=1, \dots, n$ et $y \in \mathcal{Y} - \{y_i\}$.

Pour $t = 1$ à T Faire

Calculer $W_i^t = \sum_{y \neq y_i} w_{i,y}^t$ et définir $q_t(i, y) = \frac{w_{i,y}^t}{W_i^t}$ pour tout $y \neq y_i$.

Calculer la distribution des poids : $D_t(i) = \frac{W_i^t}{\sum_{i=1}^n W_i^t}$.

Appliquer la méthode d'apprentissage en la traînant sur l'échantillon d'apprentissage O muni de la distribution D_t .

Retenir le classifieur $h_t : O \times \mathcal{Y} \rightarrow [0, 1]$.

Calculer son taux de la pseudo-perte : $\varepsilon_t = 0.5 \times \sum_{i=1}^n D_t(i)(1 - h_t(o_i, y_i) + \sum_{y \neq y_i} q_t(i, y)h_t(o_i, y))$.

Calculer la perte : $\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}$.

Définir les nouvelles valeurs du vecteur des poids : $w_{i,y}^{t+1} = w_{i,y}^t \times \beta_t^{0.5 \times (1 + h_t(o_i, y_i) - h_t(o_i, y))}$ pour tout $i=1, \dots, n$ et $y \in \mathcal{Y} - \{y_i\}$.

Fin

Résultat du vote : $h_{fin}(o) = \arg \max_{y \in \mathcal{Y}} \sum_{t=1}^T (\log(1/\beta_t)) h_t(o, y)$.

Fin

BFC : Boosting Formal Concept

Données :

- ① Soit $O = \{(o_1, y_1), \dots, (o_n, y_n)\}$ un échantillon de données étiquetées tel que $y_i \in Y = \{1, \dots, K\}$.
- ② T est le nombre de classifieurs à générer.
- ③ CFC_t : un classifieur faible qui induit une hypothèse sur chaque échantillon.

Résultat : CFC_{fin} : L'hypothèse qui est le résultat final du *Boosting*.

Début

Initialiser Q_0 à zéro ;

Initialiser la distribution des poids D_0 : $D_0(i) = (1/n)$; $i = 1, \dots, n$ à partir de O ;

Initialiser le vecteur des poids $w_{i,y}^1 = D_0(i)/(k-1)$; pour $i=1, \dots, n$ et $y \in Y - \{y_i\}$;

Pour t de 1 à T Faire

Calculer W_i^t et définir $q_t(i, y)$ pour tout $y \neq y_i$;

Calculer la distribution des poids : $D_t(i)$;

Appliquer CFC en la traînant sur O muni de D_t . Retenir le classifieur CFC_t ;

CFC_t attribue à chaque o_i une probabilité $p_t(o_i, y_i)$ estimée pour chaque classe y_i ;

Calculer le taux de la pseudo-perte : ε_t ;

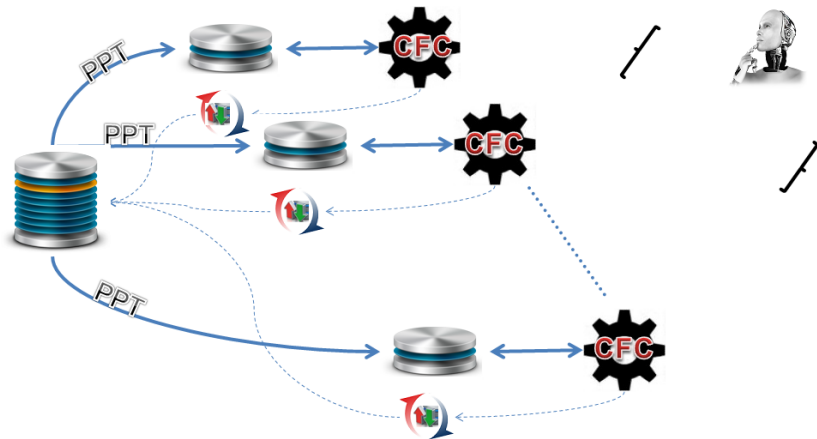
Calculer la perte : β_t ;

Définir les nouvelles valeurs du vecteur des poids : $w_{i,y}^{t+1}$ pour tout $i = 1, \dots, n$ et $y \in Y - \{y_i\}$;

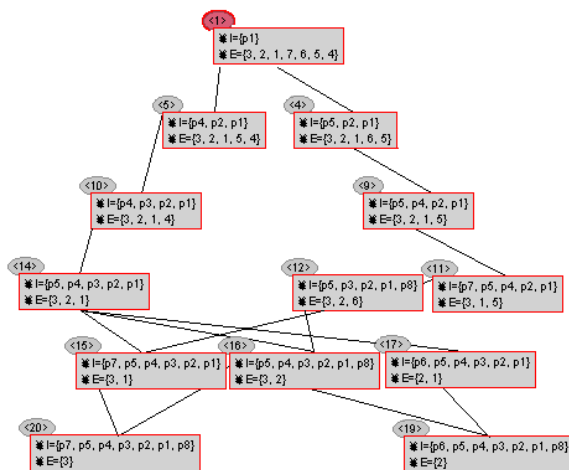
Fin

Fin

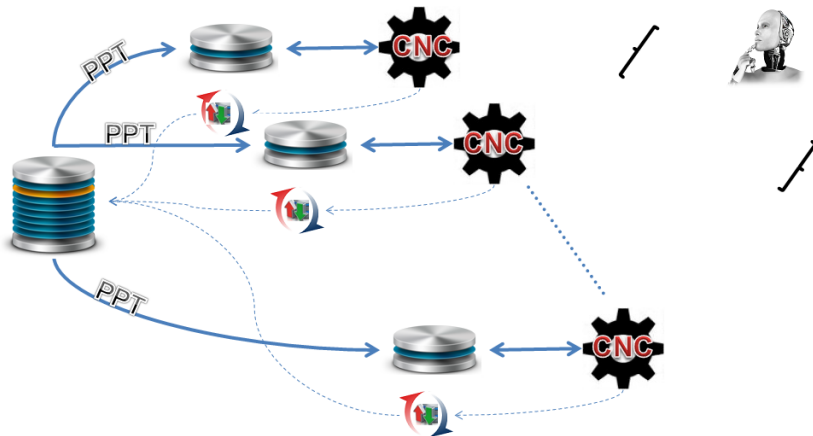
BFC : Boosting Formal Concept



BFC : Boosting Formal Concept



BNC : Boosting Nominal Concept



Diversité des classifieurs

Boosting

- Construit séquentiellement un ensemble de classifieurs de même modèle.
- Favorise la diversité des classifieurs.
- L'ajout de classifieurs pourrait avoir des effets *paralysant* à cause du *sur-apprentissage*.

Utiliser la diversité de classifieurs pour arrêter le *Boosting* au bon moment ?.

Diversité des classifieurs

- Les classifieurs devraient être différents les uns des autres.
- La diversité des *classifieurs linéaires* améliore la performance d'*AdaBoost* pour des problèmes à 2 classes.
- Plusieurs mesures existent pour calculer la diversité des sorties des classifieurs : *Coefficient de Correlation, Q Statistique, kappa..*
- Il n'y a pas une meilleure mesure de diversité bien qu'elles soient nombreuses.

Pourquoi Q Statistique ?

Q statistique a une valeur nulle pour les classifieurs indépendants.

$$Q_{j,k} = \frac{N^{11}N^{00} - N^{10}N^{01}}{N^{11}N^{00} + N^{10}N^{01}} \quad (3)$$

Avec : $N = N^{00} + N^{01} + N^{10} + N^{11}$

	h_k correct(1)	h_k incorrect(0)
h_j correct(1)	N^{11}	N^{10}
h_j incorrect(0)	N^{01}	N^{00}

Table – Le nombre d'accords et de désaccords entre deux classifieurs.

Arrêter le *sur-apprentissage* dans le *Boosting* en utilisant *Q Statistique*

La variation de cette mesure devient stationnaire lors du sur-apprentissage.



AdaBoost.HD : AdaBoost Higher Divers

Données :

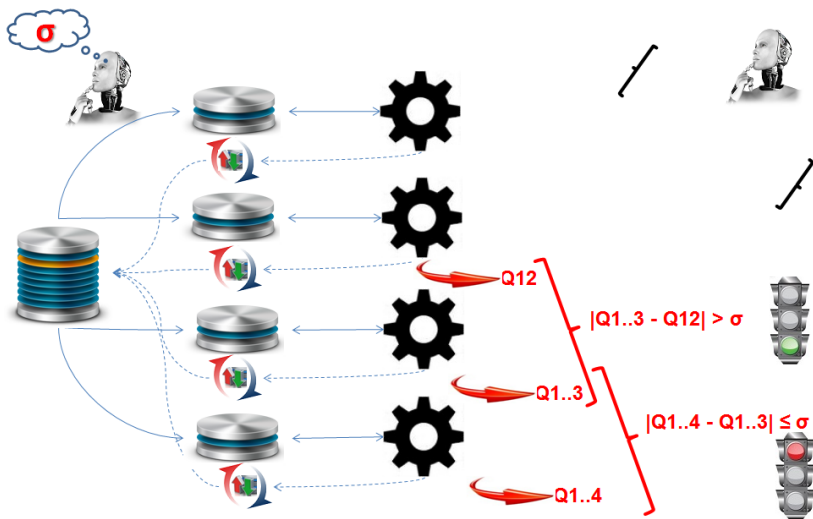
- 1 Soit $O = \{(o_1, y_1), \dots, (o_n, y_n)\}$ un échantillon de données étiquetées tel que $y_i \in Y = \{1, \dots, K\}$.
- 2 T est le nombre maximum de classifieurs à générer.
- 3 σ est un seuil d'accroissement de la mesure Q .
- 4 h_t : un classifieur faible qui induit une hypothèse sur chaque échantillon.

Résultat : h_{fin} : L'hypothèse qui est le résultat final du *Boosting*.

Début

```

... ;
Pour t de 1 à T Faire
    Calculer  $W_t^t$  et définir  $q_t(i, y)$  pour tout  $y \neq y_i$ ;
    Calculer la distribution des poids :  $D_t(i)$  ;
     $h_t$  attribuée à chaque  $o_i$  une probabilité  $p_t(o_i, y_i)$  estimée pour chaque classe  $y_i$  ;
    Mesurer la diversité  $Q_{1..t}$  de l'ensemble des classifieurs en cours ;
    Si  $|Q_{1..t} - Q_{1..(t-1)}| < \sigma$  Alors
        Arrêter le Boosting ;
    Fin
    Sinon
        /*La génération des classifieurs est poursuivie */ ;
        Calculer le taux de la pseudo-perte :  $\epsilon_t$  ;
        ... ;
    Fin
Fin
Fin
```



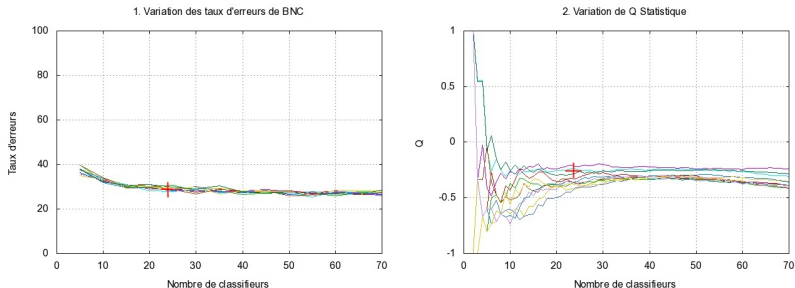


Figure – Diversité et taux d'erreurs du BNC sur la base German Credit

AdaBoost.HD construit un ensemble de 24 classifieurs différents et dépendants négativement ($Q = -0.23$) et un TE = 31.8%

. Seuil très faible, σ vaut **0.001**

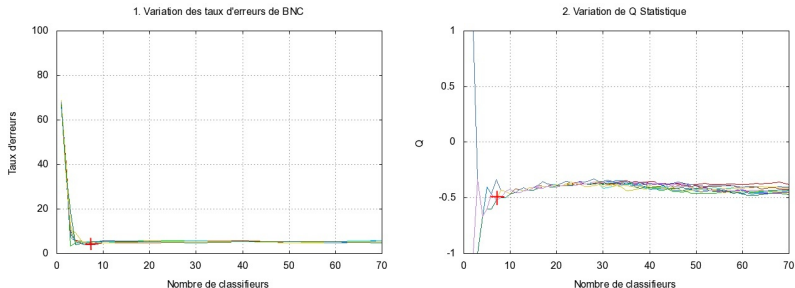


Figure – Diversité et taux d'erreurs du BNC sur la base *Iris*

Pour des données très corrélées, *AdaBoost.HD* est arrêté après la génération de 7 classifieurs très dépendants ($Q = -0.57$) et avec un TE = 4.66%.

. Seuil très faible, σ vaut **0.001**

Échantillonnage de données pour l'apprentissage

Rappel

Échantillonnage

Opération de recueil de données pour un échantillon d'individus d'une population.

Base de sondage

Une liste des individus à partir de laquelle on prélève un échantillon. Cette liste détermine la population observée.

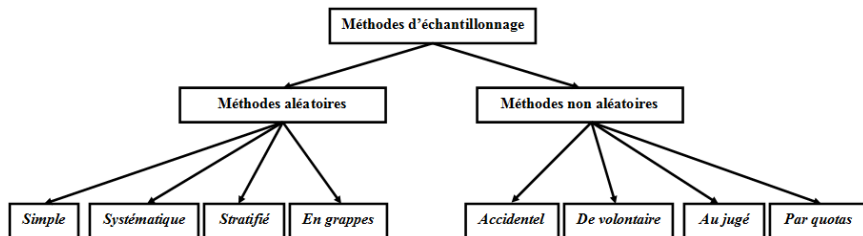
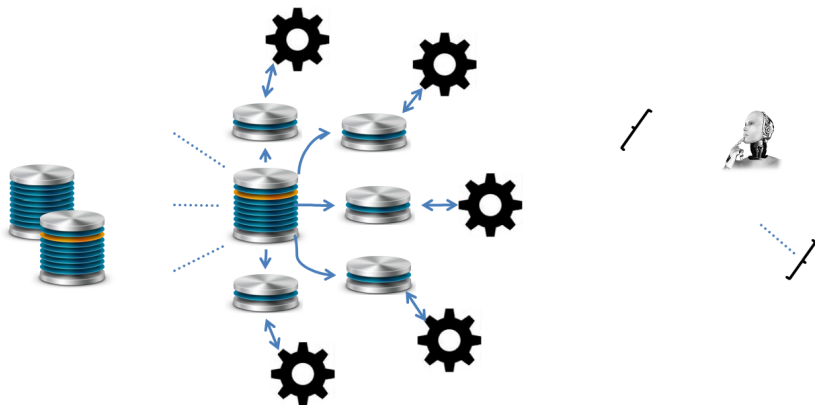
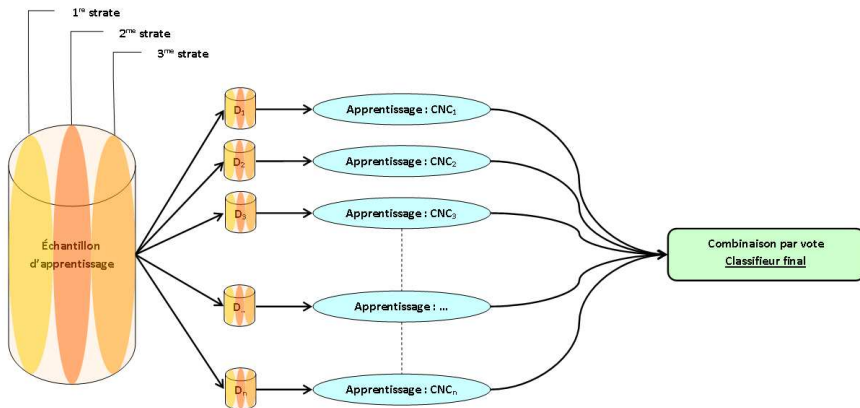


Figure – Taxonomie des méthodes d'échantillonnage

Bagging : Bootstrap aggregating



Dagging : Disjoint samples aggregating



DNC : Dagging Nominals Concepts

Données :

- 1 T est le nombre de classifieurs à générer.
- 2 Les données d'apprentissage O de taille n avec $O = \{(o_1, y_1), \dots, (o_n, y_n)\}$ étiquetées $y_i \in \{1, \dots, Y\}$.

Résultat :

- Un classifieur h_{vote} .

Début

Diviser la population en T strates;

Établir la liste la plus complete des instances de chaque strate;

Pour t de 1 à T Faire

Calculer la pourcentage P_t des instances de t ieme strate similaire à O ;

Choisir de façon aléatoire simple des instances de t ieme strate pour constituer O^{Θ_t} par respect à P_t ;

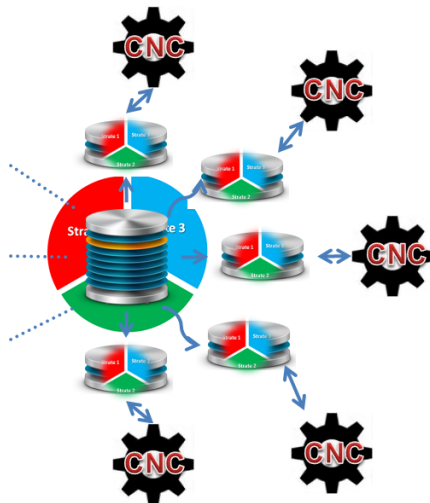
Apprendre le classifieur CNC sur O^{Θ_t} afin de générer h_t ;

Fin

Appliquer la règle du vote : $h_{vote} = \arg \max_{y \in Y} \sum_{t=1}^T h_t(o, y)$;

Fin

DNC : Dagging Nominals Concepts



Plan :

- 1 Introduction
- 2 Analyse de Concepts Formels
 - Pourquoi l'Analyse de Concepts Formels ?
 - Apprentissage à partir d'un treillis de concepts
 - Apprentissage à partir d'un demi/semi treillis de concepts
 - CFC & CNC
- 3 Les méthodes d'ensembles
 - Apprentissage séquentiel
 - Apprentissage parallèle
- 4 Conclusion

Mes contributions :

- **BFC** : **B**oosting **F**ormal **C**oncepts.
- **BNC** : **B**oosting **N**ominal **C**oncepts.
- **AdaBoost.HD** : **A**daptative **B**oosting of the **H**igher **D**ivers classifiers.
- **DNC** : **D**agging **N**ominal **C**oncepts.

Problématique Résolue ?

- Nombre élevé de connaissances extraites.
- Pertinence de connaissances extraites.
⇒ Classifier Formal Concept.
- Perte d'information.
⇒ Classifier Nominal Concept.
- Absence de l'aspect adaptatif.
⇒ Apprentissage séquentiel de classifieurs basés sur ACF.
- Sur-apprentissage.
⇒ Un critère d'arrêt d'apprentissage adaptatif.
- Performances.
⇒ Apprentissage Parallèle de classifieurs basés sur ACF.

Merci pour votre attention . . . Questions ?
Remarques ? Suggestions ? ... ?