

# Treillis de Galois et Extraction de Connaissances

---

---

Engelbert MEPHU NGUIFO

CRIL - IUT de Lens

[mephu@cril.univ-artois.fr](mailto:mephu@cril.univ-artois.fr)

<http://www.cril.univ-artois.fr/~mephu>

Tutoriel - Conférence E.G.C.'2002

Montpellier, 21 Janvier 2002

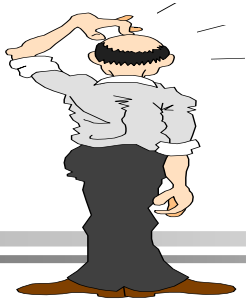
# Motivations

---

- Montrer l'intérêt d'un sujet 'nouveau'
- Faire partager un savoir
- Acquérir d'autres connaissances

Quelle démarche adoptée pour 3h ?

- Articles de recherche ?
- Généralités sur le travail de recherche ?



# SOMMAIRE

---

- Introduction - ECBD
- Treillis de Galois
- Prétraitement de données
- Règles d'association
- Classification supervisée
- Conclusion



# Introduction : Contexte

---

- Extraction de connaissances dans les bases de données (ECBD)

Processus interactif et itératif d'analyse d'un **grand ensemble de données brutes** afin d'en extraire des connaissances exploitables par l'utilisateur-analyste qui y joue un rôle central

[Kodratoff, Napoli, Zighed, dans Bulletin AFIA 2001 sur ECBD]

# Introduction : ECBD

---

- Extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) information (knowledge) or patterns from data in large databases or other information repositories

[Fayyad et al., 1996]

'Knowledge Discovery in Databases' (KDD) or 'Data Mining' (DM)

- Processus interactif et itératif d'analyse d'un **grand ensemble de données brutes** afin d'en extraire des connaissances exploitables par l'utilisateur-analyste qui y joue un rôle central



2002

[Kodratoff, Napoli, Zighed, dans Bulletin AFIA 2001 sur ECBD]  
**ECBD** ou encore 'Fouille de données'

# Introduction : ECBD

---

- Plusieurs découvertes scientifiques concerne l'ECBD
  - » Loi de Kepler, Lois de Newton, Table périodique des éléments chimiques, ...
- Statistique, Apprentissage automatique:
  - » disciplines dédiées à l'analyse de données
- Pourquoi l'ECBD? Quelles sont les différences?
  - » **Données de taille volumineuse** - du giga au tera octets
  - » Ordinateur rapide - réponse instantanée, analyse interactive
  - » Analyse multidimensionnelle, puissante et approfondie
  - » Langage de haut niveau, "déclaratif" – Facilité d'usage et Contrôlable
  - » Automatisée or semi-automatisée —fonctions de fouille de données cachées ou intégrées dans plusieurs systèmes

# Introduction : ECBD

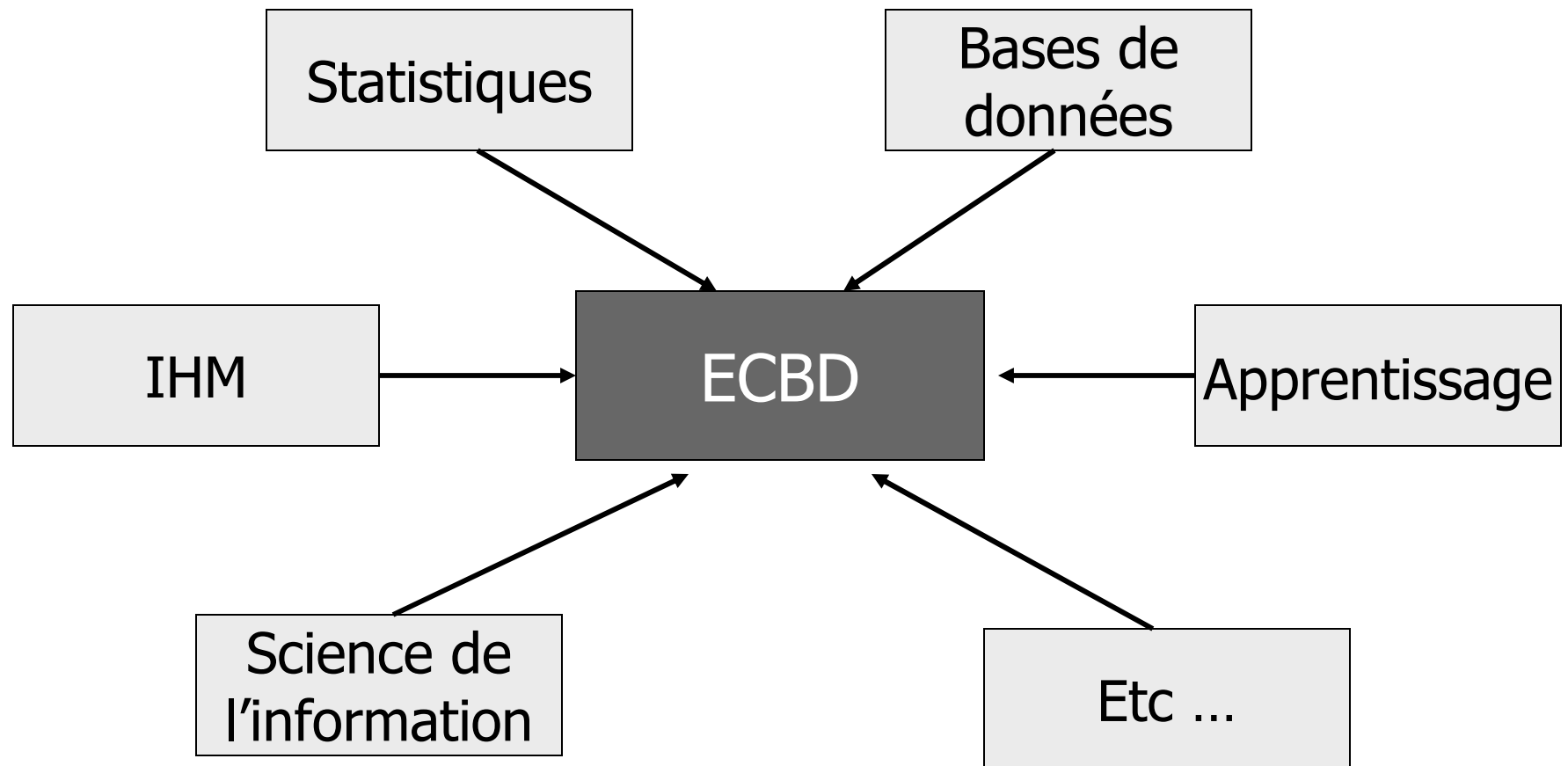
---

- **Applications diverses et variées**  
Médecine, Biologie moléculaire, Finance, Distribution, Télécommunication, ...
- **Domaines de recherche**  
Bases de données, Statistiques, Intelligence Artificielle, Interface Homme-Machine, Reconnaissance des Formes, Réseaux de Neurones, Science de l'information, ...

# Introduction : ECBD

---

---





# Introduction : ECBD

---

- 1989 IJCAI Workshop on Knowledge Discovery in Databases
  - » Knowledge Discovery in Databases (G. Piatetsky-Shapiro and W. Frawley, 1991)
- 1991-1994 Workshops on Knowledge Discovery in Databases
  - » Advances in Knowledge Discovery and Data Mining (U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 1996)
- 1995-1998 International Conferences on Knowledge Discovery in Databases and Data Mining (KDD'95-98)
  - » Journal of Data Mining and Knowledge Discovery (1997)
- 1998 ACM SIGKDD, SIGKDD'1999-2001 conferences, and SIGKDD Explorations
- More conferences on data mining
  - » PAKDD, PKDD, SIAM-Data Mining, (IEEE) ICDM, DaWaK, SPIE-DM, etc.
- **En France** : EGC janvier 2001 (Nantes), janvier 2002 (Montpellier)

# Introduction : ECBD

---



- Livres :
  - » *Data Mining*,
    - Han & Kamber, chez Morgan Kaufmann Pubs., 2001
  - » *Mastering Data Mining*,
    - Berry & Linoff, chez Wiley Computer Publishing, 2000
  - » ...
- Sites intéressants :
  - » <http://www.kddnuggets.com> : à consulter
  - » <http://www.crisp-dm.org> : CRoss-Industry Standard Process for Data Mining - effort de standardization
  - » ...

# Introduction : ECBD

---



- Outils commercialisés :
  - » **Intelligent Miner** (<http://www.ibm.com>),
  - » **Entreprise Miner** (SAS Institute),
  - » **MineSet** (Silicon Graphics Inc.),
  - » **Clementine** (Integral Solutions Ltd, racheté par SPSS),
  - » **DBMiner** (<http://www.dbminer.com> ou <http://db.cs.sfu.ca>, version libre 90 jours),
  - » ....

# Introduction : ECBD

---



- **Processus Itératif - 4 étapes :**
  - » **Nettoyage et Intégration de bases de données**  
Suppression données inconsistantes ou combinaison de données de différentes sources pour constituer un entrepôt
  - » **Prétraitement de données**  
Sélection ou transformation de données de l'entrepôt pour les rendre exploitables
  - » **Fouille de données**  
Utilisation de méthodes intelligentes pour extraire des motifs.  
Tâches: caractérisation, discrimination, **association**, **classification**, prédiction, analyse de données évolutives
  - » **Evaluation et Présentation**  
Identifier les motifs intéressants, les visualiser, et interagir

# Introduction : ECBD-TG

---

- Prétraitement, Fouille de données
  - » Treillis de Galois
    - structure mathématique,

Est-ce un cadre pertinent pour :

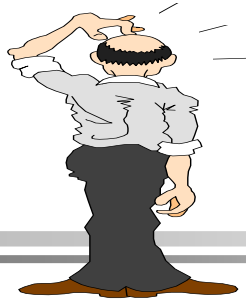
- Prétraiter les données ?
- Rechercher les règles d'association ?
- Effectuer de la classification ?

# Introduction : ECBD-TG

---



- 2001 ICCS workshop on Concept Lattices for KDD
  - » Concept Lattices-based Theory, Methods and Tools for Knowledge Discovery in Databases, Stanford (CA), July 30, 2001. <http://CEUR-WS.org/Vol-42> (E. Mephu Nguifo, V. Duquenne and M. Liquière)
  - » Special issue of JETAI - Journal of Experimental and Theoretical Artificial Intelligence – to appear Winter 2002 (E. Mephu Nguifo, V. Duquenne and M. Liquière)
- 2002 ECAI workshop on Formal Concept Analysis for KDD
  - » Advances in Formal Concept Analysis for Knowledge Discovery in Databases, Lyon (France) July 22-23, 2002 (M. Liquière, B. Ganter, V. Duquenne, E. Mephu Nguifo, and G. Stumme)



# SOMMAIRE

---

- ✓ Introduction - ECBD
- **Treillis de Galois**
  - Prétraitement de données
  - Règles d'association
  - Classification supervisée
  - Conclusion

# Treillis de Galois - Préliminaires

---

- ou Treillis de Concepts
- En anglais : Concept or Galois Lattices
- Travaux :
  - » Birkhoff's Lattice Theory : 1940, 1973
  - » Barbut & Monjardet : 1970
  - » Wille : 1982
  - » Chein, Norris, Ganter, Bordat, ...
  - » Diday, Duquenne, ...
- Concepts de base
  - » Contexte, Correspondance de Galois, Concept, Ordre

# Treillis de Galois - Définition

---

- Contexte = triplet  $(O, A, I)$  tel que:
  - »  $O$  : ensemble fini d'exemples
  - »  $A$  : ensemble fini d'attributs
  - »  $I$  : relation binaire entre  $O$  et  $A$ ,  $(I \subseteq O \times A)$

- 2 exemples :

$O \setminus A$	a	b	c
1	1	1	1
2	1	1	
3		1	1

	a	b	c	d	e	f	g	h
1	1	1	1	1	1	1	1	
2	1	1	1	1	1	1		1
3	1	1	1	1	1		1	1
4	1	1	1	1		1		
5	1	1		1	1		1	
6	1	1	1		1			
7	1		1			1		

# Treillis de Galois - Définition

---

- Correspondance de Galois:
  - » Soient  $O_i \subseteq O$  et  $A_i \subseteq A$ , on définit  $f$  et  $g$  comme suit:
  - »  $f : \mathfrak{P}(O) \rightarrow \mathfrak{P}(A)$   $f(O_i) = \{ a \in A / (o,a) \in I, \forall o \in O_i \}$  **intension**
  - »  $g : \mathfrak{P}(A) \rightarrow \mathfrak{P}(O)$   $g(A_i) = \{ o \in O / (o,a) \in I, \forall a \in A_i \}$  **extension**
  - »  $f$  et  $g$  sont 2 applications monotones décroissantes
  
  - » Soient  $h = g \circ f$  et  $h' = f \circ g$ , elles sont:
    - isotones (monotones croissantes):  $O_1 \subseteq O_2 \Rightarrow h(O_1) \subseteq h(O_2)$
    - extensives  $O_1 \subseteq h(O_1)$
    - idempotentes  $h(O_1) = h \circ h(O_1)$
  - »  $h$  (resp.  $h'$ ) est une fermeture dans  $\mathfrak{P}(O)$  ( resp.  $\mathfrak{P}(A)$  )
- $(f,g)$  = correspondance de Galois entre  $\mathfrak{P}(O)$  et  $\mathfrak{P}(A)$ .

# Treillis de Galois - Définition

- Correspondance de Galois: Exemple

- »  $O_1 = \{6, 7\} \Rightarrow f(O_1) = \{a, c\}$

- »  $A_1 = \{a, c\} \Rightarrow g(A_1) = \{1, 2, 3, 4, 6, 7\}$

	a	b	c	d	e	f	g	h
1	1	1	1	1	1	1	1	
2	1	1	1	1	1	1		1
3	1	1	1	1	1		1	1
4	1	1	1	1		1		
5	1	1		1	1		1	
6	1	1	1		1			
7	1		1			1		

*Remarque:*  $h(O_1) = g \cdot f(O_1) = g(A_1) \neq O_1$

# Treillis de Galois - Définition

---

- Concept (fermé, rectangle):
  - » Soient  $O_i \subseteq O$  et  $A_i \subseteq A$ ,
  - »  $(O_i, A_i)$  est un **concept** *si et seulement si*  $O_i$  est l'extension de  $A_i$  et  $A_i$  est l'intension  $O_i$ 
    - c-à-d:  $O_i = g(A_i)$  et  $A_i = f(O_i)$
  - » Soit  $\mathbf{L} = \{ (O_i, A_i) \in \mathfrak{P}(O) \times \mathfrak{P}(A) / O_i = g(A_i) \text{ et } A_i = f(O_i) \}$  l'ensemble des concepts
- Relation d'ordre ( $\leq$ ) sur  $\mathbf{L}$ :
  - » Sous-concept / Sur-concept (spécialisation / généralisation)
  - »  $(O_1, A_1) \leq (O_2, A_2)$  si et seulement si  $O_1 \subseteq O_2$  (ou  $A_1 \supseteq A_2$ )
- Treillis de Galois
  - »  $\mathbf{T} = (\mathbf{L}, \leq)$ , ens. des concepts muni de la relation d'ordre

# Treillis de Galois - Définition

- Concept: Exemple

»  $O_1 = \{6, 7\} \Rightarrow f(O_1) = \{a, c\}$

»  $A_1 = \{a, c\} \Rightarrow g(A_1) = \{1, 2, 3, 4, 6, 7\}$

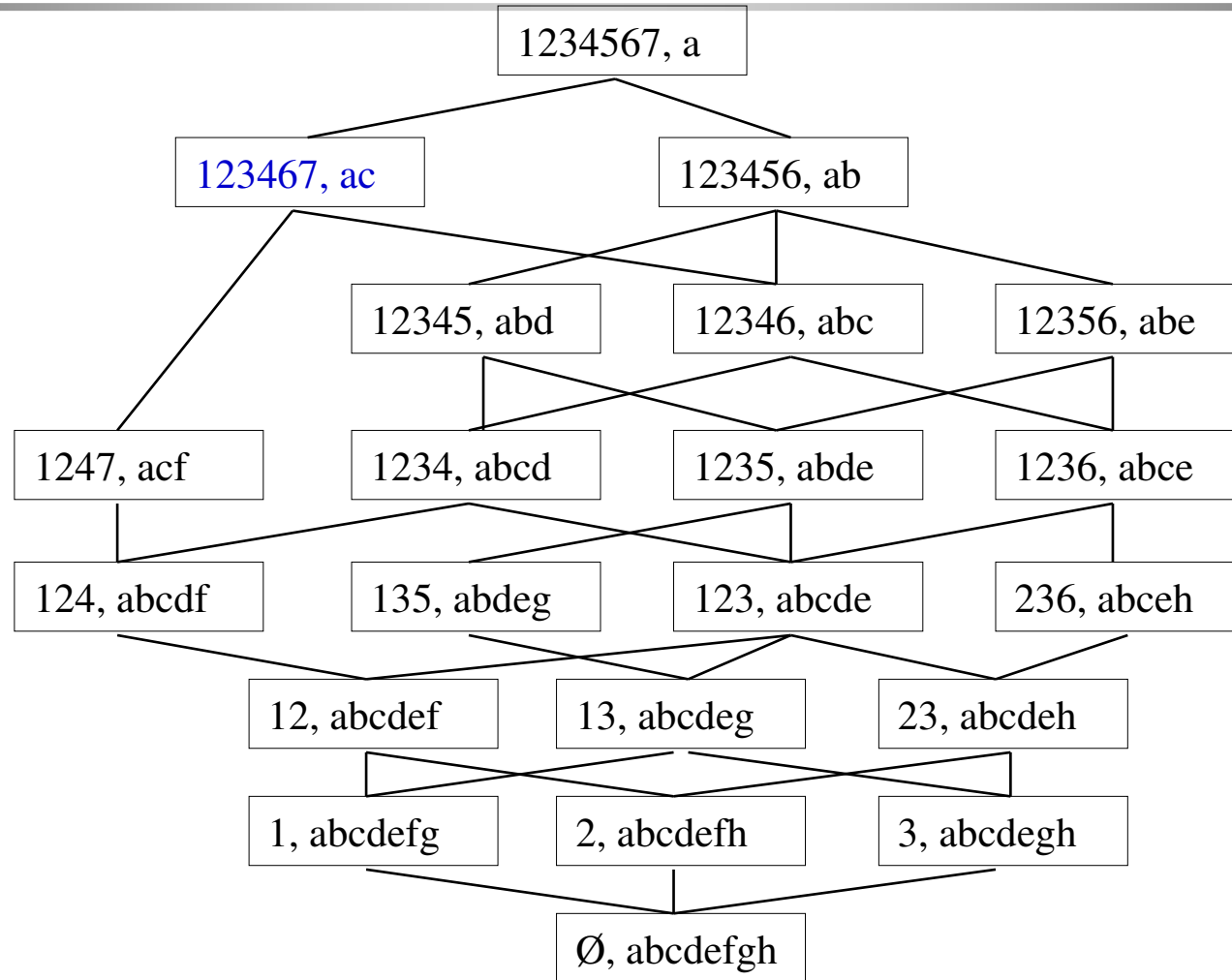
» *Remarque:*  $h(O_1) = g \cdot f(O_1) = g(A_1) \neq O_1$

»  $(\{6, 7\}, \{a, c\}) \notin L$

»  $(\{1, 2, 3, 4, 6, 7\}, \{a, c\}) \in L$

	a	b	c	d	e	f	g	h
1	1	1	1	1	1	1	1	
2	1	1	1	1	1	1		1
3	1	1	1	1	1		1	1
4	1	1	1	1		1		
5	1	1		1	1		1	
6	1	1	1		1			
7	1		1			1		

# Treillis de Galois - Définition



# Treillis de Galois - Algorithmes

---

- Non Incrémental
  - » Chein, 1969 ; Ganter, 1984 ;
  - » Bordat, 1986 ; construit le graphe de hasse
  - » Nourine et Raynaud, 1999
  - » ...
- Incrémental
  - » Norris, 1978 ;
  - » Godin et al., 1991 ; Oosthuisen, 1991 ;
  - » Carpineto et Romano, 1996
  - » ...
- Etudes comparatives d'algorithmes
  - » Guénoche, 1990, dans revue *Math. Info. Sci. Hum.*
  - » Godin et al., 1995, dans *Computation Intelligence*
  - » Kuznetsov & Obiedkov, 2001, *CLKDD proceedings*

# Treillis de Galois - Algorithmes

---

- Complexité théorique exponentielle
  - » **meilleure** : Nourine & Raynaud, [IPL 1999]
  - » **choix** : Bordat, [Math. Sci. Hum., 1986]
- Complexité est fonction du Contexte
  - ⇒ complexité sur des cas pratiques ?
  - » [Godin, 89], [Kuznetsov & Obiedkov, CLKDD'01]
- Algorithme de Bordat
  - » Approche de génération structurée (descendante, par niveau)
  - » Approche par spécialisation/généralisation
  - » Principe s'appuyant sur la relation de couverture de la rel. d'ordre  $\leq$
  - » Couverture d'un concept  $(O_i, A_i)$ , notée  $\underline{(O_i, A_i)}$ 
    - Ens des  $(O_j, A_j)$  tel que :
      - $(O_j, A_j) \leq (O_i, A_i)$  et
      - $\nexists (O_k, A_k)$  tq  $(O_j, A_j) \leq (O_k, A_k) \leq (O_i, A_i)$

# Treillis de Galois - Algorithmes

---

- Algorithme de Bordat :

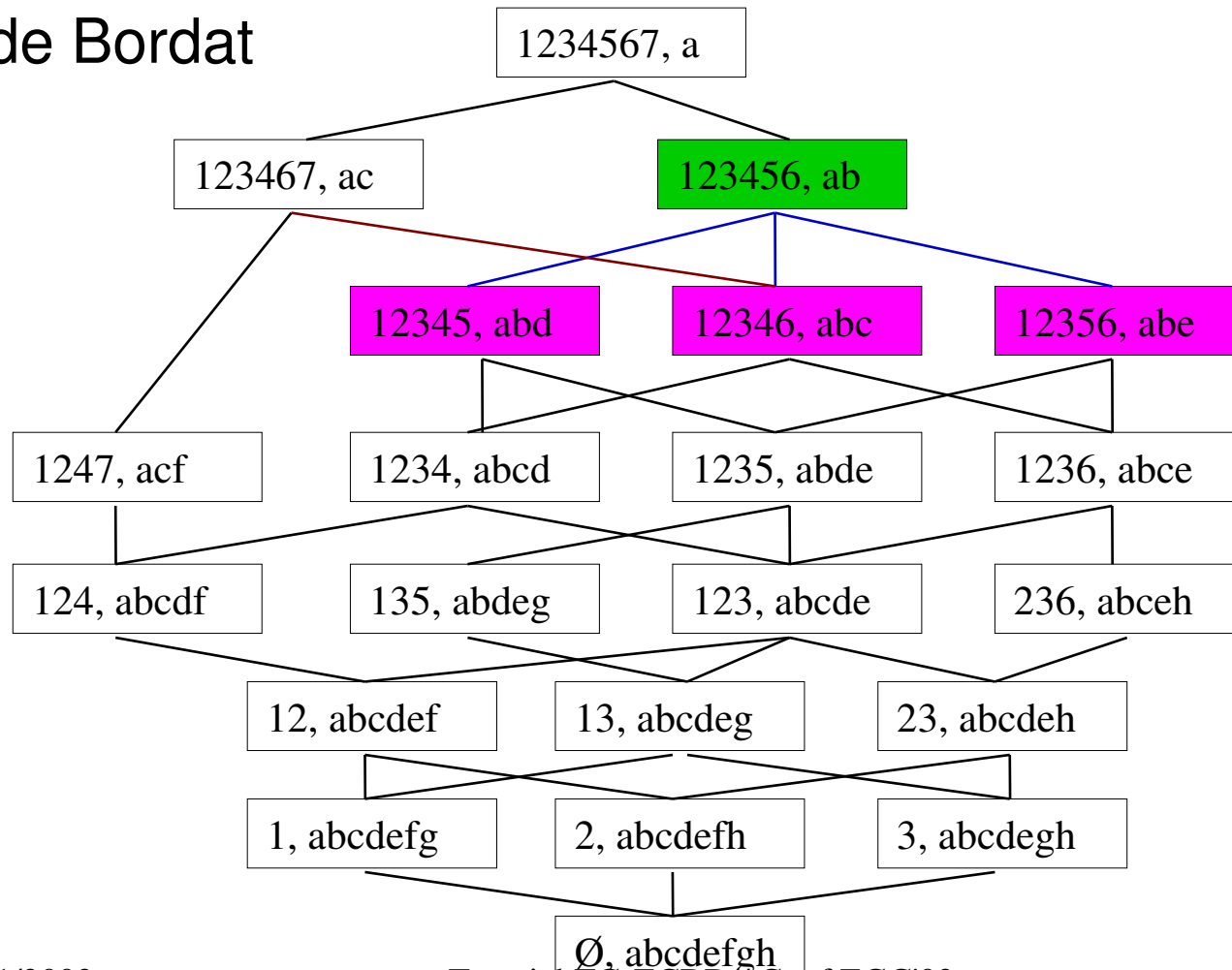
```
L = (O, f(O))
Pour chaque concept (Oi, Ai) de L
  Rechercher couverture C = (Oi, Ai)
  Pour chaque (Oj, Aj) ∈ C
    Si (Oj, Aj) ∉ L alors ajouter (Oj, Aj) à L
    Sinon rajouter un arc seulement
  Fin Pour
Fin Pour
```

**Inconvénient:** Concept engendré autant de fois qu'il a de sur-concepts

**Avantage:** Enumération des arêtes du graphe de Hasse du treillis

# Treillis de Galois - Algorithmes

Algo. de Bordat



# Treillis de Galois - Outils

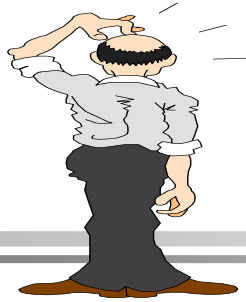
---

- Logiciels

- » GLAD (Duquenne, *ORDAL'96*)
- » TOSCANA et ANACONDA (Wille et al., 1995 >)
- » CERNATO (Sté Navicon GmbH)
- » TkConcept (Lindig, 1996)
- » SODAS (Diday & al., 2000)
- » ...

- Sites

- » <http://php.indiana.edu/~upriss/fca/fca.html>
- » <http://www.lattices.org> en cours de construction



# SOMMAIRE

---

- ✓ Introduction - ECBD
- ✓ Treillis de Galois
- **Prétraitement de données**
  - Règles d'association
  - Classification supervisée
  - Conclusion

# Prétraitement de données

---

- Objectifs
  - » Améliorer la qualité des données pour en tirer de meilleurs résultats
  - » Plusieurs techniques : Réduction [ou Sélection] ou **transformation** [ou Construction] de données (exemples ou **attributs**)
- Références : Livres (collection d'articles)
  - » Liu & Motoda, 1998, sur les attributs Kluwer Acad. Pub
  - » Liu & Motoda, 2001, sur les exemples idem
- Redescription de données
  - » Mephu Nguifo & Njiwoua, [ECML'98] et [Liu & Motoda 98]

# Prétraitement de données - Redescription

---

- Problématique :

Que faire en présence d'attributs symboliques et numériques?

1. Tout Symbolique: discrétisation des attributs numériques
2. Traitement séparée d'attributs symboliques et numériques
3. Notre proposition: **Tout Numérique**
  - Transformer les attributs symboliques en attributs numériques en s'appuyant sur le contexte de description des données

- Etat de l'art

- » Méthode Disqual : Combinaison analyse de correspondances multiples et analyse factorielle discriminante
- » **Notre approche** : utiliser treillis de Galois avec filtre sur concepts

# Prétraitement de données - Redescription

---

- Principe :

1. Générer les concepts “**pertinents**” du treillis
2. Associer à chaque attribut présent, un nouvel attribut numérique (appelé descripteur)
3. Redécrire chaque exemple avec ces descripteurs
  - Dénombrer le nombre de fois que l'exemple et l'attribut apparaissent simultanément dans un concept
4. Appliquer une technique de traitement de données numériques

- Résultat :

- » Contexte à valeurs numériques discrètes, bornées par le nombre de concepts «pertinents»
- » Construction de nouveaux attributs,  $A^*$

# Prétraitement de données - Redescription

---

- Génération de concepts « pertinents » :
  1. Utilisation de fonctions de sélection
    - Vote majoritaire (ou support)
    - Entropie
    - Loi de succession de Laplace
    - Etc ...
  2. Utilisation d'un seuil pour la sélection
- Résultat :
  - »  $L = \{ (O_i, A_i), \text{concepts «pertinents»} \}$
  - »  $P = \{ A_i, \text{hypothèses «pertinentes»} \}$
  - » Hypothèse = intension du concept, exprimée sous forme de conjonction d'attributs

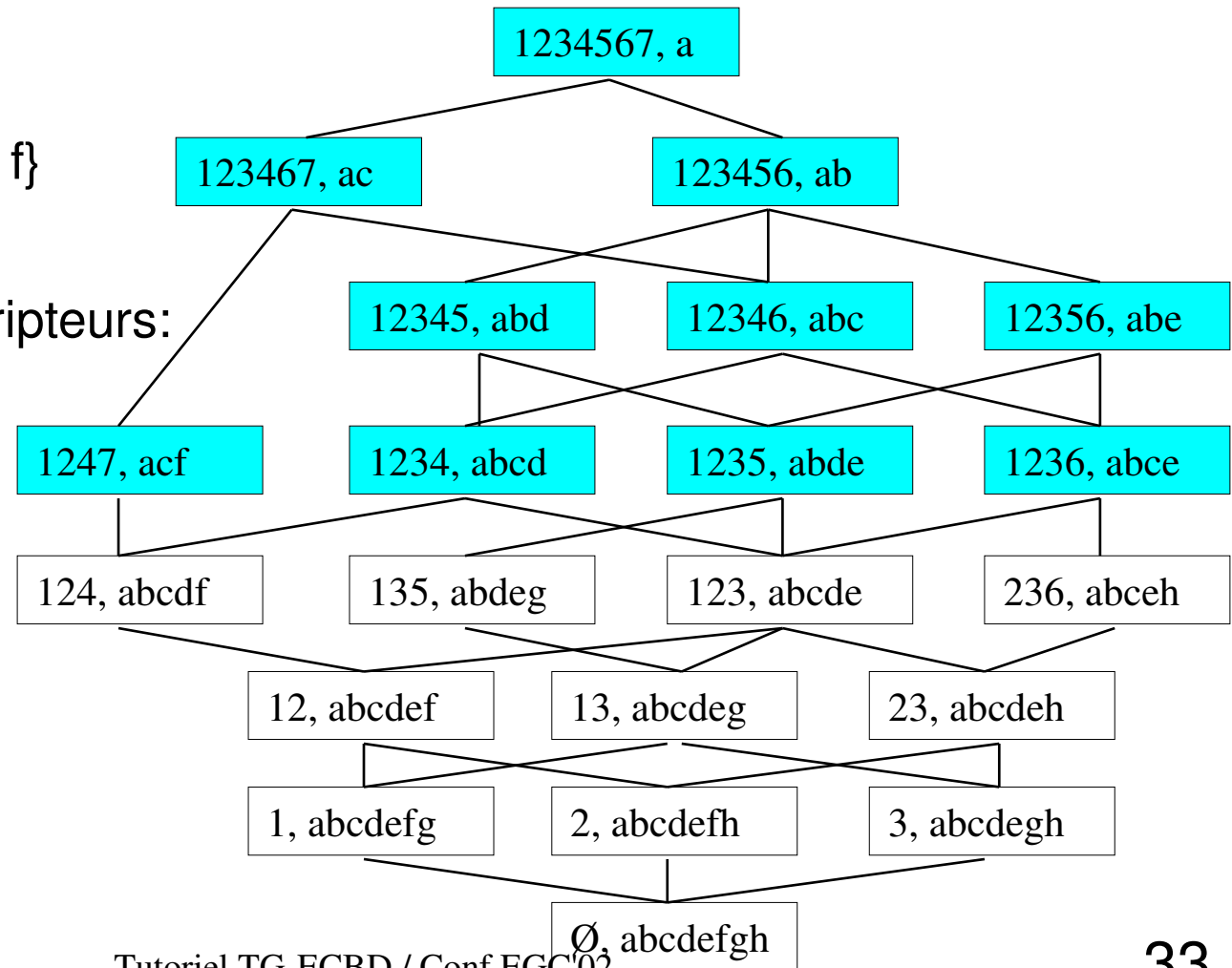
# Prétraitement de données : Redescription

- Exemple:

$A^* = \{a, b, c, d, e, f\}$

6 nouveaux descripteurs:

- $d1 \approx a$
- $d2 \approx b$
- $d3 \approx c$
- $d4 \approx d$
- $d5 \approx e$
- $d6 \approx f$



# Prétraitement de données - Redescription

- Algorithme :

```
Redescription (O, D, P)
- Renvoie O x D, la matrice redécrite
Début
  Pour tout  $o_i \in O$  Faire
    Pour tout  $d_k \in D$  Faire
       $d_{ik} \leftarrow 0$ 
    Fin Pour
  Fin Pour
  Pour chaque exemple  $o_i \in O$ 
     $P_i \leftarrow \{ r \in P / o_i \text{ vérifie } r \}$ 
    Pour chaque hypothèse  $r \in P_i$ 
      Pour chaque attribut  $a_j$  de l'hypothèse  $r$ 
        Rechercher le descripteur  $d_{ik}$  associé à  $a_j$ 
         $d_{ik} \leftarrow d_{ik} + 1$ 
      Fin Pour
    Fin Pour
  Fin Pour
Fin
```

# Prétraitement de données :

## Redescription

- Exemple:

$A^* = \{a, b, c, d, e, f\}$

$D = \{d1, d2, d3, d4, d5, d6\}$

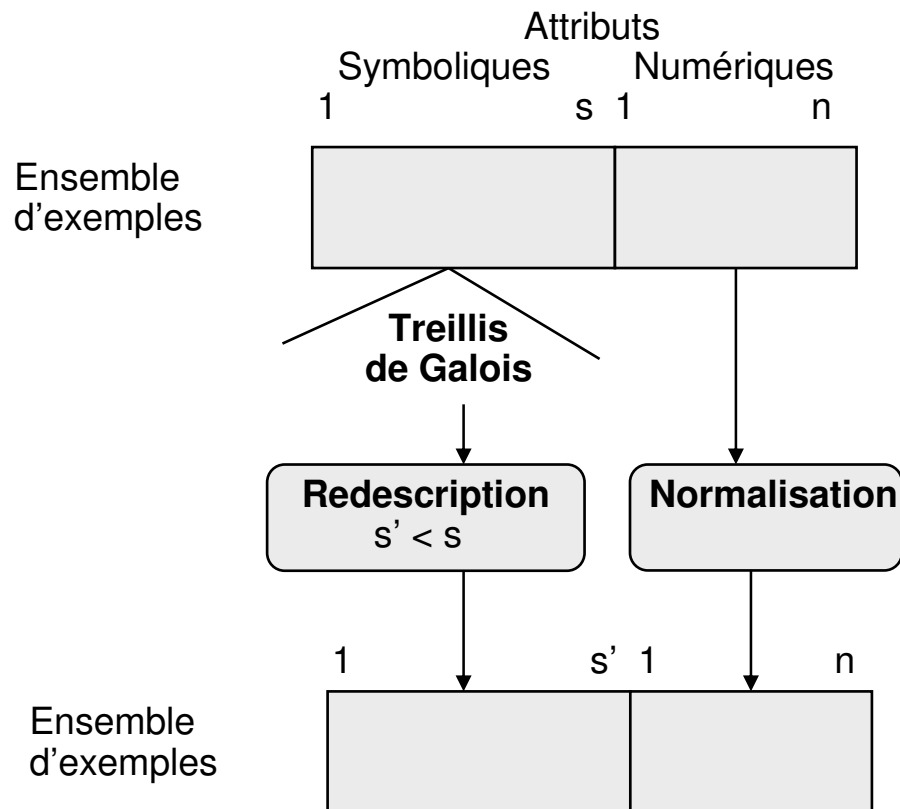
	a	b	c	d	e	f	g	h
1	1	1	1	1	1	1	1	
2	1	1	1	1	1	1		1
3	1	1	1	1	1		1	1
4	1	1	1	1		1		
5	1	1		1	1		1	
6	1	1	1		1			
7	1		1			1		



	d1	d2	d3	d4	d5	d6		
1	10	7	5	3	3	1		
2	10	7	5	3	3	1		
3	9	7	4	3	3			
4	7	4	4	2		1		
5	5	4		2	2			
6	6	4	2		2			
7	3		2			1		

# Prétraitement de données - Redescription

- Vue d'ensemble :



# Prétraitement de données - Redescription

---

- Que faire ensuite ?
  1. Appliquer toute méthode d'ECBD reposant sur des données numériques
  2. Techniques PPV en classification, ...  
Distances euclidienne, manhattan, mahalanobis, ...
  3. **Concevoir une nouvelle méthode de classification**  
IGLUE, CIBLe
- Expérimentations :

Jeu de données de test (Monks 1-2-3, Small soybean, Votes, Breast cancer) du "UCI Repository of ML DB"  
[Mephu Nguifo & Njiwoua, 1998], ECML et Livre Liu & Motoda  
[Njiwoua, 2000], Thèse de doctorat

# Prétraitement de données - Redescription

---

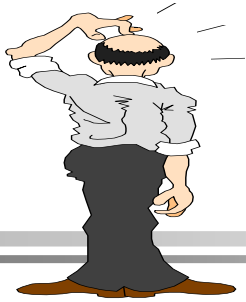
- Conclusion :

- Exemple redécrit et Exemple non redécrit ( $d_{ik} = 0$ )
- Extension aux contextes multivalués, et multiclassés
- Redescription étendue : vérification partielle

Un exemple  $o_i$  vérifie partiellement une hypothèse  $r$  avec un pourcentage égal à  $p/q$  si,  $r$  est de longueur  $q > 0$  et  $o_i$  possède  $p$  attributs de  $r$ .

- Technique pouvant être généralisée à tout système qui fait de l'induction de règles

Hypothèse  $\approx$  prémisse d'une règle



# SOMMAIRE

---

- ✓ Introduction
- ✓ Treillis de Galois
- ✓ Prétraitement de données
- Règles d'association
  - Classification supervisée
  - Conclusion



# Règles d'association :

---

- Objectif:

- » Recherche de relations d'association ou de corrélation intéressantes parmi un grand ensemble de données.

- Applications:

- » Analyse du panier d'un client en grande distribution

Quel groupe ou ensemble de produits sont fréquemment achetés ensemble par un client lors d'un passage au magasin ?

⇒ Disposition de produits à l'étalage

- » Exemple : Lait et Pain

Lorsqu'un client achète du lait, achete-t-il aussi du pain ? Si oui avec quelle fréquence? → 2 Mesures : Support, Confiance

# Règles d'association :

---

- Définition:

- » Item - Attribut                      ex: un produit
- » Ensemble d'items - Ensemble d'items fréquents
- » Transaction : Ensemble d'items, Exemple                      ex: un panier
- » Soient A et B deux sous-ensembles d'items,  
une **règle d'association** est une implication de la forme  $A \Rightarrow B$   
avec  $A \cap B = \emptyset$ .
- » Deux mesures :
  - Support : pourcentage de transactions qui contiennent A U B (à la fois A et B)      support  $(A \Rightarrow B) = P(A \cup B)$ .
  - Confiance : pourcentage de transactions contenant A qui contiennent aussi B      confiance  $(A \Rightarrow B) = P(B / A)$ .

# Règles d'association :

---

- Démarche:
  1. Rechercher tous les ensembles d'items **fréquents**, c-à-d dont le support est supérieur à un seuil minimum
  2. Générer les règles d'association **fortes** à partir des ensembles d'items fréquents, c-à-d dont le seuil minimum du support et le seuil minimum de confiance sont satisfaits
- » Etape 2 est le plus facile
- » Performance du processus de génération des règles d'association repose sur la 1ère étape.
- » Algorithme : Apriori [[Agrawal](#), [Mannila](#), [Srikant](#), [Toivonen](#) et [Verkamo](#), 1994, 1994, 1996]

# Règles d'association :

---

- Plusieurs types basés sur:
  - » Types de valeur  
Booléennes, Quantitatives
  - » Dimensions des données  
Simple, Multiple                    ex: tenir compte de +sieurs propriétés
  - » Niveaux d'abstraction  
Simple, Multiple                    ex: prise en compte d'une hiérarchie
  - » Autres extensions:
    - Ensembles d'items maximum (ou "Maxpatterns")
    - Ensembles **fermés** d'items (ou "frequent closed itemsets")
    - Contraintes sur les règles d'associations
    - Méta-règles pour guider la génération de règles d'association

# Règles d'association :

- Exemple:

- » Transactions : ensemble  $O = \{1, 2, 3, 4, 5, 6, 7\}$

- » Items : ensemble  $A = \{a, b, c, d, e, f, g\}$

Valeurs booléennes, Dimension simple, Abstraction simple

Support( $a \Rightarrow b$ ) = 6/7

Confiance( $a \Rightarrow b$ ) = 6/7

- » Support( $b \Rightarrow c$ ) = 5/7

- » Confiance( $b \Rightarrow c$ ) = 5/6

Support( $g \Rightarrow h$ ) = 1/7

Confiance( $g \Rightarrow h$ ) = 1/3

- » Support( $h \Rightarrow g$ ) = 1/7

- » Confiance( $h \Rightarrow g$ ) = 1/2

	a	b	c	d	e	f	g	h
1	1	1	1	1	1	1	1	
2	1	1	1	1	1	1		1
3	1	1	1	1	1		1	1
4	1	1	1	1		1		
5	1	1		1	1		1	
6	1	1	1		1			
7	1		1			1		

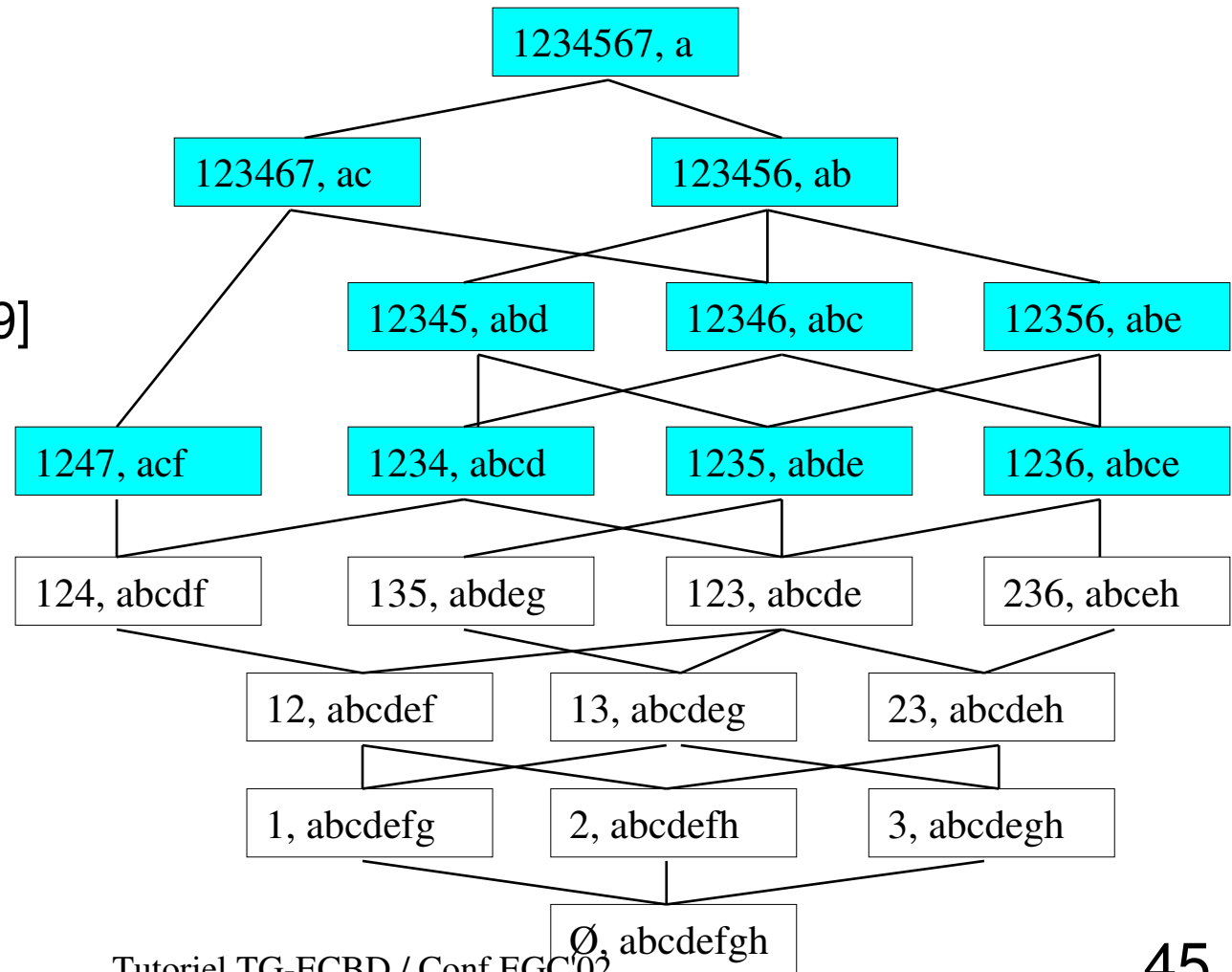
# Règles d'association :

- Exemple:  
Seuil support = 4

[Lakhal et al, 1999]

Algorithmes:

- Close
- Closet
- Charm
- Titanic
- ...



# Règles d'association :

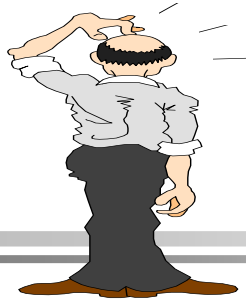
---

- Génération des ensembles de fermés fréquents:
  - » Bayardo, 1998, *ACM SIGMOD ICMD*.  
“Efficiently mining long patterns from databases”  
Pb lors passage des fermés fréquents à tous les ens d'items fréquents car  
génération à partir des bases de données
  - » *Pasquier, Bastide, Taouil & Lakhal, 1999, ICDT*  
“Discovering frequent closed itemsets for association rules”  
Algorithmes CLOSE, A-CLOSE,  
Thèses Pasquier 2000, Bastide 2000    Univ de clermont-ferrand
  - » Boulicaut & Bykowski, 2000, PAKDD conf.  
“Frequent closures as a concise representation for binary data mining”
  - » ....

# Règles d'association :

---

- Génération des règles à partir des fermés :
  - » Duquenne & Guigues, 1986, *Maths. et Sci. Hum.*  
Famille minimale d'implications informatives dans un tableau binaire
  - » Luxenburger, 1991, *Maths. et Sci. Hum.*  
Implications partielles dans un contexte
  - » Pasquier, Bastide, Taouil & Lakhal, 1999, *Information Systems*  
Adaptation de Duquenne-Guigues'86 et Luxenburger'91
  - » Bastide, Pasquier, Taouil, Stumme & Lakhal, 2000, *DOOD conf*  
Règles d'associations minimales et non redondantes
  - » ....



# SOMMAIRE

---

- ✓ Introduction - ECBD
- ✓ Treillis de Galois
- ✓ Prétraitement de données
- ✓ Règles d'association
- **Classification Supervisée**
- Conclusion



# Classification supervisée

---

- Types :
  - » Classification non supervisée (ou “Clustering”)
  - » **Classification supervisée** (ou “Classification” en anglais)
- Définition : Classification supervisée
  - » Processus à deux phases:
    1. Apprentissage : construire un modèle (ou classifieur) qui décrit un ensemble prédéterminé de classes de données, et
    2. Classement : utiliser le classifieur pour affecter une classe à un nouvel objet
- Domaines concernés:

Apprentissage automatique, Réseaux de neurones, Statistiques, Reconnaissance des formes, etc ...

# Classification supervisée

---

- Applications :

Attribution de crédit bancaire, Diagnostic Médical, Marketing Sélectif, Reconnaissance de gènes en Biologie, Prédiction de sites archéologiques, Prédiction du Ballon d'Or Européen (Football), .....

- Plusieurs techniques:

Induction d'arbres de décision, Réseaux de neurones, Réseaux bayésiens, Algorithmes génétiques, Apprentissage à partir de d'instances, k-PPV, [Induction à partir des treillis](#), Induction de règles de décision, ...

# Classification supervisée

---

- **Problème d'apprentissage (supervisée):**

**Données :**

- »  $f$  : fonction caractéristique de l'ensemble d'apprentissage; **inconnue**
- »  $O$  : ensemble d'apprentissage de taille fini,  $n \in \mathbb{N}$ , suite de couples  $(x_i, y_i)$  - exemple ou tuple ou objet ou instance ou observation
- »  $(x_i, y_i)$   $1 \leq i \leq n$ , exemple d'apprentissage tel que  $y_i = f(x_i)$
- »  $y_i$  indique la classe des exemples, nombre fini, valeur symbolique
- »  $A$  : ensemble d'attributs (propriété ou descripteur),  $m \in \mathbb{N}$
- »  $x_i = (x_{i1}, \dots, x_{im})$ , tel que  $x_{ij}$  = valeur de  $x_i$  pour l'attribut  $j$ .

**But :**

- » Construire un modèle (classifieur)  $f^\wedge$  qui **approxime** au mieux la fonction  $f$  à partir d'un ensemble d'exemples sélectionnés de manière aléatoire dans  $O$

# Classification supervisée

---

- Apprenti : qui apprend ?
- Domaine : apprendre quoi ?
- Information initiale : à partir de quoi ?
  - » Exemples
  - » Questions à un Maître
  - » Expérimentation
- Connaissance à priori : Que sais-je ?
- Critères de performance : Comment valider ?
  - » Batch ou On-line,                      Forme Connaissance apprise
  - » Taux d'erreur (Accuracy),                      Complexité (Efficacité)



# Classification supervisée

---

- **Problème de classement :**

**Données :**

»  $f^\wedge$  : classifieur; **modèle appris**

»  $x_k$  : exemple

**But :**

» Déterminer  $y^\wedge_k = f^\wedge(x_k)$ , classe d'un nouvel exemple  $x_k$  :

**Question :**

» Comment apprécier la différence entre  $f$  et  $f^\wedge$  ?

– Réponse: calcul du taux de précision ou du taux d'erreur

# Classification supervisée

---

- Taux de précision du classifieur :
  - » Pourcentage des exemples de l'ensemble test qui sont correctement classés par le modèle
  - » Taux d'erreur =  $1 - \text{Taux de précision}$
- Ensemble d'exemples dont on connaît les classes, découpé en 2 (technique du "holdout") :
  - » Un ensemble utilisé dans la phase d'apprentissage
  - » Un ensemble de test utilisé dans la phase de classement
- Plusieurs autres techniques de découpage, issues des statistiques : (voir [Dietterich, *RR*'97], pour comparaison)
  - » Validation croisée, Resubstitution, "Leave-one-out"

# Classification supervisée

---

- Critères de comparaison de classifieurs :
  1. **Taux de précision** : capacité à prédire correctement
  2. **Temps de calcul** : temps nécessaire pour apprendre et tester  $f^{\wedge}$
  3. Robustesse : précision en présence de bruit
  4. **Volume de données** : efficacité en présence de données de grande taille
  5. Compréhensibilité : Niveau de compréhension et de finesse
- Problèmes
  - » Critères 1 et 2 “mesurables”
  - » Critère 4 important pour l'ECBD
  - » Critères 3 et 5 “laissés à l'appréciation” de l'utilisateur-analyste

# Classification Supervisée

---

- Exemple :  
*Ballon d'or Football*
  - »  $O_+ = \{\text{Platini, Weah}\}$
  - »  $O_- = \{\text{Desailly}\}$
  - »  $O? = \{\text{Anelka}\}$
  - »  $A = \{\text{JouerNordFrance, JouerEnItalie, JouerEquipeFrance}\}$

O\A	a	b	c	Classe
1:Platini	1	1	1	oui
2:Weah	1	1		oui
3:Desailly		1	1	non
4:Anelka	1		1	?

# Classification supervisée

---

- **Arbres de décision :**
  - » Simplicité, Efficacité (complexité polynomiale)
  - » Concepts disjonctifs
  - » Représentation restrictive (attribut-valeur): discrétisation possible
  - » Génération de règles de type Si-Alors
  
  - » Problèmes : Duplication des nœuds, Fragmentation de données,
  
  - » Biais de la mesure de sélection des attributs :
    - gain d'information, gain ratio, gini index, chi2, ...
  
  - » Algorithmes :
    - CLS [1966], CART [1984], **ID3** [ML'86], **C4.5** [1993], ...
    - SLIQ [EDBT'96], SPRINT [VLDB'96], ... pour les grandes bases de données

# Classification supervisée

---

- **Arbres de décision :**

- » Principe:

- Chaque noeud interne teste un attribut
- Chaque branche = valeur possible de cet attribut
- Chaque feuille fournit une classification
- Chaque chemin dans l'arbre correspond à une règle
- Ordre sur les attributs ~ pouvoir de discrimination

- » Algorithme de base :

1. Choisir le "meilleur" attribut
2. Etendre l'arbre en rajoutant une nouvelle branche pour chaque valeur de l'attribut
3. Répartir les exemples d'app. sur les feuilles de l'arbre
4. Si aucun exemple n'est mal classé alors arrêt, sinon répéter les étapes 1-4 pour les feuilles

# Classification supervisée

---

- **Arbres de décision :**

- » Problème: Quel est le meilleur entre  $a_i$  et  $a_j$  ?

- » Solution :

- Mesure d'entropie  $E(I)$  -> meilleure préclassification

- Gain d'Information,  $\text{Gain}(A,I)$ , en testant l'attribut A

- » Mesure d'entropie:

$$E(I) = - (p/(p+n)) \log_2(p/(p+n)) - (n/(p+n)) \log_2(n/(p+n))$$

I : ensemble d'exemples

p : nombre d'exemples positifs      n : nombre d'exemples négatifs

- » Gain d'Information: Différence entre l'incertitude avant et après la sélection de l'attribut

$$\text{Gain}(A,I) = E(I) - \sum_j ( ((p_j+n_j)/(p+n)) * E(I_j) )$$

le jème descendant de I est l'ens. d'exples avec la valeur  $v_j$  pour A

**Sélection de l'attribut qui maximise le gain d'information**

# Classification supervisée

---

- Pourquoi les treillis de Galois ?
  - Complexité exponentielle !
  - + Cadre pour la classification supervisée et non supervisée  
Concept  $\approx$  Extension + Intension
  - + Exploration d'une alternative aux arbres de décision
  - + Structure redondante  $\rightarrow$  duplication supprimée
  - + Espace de recherche exhaustif et concis
  - + Représentation géométrique intuitive – organisation hiérarchique
  - + Propriétés de symétrie et d'invariance
  - + Règles de la forme Si-Alors
- ? Précision des méthodes existantes

# Classification supervisée :

## Treillis de Galois

---

- Systèmes :
  - » CHARADE [Ganascia, 87, *IJCAI*]
  - » GRAND [Oosthuisen, 88, PhD thesis, Glasgow]
  - » LEGAL [Liquière & Mephu, 90, *JFA*]
  - » Travaux Godin *et al.*, 91
  - » GALOIS [Carpineto & Romano, 93, *ICML*]
  - » RULEARNER [Sahami, 95, *ECML*]
  - » ...
  - » GLUE, IGLUE, CIBLe [Njiwoua & Mephu, ...]
  - » Flexible-LEGAL [Zegaoui & Mephu, 99, *SFC*]

# Classification supervisée :

## Le système LEGAL

---

- Principe apprentissage :
  - » Sélection quantitative
    - **Validité** : une hypothèse est valide si elle est vérifiée par “assez” d'exemples positifs
    - **Quasi-cohérence** : une hypothèse est quasi-cohérente si elle est vérifiée par “peu” d'exemples négatifs
    - ❖ Une hypothèse est **sélectionnée** si elle est valide et quasi-cohérente.

Hypothèse  $\approx$  intension d'un concept du treillis

Un concept du treillis est sélectionné si son intension l'est
  - » Autres critères
    - Minimalité, Maximalité

# Classification supervisée :

## Le système LEGAL

---

- Principe apprentissage :
  - » Construction d'un sup-demi-treillis
    - Approche descendante
    - Eviter le sur-apprentissage
    - Adaptation Algorithme Bordat 86 tq seuls les concepts valides sont générés
  - » Propriétés :
    - Si un nœud n'est pas valide, alors tous ses successeurs (sous-nœud) ne le sont pas.
    - Si un nœud est valide alors tous ses prédecesseurs (sur-nœud) le sont.
  - » Paramètres
    - Seuils de validité et de quasi-cohérence choisis par l'utilisateur

# Classification supervisée :

## Le système LEGAL

---

- Algorithme Apprentissage :

- »  $L = (O, \emptyset)$
- » Pour chaque concept  $(O_i, A_i)$  de L
  - Rechercher couverture  $C = \underline{(O_i, A_i)}$
  - Pour chaque  $(O_j, A_j) \in C$ 
    - Si Validité  $((O_j, A_j))$  alors
      - » Si  $(O_j, A_j) \notin L$  alors ajouter  $(O_j, A_j)$  à L
      - » Sinon rajouter un arc seulement
  - Fin Pour
- » Fin Pour

Seuls les nœuds valides sont générés

# Classification supervisée :

## Le système LEGAL

---

- Principe classement :
  - » Vote majoritaire
    - Un exemple est considéré comme un **exemple positif** s'il vérifie "suffisamment" hypothèses pertinentes --- Justification
    - Un exemple est considéré comme un **exemple négatif** s'il vérifie "peu" hypothèses pertinentes --- Refutation
    - Dans les autres cas, le système est silencieux.
  - » Paramètres :
    - Seuils de justification et de refutation sont choisis par l'utilisateur, ou peuvent être calculés par le système

# Classification supervisée :

## Le système LEGAL

---

---

- Variantes :
  - » Maximalité des concepts : les plus généraux, rapidité
  - » Minimalité des concepts : les plus spécifiques
  - » LEGAL-E :
    - Seuls les exemples positifs sont utilisés pour générer les noeuds du treillis
  - » LEGAL-F :
    - Intégrer les seuils de validité pour sélectionner les attributs
  - » NoLEGAL :
    - Représentation sous forme attribut-valeur nominale
  - » FlexibleLEGAL :
    - Introduction des sous-ensembles flous

# Classification supervisée :

## Le système LEGAL

- Exemple 1 :

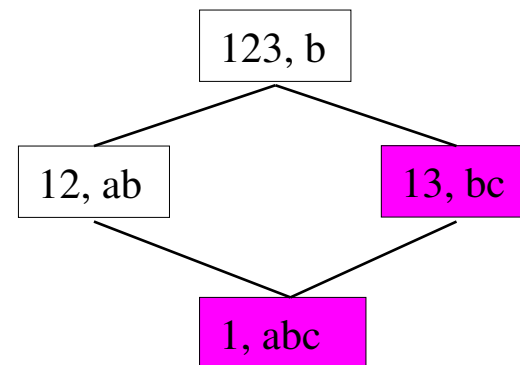
A = { a:JouerNordFrance,  
 b:JouerEnItalie,  
 c:JouerEquipeFrance }

Validité = 100%

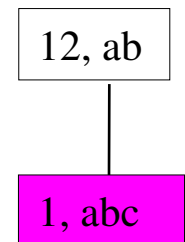
Quasi-cohérence = 0%

O\A	a	b	c	Classe
1:Platini	1	1	1	oui
2:Weah	1	1		oui
3:Desailly		1	1	non
4:Anelka	1		1	?

LEGAL



LEGAL-E



Arbre de décision

Si *JouerNordFrance* alors Ballon d'Or

# Classification supervisée :

## Le système LEGAL

- Exemple 2 :

LEGAL

	a	b	c	d	e	f	g	h	$y_i$
1	1	1	1	1	1	1	1		+
2	1	1	1	1	1	1		1	+
3	1	1	1	1	1		1	1	+
4	1	1	1	1		1			+
5	1	1		1	1		1		-
6	1	1	1		1				-
7	1		1			1			-

LEGAL-E

	a	b	c	d	e	f	g	h	$y_i$
1	1	1	1	1	1	1	1		+
2	1	1	1	1	1	1		1	+
3	1	1	1	1	1		1	1	+
4	1	1	1	1		1			+
5	1	1		1	1		1		-
6	1	1	1		1				-
7	1		1			1			-

# Classification supervisée :

## Le système LEGAL

- Exemple 2 :
- Seuil Validité = 3/4
- S. Quasi-cohérence = 1/3

Valide, non quasi-cohérent

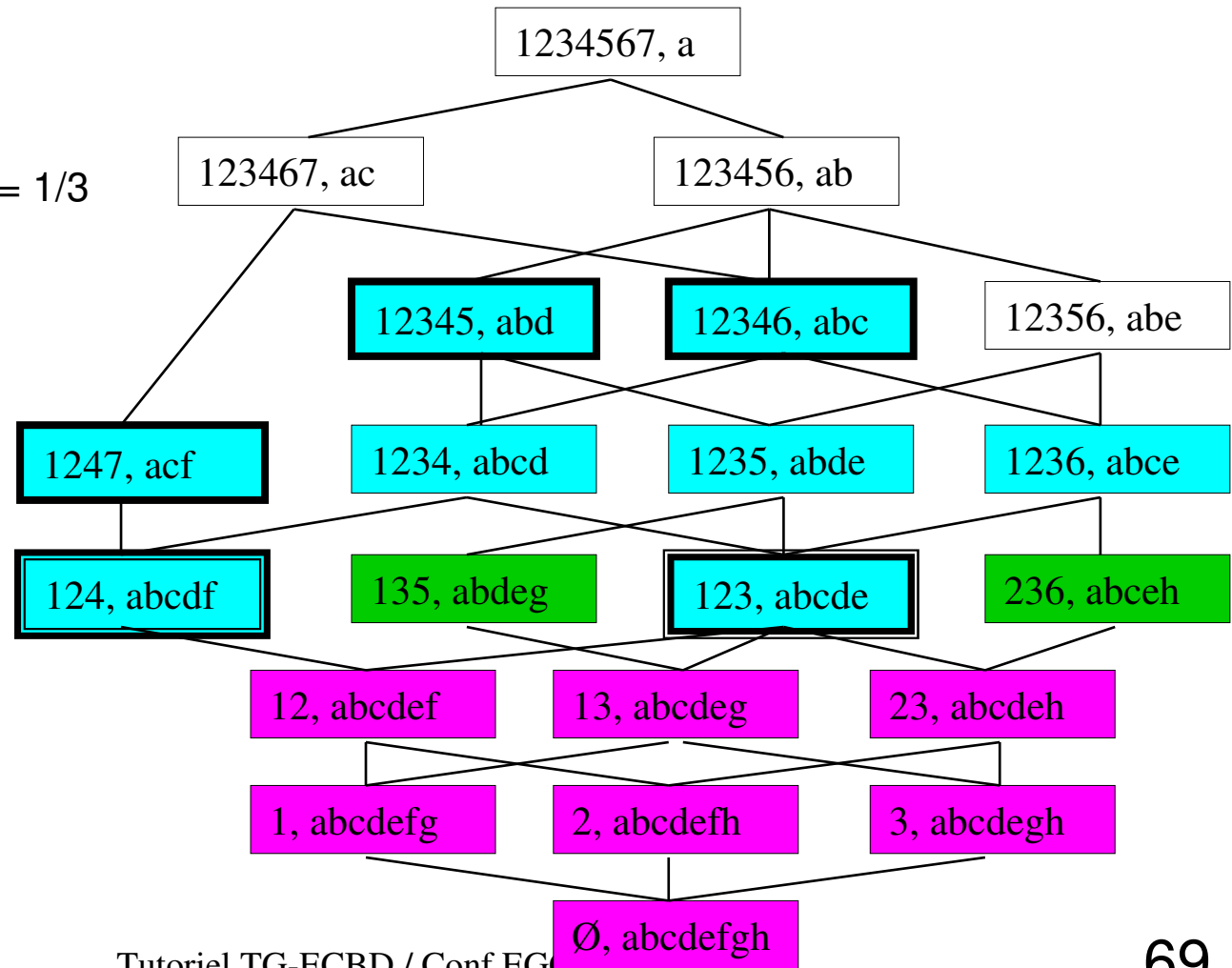
Valide, quasi-cohérent

V, QC, maximal

V, QC, minimal

Non valide, mais généré

Non généré

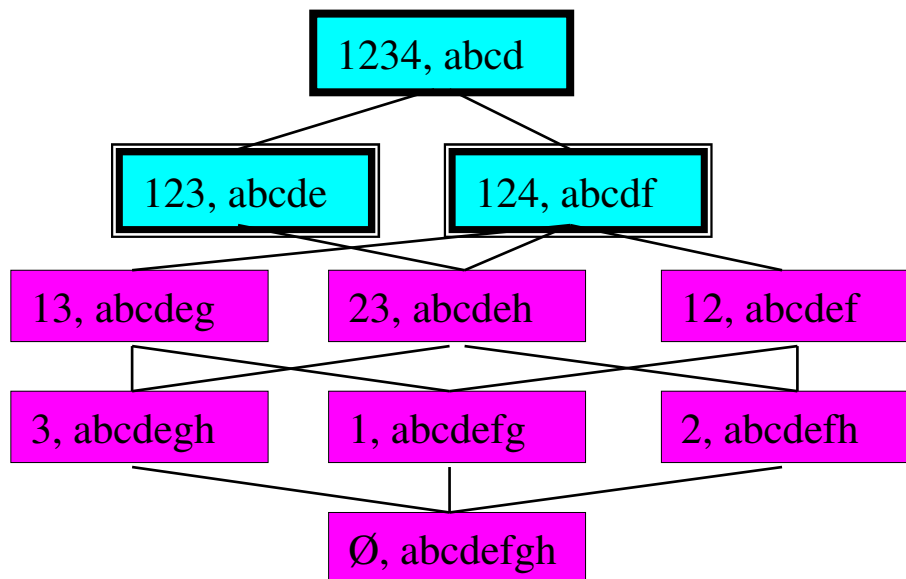


# Classification supervisée :

## Le système LEGAL-E

- Exemple 2 :
- Seuil Validité = 3/4
- S. Quasi-cohérence = 1/3

	a	b	c	d	e	f	g	h	$y_i$
1	1	1	1	1	1	1	1		+
2	1	1	1	1	1	1		1	+
3	1	1	1	1	1		1	1	+
4	1	1	1	1		1			+
5	1	1		1	1		1		-
6	1	1	1		1				-
7	1		1			1			-



Valide, non quasi-cohérent

Valide, quasi-cohérent

V, QC, minimal

V, QC, maximal

Non valide, mais généré

Non généré

# Classification supervisée :

## Le système LEGAL

---

- Remarques :
  - » Difficulté en présence de contexte de taille très grande
    - Complexité exponentielle
    - Exhaustivité du treillis
  - » Solutions :
    - Biais d'apprentissage (limitant espace recherche) : validité, quasi-cohérence, exemples positifs, ....., mais exhaustivité
    - **Approximation du treillis pour limiter l'exhaustivité**
      - Treillis + sous-ensembles flous → Treillis de concepts flexibles
      - Systeme Flexible-LEGAL

# Classification supervisée :

## Le système Flexible-LEGAL

---

- Théorie des sous-ensembles flous:
  - » Soit  $O$ , un ensemble de référence,  $o_i \in O$
  - » Un sous-ens classique  $O_x$  de  $O$  est défini par  $\theta$  tel que :  
$$\theta(o_i) = 0 \text{ si } o_i \notin O_x \quad \text{et} \quad \theta(o_i) = 1 \text{ si } o_i \in O_x$$
  - » Un **sous-ens flou**  $O_y$  de  $O$  est défini par une fonction d'appartenance  $\mu$  qui assigne à chaque élément  $o_i$  de  $O$ , un nombre réel  $\in [0,1]$ , décrivant le degré d'appartenance de  $o_i$  à  $O_y$
  - » **Noyau**,  $N(O_y) = \{o_i \in O, \mu_{O_y}(o_i) = 1\}$
  - » **Support**,  $S(O_y) = \{o_i \in O, \mu_{O_y}(o_i) \neq 0\}$
  - » Hauteur de  $O_y$ ,  $h(O_y) =$  plus grande valeur de  $\mu_{O_y}$
  - »  $O_y$  est normalisé si  $h(O_y) = 1$

# Classification supervisée :

## Le système Flexible-LEGAL

---

- Principe Apprentissage et Classement:
  - » Idem LEGAL
  - » Différence avec LEGAL: Génération des nœuds du treillis
- Principe génération du treillis
  - » Si la différence entre les exemples vérifiant l'attribut  $a_i$  et l'attribut  $a_j$ , est insignifiante, alors  $a_i$  et  $a_j$  sont similaires
  - » Mesure de similarité entre attributs, *diff*  
Si  $diff(g(a_i), g(a_j)) \leq \delta$  alors  $a_i$  et  $a_j$  similaires
- Paramètre  $\delta$  : seuil similarité choisi par l'utilisateur

# Classification supervisée :

## Le système Flexible-LEGAL

---

- Fonction d'appartenance à un concept,  $\mu$ :
  - » Soit  $(O_1, A_1)$ , un concept flexible
  - »  $o_i \in O_1, m = |A_1|$
  - »  $m_{o_i}$  = nombre d'attributs de  $A_1$  vérifiés par  $o_i$
  - » Fonction appartenance,  $\mu(o_i) = m_{o_i} / m$

Concept flexible  $\approx$   
support d'un sous-ensemble flou + intension

# Classification supervisée :

## Le système Flexible-LEGAL

- Exemple 2 :

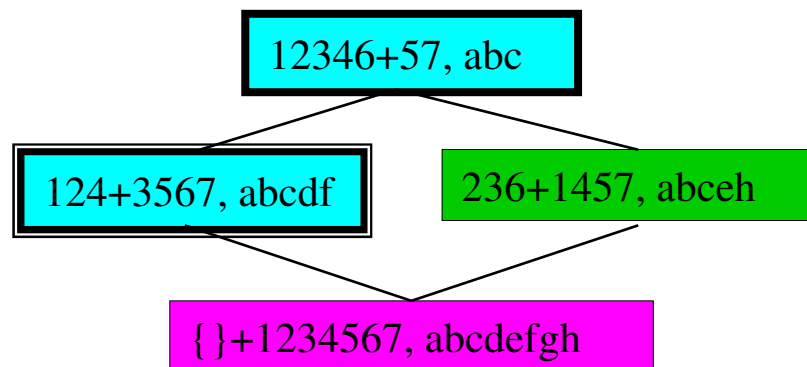
Seuil Validité =  $\frac{3}{4}$  S. Quasi-cohérence =  $\frac{1}{3}$

S. Similarité =  $\frac{1}{7}$

$C_1 = (12346+57, abc)$   $\mu(3) = 100\%$   $\mu(5) = 66\%$

Noyau( $C_1$ ) = {12346} Support( $C_1$ ) = {1234567}

	a	b	c	d	e	f	g	h	$y_i$
1	1	1	1	1	1	1	1		+
2	1	1	1	1	1	1		1	+
3	1	1	1	1	1		1	1	+
4	1	1	1	1		1			+
5	1	1		1	1		1		-
6	1	1	1		1				-
7	1		1			1			-



Valide, non quasi-cohérent

Valide, quasi-cohérent

V, QC, maximal

V, QC, minimal

Non valide, mais généré

Non généré

# Classification supervisée :

## Le système Flexible-LEGAL

---

- Expérimentations:
  - » 5 Jeux de données artificielles (3) et réelles (2) de UCI Irvine
  - » Validation croisée d'ordre 5
  - » Mesure du temps CPU, hauteur treillis, nombre de concepts, et taux de précision
  - » Variation de seuil de similarité,  $\delta$  : 0%, 25%, 35%
- Observations:
  - » Sur un des problèmes réels, **gain d'un facteur 10 en temps CPU**, d'un **facteur 30 en espace mémoire**, avec un **taux de précision meilleur**, par rapport à LEGAL
  - » Pas de variation sur les données artificielles
  - » *Difficulté de choix de  $\delta$*

# Classification supervisée

---

- Conclusion sur LEGAL et variantes
  - » **Logique majoritaire** : élimine les concepts dont l'extension n'est pas suffisamment grand mais pouvant être discriminants

Solution : Mesures d'information (Entropie, Loi de succession de Laplace)

→ Systèmes GLUE, IGLUE et CIBLe

# Classification supervisée :

## IGLUE - CIBLe

---

- Double objectif :
  - » Introduire une mesure d'information pour sélectionner les hypothèses
  - » Combiner une approche inductive reposant sur le treillis (redescription) et une approche d'apprentissage à partir d'instances pour faire de l'induction constructive
- ⇒ Mise au point d'une technique de sélection dynamique d'instances représentatives pour l'apprentissage à partir d'instances

# Classification supervisée :

## IGLUE - CIBLe

---

- Apprentissage à partir d'instances :
  - » En anglais: Instance-based learning ou Lazy learning
  - » Simplicité, Induction paresseuse
  - » Principe:
    - Donnée: instances + leurs classes
    - L'apprentissage consiste à stocker les instances représentatives (ou prototypes) des classes.
    - Une mesure de similarité ou de distance est définie entre instances
    - La phase de classement fait appel à la technique des plus proches voisins (PPV) pour affecter une classe à un nouvel exemple
  - » Notions de voisinage, de proximité
  - » Appropriée pour les données numériques
- ⇒ **Limites**: influence mesure de similarité, difficulté de prise en compte attributs symboliques, complexité de la phase de classement

# Classification supervisée :

## IGLUE - CIBLe

---

- Principe commun:
  - » Construction du Sup-demi-treillis, et génération de concepts pertinents à l'aide d'une fonction de sélection
  - » Redescription du contexte initial
  - » Classement avec la technique du PPV, en choisissant une mesure de similarité/distance pour données numériques
- Différences
  - ⇒ Construction du demi-treillis: Contexte binaire et à 1 classe pour IGLUE, alors que CIBLe traite les contextes multivalués et multi-classes
  - ⇒ Redescription : appariement complet pour IGLUE, appariement complet ou partiel dans CIBLe
  - ⇒ Classement : Utilisation d'une méthode de sélection dynamique de prototypes dans CIBLe

# Classification supervisée :

## IGLUE - CIBLe

---

- Expérimentations (voir thèse Njiwoua, 00, Univ d'Artois):
  - » Validation croisée sur 37 ensembles de l'UCI
  - » Mesure temps cpu et taux de précision
  - » Test de plusieurs fonctions de sélection et de mesures de similarité
  - » Comparaison avec plusieurs méthodes : C4.5, K\*, IBi, KNNFP, PEBLS
- Observations:
  - ⇒ Résultats comparables à ceux des méthodes standard
  - ⇒ Robustesse de l'approche
  - ⇒ Taux de précision généralement meilleur avec comme fonction de sélection la loi de succession de Laplace qu'avec l'entropie
  - ⇒ Appariement partiel meilleur appariement complet
  - ⇒ Sur certains cas, taux de précision de IBi, C4.5, KNNFP sont meilleurs sur le contexte redécrit que sur le contexte initial
  - 👉 Données hybrides – Fusion attributs numériques (redécrits et initiaux) ?

# Classification Supervisée

---

- Conclusion :

1. Fonction au cœur de l'EBCD

2. Plusieurs systèmes s'appuyant sur le treillis de Galois développés et évalués

3. Théorème «No Free Lunch» [Schaffer 94, *ICML*]

⇒ Treillis de Galois: cadre pertinent pour la classification



# SOMMAIRE

---

- ✓ Introduction - ECBD
- ✓ Treillis de Galois
- ✓ Prétraitement de données
- ✓ Règles d'association
- ✓ Classification supervisée
- Conclusion



# Conclusion

---

- Travaux :
  1. **Prétraitement de données**
  2. **Règles d'association**
  3. **Classification supervisée**
  
  4. Classification non supervisée
    - » ...
  5. Extension sur les Types de données
    - » Diday & al, « objets symboliques »
    - » Wolff, CLKDD2001, « temporal concept analysis »
    - » ...

# Conclusion

---

- Applications :
  - » Indexation documentaire :
    - Godin & al., 1986, *Information Sciences*  
«Lattice Model of Browsable Data Spaces»
    - Carpineto & Romano, 1996, *Machine Learning*  
«A lattice conceptual clustering system and its application to browsing retrieval
    - Cole, Eklund & Stumme, 2002, preprint **WEB**  
« Document retrieval for email search and discovery using formal concept analysis »
    - ...
  - » Bioinformatique :
    - Thèse Mephu, 1993, Univ. de Montpellier
    - Duquenne & al., 2001, CLKDD proceedings  
« Structuration of phenotypes/genotypes through Galois lattices and Implications »

# Conclusion

---

- Perspectives :
  - » Treillis de Galois: cadre pertinent pour l'ECBD ?
  - » Problème: Taille des données ?  
Solutions: Echantillonnage, Usage de mémoire secondaire, Parallélisme, ...
  - » Pistes à explorer :
    - Algorithmes (efficacité à améliorer)
    - Approximations
    - Usage de connaissance à priori
    - ...

# FIN

---

---

● ....

