Introduction à l'Extraction de Connaissances

Chapitre IV : les méthodes

Part 2

Extraction de Règles (de classification, d'association)

Mesures d'évaluation des règles

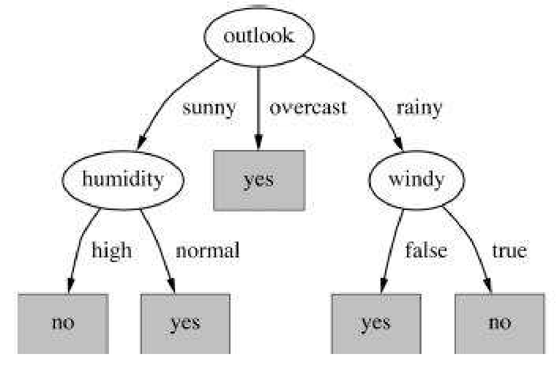
Alexandre Saidi
Master -Informatique
ECL - LIRIS - CNRS

Octobre-Novembre 2017

Mesures simples d'évaluation de modèles

Rappel du chapitre précédent :

l'arbre de décision (AD) pour "météo" (pour ID3 et C4.5) :



Mesures simples d'évaluation de modèles (suite)

- Le l'importance de la méthode d'évaluation (pour l'exemple "météo") :
  - Si on utilise toutes les données pour apprendre, l'erreur sera nulle.
  - Par contre, avec 10-XV, on aura une erreur de $\frac{2}{14}$

  ➛ On obtient des détails sur ces erreur par une **matrice de confusion**.

- Pour la BD "météo" et 3 manières de validation

Matrice si toute la BD. utilisée :

```
a b <- classified as
9 0 | a = yes
0 5 | b = no
```

Matrice si 10-XV :

```
a b <- classified as
8 1 | a = yes
1 4 | b = no
```

Matrice si $\frac{2}{3}$ pour apprendre et $\frac{1}{3}$ pour le test :

```
a b <- classified as
3 0 | a = yes
2 0 | b = no
```

☞ Pour le 3e cas (à droite), on ne considère que **l'ensemble de test**.

Mesures simples d'évaluation de modèles (suite)

- Une matrice de confusion est une table de contingences.

  Les valeurs actuelles (BD) sont celles qui sont dans la BD (qu'on aimerait voire prédites) et les valeurs "modèle" sont les conclusions (prédictions) du modèle appris.

		Valeurs actuelles (BD)		
		Positifs	Négatifs	Total
Valeurs	Positifs →	Vrais positifs (TP)	Faux Positifs (FP)	les positifs (modèle)
Prédites	Négatifs →	Faux Négatifs (FN)	Vrais Négatifs (TN)	les négatifs (modèle)
(Modèle)	Total	les positifs (BD)	les négatifs (BD)	

- Par exemple (sortie Weka) :

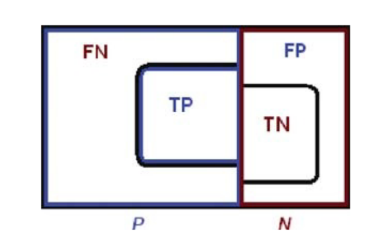
```
 a   | b   |    <-- classified as
--------------------------------------
 8   | 1   |     a   = yes
 1   | 4   |     b   = no
```

Mesures simples d'évaluation de modèles (suite)

Petite introduction à l'évaluation (ici pour les ADs + matrice de confusion).

- **Quelques unes des mesures :**

Soit TN : True Negative,
FP : False Positive, ...
P =TP+FN, N = TN+FP



- Sensibilité = TP/P = TP/ TP + FN
  est aussi appelée *Recall = rappel*
- Spécifité = TN/N = TN/ FP + TN
- Justesse = TP + TN / P + N
  ou parfois (si pas de TN) :
  justesse = TP/ P + N
- Précision = TP/ TP + FP

Remarque : dans une tâche d'extraction d'information (IR, par ex. Google), on a :
TP + FN = P = les documents **relevants** (dans la BD)
TP + FP= les documents **extraits** (présentés comme les bonnes réponses).

Mesures simples d'évaluation de modèles (suite)

Remarques :

- $Recall\ (Rappel) = \frac{TP(Modele)}{TP+FN(B.D.)} = \frac{TP(Modele)}{P(B.D.)}$ (une mesure Modèle vs. la B.D.)
  - La *Sensibilité* (*Rappel*) se dit en anglais *Recall* (*Sensitivity*)
  - Pour une classe c : $Recall_c\ (rappel_c) = \frac{\text{nbr. d'instances correctement attribuées à la classe } c \text{ (Modèle)}}{\text{nbr. d'instances appartenant à la classe } c \text{ (B.D.)}}$
  - Si je demande à Google "r=combien de moutons à 5 pattes ?", on aura :

    $$Recall_r\ (Rappel_r) = \frac{\text{nbr. de réponse correctes trouvées par Google (Modèle)}}{\text{nbr. de moutons à 5 pattes qui existent (B.D.)}}$$
  - Si le taux de *Rappel* est bas, on parle du Silence (l'opposé du *Rappel*).

- $Precision = \frac{TP(Modele)}{TP+FP(Modele)}$ (une mesure sur le Modèle)
  - Pour une classe c : $Precision_c = \frac{\text{nbr. d'instances correctement attribuées à la classe } c \text{ (Modèle)}}{\text{nbr. d'instances attribuées à la classe } c \text{ (Modèle)}}$
  - Pour la requête r ci-dessus :

    $$Recall_r\ (Rappel_r) = \frac{\text{nbr. de réponse correctes trouvées par Google (Modèle)}}{\text{nbr. de réponses de Google : ce que Google croit être la bonne réponse (Modèle)}}$$

➤ Dans la *Précision* (et Google), la part FP (les mauvaises réponses de Google) est du Bruit.

Mesures simples d'évaluation de modèles (suite)

- Pour un cas **multi-classes** (ensemble $C = \{c_1, c_2, ..., c_k\}$)

$$Rappel = \frac{\displaystyle\sum_{i=1}^{k} Rappel_i}{k}$$

$$Precision = \frac{\displaystyle\sum_{i=1}^{k} Precision_i}{k}$$

- Un modèle *parfait* aura $Rappel = Precision = 1$:

  Sachant que $Rappel\ (Recall) = \frac{TP}{TP+FN}$ et $Precision = \frac{TP}{TP+FP}$

➢ Il trouvera la totalité des instances pertinentes (cf. *Rappel* où $FN = 0$) et ne fait aucune erreur (cf. *Précision* où $FP = 0$).

Mesures simples d'évaluation de modèles (suite)

- Résumé des mesures (+ quelques détails) :

true positive (TP)

eqv. with hit

true negative (TN)

eqv. with correct rejection

false positive (FP)

eqv. with false alarm, Type I error

false negative (FN)

eqv. with miss, Type II error

sensitivity or true positive rate (TPR)

eqv. with hit rate, recall

$$TPR = TP/P = TP/(TP + FN)$$

specificity (SPC) or True Negative Rate

$$SPC = TN/N = TN/(FP + TN)$$

precision or positive predictive value (PPV)

$$PPV = TP/(TP + FP)$$

negative predictive value (NPV)

$$NPV = TN/(TN + FN)$$

fall-out or false positive rate (FPR)

$$FPR = FP/N = FP/(FP + TN) = 1 - SPC$$

false discovery rate (FDR)

$$FDR = FP/(FP + TP) = 1 - PPV$$

Miss Rate or False Negative Rate (FNR)

$$FNR = FN/P = FN/(FN + TP)$$

accuracy (ACC)

$$ACC = (TP + TN)/(P + N)$$

F1 score

is the harmonic mean of precision and sensitivity

$$F1 = 2TP/(2TP + FP + FN)$$

Matthews correlation coefficient (MCC)

$$\frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Informedness = Sensitivity + Specificity - 1

Markedness = Precision + NPV - 1

Mesures simples d'évaluation de modèles (suite)

		Condition (as determined by "Gold standard")			
	Total population	Condition positive	Condition negative	Prevalence = Σ Condition positive / Σ Total population	
Test outcome	Test outcome positive	**True positive**	**False positive** (Type I error)	Positive predictive value (PPV, Precision) = Σ True positive / Σ Test outcome positive	False discovery rate (FDR) = Σ False positive / Σ Test outcome positive
	Test outcome negative	**False negative** (Type II error)	**True negative**	False omission rate (FOR) = Σ False negative / Σ Test outcome negative	Negative predictive value (NPV) = Σ True negative / Σ Test outcome negative
	Positive likelihood ratio (**LR+**) = TPR/FPR	True positive rate (TPR, Sensitivity, Recall) = Σ True positive / Σ Condition positive	False positive rate (FPR, Fall-out) = Σ False positive / Σ Condition negative	Accuracy (ACC) = Σ True positive + Σ True negative / Σ Total population	
	Negative likelihood ratio (**LR−**) = FNR/TNR	False negative rate (FNR) = Σ False negative / Σ Condition positive	True negative rate (TNR, Specificity, SPC) = Σ True negative / Σ Condition negative		
	Diagnostic odds ratio (**DOR**) = LR+/LR−				

A propos de la matrice de Confusion

- Pour un cas bi-classes, les valeurs TP, TN, FP et FN sont directement accessibles.
- Par contre, pour une classification avec plus de 2 classes, comment procéder ?
- Exemple des sorties d'un modèle avec les codes couleurs :
  - ➛ noire : Bien classés, orange : Chats pris pour autre chose, magenta : Chiens pris pour autre chose, bleu : Loups pris pour autre chose.

		Actuels (B.D.)		
		Chat	Chien	Loup
Prédictions	Chat	5	2	0
	Chien	3	3	2
	Loup	0	1	11

- Constat : sur 8 Chats, 5 ont été bien classés (TP) et 3 ont été pris pour des Chiens.
- Parmi les 6 Chiens, 3 ont été bien classés (TP) et 2+1=3 autres non.
- Parmi les 13 loups, 11 ont été bien classés (TP) et 2 autres non.

A propos de la matrice de Confusion (suite)

Calcul de TP,TN,FP et FN pour le **Chat** (suivre les couleurs) :

- Rappel de la table de confusion des 3 classes :

		Actuels (B.D.)		
		Chat	Chien	Loup
Prédictions	Chat	5	2	0
	Chien	3	3	2
	Loup	0	1	11

- Considérez *Chat* × *Chien* mais on a besoin des non-Chats (= Chien + Loup)

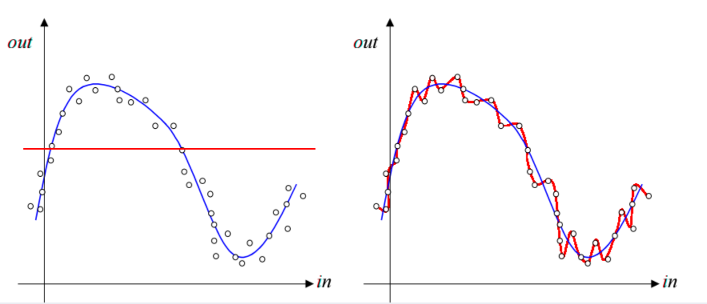
	Chat	**Chien**	
Chat	TP=5 Chiens	FP= 2 + 0	← Positifs (Modèle)
Chien	FN= 3 + 0	TN= 3+11 + 1+2 =17	← Négatifs (Modèle)

- Il y a dans cette table :
  - 5 Chats bien classés (TP) et 3 des (vrais) Chats sont pris pour des Chiens (FN)
  - 2 Chiens Pris pour Chats (FP)
  - Les TN pour notre matrice (on prend Chats comme repère) :
    - ➛ 11 Loups et 3 Chiens sont bien classés (donc TN pour Chats),
    - ➛ 2 Loups sont pris pour des Chiens (TN pour Chats) car ils ne sont pas pris pour des Chats ; 1 Chien est pris Loup (TN pour la même raison.)

- De même pour le *Chien* et le *Loup* : on crée les matrices de confusion 2 à 2.

Biais / Variance : le compromis

- Les données = les points (les ronds) sur lesquelles on trouve 2 modèles.



- A gauche (la droite rouge) : on ignore presque les données ! Le modèle obtenu
  ➛ Erreur élevée (biais important) mais peu de variance (pour les BDs diff.)
- A droite : on colle trop aux données (la courbe rouge passe par ttes les données) !
  ➛ Erreur faible (biais nul) mais variance élevée (pour les BDs diff.)

Pbs. : Underfitting/Overfitting dans AD

- Le compromis Under/Over-Fitting est relié au Biais / Variance.

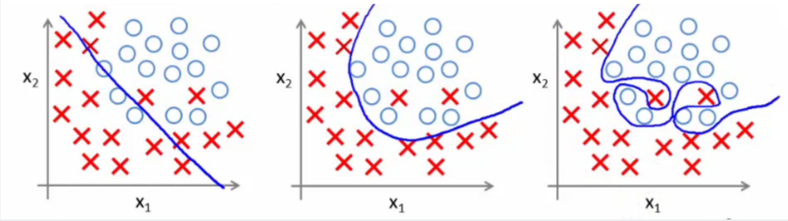


FIGURE 1: UnderFitting + Biais / - Variance — Juste Biais / variance plus équilibré — Overfitting - Biais / + Variance

☞ Méfiez-vous des modèles sans erreur !

- Le bon compromis minimise le biais et la variance (l'over/under-fitting).
- Détection par de solides mesures d'évaluation (voir plus loin)
- Au chapitre précédent (CART et classe numérique) :
  ➢ voir minimisation de $Biais^2 + Variance$ en minimisant σ^2

Pbs. : Underfitting/Overfitting dans AD (suite)

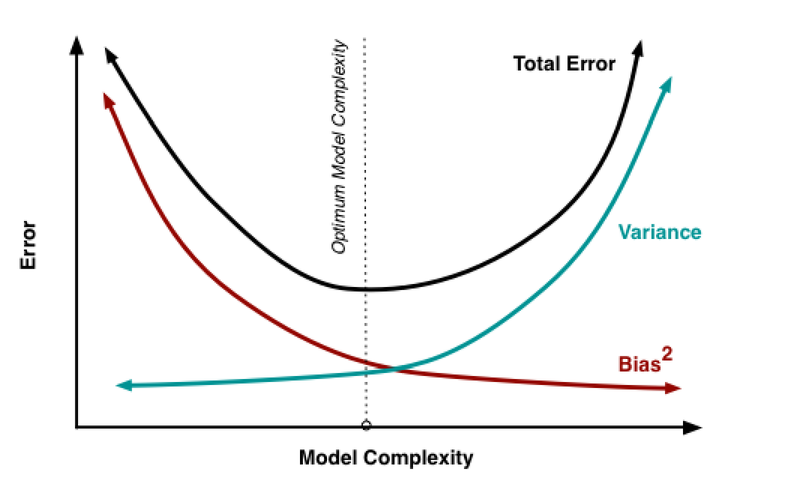


FIGURE 2: Contribution du couple Biais / Variance à l'erreur du modèle

Pbs. : Underfitting/Overfitting dans AD (suite)

- Il y a **Underfitting** lorsque le modèle est trop simple et que le taux d'erreur d'apprentissage/test est élevé (cf. le nuage de points)
  - ➙ Un nouveau point peut avoir une chance sur 2 d'être mal classé
  - ➙ Les méthodes 0-R ou 1-R sont des exemples (utilité en exploration).

- Parallèlement, l'**Overfitting** (cf. dans les Arbres de Décision) est souvent provoqué à cause d'un modèle trop complexe.

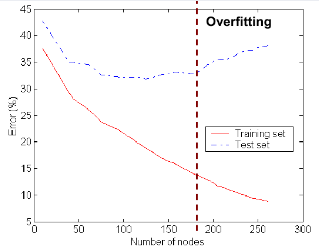


FIGURE 3: Illustration des erreurs d'un cas d'overfitting

Pbs. : Underfitting/Overfitting dans AD (suite)

- L'Under-fitting peut aussi être provoqué par un manque (peu) de données.

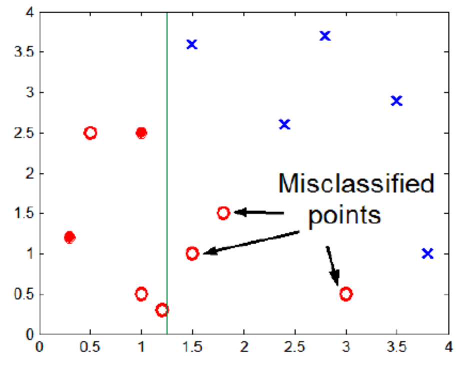


FIGURE 4: L'insuffisance des données provoque une classification erronée

☞ Ici, on peut aussi bien sur-apprendre si on sépare les rouges des bleus.

Pbs. : Underfitting/Overfitting dans AD (suite)

- *Overfitting* sur des données bruitées (bruits non détectés) :

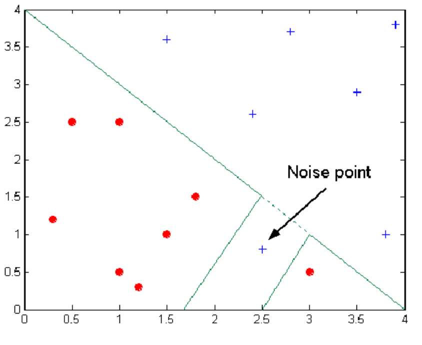
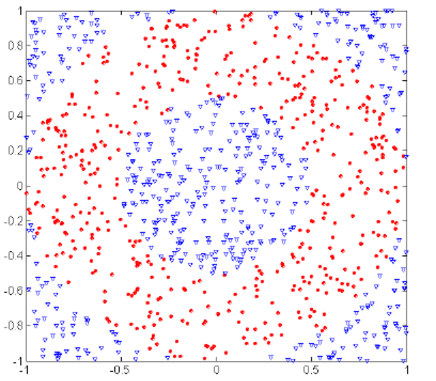


FIGURE 5: Les frontières de la décision faussées par la présence de bruits

Pbs. : Underfitting/Overfitting dans AD (suite)

Under ou Over fitting (axes x_1 et x_2) ? :



On a 500 cercles (rouges) et 500 triangles (bleus)

∘ critère pour les cercles rouges :

$$0.5 \leq \sqrt{(x_1^2 + x_2^2)} \leq 1$$

∘ critère pour les triangles :

$$\sqrt{(x_1^2 + x_2^2)} < 0.5$$

$$\text{ou} \quad \sqrt{(x_1^2 + x_2^2)} > 1$$

☞ Spécificité du nuage.

- Critères trop simplistes ! quid d'un nouveau point ?
  ➙ Ces critères distinguent-ils proprement un Cercle d'un Triangle ?

Apprentissage : un bilan intermédiaire

Rappel des méthodes étudiées :

- **0R et 1R**
  ➙ Aucun ou Un seul attribut discriminant, taux d'erreur élevé
- **Bayes naïve**
  ➙ Hypothèses, veto, Laplace, simple et bons résultats
- **Arbre de décision**
  ➙ Entropie (ID3), Ratio de Gain (C4.5), Gini (CART), ...
- **Cette partie : construction de règles**
  ➙ PRISM (règles de classification)
  ➙ A PRIORI (règles d'association)

Construction de règles de classification

Comment produire des règles de classification :

- On peut traduire un Arbre de Décision (ID3, C4.5, CART, etc.) en règles de classification
  - La conversion (correcte/complète) n'est pas toujours triviale,
  - Taux d'erreur = celui de l'AD
  - Élagage en fonction du taux d'erreur accepté / la complexité de l'AD.

- Une approche alternative : règles de **couverture** ("covering") :
  - **Principe** : pour chaque classe C , trouver directement les règles pour couvrir toutes les instances dans C (en évitant celles qui ne sont pas dans C).
  - A chaque étape, on identifie des règles qui "couvrent" un groupe d'instances.

../..

Construction de règles de classification (suite)

Exemple introductif : on traite chaque classe

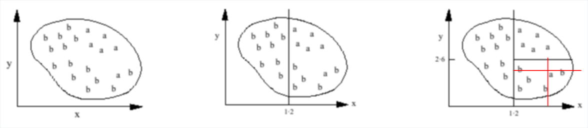


FIGURE 6: Génération directe de règles (covering)

- On peut directement générer des règles pour la classe 'a' (figs. de gche à dte) :

(if true then) classe = b.	(**0-R** : schéma de gauche, 'b' majoritaire)

if $x > 1.2$ then classe = a.	(**1-R** : droite au milieu, couvre beaucoup de 'b')

- Si on vise seulement les 'a' :

if $x > 1.2$ and $y > 2.6$ then classe = a.	(Fig. droite, tout sauf un 'a')
if $x > 1.4$ and $y < 2.4$ then classe = a.	(.. et pour le 'a' isolé mais couvre un 'b' !)

Construction de règles de classification (suite)

Règles pour les deux classes 'b' et 'a' :

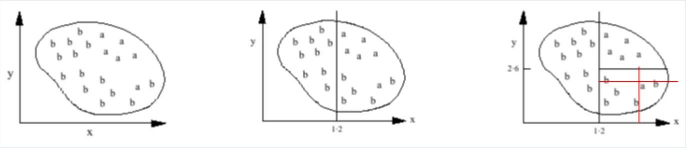


FIGURE 7: Génération directe de règles (covering) (Ibid)

Règles pour les 2 régions des 'b's :

if $x \leq 1.2$ then classe = b.	
if $x > 1.2$ and $y \leq 2.6$ then classe = b.	(couvre un 'a' tout à droite)

Pour la région des 'a', on avait eu :

if $x > 1.2$ and $y > 2.6$ then classe = a.	
if $x > 1.4$ and $y < 2.4$ then classe = a.	(couvre un 'b' tout à droite)

Mais il y a un 'b' couvert par les règles pour 'a' et vice versa !

➙ Solution : ajouter d'autres tests et règles pour arriver à 0 erreur ...
C'est le principe de **Covering**.

Un algorithme simple de couverture

- Principe (simple) des algorithmes de "couverture" :
  ➙ ajout de tests aux règles en construction en **maximisant la justesse**.

- Par comparaison : ID3 ajoute des tests sur l'arbre en construction, en **maximisant la pureté** (séparation des classes).

- Dans les deux cas : trouver un attribut pour la division.
  ➙ critères différents de choix d'attribut.

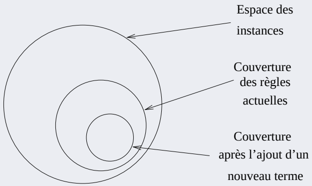


FIGURE 8: L'espace d'instances pendant l'application de l'algorithme de couverture

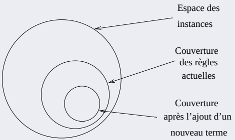
Un algorithme simple de couverture (suite)

- Un algorithme comme **ID3** (stratégie *Diviser pour régner*) choisit un **attribut qui maximise** le *gain d'information*, toutes classes confondues

- Un algorithme de "couverture" (*Séparer pour régner*) choisit **une paire attribut-valeur** pour **maximiser** la probabilité de la classification,
  ➙ une classe à la fois.

- Figure ci-dessous : l'ajout d'un nouveau terme restreint la couverture de la règle
  - L'idée : **inclure** autant que possible des instances de la classe souhaitée et **exclure** autant que possible d'instances d'autres classes.
  - Mesure : une nouvelle règle couvre un total de **t** instances dont **p** sont **positifs** et **t-p** dans d'autres classes

    ➙ Choisir un nouveau terme qui maximise le ratio **p/t**.
  - Fin de l'algorithme : le ratio $\mathbf{p/t = 1}$ ou plus aucune division possible



Exemple des lentilles

Un exemple : BD. de "lentilles de contact" :

Age (ophtalmo.)	Prescription (diagnostique)	Anomalie de réfraction	Effet Lacrymal	Type lentilles
Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	No	Reduced	None
Young	Myope	No	Normal	Soft
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	No	Reduced	None
Young	Hypermetrope	No	Normal	Soft
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	No	Reduced	None
Pre-presbyopic	Myope	No	Normal	Soft
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	No	Reduced	None
Pre-presbyopic	Hypermetrope	No	Normal	Soft
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	No	Reduced	None
Presbyopic	Myope	No	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	No	Reduced	None
Presbyopic	Hypermetrope	No	Normal	Soft
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

TABLE 1: Lentilles de contact

Extraction des règles de couverture

Règles pour le problème de "lentilles de contact" :

- On cherche une règle de la forme ***If*** $\boxed{?}$ ***Then*** ***recommandation=hard***.

Tests possibles pour le terme inconnu $\boxed{?}$ (9 cas) : p/t

(1)	age=young	2/8
(2)	age=pre-presbyte	1/8
(3)	age=presbyte	1/8
(4)	spectacle prescription =myope	3/12
(5)	spectacle prescription =hypermetrope	1/12
(6)	astigmatisme=no	0/12
(7)	astigmatisme=yes	4/12
(8)	tear production rate=reduced	0/12
(9)	tear production rate=normal	4/12

- Age=young avec 2/8 :

➜ 8 exemples impliqués dont 2 ont "recommandation = hard".

Extraction des règles de couverture (suite)

- Meilleur taux : 4/12 (choix aléatoire entre le 7e et le 9e)
- Donne la règle :

  if astigmatisme=yes then recommandation=hard. (4/12)

- Le sous ensemble des instances couvertes par cette règle :

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

TABLE 2: Les instance couvertes par la règle modifiée (Astigmatisme=Yes)

- Règle pas trop juste (seulement 4/12 cas couverts) : erreur=2/3

Extraction des règles de couverture (suite)

- On raffine en ajoutant un autre terme inconnu à compléter :

If astigmatisme=yes $\boxed{\textbf{\textit{and ?}}}$ ***Then recommandation=hard.***

- Pour le terme '?', on a 7 choix

age=young	2/4
age=pre-presbyte	1/4
age=presbyte	1/4
spectacle prescription =myope	3/6
spectacle prescription =hypermetrope	1/6
tear production rate=reduced	0/6
tear production rate=normal	4/6

- Le dernier l'emporte (4/6)

Extraction des règles de couverture (suite)

- La règle : $\boxed{\begin{array}{l} \textit{If astigmatisme} = \textit{yes and tear production rate} = \textit{normal} \\ \textit{Then recommandation} = \textit{hard.} \qquad 4/6 \end{array}}$

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Normal	None

TABLE 3: Les instance couvertes par la nouvelle règle

- On ne s'arrête pas (avec une erreur de 1/3) :
  - Il faut des règles exactes (mêmes si complexes)
  - Les possibilités pour le nouveau terme '?' dans :

If astigmatisme=yes and tear production =normal $\boxed{\textbf{\textit{and ?}}}$ ***Then recommandation=hard.***

../..

Extraction des règles de couverture (suite)

- Pour le terme '?' de la règle , on a 5 choix :

(1)	age=young	2/2
(2)	age=pre-presbyte	1/2
(3)	age=presbyte	1/2
(4)	spectacle prescription =myope	3/3
(5)	spectacle prescription =hypermetrope	1/3

Choix entre (1) et (4) → on prend 3/3 → couverture plus large

- La règle finale à 100% juste (mais ne couvre pas tous les *hard*) :

if astigmatisme = yes and tear production rate = normal and spectacle prescription = myope then recommandation = hard 3/3

- Elle ne couvre que 3 cas sur 4 du total des "recommandation=hard"
  ➙ Faut couvrir tous les cas.

☞ L'ajout d'un autre test (il ne reste que *Age*) à cette règle n'apporte rien,
  ➙ On recommence la "création" d'une nouvelle règle qui conclura sur **hard**. ../..

Extraction des règles de couverture (suite)

Le schéma (complémentaire) recherché : $\boxed{\textit{if ? then recommandation=hard.}}$

- En suivant le même processus (sur la BD réduite) :

  1) ➙ *age=young* est le meilleur choix (8 cas dont 2 'hard')

  ➙ Une instance des 2 cas 'hard' a été couverte par la 1e règle

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	No	Reduced	None
Young	Myope	No	Normal	Soft
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	No	Reduced	None
Young	Hypermetrope	No	Normal	Soft
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard

FIGURE 9: Données lentilles pour age=young (2 "hard" dont un déjà couvert)

➙ Règle de la forme *if age=young &* $\boxed{?}$ *then recommandation = hard*

2) → Le meilleur choix $\boxed{?}$: *astigmatisme=yes* pour 1/3 (ballottage)

3) → Puis *production rate=normal* (couv. 1/1) pour complétér $\boxed{?}$ ../..

Extraction des règles de couverture (suite)

La règle obtenue :

if age=young and astigmatisme=yes and production rate=normal then recommandation=hard.

- Couvre 3 instances de l'ensemble d'origine dont 2 couvertes par la 1^e règle
  ➙ pas de problème car la classe est la même.

Suite : On recommence pour les 2 autres classes :
➙ Idem pour "recommandation=soft" puis "recommandation=none"

- **On peut aussi prendre** une règle par défaut pour "Recomm.=none".

Petit Bilan :

- **La méthode exposée = méthode PRISM**
- Mesure la justesse d'une règle par le ratio **p/t** (ex. correctes/impliqués).
- Toute règle doit être "parfaite" (0 erreur) :
  ➙ ajout de tous les tests (et règles) nécessaires.

PRISM : Algorithme de principe

Pour créer les règles $LHS \Rightarrow RHS$:

. Pour toute classe C
- Initialiser A $\leftarrow$ ensemble des instances
- Tant que A contient des instances de la classe C
  + Créer la règle R avec un LHS=vide qui prédit C (R : If LHS Then C)
  + Répéter jusqu'à R parfaite (ou plus aucun attribut à utiliser) :
    * Pour tout attribut A non utilisé dans R et pour toute valeur v :
      . Ajouter la condition $A = v$ à LHS de R telle que $p \neq 0$
        et p/t maximum (si conflit, choisir le plus grand p)
  + Supprimer de A les instances couvertes par R

FIGURE 10: Pseudo algorithme basique pour l'apprentissage de règles

N.B. : à chaque itération principale, A est initialisé à toutes les instances :

- Les ensembles couvertes par 2 règles ne sont pas disjointes.
- Si on permute les 2 premières lignes de cet algorithme, on introduit un ordre dans les règles.

Règles vs. Listes de Décision

- Dans PRISM, la boucle principale traite toutes les classes :
  - Pas d'ordre de dépendance (les classes traitées une par une)
  - Meilleure modularité par ces (règles de) Connaissances (appelées *pépites* ou *nuggets*).

- Dans PRISM, si l'on enlève l'initialisation de A de la boucle principale :
  - L'algorithme génère une **Liste de Décision** pour UNE classe (**ordre**)
  - Donne une version légèrement modifiée de PRISM (v. chaps svts).

- **Problèmes** d'une liste de décision : recouvrement par des règles (overlapping)

  → *règle par défaut* nécessaire, gestion de conflit, ...

  → Privilégier les règles qui couvrent un maximum d'instances.

Règles vs. Arbre

Les règles sont plus efficaces vs. les Arbres de Décision (ADs) :

➙ Les ADs souffrant du problème de "duplication de sous arbres".

Exemple cas bi-classes :

- 4 attributs $(x, y, z, w) \in \{1, 2, 3\}$
- deux classes a et b.
- L'arbre (complexe) construit :

➙ triangle gris à dte. = le sous-arbre à 3 niveaux gris à gauche en bas.

➙ Une manière pénible d'exprimer un concept pourtant simple :

Si x=1 et y=1 Alors classe=a ;
Si x=1 et w=1 Alors classe=a ;
Sinon classe=b ;

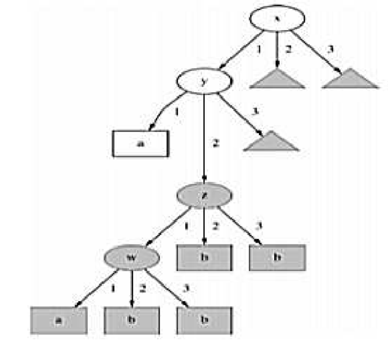


TABLE 4: L'arbre avec des sous-arbres dupliqués

- A l'inverse, on peut trouver un Arbre de Décision pour un ensemble donné de règles de classification.

Règles vs. Arbre (suite)

Comparaison des stratégies obtenant Règles vs. Arbres :

- *Règles de couverture*
  → on considère une classe à la fois,

- *Arbre de décision (AD)*
  → on considère toutes les classes en même temps
  (pour maximiser la "pureté" des partitions)

- Situations similaires pour une stratégie descendante *Divide-and-Conquer* (cf. A.D.) et pour un algorithme de *couverture* (Covering) :
  - Scinder d'abord l'ensemble de données en fonction d'un attribut
  - Scinder éventuellement sur un 2nd attribut (x et y de l'exemple)

  ➙ Des similarités (entre Arbres et Règles) peuvent apparaître ...

Règles vs. Arbre (suite)

Exemple de similarité entre Règles & Arbres :

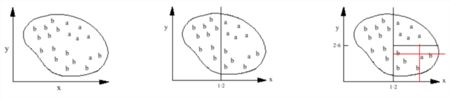
if $x \leq 1.2$ *then classe* $= b$.

if $x > 1.2$ *and* $y \leq 2.6$ *then classe* $= b$.

then classe $= b$.

if $x > 1.2$ *and* $y > 2.6$ *then classe* $= a$

then classe $= a$



AD des région 'a' et 'b' :

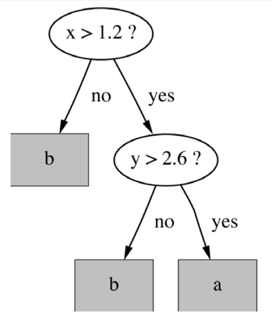


TABLE 5: l'Arbre de Décision associé à l'exemple précédent (2 règles pour 'b' et une pour 'a', erreur=1)

Addendum : La méthode Ripper

- Méthode industrielle et performante pour obtenir des règles de classif.

- Génération de règles de classification par élagage pour réduction incrémentale d'erreurs (*Incremental reduced-error Pruning*).

- Le principe de l'algorithme :

```
- Initialiser A ← toutes les instances ;
- Scinder A en deux ensembles Croissance et Élagage avec un ratio 2 : 1
- Pour chaque classe C pour laquelle Croissance et Élagage contiennent une instance
      - Utiliser un algorithme basique de couverture pour créer la meilleure règle parfaite pour C
      - Calculer l'intérêt W(R) pour la règle dans Élagage et W(R−) pour la même règle privée de sa toute
            dernière condition (prémisse) ;
      - Tant que W(R−) > W(R) :
            supprimer la condition finale de la règle et répéter l'étape précédente
      - Pour les règles générées, retenir celle avec la plus grande valeur de W(R)
      - Enlever les instances de A couvertes par cette règle
```

- Dans l'algorithme ci-dessus, les classes sont examinées une par une dans l'ordre de leur taille (nombre d'instances de la classe).

- Un premier ensemble de règles est construit pour la classe en utilisant un algorithme comme ci-dessus.

Addendum : La méthode Ripper (suite)

• Une condition d'arrêt supplémentaire dépendante de la longueur de la description (*DL : Description Length*) est introduite.

• Le calcul du DL est assez complexe prenant en compte le nombre de bits nécessaires pour communiquer un ensemble d'instances et un ensemble de règles avec k conditions (+ d'autres informations).

• Une fois produites les règles pour une classe, chaque règle est reconsidérée et deux variantes seront produites en utilisant encore un algorithme comme ci-dessus mais cette fois, les instances couvertes par les autres règles de la classe seront enlevées de l'ensemble *Élagage*.

• Un taux de succès calculé sur les instances restantes est utilisé comme critère d'élagage.

• Si l'une des variantes donne un meilleur DL, celle-ci remplace la règle.
• On passe ensuite à la construction des règles pour les instances non couvertes.
• Un test final a lieu pour s'assurer que chaque règle contribue à la réduction de DL avant de passer à la classe suivante.

Addendum : La méthode Ripper (suite)

Les mesures utilisées par Ripper :

$G = Pr[log(p/t) - log(P/T)]$ *le Gain d'information*

$W = \dfrac{p+1}{t+k}$ *l'intérêt* de la règle

$A = \dfrac{p+n'}{T}$ *La justesse* de la règle

Avec :

- k le nombre de classes , 2 pour bi-classes
- p le nombre d'instances positivement couvertes par la règles (la valeur TP)
- n le nombre d'instances négativement couvertes par la règles (la valeur FN)
- $t = p + n$ le nombre total d'instances couvertes par la règles
- P le nombre d'instances positives de la classe
- N le nombre d'instances négatives de la classe
- $T = P + N$ le nombre total d'instances de la classe

Addendum : La méthode Ripper (suite)

Algorithme plus détaillé et optimisé de Ripper :

Initialiser $A \leftarrow$ toutes les instances ;
Pour chaque classe C, de la plus petite (en nbr. d'instances) à la + grande :

Phase Construction :
- Scinder A en deux ensembles *Croissance* et *Élagage* avec un ratio 2 : 1
- Répéter jusqu'à l'une des conditions suivantes :
  - (a) il n'y a plus d'exemple de la classe C à couvrir OU
  - (b) Longueur de Description [a] de l'ensemble $\{Regles \cup Exemples\}$ est plus longue de 64 bits que la plus petite DL atteinte jusque là OU
  - (c) Le taux d'erreur dépasse 50%

  Phase Croissance :
  - Faire croître une règle jusque 100% de *justesse*, en y ajoutant des conditions qui viennent de toutes combinaisons de (attribut × valeur) avec le plus grand *gain* d'information G

  Phase Élagage : Supprimer des conditions dans l'ordre dernière-à-première tant que la valeur W (coût, intérêt) de la règle augmente

Phase Optimisation :

Génération des Variantes :
Pour toute règle R pour la classe C,
- Scinder de nouveau A en deux ensembles *Croissance* et *Élagage*
- Supprimer toute instance de *Élagage* qui est couverte par une autre règle pour C
- Utiliser les ensembles *Croissance* et *Élagage* pour générer et élaguer deux règles R_1, R_2 concurrentes à partir des données nouvellement scindées :
  - R_1 est une nouvelle règle, construite *ex nihilo*
  - R_2 est générée en ajoutant petit à petit des antécédents à R
- Élaguer à l'aide du métrique A à la place de W (voir ci-dessus) sur cet ens. réduit de données.

Choix des représentants : Remplacer la règle R par celle d'entre R, R_1, R_2 avec un DL minimum ../..

a. DL : *Description Length*, v. chapitre 5 du cours

Addendum : La méthode Ripper (suite)

- Suite de l'algorithme :

> **Ramassage :**
> B'il reste des instances de la classe C non encore couvertes, aller à *Phase Construction* pour générer des règles pour ces instances ;
>
> **Nettoyage :**
> - Calculer DL pour toutes l'ensemble de toutes les règles ET pour ce même ensemble privé de chacune de ses règles à tout de rôle ; supprimer toute règle qui augmente le DL.
> - Supprimer les instances couvertes les règles retenues (fraîchement générées).

- On utilisera cette méthode dans les BEs.

Règles d'association

- Généralisent les règles de classification
- Au format *Gauche* $\Rightarrow$ *Droit*
  (LHS $\Rightarrow$ RHS qui se lit : *si LHS alors RHS*) :
  - Ces règles porteront sur des sous-ensembles d'attributs
  - Toute expression d'attributs en partie droite (RHS) est possible
  - Une règle d'association peut prédire les valeurs de plusieurs attributs.

Exemples : *Si Tomates & Fromage Alors Pain & Vin.*
Si Viande & Pâtes Alors ~ Poisson & Tomates.

Critères basiques de sélection :

- Notion de **support** et **Fréquence** pour les ensembles d'attributs ;
- Notion de **confiance** pour les règles intéressantes.

Critères d'évaluation : plusieurs mesures.

Règles d'association (suite)

- **Terminologie :**
  - **item** : une paire attribut-valeur
  - **itemset** : tous les items figurant dans une règle (LHS $\Rightarrow$ RHS)
  - *itemsets fréquents* : itemset répété $\geq$ certain seuil :
    ➙ Le seuil des **supports** (S) = minimum d'occurrences d'un itemset,
    ➙ Ou des **fréquences** (f) $= \dfrac{S}{|D|}$ $|.|$=taille d'un ensemble.
- **But** : Trouver les *itemsets fréquents* : ceux dont *support* $\geq S$
  ➙ Pour générer les règles qui s'appliquent aux plus grands nombres d'instances.
- Terminologie venant de l'analyse du panier/caddie (*market basket analyse*) :
  ➙ Les items sont des articles dans le caddie
  ➙ Le manager du supermarché cherche des associations dans les achats.
- Voir plus loin pour d'autres définitions.

Génération des Itemsets fréquents

Modus operandi (déjà optimisé) via un exemple :

- Soit une BD. de 5 transitions
- Soit le seuil de supports $S = 2$ (ou de fréquences $f = \frac{2}{5} = 40\%$)
- On constitue l'ensemble des candidats de taille 1 (*1-itemsets*) (C_1)

BD trans. ➞ **Ens.** C_1

Id Trans.	Les items
50	4, 6, 7
100	4, 5
120	2, 3, 5
150	3, 4, 5
200	1, 2, 3, 5

Itemset	Supp.
{1}	1
{2}	2
{3}	3
{4}	3
{5}	4
{6}	1
{7}	1

- On ne retient pas les rouges pour la suite.

Génération des Itemsets fréquents (suite)

Exemple (suite) : avec support $S = 2$

- On constitue l'ensemble des **1-frequent-itemsets** (L_1) depuis C_1

BD trans. → **Ens.** C_1 → **Ens.** L_1

Id Trans.	Les items
50	4, 6, 7
100	4, 5
120	2, 3, 5
150	3, 4, 5
200	1, 2, 3, 5

Itemset	Supp.
{1}	1
{2}	2
{3}	3
{4}	3
{5}	4
{6}	1
{7}	1

Itemset	Supp.
{2}	2
{3}	3
{4}	3
{5}	4

- $L1$ ne retient que les *1-itemsets* qui satisfont la contrainte du support S.

Génération des Itemsets fréquents (suite)

Exemple (suite) : construction de l'ensemble C_2

BD trans.

Id Trans.	Les items
50	4, 6, 7
100	4, 5
120	2, 3, 5
150	3, 4, 5
200	1, 2, 3, 5

Ens. C_1

Itemset	Supp.
{1}	1
{2}	2
{3}	3
{4}	3
{5}	4
{6}	1
{7}	1

Ens. L_1

Itemset	Supp.
{2}	2
{3}	3
{4}	3
{5}	4

Ens. C_2

Itemset	Supp.
{2, 3}	2
{2, 4}	0
{2, 5}	2
{3, 4}	1
{3, 5}	3
{4, 5}	2

- $L1$ ne retient que les *1-itemsets* de $C1$ dont le nombre d'occ. $\geq S$.
  ➙ sans cette "élimination", on combinerait TOUTES paires d'itemset de $C1$ puis on éliminerait les non-fréquents dans $C2$! (voir plus loin la complexité).
- C_2 construit en croisant L_1 <u>avec lui même</u> suivie d'une vérification dans la BD.

Une 1ère optimisation : pour construire $C_{k>2}$, on utilise L_{k-1}

- *A priori*, un comptage dans la BD sera nécessaire (voir plus loin).

☞ Pour un k-itemsets ($k > 1$), on exige la présence simultanée et fréquente des k items.

Génération des Itemsets fréquents (suite)

Exemple (suite) : obtention de l'ensemble L_2 par filtrage de C_2

BD.

Id Trans.	Les items
50	4, 6, 7
100	4, 5
120	2, 3, 5
150	3, 4, 5
200	1, 2, 3, 5

C_1

Itemset	Supp.
{1}	1
{2}	2
{3}	3
{4}	3
{5}	4
{6}	1
{7}	1

L_1

Itemset	Supp.
{2}	2
{3}	3
{4}	3
{5}	4

C_2

Itemset	Supp.
{2, 3}	2
{2, 4}	0
{2, 5}	2
{3, 4}	1
{3, 5}	3
{4, 5}	2

L_2

Itemset	Supp.
{2, 3}	2
{2, 5}	2
{3, 5}	3
{4, 5}	2

- ➙ Les itemsets rouges sont éliminés.
- ➙ Le filtrage de $C2$ se fait par un comptage dans la BD.

Génération des Itemsets fréquents (suite)

Exemple (suite) : construction de l'ensemble C_3.

BD.

Id Trans.	Les items
50	4, 6, 7
100	4, 5
120	2, 3, 5
150	3, 4, 5
200	1, 2, 3, 5

C_1

Itemset	Supp.
{1}	1
{2}	2
{3}	3
{4}	3
{5}	4
{6}	1
{7}	1

L_1

Itemset	Supp.
{2}	2
{3}	3
{4}	3
{5}	4

C_2

Itemset	Supp.
{2, 3}	2
{2, 4}	0
{2, 5}	2
{3, 4}	1
{3, 5}	3
{4, 5}	2

L_2

Itemset	Supp.
{2, 3}	2
{2, 5}	2
{3, 5}	3
{4, 5}	2

C_3

Itemset	Supp.
{2, 3, 5}	2
{2, 4, 5}	0
{3, 4, 5}	1

- Rappel : C_3 est construit à l'aide de L_2 seulement.

☞ Dans C_3, l'itemset {3,4,5} est "envisagé" (par la présence de {4,5} et de {3,5}) avant d'être rejeté (pour support insuffisant).

➙ Idem pour {2,4,5}

Génération des Itemsets fréquents (suite)

Exemple (suite) : obtention de l'ensemble L_3 à partir de C_3

BD.

Id Trans.	Les items
50	4, 6, 7
100	4, 5
120	2, 3, 5
150	3, 4, 5
200	1, 2, 3, 5

C_1

Itemset	Supp.
{1}	1
{2}	2
{3}	3
{4}	3
{5}	4
{6}	1
{7}	1

L_1

Itemset	Supp.
{2}	2
{3}	3
{4}	3
{5}	4

C_2

Itemset	Supp.
{2, 3}	2
{2, 4}	0
{2, 5}	2
{3, 4}	1
{3, 5}	3
{4, 5}	2

L_2

Itemset	Supp.
{2, 3}	2
{2, 5}	2
{3, 5}	3
{4, 5}	2

C_3

Itemset	Supp.
{2, 3, 5}	2
{2, 4, 5}	0
{3, 4, 5}	1

L_3

Itemset	Supp.
{2, 3, 5}	2

➙ Les calculs des itemsets fréquents s'arrête ici.

Représentation par treillis

Optimisation de représentation en mémoire :

- Soit un ensemble d'attributs $I = \{A, B, C, D, A, F, G\}$.
- Une représentation de tous les itemsets possibles à partir de I en page svte.
- A noter que ce treillis n'a pas forcément besoin de contenir la représentation effective (en mémoire) de chaque itemset : on reconstitue les itemsets par calcul.
    - On utilise les numéros des lignes / colonnes (1..7) ainsi que le label de chaque case (une lettre $\in I$ en couleur magenta).
    - Le contenu (les itemsets) de chaque case est déduit à partir de son label et les itemsets de ses cases parentes (cf. ceux des cases (5,2), (5,3) ou (6,2)).

- **Remplissage** : un passage dans la BD :

A la lecture de chaque instance (e.g. $ABDE$), on incrémente le compte d'occurrence de ABDE et de tous ses sous-itemsets, jusqu'aux singletons.

➙ Une fois la BD lue, on a par cette matrice une représentation compacte de cette BD et on peut commencer par éliminer les non-fréquents

➙ ... Vers Clos et MAXI.

➙ Chaque itemset donne accès à ces sous/super itemsets

A {A}	B {B}	C {C}	D {D}	A {A}	F {F}	G {G}
B {A, B}	C {A, C} {B,C}	D {A, D} {B,D} {C,D}	A {A,A} {B,A} {C,A} {D,A}	F {A,F} {B,F} {C,F} {D,F} {A,F}	G {A,G} {B,G} {C,G} {D,G} {A,G} {F,G}	∅
C {A,B,C}	D {A,B,D} {A,C,D} {B,C,D}	A {A,B,A} {A,C,A} {B,C,A} {A,D,A} {B,D,A} {C,D,A}	F {A,B,F} {A,C,F} {B,C,F} {A,D,F} {B,D,F} {C,D,F} {A,A,F} {B,A,F} {C,A,F} {D,A,F}	G {A,B,G} {A,C,G} {B,C,G} {A,D,G} {B,D,G} {C,D,G} {A,A,G} {B,A,G} {C,A,G} {D,A,G} {A,F,G} {B,F,G} {C,F,G} {D,F,G} {A,F,G}	∅	∅
D {A,B, C,D}	A {A,B,C,A} {A,B,D,A} {A,C,D,A} {B,C,D,A}	F {A,B,C,F} {A,B,D,F} {A,C,D,F} {B,C,D,F} {A,B,A,F} {A,C,A,F} {B,C,A,F} {A,D,A,F} {B,D,A,F} {C,D,A,F}	G {A,B,C,G} {A,B,D,G} {A,C,D,G} {B,C,D,G} {A,B,A,G} {A,C,A,G} {B,C,A,G} {A,D,A,G} {B,D,A,G} {C,D,A,G} {A,B,F,G} {A,C,F,G} {B,C,F,G} {A,D,F,G} {B,D,F,G} {C,D,F,G} {A,A,F,G} {B,A,F,G} {C,A,F,G} {D,A,F,G}	∅	∅	∅

Suite du tableau ../..

A {A}	B {B}	C {C}	D {D}	A {A}	F {F}	G {G}
...	...	...	...	...	...	...
A {A,B, C,D,A}	F AJOUTER "F" AUX ITEMSETS DE {(4,1)} ∪ {(4,2)}	G AJOUTER "G" AUX ITEMSETS DE {(4,1)} ∪ {(4,2)} ∪ {(4,3)}	∅	∅	∅	∅
F {A,B,C, D,A,F}	G AJOUTER "F" À {(5,1)} ∪ {(5,2)}	∅	∅	∅	∅	∅
G {A,B,C, D,A,F,G}	∅	∅	∅	∅	∅	∅

- On constate que la reconstitution du contenu de chaque case peut se faire par calculs (étant donne ligne/colonne) et leur rang suivant un ordre lexicographique.

- On peut également utiliser le *hashage* si on décide de représenter les itemsets.

Itemsets pour l'exemple Météo

Rappel de la table de l'exemple "Météo" :

Outlook	**Temp.**	**Humidity**	**Windy**	**Play**
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Table 6: BD exemple "Météo"

Itemsets pour l'exemple Météo (suite)

Itemsets (Météo) pour S=2 :

	UN-Itemsets	DEUX-Itemsets	TROIS-Itemsets	QUATRE-Itemsets
1	Outlook=Sunny (5)	Outlook=Sunny Temperature = Mild (2)	Outlook=Sunny Temperature = Hot Humidity = High (2)	Outlook=Sunny Temperature = Hot Humidity = High Play = no (2)
2	Outlook=Overcast (4)	Outlook=Sunny Temperature = Hot (2)	Outlook=Sunny Temperature = Hot Play=no (2)	Outlook=Sunny Windy = False Humidity = High Play = no (2)
4	Temperature = Cool (4)	Outlook=Sunny Humidity = High (3)	Outlook=Sunny Humidity = High Windy=false (2)	Outlook=Rainy Temperature = Mild Windy=false (2) Play = yes (2)
7	Humidity=Normale (7)	Outlook=Sunny Play = yes (2)	Outlook=Overcast Temperature = Hot Play = no (2)	
12	Play=no (5)	Outlook=Overcast Windy=true (2)	Outlook=Overcast Windy = false Play = yes (2)	
...		...	...	
47		Windy=false Play=no (2)		

TABLE 7: Les Item-sets pour l'exemple "Météo" avec un *support* ≥ 2

Génération des itemsets (Météo)

- Hypothèse : on veut des règles d'association avec un **support $\geq$ 2**.
  - Au départ, toutes combinaisons attribut-valeur (UN-itemset) avec support ≥ 2
  - Ensuite, on combine chacun des 1-itemsets avec un attribut **différent**
    ➙ Jamais 2 fois le même attribut avec des valeurs différentes (e.g. *outlook=sunny & overcast*) dans un k-itemset
    ➙ N'arrive jamais dans une vraie instance.

- <u>Respect du support minimum</u> (BD Météo) :
  - 12 Un-itemsets : ne nous intéressent pas !
  - **47** Deux-itemsets (*2-fréquent itemsets*) générés
  - **39** Trois-itemsets,
  - **6** Quatre-itemsets (support ≥ 2)

N.B. : Pas de *Cinq-frequent* itemset dans Météo
$\rightarrow$ ne peut correspondre qu'à une instance répétée
(support = 2 veut dire : le 5-itemset est forcément un doublon !).

Support, Fréquence et Confiance

Rappels et compléments des seuils :

- Base de données D, Itemset I,
- Support d'un itemset I : ensemble des transactions de D qui contiennent I.
- Fréquence d'un itemset I : $Freq(I, D) = \frac{Support(I,D)}{|D|}$
- Confiance d'une règle d'association (pour $L_l = k$-fréquent itemsets)
- Supposons que l'ensemble L_k de k-itemsets fréquents soit construit (cf. ci-dessus).

  Pour un itemset $I \in L_k$ et un de ses sous-itemset $X \subset I, X \neq \emptyset$:
  - Pour I, on envisage toutes règles $(LHS \Rightarrow RHS)$ de la forme $X \Rightarrow I \setminus X$
  - $Conf(X \Rightarrow I \setminus X) = \frac{Freq(I,D)}{Freq(X,D)}$
- Si $Conf_{(LHS \Rightarrow RHS)} \geq min\_conf$, la règle est retenue.
- On obtiendra des règles de la forme

$$\boxed{\{\text{Outlook=Sunny, Temp=Hot}\} \rightarrow \{\text{Play=Yes}\} \text{ avec } \textit{Conf=80\%}}$$

Règles d'association pour la BD "météo"

- On obtient de chaque itemset fréquent un ensemble (éventuellement vide de) règles avec la confiance min spécifiée.

- Exemple : le Trois-itemset

{ Humidity=normal, windy=false, play=yes } (4)

donne 7 ($2^N - 1$) règles potentielles :

if humidité=normale and windy=false then play=yes	(4/4)
if humidité=normale and play=yes then windy=false	(4/6)
if windy=false and play=yes then humidité=normale	(4/6)
if humidité=normale then windy=false and play=yes	(4/7)
if windy=false then humidité=normale and play=yes	(4/8)
if play=yes then humidité=normale and windy=false	(4/9)
if True then humidité=normale and windy=false and play=yes	l'itemset : (4/14)

Règles d'association pour la BD "météo" (suite)

Soit le 3-itemset précédent {*Humidity=normal, windy=false, play=yes*} *(support = 4)*

- La **confiance** de chaque règle ($LHS \Rightarrow RHS$) obtenue est une fraction **p/t** :
  - ➙ **p** = nombre d'occurrences du 3-itemset (support de ($LHS \cup RHS$))
  - ➙ **t** = nombre d'occurrences du prémisse de la règle (support de LHS)
- Par exemple, pour la règle tirée de cet 3-itemset (avec $\frac{p}{t} = \frac{4}{6}$) :
  if humidité=normale and play=yes then windy=false *(4/6)*
  - ➙ 4 instances vérifient le 3-itemset ci-dessus et 6 instances qui vérifient *humidité=normale* <u>*and*</u> *play=yes*.

N.B. : pour la dernière règle :

if True then humidité=normale and windy=false and play=yes *(4/14)*

- ➙ l'antécédent (True) est vrai pour les 14 exemples (et la conclusion 4 fois).
- ➙ C'est l'itemset lui-même !
- ➙ On n'exploite pas ce type de règles !

Règles d'association pour la BD "météo" (suite)

- Si **confiance=100%**, la 1ère parmi les 7 possibles est acceptée :

  ➜ *if humidité=normale and windy=false then play=yes* *(4/4)*

- Un autre exemple : le 4-itemset (de la table 7 précédente) :
  {*Temp=cool, Humidity=normal, Windy=false, Play=yes*}. *(2)*

  ➜ Si confiance=100%, la recherche des sous-itemsets fréquents de ce 4-itemset donne 3 **itemsets fréquents** utilisés comme antécédents d'une règle :

temp=cool, windy=false	*(2)*
temp=cool, humidity=normal,windy=false	*(2)*
temp=cool, windy=false, play=yes.	*(2)*

  ➜ Qui conduisent à 3 règles pour lesquelles la **confiance=100%** :

temp=cool, windy=false $\Longrightarrow$ *humidity=normal, Play=yes*	*100%*
temp=cool, humidity=normal, windy=false $\Longrightarrow$ *Play=yes*	*100%*
temp=cool, windy=false, play=yes $\Longrightarrow$ *humidity=normal*	*100%*

  ➜ support(4-itemset)=2, support pour les 3 trois-itemsets=2 ➜ conf=100%

Règles d'association pour la BD "météo" (suite)

Bilan des règles d'association de l'exemple météo :
(BD avec 14 instances)

	Prémisse **(LHS)**	⇒ Conclusion **(RHS)**	Support	Confiance
1	Humidity=normal & Windy=false	⇒ Play=yes	4	100%
2	Temperature=Cool	⇒ Humidity=normal	4	100%
3	Outlook=Overcast	⇒ Play=yes	4	100%
4	Temperature=Cold & Play=yes	⇒ Humidity=normal	3	100%
5	Outlook=Rainy & Windy=false	⇒ Play=yes	3	100%
...	...	⇒ ...	...	100%
58	Outlook=Sunny & Temperature=Hot	⇒ Humidity=High	2	100%

➙ **Au total** :

3 règles avec un support=4 ;

5 avec support=3 et

50 avec support=2

Toutes avec une confiance de 100%.

Optimisation : Génération efficace des itemsets

- Comment trouver tous les itemsets **fréquents** (support / fréquence) ?
  - Trouver les 1-itemsets fréquents (& respecter le support) → facile !
  - **Idée :** utiliser les $k - frequents$ pour générer $(k+1) - frequents$ ...

- La propriétés **d'anti-monotonicité** permet de réduire le nombre d'itemsets recherchés pour trouver des fréquents :

Si un ensemble ne satisfait pas une propriété, alors aucun de ses sur-ensembles (super-sets) ne pourra satisfaire la même propriété

Ex : si $\{A, B\}$ n'est pas fréquent, ni $\{A, B, C\}$ ni $\{A, B, D\}$ ne le seront.

- Deux autres propriétés également exploitées (cf. les exs. précédents) :

Propriété 1- Si, dans L_k , on a deux itemsets de taille k :

$$r = \{r_1, r_2, ..., r_{k-1}, r_k\} \text{ et } s = \{s_1, s_2, ..., s_{k-1}, s_k\}$$

$$\text{tels que } r_i = s_i \quad \forall i \in 1, ..., k-1,$$

Alors "envisager" l'itemset candidat $\{r_1, r_2, ..., r_{k-1}, r_k, s_k\}$ dans C_{k+1}

➙ Exemple : si $\{X, Y\}, \{X, Z\} \in L_2$ alors "envisager" $\{X, Y, Z\} \in C_3$

Optimisation : Génération efficace des itemsets (suite)

Rappel : **Propriété 1-** Si, dans L_k , on a deux itemsets de taille k :
$r = \{r_1, r_2, ..., r_{k-1}, r_k\}$ et $s = \{s_1, s_2, ..., s_{k-1}, s_k\}$
tels que $r_i = s_i \quad \forall i \in 1, ..., k-1$,
Alors on ajoute à C_{k+1} l'itemset $\{r_1, r_2, ..., r_{k-1}, r_k, s_k\}$

- Cette propriété permet la construction efficace des k-itemsets
  ➙ candidats pouvant être retenus après comptage dans la BD.

- La 2e propriété permet l'élagage des candidats :
  ➙ Si **{A, B}** est un itemset fréquent alors **(A)** et **(B)** doivent être fréquents

Propriété 2- Si X est k-fréquent itemset, tous les (k-1)-itemsets (sous ensembles de X) sont aussi fréquents

Exemple (suite) : on a envisagé $I = \{X, Y, Z\} \in C_3$;
- La propriété-1 (ci-dessus) permet d'envisager I car $\{X, Y\}, \{X, Z\} \in L_2$
- On ne retient I dans L_3 que si $\{Y, Z\} \in L_2$ (n'évite pas de recompter dans la BD.)

- Le calcul du support d'un k-itemset candidat nécessite le comptage dans la BD.
  ☞ Savoir que $I_1, I_2 \in L_2$ sont fréquents ne veut dire que $I_1 \cup I_2$ est fréquent.

Optimisation : Génération efficace des itemsets (suite)

Génération des itemsets candidats : un exemple

- Soient cinq 3-itemsets (A B C), (A B D), (A C D), (A C A) et (B C D) fréquents avec par exemple A : "outlook=sunny"...

- Dans un ordre lexicographique :
  ➙ considérer les couples de triplet avec les 2 premiers membres égaux :

  → Les candidats 4-itemsets $\in C_4$:
  - (A B C D) $\Longrightarrow$ OK grâce à (B C D) : accepte le candidat (A B C D)
  - Mais pas (A C D A) : absence de (C D A) et (A D A)

☞ L'union de (A B C) et (A B D) donne (A B C D) un candidat dans C_4 d'autant que ses autres 3-itemsets (A C D) et (B C D) sont fréquents.

➙ Pour savoir si $(ABCD) \in L_4$, il faut compter dans la BD. !

Pour obtenir les supports : comptage final dans l'ensemble d'apprentissage

☞ Pourquoi faut-il recompter pour savoir si (A B C D) est fréquent ? ../..

Optimisation : Génération efficace des itemsets (suite)

Un exemple : soit la BD

A	B	C
A	B	-
A	B	-
A	B	C
A	-	C
-	B	C
-	B	C

- Soit S=3

Le support est également appelé **coverage** (couverture)

- On a *support*($\{A, B\}$)=*4*, *support*($\{A, C\}$)=*3*, *support*($\{B, C\}$)=*4*.

➙ {A, B}, {A, C}, {B, C} sont fréquents mais {A, B, C} ne l'est pas.

- {A, B} et {A, C} seuls (dans L_2) permettent d'envisager {A, B, C} (dans C_3) mais on ne le retient pas dans L_3.

☞ **Rappel** : par contre, si {A, B, C} était fréquent, ses trois 2-itemsets l'auraient été !

Optimisation : Génération efficace des itemsets (suite)

Efficacité du traitement : utilisation d'une table de hachage

- Efficacité : après le comptage dans la BD d'un itemset, le résultat est stocké dans une **table de hachage**.
- Soit les (k-1)-itemsets stockés dans une table de hachage
- De l'exemple plus haut, on envisage chacun des 4-itemsets de l'ensemble et on vérifie que les 3-itemsets associés sont dans la table de hachage.

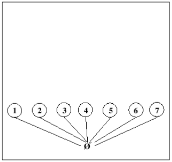
  Avec les fréquents : (A B C), (A B D), (A C D), (A C A) et (B C D)

  ➙ Les 4-itemset *(A B C D)* et *(A C D A)* sont candidats (dans C_4).
  ➙ Si tous les 3-itemsets sont ok (dans la hash-table) alors les 4-itemsets (ici $\{A\ B\ C\ D\}$) sont envisagés.

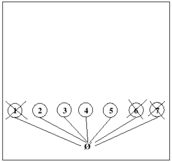
- La vérification du support (couverture) se fait uniquement dans l'ensemble d'apprentissage utilisant une table de hachage
- La présentation par Treillis pour mieux comprendre ce processus ../..

Itemsets vus par des Treillis

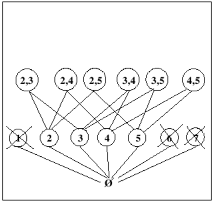
La présentation par Treillis :

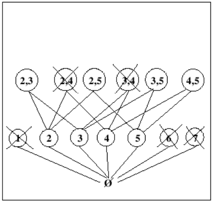


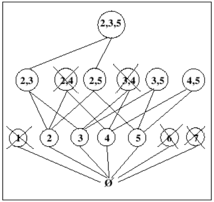
$C_1 = \{1, 2, 3, 4, 5, 6, 7\}$

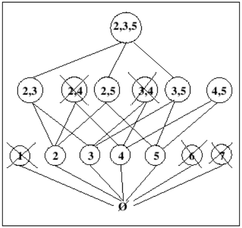


$L_1 = \{2, 3, 4, 5\}$









- On peut constater l'examen plus efficace des ensembles candidats.

Itemsets vus par des Treillis (suite)

- **On n'aime pas**
  - Repasser dans la DB pour les itemsets de taille différentes
  - Pour le support des itemsets et pour la confiance des règles
  - ➢ L'utilisation d'une table de hachage est une aide appréciable.

- **Une stratégie utilisée** :
  générer (k+2)-itemsets après la génération des (k+1)-itemsets
  - Intéressant si BD trop large (et problème taille mémoire lors des passes)
  - Passage en même temps dans la DB pour 2 tailles consécutives d'itemsets
  - Génération k+2 avant de vérifier le support/confiance des k+1
  - Inconvénient :
    (k+2)-itemsets plus que nécessaire mais évite des passes
    ➙ C'est un compromis

Génération de règles : rappel méthode simple

- Règles générées à partir d'itemsets fréquents.
- Il faut des règles $LHS \Rightarrow RHS$ avec un **maximum** de confiance.
- On utilise le support de LHS et celui de $LHS \cup RHS$
  ➙ Ces *supports* sont obtenus de la **table de hachage** (voir la table 7)

  ➙ La **confiance** de la règle obtenue : $\frac{|LHS \cup RHS|}{|LHS|}$ où $LHS \cap RHS = \oslash$

- La méthode "**brut-force**" de création d'une règle LHS ⇒ RHS :
  ➙ De complexité $2^N - 1$, $N =$ taille de l'itemset

- Brut-Force : si plusieurs termes dans l'itemset
  Placer chaque sous-ensemble l'itemset en RHS, le reste comme antécédent ...

  ➙ Méthode inefficace !

../..

Règles : vers une meilleure méthode

☞ **Rappel** : après le comptage d'un itemset dans la BD, on stock le couple dans une table de hachage.

- On sait que si la règle $\boxed{A\ B \Rightarrow C\ D}$ est valide
  Alors les règles à simple conséquence $\boxed{A\ B \Rightarrow C}$ et $\boxed{A\ B \Rightarrow D}$ le sont aussi.
  - ➙ Puisque LHS reste le même et le support de ABC (et de ABD) est (potentiellement) supérieur à celui de ABCD
  - ➙ La confiance pour ces deux règles à 1-conséquence risque d'augmenter.

- **<u>Par contre</u>** :
  Si l'une des règles $\boxed{A\ B \Rightarrow C}$ ou $\boxed{A\ B \Rightarrow D}$ N'EST PAS valide
  Alors la règle $\boxed{A\ B \Rightarrow C\ D}$ **N'EST sûrement PAS** valide

- Le principe de construction des règles à *c+1-conséquences* à partir des règles à *c-conséquences* (voir l'exemple suivant) :
  Envisager la règle *c+1-conséquences* sur la base des *c-conséquences* ../..

Règles : vers une meilleure méthode (suite)

- Mieux (donc) : pour un même itemset
  construire les règles *(c+1)-conséquences* à partir des *c-conséquences*
- **Exemple :** pour l'itemset {windy=false and play=no, outlook=sunny , humidity=high}

  Si les 2 règles à 1-conséquence (noter {windy=false, play=no} répétés dans les *LHS*)

  if humidity=high and windy=false and play=no Then outlook=sunny. *(2/2)*
  if outlook=sunny and windy=false and play=no Then humidity=high. *(2/2)*

  sont valides avec le minimum requis de couverture et de confiance

  Alors la règle à 2-conséquences suivante l'est aussi :
  If windy=false and play=no Then outlook=sunny and humidity=high. *(2/2)*

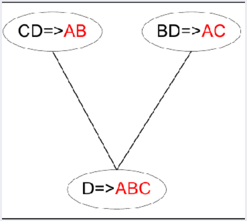
> Une règle à *(c+1)-conséquences* est valide si **toutes** ses règles (du même itemset) à *c-conséquences* sont valides (support et confiance).

☞ **L'inverse est évident** : si l'une de ces 2 règles 1-conséquence n'est pas valide, ne pas considérer la règle à double conséquences.

Exemples

- **Rappel du modus operandi** :
  - Construire (envisager) des règles à k+1-conséquences à partir des règles à k-conséquence,
  - La confiance de chaque règle candidate est calculée via la table de Hachage
  - En général, on vérifier bien moins de règles que dans la méthode "brut force".

- La règle à 3-conséquences proposée à partir des deux règles 2-conséquences.
  ➛ On doit juste extraire le support de "D" (on connaît celui du 4-itemset)



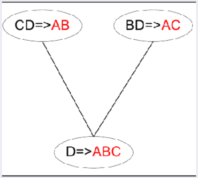
Exemples (suite)

Une technique équivalente :

Rappel : une règle *(k+1)-conséquences* est valide si toutes ses sous-règles *k-conséquences* sont valides (ont la *confiance* requise).

- Soit l'exemple vu plus haut :

Considérer la génération des règles candidates en fusionnant 2 règles qui partagent un même préfixe dans leur prémisses :



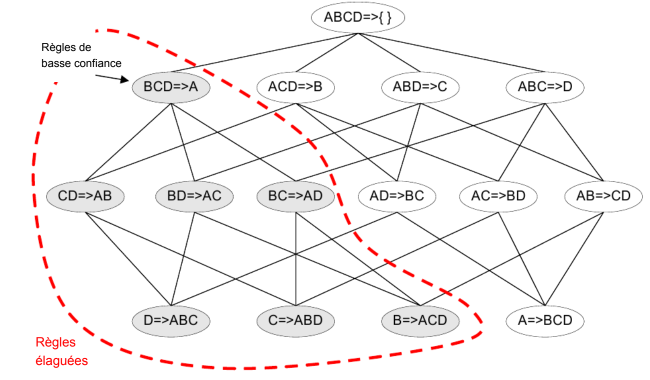
1- Fusionner $(CD \Rightarrow AB,\ BD \Rightarrow AC)$ donnant le candidat $(D \Rightarrow ABC)$.

2- Élaguer $(D \Rightarrow ABC)$ si TOUS ses sous-ensembles $(AD \Rightarrow BC)$ et $(BC \Rightarrow AD)$ n'ont pas la confiance prévue.

☞ Ce principe d'élagage permet une optimisation appréciable. $\cdots \rightsquigarrow$

Exemples (suite)

Importance de l'élagage (pour l'optimisation) :



- Si $BCD \Rightarrow A$ (*1-conséquence*) est de confiance faible
  ➜ Aucune des règles à *2-conséquence* (avec LHS $\subseteq \{B, C, D\}$) ne sera meilleure.

Exemples (suite)

Élagage ou pas : une question de *confiance* (des règles)

- En général, la mesure de confiance n'a pas la propriété d'*antimonotonie* :

➙ *conf(ABC → D)* peut être plus grande ou plus petite que *conf(AB → D)*.

- Mais la confiance des règles engendrées par le même itemset fréquent possède cette propriété :

Par exemple, pour $L = \{ABCD\}$ fréquent, on a :

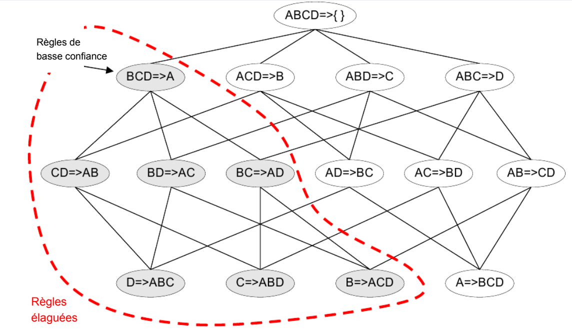
$$conf(ABC \rightarrow D) \geq conf(AB \rightarrow CD) \geq conf(A \rightarrow BCD)$$

➙ La propriété vient en fait de celle du support (de la partie LHS).
➙ La confiance est anti-monotone p/r au nombre d'items dans la RHS de la règle.

- Un exemple du cas complémentaire $\cdots \rightsquigarrow$

Exemples (suite)

Exemple : dans la hiérarchie suivante (voir la partie droite de la figure) :



- Si $A \Rightarrow BCD$ est d'une confiance suffisante (en bas à droite de la figure),
  ➛ toutes les règles qui en découlent (toute la partie droite de la hiérarchie) auront la confiance requise.

Remarques pratiques

Remarques pratiques pour fixer les seuils :

- On veut souvent N règles (e.g. 50) avec
  - le maximum de support (trouvable par itérations)
  - respectant le minimum de confiance prédéfini

  ➞ Génération de règles par réduction <u>incrémentale</u> du support
    - commencer par un support élevé,
    - réduction incrémentale pour atteindre le nombre de règles souhaité

  ➞ Le temps de calcul dépend du minimum du support souhaité
  ➞ La confiance n'affecte pas le nombre de passage dans la DB

- Le rôle des experts est non négligeable.

Méthode A Priori

La méthode (ci-dessus) de génération des Règles d'association
➙ la méthode A PRIORI

• Pour calculer des règles d'association souvent recherchées dans les bases de données très larges (e.g. domaine *Market Basket*)
➙ Dans ce domaine, les algorithmes <u>efficaces</u> sont importants.

Remarque : méthode A PRIORI développée pour les données *Market Basket* où :

➙ Les attributs = items dans le caddie avec beaucoup de valeurs manquantes

➙ Les instances = transactions, avec beaucoup d'attributs booléens (présent/non)

☞ La *confiance* n'est pas forcément la meilleure mesure

➙ E.g. le "lait" se répète dans presque toutes les transactions

➙ Il y a d'autres mesures plus adaptées (voir plus loin) ...

A Priori : exemple de caddie

Rappel algorithme A Priori

- C_k : ensembles des itemsets candidats de taille k.
- L_k : ensembles des itemsets fréquents de taille k, $L_k \subseteq C_k$.
  - Générer L_1 (en ne conservant que les singletons fréquents)
  - Répéter les étapes $k = 2...$ jusqu'à ce que plus aucun itemset fréquent n'existe :
    - Génération des itemsets candidats C_k à partir de L_{k-1}
    - Elager $I \in C_k$ si I contient un sous-ensemble non fréquent dans L_{k-1} (antimonotonicité)
    - Comptage effectif de la fréquence des candidats dans la BD.
    - Eliminer les candidats NON fréquents

Encore un exemple :

•Application à un exemple de caddie (Basket Analysis) ../..

A Priori : exemple de caddie (suite)

Exemple caddie :

- **On cherche** les règles telles que :

$$\{couches\} \Rightarrow \{bière\},$$
$$\{lait,\ pain\} \Rightarrow \{oeufs,\ cola\},$$
$$\{bière,\ pain\} \Rightarrow \{lait\}$$

TID	Items
1	*pain, lait*
2	*pain, couches, bière, oeufs*
3	*lait, couches, bière, cola*
4	*pain, lait, couches, bière*
5	*pain, lait, couches, cola*

Ces règles ne dénotent pas de causalité mais plutôt des corrélations (co-occurrences) orientées respectant un seuil de *confiance*.

☞ **Weka** et la représentation de ces BDs :
- ◦ Binarisation
- ◦ Association d'un entier à chaque couple Attribut-Valeur.

A Priori : exemple de caddie (suite)

- **Les itemsets :** Soit la BD (**support minimal= 3**) :

TID	Items
1	pain, lait
2	pain, couches, bière, oeufs
3	lait, couches, bière, cola
4	pain, lait, couches, bière
5	pain, lait, couches, cola

C_1 / L_1

Item	support
pain	4
cola	2
lait	4
couches	3
bière	3
oeufs	1

→

C_2 / L_2

Itemset	support
{pain, lait}	3
{pain, bière}	2
{pain, couches}	3
{lait, bière}	2
{lait, couches}	3
{bière, couches}	3

→

L_3

Itemset	support
{pain, lait, couches}	3

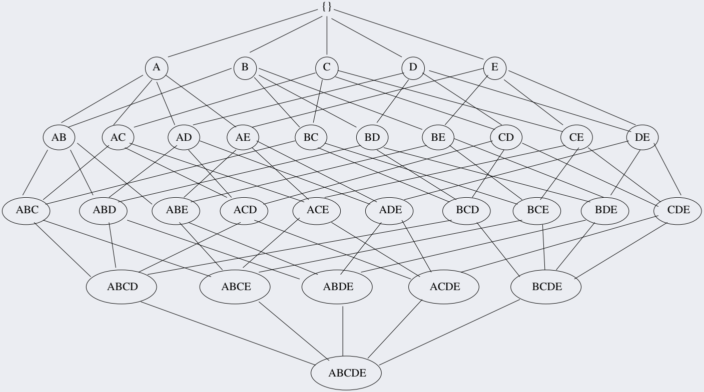
- Avec l'élagage à base du support, on considère 13 itemsets au lieu de 41.
- Si on prend L_3, les règles de conf=100% possibles sont (l'exemple est simple) :

$$\begin{array}{ll} couches \Rightarrow \{pain, lait\} & conf = \frac{3}{3} = 1 \\ \{lait, couches\} \Rightarrow pain & conf = \frac{3}{3} = 1 \\ \{lait, pain\} \Rightarrow couches & conf = \frac{3}{3} = 1 \\ \{couches, pain\} \Rightarrow couches & conf = \frac{3}{3} = 1 \end{array}$$

☞ $pain \Rightarrow \{lait, couches\}$ et $lait \Rightarrow \{pain, couches\}$ ont une *confinace* $= \frac{3}{4}$

Complexité de génération des règles d'association

Soit le treillis des itemsets possibles avec les attributs $\{A, B, C, D, A\}$



- Pour d items (attributs), il y a 2^d itemset candidats possibles

Complexité de génération des règles d'association (suite)

Mesure de la complexité de génération des règles d'association :

- Soit d items uniques
  - ➙ On aura 2^d itemsets possibles.
  - ➙ Le nombre total des Règles d'association sera :

$$R = \sum_{k=1}^{d-1} \left[\binom{d}{k} \times \sum_{j=1}^{d-k} \binom{d-k}{j} \right]$$

$$= 3^d - 2^{d+1} + 1$$

- ➙ Pour $d = 6$, on a R=602 règles !
- ➙ Pour $d = 9$, environ 20000 règles !

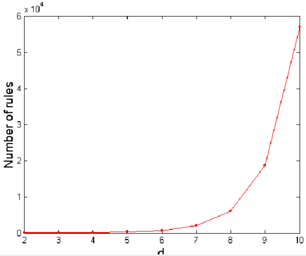


TABLE 8: complexité de génération

Solutions et Techniques utilisées

Agir sur la complexité de génération des Fréquents :

- 1- Réduire le nombre de transactions (soit N) lors d'apprentissage

  - Réduire la taille de N lorsque la taille des itemsets augmente

  ➙ Utilisé par les méthodes **DHP** (Direct Hashing and Pruning = une extension de *A Priori*) et les algorithmes de développement *verticaux* (cf. d'Arbre de Décision). Voir aussi Chap. 6

- 2- Réduire le nombre de comparaisons

  - Utilisation efficace des structures de données (e.g. *Hash-tables*) pour stocker les candidats/transactions

  - Réduire le besoin de comparer tous les candidats contre toutes les transactions (via des techniques d'optimisation)

☞ La représentation par matrice hachée (vue plus haut) permet des améliorations.

Solutions et Techniques utilisées (suite)

- 3- Réduire (rapidement) le nombre des candidats

  - Recherche complète : 2^d itemsets possibles (pour d attributs)

    ➜ Utilisez des techniques pour réduire ce nombre (cf. C_{k+1} par L_k).

Principe (utilisé dans l'algorithme A Priori) :

> Si un itemset est fréquent, alors tous ses sous-ensembles doivent également être fréquents

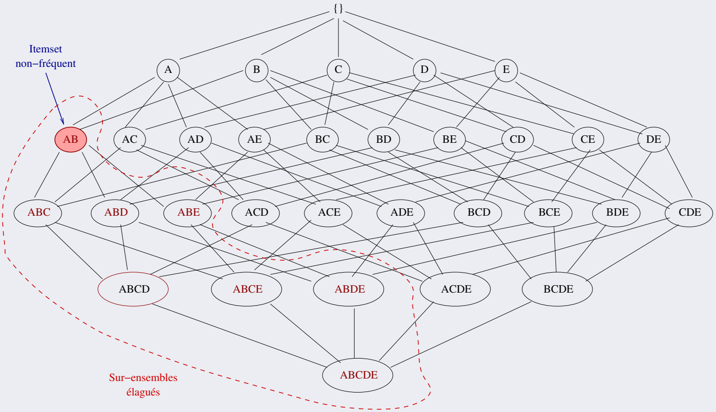
- La méthode *A Priori* utilise la propriété de la mesure du Support (et de la fréquence) :

$$\forall X, Y : (X \subseteq Y) \rightarrow s(X) \geq s(Y)$$

➜ Le Support d'un itemset ne dépasse jamais le support de ses sous-ensembles

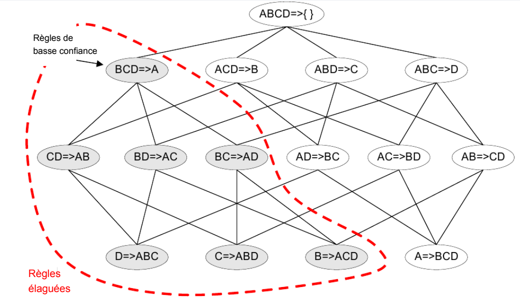
Solutions et Techniques utilisées (suite)

A l'inverse : {A,B} non fréquent permet d'éliminer ses sur-ensembles (super-itemsets)



Solutions et Techniques utilisées (suite)

- 4- Générer efficacement les règles (rappel) :



- Si $BCD \Rightarrow A$ (*1-conséquence*) est de confiance faible
  ➙ Aucune des règles à *2-conséquence* ne sera meilleure.
  ➙ Propriété d'antimonotonicité en LHS

- Également, si $A \implies BCD$ est de confiance (à droite), toute la partie droite l'est aussi

Représentation compacte des Itemsets

- D'autres techniques de réduction de la complexité
- Constat :

TID	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

- Le nombre des itemsets fréquents : $3 \times \sum_{k=1}^{10} \binom{10}{k}$

➙ **Besoin d'une représentation compacte : Maximal et Clos.**

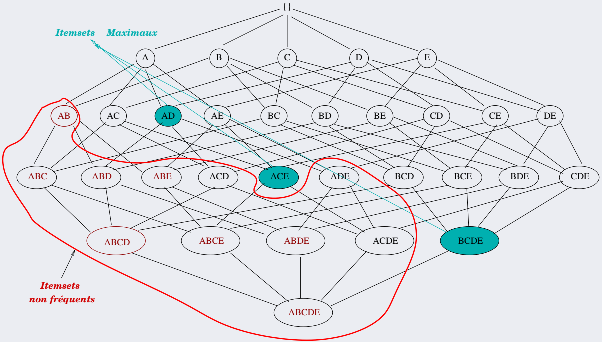
Itemsets Maxima

Objectif : trouver les itemset de longueur maximale (+ réduire le nbr. des fréq.)

Rappel : si I est fréquent, ses sous-ensembles le sont aussi
(avec une fréquence potentiellement supérieure à celle de I)

• **Définition** : un itemset I est **Maximal** si aucun de ses super-itemsets immédiats (sur-ensembles de I de taille +1) n'est *fréquent*.

- Les itemsets maxima (en bleu) n'ont (par def.) pas un sur-ensemble fréquent.
- Un *Maximal* est un *fréquent* duquel on peut déduire tous ses *sous-fréquents* sans en connaître le support exact (mais potentiellement plus grand).
- AC n'est pas maximal car ACE est fréquent.



Itemsets Clos

- **Définition** : un itemset I est **Clos** si aucun de ses super-itemsets (sur-ensembles) immédiats n'a exactement *le même support.*

  ➙ Un itemset I n'est donc pas clos si au moins l'un de ses sur-ensembles immédiats a le même support (la couleur ➙ clos) :

La BD.

TID	Items
1	A, B
2	B, C, D
3	A, B, C, D
4	A, B, D
5	A, B, C, D

➙

C_1 et C_2

Itemset	support
A	4
B	5
C	3
D	4

AB	4
AC	2
AD	3
BC	3
BD	4
CD	3

➙

C_3 et C_4

Itemset	support
ABC	2
ABD	3
ACD	2
BCD	3

ABCD	2

- Les itemsets marqués en couleur sont clos :
  ils n'ont pas de super-itemset **du même support**.

Itemsets Clos (suite)

Rappel de l'exemple : parmi tous les itemsets fréquents (tables C_1, C_2, C_3 et C_4 ci-dessous), ceux qui sont marqués en couleur sont **clos** car il n'y a aucun autre itemset **du même support** qui les <u>*subsume*</u>.

• Pour les autres (non marqués par la couleur), ils ne sont pas clos car il y a au moins un itemset du même support qui les <u>*subsume*</u>.

P. Ex. : AC : 2 n'est pas clos car ABC : 2 existe (lui-même *subsumé* par $ABCD$: 2).

C_1 et C_2

Itemset	support
A	4
B	5
C	3
D	4
AB	4
AC	2
AD	3
BC	3
BD	4
CD	3

→

C_3 et C_4

Itemset	support
ABC	2
ABD	3
ACD	2
BCD	3
ABCD	2

Itemsets Clos (suite)

L'ensemble Maximal est inclus dans Clos :
lorsque un itemset n'est pas clos, il ne sera pas Maximal (voir Fig. ci-dessous).

- Dans la table ci-dessus, parmi les itemsets clos marqués, seul $ABCD : 2$ est maximal (car aucun autre fréquent le subsume).
  - ➛ *De facto*, un itemset au sommet du treillis des itemsets est Maximal (et donc clos) s'il est fréquent car il n'a pas de sur-ensemble.

C_1 et C_2

Itemset	support
A	4
B	5
C	3
D	4
AB	4
AC	2
AD	3
BC	3
BD	4
CD	3

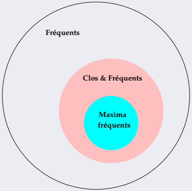
➛

C_3 et C_4

Itemset	support
ABC	2
ABD	3
ACD	2
BCD	3
ABCD	2

Maximal

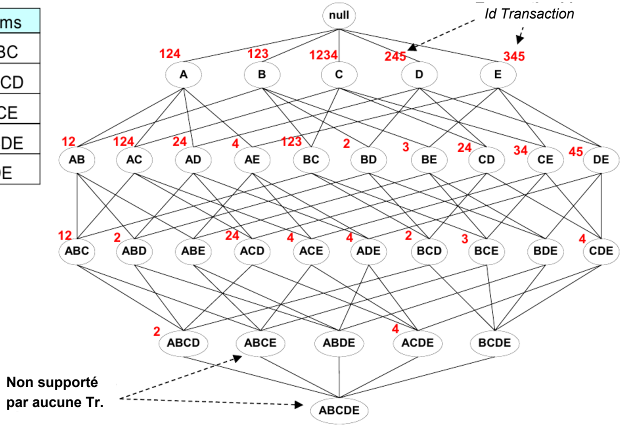
Itemset	support
ABCD	2



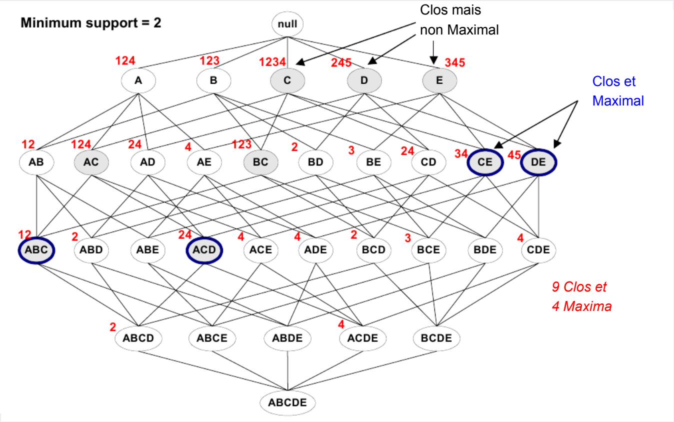
Maximal vs. Clos : exemple

- Soit le treillis de $ABCDE$:

TID	Items
1	ABC
2	ABCD
3	BCE
4	ACDE
5	DE



Maximal vs. Clos : exemple (suite)

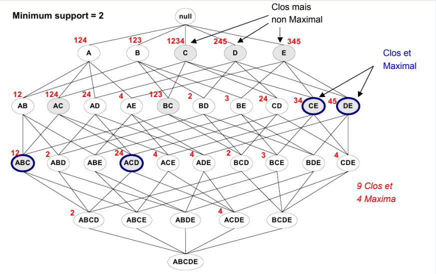


- *CE*, *DE*, *ABC*, *ACD* (de support=2) sont des *Maxima* car aucun de leurs sur-ensembles n'est fréquent. Il sont des *Clos* car aucun de leurs sur-ensembles n'a le même support.
- *D*, *A*, *AC*, *BC* (support=3) et *C* (support=4) sont des *Clos* car aucun de leurs sur-ensembles n'a le même support. Ils ne sont pas *Maxima* car ils ont d'autres sur-ensembles fréquents.

Maximal vs. Clos : exemple (suite)

Comment faire ? :

∘ D'abord les Clos (même support) puis les Maxima (fréquent).
∘ Considérer la base (le Top) du treillis et éliminer les non fréquents (ici Supp.=2).
➙ élimine ici tout itemset de taille ≥ 4.
∘ De ceux de longueur = 3, il ne reste que $\{ABC, ACD\}$.
∘ ABC élimine AB (même support), ACD élimine AD et CD (même support).
∘ On a fini avec la longueur=3 et on conserve $\{AC, BC, CE, DE\}$ du niveau (de longueur=2).



∘ AC élimine A (même support), BC élimine B.

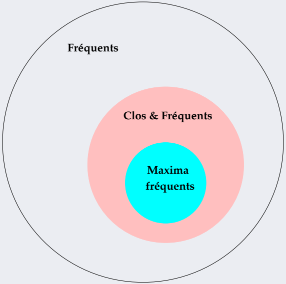
Fin du calcul des Clos.

∘ Pour les Maxima, on part de l'ensemble des 9 Clos.
∘ D'emblée, les Clos de longueur=3 ($\{ABC, ACD\}$) seront des Maxima.
∘ ABC élimine $\{AC, BC\}$ des itemsets de longueur=2 (fréquents).
∘ ACD n'élimine rien et du niveau de longueur=2, il reste $\{CE, DE\}$ considérés comme Maxima (car rien ne les a éliminé).
∘ CE éliminera $\{C, A\}$ du niveau de longueur=1, DE éliminera D en plus.

∘ L'ensemble des Maxima = $\{ABC, ACD, CE, DE\}$.

Fin des calculs.

Maximal vs. Clos : exemple (suite)



- Pourquoi préférer $ABCD$ à AB (*clos* ou *maximal* vs simples fréquents) ?
  - Pour éviter l'explosion combinatoire ; le clos (maximal) a le même pouvoir d'expression ;
  - Le *clos* (maximal) ne contredit pas le *simple* (mais l'inverse n'est pas vrai)
  - Le *clos* (maximal) est plus représentatif,
  - Les *clos* (maxima) évitent un nombre important de règles.

Algorithme Clos

Fonction Clos (version qui ajoute des Clos au fur et à mesure) :

Entrées : IS_Freq ensemble des itemsets fréquents.
Sorties : IS_Clos ensemble des itemsets (fréquents et) Clos.

1. $IS\_Clos \leftarrow \varnothing$;
2. Ordonner IS_Freq dans l'ordre descendant des tailles des itemsets (le plus long d'abord)
3. $\forall I \in IS\_Freq$

   | $IS\_Clos \leftarrow IS\_Clos \cup \{I\}$
   | Si $\exists I' \in IS\_Freq$ tel que $I' \subseteq I$ et $support(I) = support(I')$
   | | Alors I' ne sera pas un Clos
   | | Sinon $IS\_Clos \leftarrow IS\_Clos \cup \{I'\}$;
   | Fin si

   Fin Pour

Algorithme Clos optimisé

Fonction Clos optimisée (version qui enlève les Non Clos au fur et à mesure) :

Entrées : IS_Freq ensemble des itemsets fréquents.
Sorties : IS_Clos ensemble des itemsets fréquents et Clos.

1. $IS\_Clos \leftarrow IS\_Freq$;
2. Ordonner IS_Clos dans l'ordre descendant des tailles des itemsets (le plus long d'abord)
3. Tant que $IS\_Clos \neq \varnothing$ `(`IS_Clos `modifié au fur et à mesure)`
   | Choisir $I \in IS\_Clos$ `le plus grand restant`
   | Si $\exists I' \in IS\_Clos - \{I\}$ tel que $I' \subseteq I$ et $support(I) = support(I')$
   | Alors $IS\_Clos \leftarrow IS\_Clos - \{I'\}$ `(supprimer` I' `de` IS_Clos`)`
   | Fin si

   Fin Tant que

Algorithme Maximal

Fonction Maximal :

Entrées : IS_Clos ensemble des itemsets fréquents **Clos**

Sorties : $IS\_Clos\_Maximal$ ensemble des itemsets fréquents Clos et Maximal.

1. $IS\_Clos\_Maximal \leftarrow IS\_Clos$;
2. Ordonner $IS\_Clos\_Maximal$ dans l'ordre descendant des tailles des itemsets (le plus long d'abord)
3. Tant que $IS\_Clos\_Maximal \neq \varnothing$

   | Choisir $I \in IS\_Clos\_Maximal$ `le plus grand restant`

   | Si $\exists I' \in IS\_Clos\_Maximal - \{I\}$ tel que $I' \subseteq I$

   | Alors $IS\_Clos\_Maximal \leftarrow IS\_Clos\_Maximal - \{I'\}$

   | Fin si

   Fin Tant que

Calcul Clos/Maximal : exemple météo

Un exemple (météo) :

- Rappel de la BD. :

Num	Temps(B)	Temperature(T)	Humidité(H)	Vent(V)	Classe
1	ensoleillé	Elevée	Elevée	non	N
2	ensoleillé	Elevée	Elevée	oui	N
3	nuageux	Elevée	Elevée	non	P
4	pluvieux	Moyenne	Elevée	non	P
5	pluvieux	Faible	Normale	non	P
6	pluvieux	Faible	Normale	oui	N
7	nuageux	Faible	Normale	oui	P
8	ensoleillé	Moyenne	Elevée	non	N
9	ensoleillé	Faible	Normale	non	P
10	pluvieux	Moyenne	Normale	non	P
11	ensoleillé	Moyenne	Normale	oui	P
12	nuageux	Moyenne	Elevée	oui	P
13	nuageux	Elevée	Normale	non	P
14	pluvieux	Moyenne	Elevée	oui	N

Calcul Clos/Maximal : exemple météo (suite)

- On fixe la fréquence à *f=40%* (support entre 5 et 6 instances sur 14)
- La recherche des itemsets fréquents donne :
  - 6 singletons
  - 2 couples : {*Humidité=normale, jouer=oui*} et {*vent=oui, jouer=oui*}
    ➙ tous deux de fréquence 0.43
- Tous ces 8 itemsets sont Clos : pas de sur-ensemble du même support.
- Les itemsets Maxima sont :
  - 3 singletons : {*Température=moyenne*} *et* {*Vent=oui*} *de fréquence 0.43,* {*Humidité=forte*} *de fréquence 0.5*
  - 2 couples : {*Humidité=normale, jouer=oui*} *et* {*Vent=oui, jouer=oui*}
    ➙ de fréquence 0.43

Calcul Clos/Maximal : exemple météo (suite)

- Rappel :

  {*Humidité=normale, jouer=oui*} *et* {*Vent=oui, jouer=oui*}
  ➙ de fréquence 0.43

- Les règles d'association de confiance $C \geq 0.75$:

  - Les singletons ne donnent rien.
  - Les règles obtenues sont :

$$\boxed{\begin{array}{ll} \quad \textit{Humidité=normale} \Rightarrow \textit{jouer=oui} & \text{(B=0.43, C=0.86)} \\ \text{et} & \\ \quad \textit{Vent=oui} \Rightarrow \textit{jouer=oui} & \text{(B=0.43, C=0.75)} \end{array}}$$

Remarques sur Clos/Maximal

- **Pourquoi préférer** des itemsets plus grands aux plus petits ?
  - La taille (grande) des zones couvertes par les Clos et Maxima dans la BD permet de mieux représenter les connaissances (voir l'ex. suivant).
  - Puis produire des règles avec une condition (la + simple) selon la Conf. :
    ➙ la règle serait plus simple à exploiter !

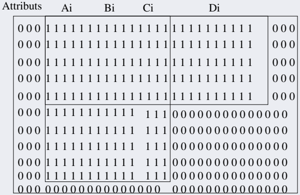
$$\begin{array}{rl} & lait \implies Tomates, Patates, Caviar, Truffes, Champagne \\ \text{mieux que} & lait, Tomates, Patates \implies Caviar, Truffes, Champagne \end{array}$$

- Rappel :

TID	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Remarques sur Clos/Maximal (suite)

- Dans le cas des DBs de la forme :



  - Les zones des "Fréquents" et "Clos(es)" : $(A_i \cup B_i \cup C_i)$ et $(A_i \cup B_i \cup C_i \cup D_i)$
  - La zone "Clos(e)" et "Maximale" : attributs $(A_i \cup B_i \cup C_i \cup D_i)$

    ➙ Selon le nombre de règles souhaitées, on peut exploiter seulement le Maxima ou les Clos (les deux zones contiennent des "Fréquents").

N.B. : les Maxima tels que $ABCD$ permettent d'obtenir des règles du type $A \Rightarrow BCD$ qui représentent plus de connaissances que celles obtenues p. ex. de ABC.

➙ Par contre, les degrés de confiance seront différents, pour les règles issues des fréquents (seuls), des Clos ou des Maxima.

Itemsets et Support variable

Constat : dans certains de cas rencontrés (réels), le support des itemsets baisse exponentiellement avec la taille des itemsets.

- **Que faire** ? (cadre statistique des "Mixture Models")

  1- Ne pas affecter le même support à tous les items (singletons)

  2- Définir un minimum pour chaque item :
  - e.g. $support(pain) > support(vin) > support(saumon) > support(caviar)$
  - Le support d'un itemset sera alors le minimum des supports de ses items.

  3- Exiger la présence d'un item dans les itemsets (cas rare).

- **Inconvénients** : ce support n'est plus anti-monotone !
  - Les itemsets contenant *caviar* deviennent plus *fréquents* que ceux sans.
  - Potentiellement, $\{pain,\ vin\}$ ne sera pas Clos (donc non retenu)
    alors que $\{pain,\ vin,\ saumon\}$ le deviendrait.

- On peut prévoir un intervalle de support (min .. max)
  ➙ Changer l'algorithme *"A Priori"* (vu plus haut)

../..

Itemsets et Support variable (suite)

Prise en compte dans la méthode *A Priori* :

- L'approche "traditionnelle" (algorithme *A priori*) :
  - Générer la liste des candidats C_{k+1} en fusionnant deux itemsets fréquents de L_k ;
  - Eliminer le candidat si l'un de ses sous itemsets de taille k n'est pas fréquent.

- **Modification** : supports variables

  ➙ Supposons avoir les items avec leur fréquences minimales :
  Lait : 5%, Cola : 3%, Brocoli : 0.1%, Saumon : 0.5%.

  ➙ Ordonner ascendant ces items selon leur fréquence :
  Brocoli, Saumon, Cola, Lait.

  ➙ Pour un itemset I contenant ces items :
  $$\text{Min(fréquence(I))} = Min(freq(Item_i)), item_i \in I$$

- Soit L_1 l'ensemble des items fréquents (étape 1, singletons)
  F_1 : ensemble d'items dont la fréquence $\geq$ Min(fréquence)
  C_2 sera la liste des candidats (étape 2) générées àpd F_1 (plutot que L_1). ../..

Itemsets et Support variable (suite)

Modification de l'algorithme *A Priori* (suite) :

L'étape d'élagage change :

- Élaguer un itemset seulement si l'un de ses sous-itemsets est non-fréquents et contient le premier item (ens. ordonné) (ici Brocoli)

➙ Ex. : soit le candidat ={Brocoli, Cola, Lait}

➙ {Brocoli, Cola} et {Brocoli, Lait} sont fréquents s'ils ont au moins la fréquence *min* de Brocoli.

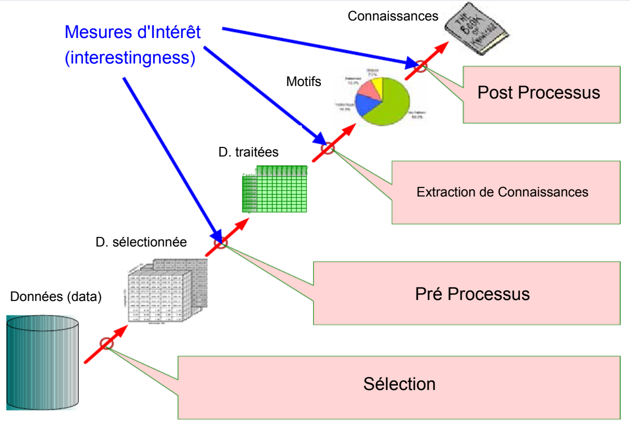
➙ {Cola, Lait} sera éliminé si sa fréquence < fréq(Cola).

➙ I.e. : si un sous itemset du candidat contenant *Brocoli* était non-fréquent, on supprimerait le candidat

➙ sans quoi la fréquence du sous-itemset tomberait trop bas.

Évaluation des règles

Mesure d'intérêt dans le processus ECD



Évaluation des règles (suite)

- **Un problème** : les algorithmes de création de règles d'association produisent beaucoup de règles
  - P. ex., pour 6 attributs, 2^6 itemsets et 602 règles possibles.
  - Toutes ne sont pas intéressantes, voire, certaines sont redondantes
    - ➙ E.g. : redondance si $\{A, B, C\} \Rightarrow \{D\}$ et $\{A, B\} \Rightarrow \{D\}$
- Notations (rappels) : $(N = card(BD)$ et $N_{A \Rightarrow B} = card(A \cup B))$
  - $S(A \Rightarrow B) = S(A \cup B) \propto \frac{N_{A \Rightarrow B}}{N}$ rappel : $A \cap B = \emptyset$
  - $F(A \Rightarrow B) = \frac{N_{A \Rightarrow B}}{N} = \frac{N_{(A \cup B)}}{N}$ N omis si toutes mesures faites sur la même BD
  - $C(A \Rightarrow B) = \frac{S(A \Rightarrow B)}{S(A)} = \frac{F(A \Rightarrow B)}{F(A)}$
- Les **mesures** basiques *Fréquence* (F) et *Confiance* (C) ne suffisent pas.
  - S (ou F) : intérêt de toute *association* pouvant donner des règles
  - C : intérêt d'une règle (celui d'une de ces associations, c-à-d. 1 règle).
- Un recours : **Clos** et **Maximal** pour la génération des itemsets.
- On utilise également d'autres **mesures d'intérêt** des règles. ...

Mesures de qualité des règles d'Ass.

Préambule : *(in)dépendance statistique*

Un exemple :soit une population de 1000 étudiants dont

- ➙ 600 savent nager ($Nage$)
- ➙ 700 savent faire du vélo ($Velo$)
- ➙ 420 savent faire les deux ($Nage, Velo$)

- $P(Nage \wedge Velo) = \frac{420}{1000} = 0.42$ et
  $P(Nage) \times P(Velo) = 0.6 \times 0.7 = 0.42$ identique

- Si $P(Nage \wedge Velo) = P(Nage) \times P(Velo)$ alors **indépendance statistique**
  Si $P(Nage \wedge Velo) > P(Nage) \times P(Velo)$ alors **corrélation positive**
  Si $P(Nage \wedge Velo) < P(Nage) \times P(Velo)$ alors **anti-corrélation (nég)**

- On appelle **Lift** le rapport entre $\frac{P(Nage \wedge Velo)}{P(Nage) \times P(Velo)}$
  → Elle mesure le rapport entre ce qui est observé dans la B.D. ($A \wedge B$ ensemble) vs. sont-ce là par hasard (indép. l'un de l'autre).

☞ Les mesures comme *Lift* et χ^2 permettent de révéler ces relations. ...

Mesures de qualité des règles d'Ass. (suite)

- Freq. / Supp. et Conf. sont simples mais pas toujours suffisantes.
- lift est une mesure importante en complément de F et C $(F(A \Rightarrow B) = F(A \cup B))$

$$lift_{(A \Rightarrow B)} = \frac{C(A \Rightarrow B)}{F(B)} = \frac{P(B|A)}{P(B)} = \frac{\frac{F(A \Rightarrow B)}{F(A)}}{F(B)} = \frac{F(A \Rightarrow B)}{F(A).F(B)} = \frac{P(AB)}{P(A).P(B)} = lift_{(B \Rightarrow A)}$$

> *lift* est le ratio entre les fréquences relatifs de $(A \cup B)$ et la fréquence relative du même événement si A et B devaient être indépendants.

  - $lift = 1$ représente une indépendance
  - $lift > 1$ représente une association positive
  - $lift < 1$ montre une association négative.

- Lift est également appelé l'**intérêt** (*interestingness*) et se généralise à l'itemset.

$$interet(X, Y) = Lift_{(X,Y)} = \frac{P(Y|X)}{P(Y)} = \frac{P(Y|X).P(X)}{P(Y).P(X)} = \frac{P(X,Y)}{P(X).P(Y)}$$

> Si X et Y sont statistiquement indépendants, alors $Lift(X, Y) = \frac{P(Y|X)}{P(Y)} = 1$

Mesures de qualité des règles d'Ass. (suite)

Exemple de Lift (sur la BD. Météo) :

- Rappel des règles obtenues à partir des Maxima :

R1 : *Humidité=normale* ⇒ *jouer=oui* (F=0.43, C=0.86)
R2 : *Vent=oui* ⇒ *jouer=oui* (F=0.43, C=0.75)

$$lift(R_1) = \frac{C(\{Humidite=normale \Rightarrow jouer=oui\})}{F(Jouer=oui)} = \frac{0.86}{0.643} = 1.34$$

$$lift(R_2) = \frac{C(\{Vent=normale \Rightarrow jouer=oui\})}{F(Jouer=oui)} = \frac{0.75}{0.643} = 1.16$$

☞ On calcule *lift* sur les règles qui satisfont la confiance C.

Exemple : pour la règle *jouer=oui* ⇒ *Humidité=normale*

- ◦ $lift = 1.34$ association positive
- ◦ Mais $C = 0.667$
- ➙ Elle n'était pas retenue (pour $C = 0.75$ imposée.)

- **Lift** permet de mesurer la puissance d'un modèle prédictif par rapport au choix aléatoire.

Mesures de qualité des règles d'Ass. (suite)

Remarque sur Lift :

- On a obtenue deux règles à partir des Maxima :

R1 : *Humidité=normale* $\Rightarrow$ *jouer=oui* (F=0.43, C=0.86)
R2 : *Vent=oui* $\Rightarrow$ *jouer=oui* (F=0.43, C=0.75)

$$lift(R_1) = \frac{C(\{Humidite{=}normale \Rightarrow jouer{=}oui\})}{F(Jouer{=}oui)} = \frac{0.86}{0.643} = 1.34$$

$$lift(R_2) = \frac{C(\{Vent{=}normale \Rightarrow jouer{=}oui\})}{F(Jouer{=}oui)} = \frac{0.75}{0.643} = 1.16$$

- $Lift_{(A,B)}$ représente le ratio de la fréquence de l'observé (celui de $A \cup B$) sur celle attendue de A et de B s'ils devaient être indépendants (ou *random*).
  - Pour la règle R1 ci-dessus, *lift* = 1.34 représente une (meilleure) association positive de RHS à LHS.
  - (En plus,) la Conf. de R1 est également plus élevée.

Intervalle de confiance du lift

- But : mieux obtenir le seuil d'inférence pour les règles d'association
  - permet la sélection des meilleures règles
  - celles avec un pouvoir d'inférence plus élevée.
- La taille de cet intervalle exprimera le degré d'incertitude de l'intérêt de la règle.
- Rappel : $lift(A \Rightarrow B) = \frac{F(A \Rightarrow B)}{F(A).F(B)}$
- Un intervalle de confiance par $log(lift)$:

$$log(lift) \pm Z_{1-\alpha/2} \sqrt{\frac{1}{F(A \Rightarrow B)} - \frac{1}{N} + \frac{1}{F(A)} + \frac{1}{F(B)}}$$

➙ α est le degré de confiance placée dans l'intervalle (différent de C)
➙ $Z_{1-\alpha/2}$ obtenu par la table de la distribution normale (voir plus loin).

Intervalle de confiance du lift (suite)

$$log(lift) \pm Z_{1-\alpha/2} \sqrt{\frac{1}{F(A \Rightarrow B)} - \frac{1}{N} + \frac{1}{F(A)} + \frac{1}{F(B)}}$$

- La taille de l'intervalle exprime le degré d'incertitude de l'intérêt d'une règle :

➙ Cette incertitude décroît lorsque la fréquence de la règle $A \Rightarrow B$ croit d'une manière balancée (i.e. les fréquence de A et B augmentent ensemble).

- L'intervalle de confiance = la *signification statistique* d'une règle :

☞ Si l'intervalle de confiance contient la valeur 1, alors *lift* peut devenir $= 1$ (malgré sa valeur calculée) alors la règle n'est pas intéressante.

➙ i.e. on ne peut tirer une association orientée (causalité ?) si A et B sont indépendants !

Rappel :

$$lift(A \Rightarrow B) = \frac{F(A \cup B)}{F(A).F(B)} = \frac{P(A,B)}{P(A).P(B)}$$

➙ $lift(A \Rightarrow B) = 1$ veut dire : A et B sont indépendants.

Interprétation probabiliste

Interprétation probabiliste des mesures d'intérêt

- La fréquence d'un itemset $\{A, B\}$ (qui pourra donner une règle) $= P(A \wedge B)$.

  N.B. : considérer $P(A \wedge B)$ comme $P(A = a \wedge B = b)$.
  ☞ Rappel : $P(A \wedge B) = F(A \cup B)$

- Une règle R : $A \Rightarrow B$ exprime $P(B|A)$:

$$confiance(A \Rightarrow B) = \frac{P(A \wedge B)}{P(A)} = \frac{P(B|A).P(A)}{P(A)} = P(B|A)$$

- On a aussi : $confiance(A \Rightarrow B) = \frac{P(A \wedge B)}{P(A)} = \frac{P(A|B).P(B)}{P(A)} = P(B).Lift_{A \Rightarrow B}$

  Rappel : $Lift(A \Rightarrow B) = \frac{P(B|A)}{P(B)}$

  Rappel : $P(A)$ est **la marginale** de A.

- Pour une confiance identique, on préfère retenir les règles avec un *Lift* supérieur.

- N.B. : en terme de proba, la confiance est exprimée pour l'itemset entier par :
  ➙ $confiance(A, B) = max(P(B|A), P(A|B))$ Si $conf(A, B)$ au lieu de $conf(A \Rightarrow B)$
  indépendamment de la règle finale retenue.

Interprétation probabiliste (suite)

En statistique, Lift a plusieurs utilisations.

- **Lift** permet de mesurer la puissance d'un modèle prédictif par rapport à un choix aléatoire.

- Par exemple :
  - Dans les enquêtes ou dans les campagnes de pub, si par exemple le taux de réponse général (donc un individu pris au hasard), est de 10%
  - Et si on a trouvé une partie A de la population (appelée *cible*) qui répondent à 30%, cela donnera un lift de 3.

  ➙ On trie les tranches (les *quantiles*) selon la valeur Lift pour viser les quantiles de meilleur Lift si on ne souhaite pas investir dans une campagne totale.

- Lift at p% : si la sous-population A (ci-dessus) est dans les premiers 15%, on dira **Lift at 15 = 3**.4

  ➢ Ou on voudra savoir p. ex. le *Lift at 50%* (Lift=1 est appelé "no Lift")

- Lift est équivalent à la *précision moyenne* (*average precision* : $\frac{TP}{TP+TN}$) :

  ➢ la probabilité de la (bonne / positive) réponse du modèle.

Conviction

Une autre mesure importante : la conviction (de/en une règle)

- La conviction d'une règle est définie par $conv_{A\Rightarrow B} = \frac{1-Freq(B)}{1-conf_{A\Rightarrow B}}$
- Elle est interprétée par le ratio entre (s'agissant de A et de B, voir ex. ci-dessous) :
  - le support attendu de A **sans** que B soit présent

    1-Freq(B) = la probabilité de se tromper dans la prédiction de B
  - et la probabilité de la "mauvaise" prédiction B sachant A.

Exemple : pour R1 ci-dessus : *[Humidité=normale ⇒ jouer=oui (F=0.43, C=0.86, Lift=1.34)]*

$$conv(R1) = \frac{1-0{,}643}{1-0{,}86} = 2.55$$

On fait mieux $2,55$ fois par rapport au hasard que A et B soient ainsi associés.

◦ Autrement dit : la règle serait incorrecte à 155% ($2,55$ fois) si l'association entre A et B devait vraiment être purement un hasard.

☞ Contre le hasard supposé de la réunion de A et de B, la BD. permet au contraire d'affirmer ($2,55$ fois plus "fort") que cette association n'est pas fortuite.

- N.B. : pour l'itemset $\{A, B\}$, la conviction est une probabilité donnée par :

$$conv_{A\Rightarrow B} = max\left(\frac{P(A).P(\overline{B})}{P(A\overline{B})}, \frac{P(B).P(\overline{A})}{P(B\overline{A})}\right)$$

Table de contingence & autres mesures

- Un outil de base (pour les mesures) : table de contingence.
  - Permet de construire rapidement des mesures de qualité.
  - Et d'expliquer différentes mesures d'intérêt...

- Ex. naïf : un marchand cherche des liens entre *tomates* et *carottes*
  - $\{tomates\} \Rightarrow \{carottes\}$? $\{carottes\} \Rightarrow \{tomatess\}$?,
  - $\{\neg tomates\} \Rightarrow \{carottes\}$? $\{carottes\} \Rightarrow \{\neg tomatess\}$?, ...

- Étant donnée la règle $X \Rightarrow Y$, on construit une table de contingence. :

	Y	$\neg Y$	
X	f_{11}	f_{10}	f_{1+}
$\neg X$	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	$\|T\|$

f_{11} = fréquence de X et de Y
f_{10} = fréquence de X et de $\neg Y$
f_{01} = fréquence de $\neg X$ et de Y
f_{10} = fréquence de $\neg X$ et de $\neg Y$
$|T|$ = totaux (1 ou 100, ...)

- La table de contingence est utilisée par plusieurs mesures d'intérêt
  ➙ Ces mesures pallient les insuffisances de F et C.

Table de contingence & autres mesures (suite)

Ex. : soit la table de contingence (proba de conso. de **Café** et de **Thé**) :

Conso.	Café	¬ café	
Thé	15	5	20
¬ Thé	75	5	80
	90	10	100

- Soit la règle d'association R : **Thé** $\Rightarrow$ **Café**. (i.e. *Café* | *Thé*)
  - *confiance(R)* = P(Café | Thé) = 0.75 $\left[= \frac{15}{20} = \frac{F(\{Thé, Café\})}{marginal(Thé)}\right]$
  - Or, $P(Café) = 0.9$! étrange ! (sauf le Thé dessert le Café)
  - Raison : la mesure *confiance* délaisse *P(Café)*.
  - La confiance est élevée mais la règle *Thé* $\Rightarrow$ *Café* est trompeuse car :

$$\text{P(Café} \mid \neg \text{Thé)} = 0.9375 \qquad \left[= \frac{75}{80} = \frac{F(\{\neg Thé, Café\})}{marginal(\neg Thé)}\right]$$

➙ Vaut mieux peut-être ne pas servir du Thé ?! Et la marginale (=20) du Thé ?

Comparaison au Complémentaire d'une règle

- La table de contingence permet également de calculer d'autres mesures.
- On peut comparer toute règle $A \Rightarrow B$ à son complémentaire $A \Rightarrow \overline{B}$
  - $\overline{B}$ est le complément de B (Vrai quand B=faux et vice versa).
  - Pour les règles d'assoc., $\overline{B}$ veut dire en particulier **absence de B**
  - $A \Rightarrow \overline{B}$ est appelé le *contre-exemple* de $A \Rightarrow B$
- Un exemple : Café $\Rightarrow \overline{Thé}$

  Avec Conf=0.83 et $Lift = 1.06$ (positive).

Comparaison au Complémentaire d'une règle (suite)

- Pour évaluer le rapport entre deux types de règles : $(A \Rightarrow B)$ et $(A \Rightarrow \overline{B})$

  On utilise une autre mesure indicative d'intérêt : **Odd**

$$\boxed{Odds(A \Rightarrow B) = \frac{F(A \Rightarrow B)}{F(A \Rightarrow \overline{B})}}$$

$F(A \Rightarrow \overline{B}) \neq 1 - F(A \Rightarrow B)$

- *Odd* permet de mieux mesurer la validité/intérêt d'un ensemble de règles

- <u>N.B.</u> : on lit $A \Rightarrow B$ comme $P(B|A)$ et $A \Rightarrow \overline{B}$ comme $P(\overline{B}|A)$

  ➢ *Odd* exprime le rapport entre les deux.

  ➢ P. ex. P(¬Thé ⇒ Café) vs. P(¬Thé ⇒ ¬ Café) !

- Interprétation Bayésienne de *Odd* : $\frac{P(X|Y)}{P(\overline{X}|Y)}$ (appelé Conditionnal Bayes Odd)

  - $Odds(A \Rightarrow B) = \frac{P(B|A)}{P(\overline{B}|A)}$: le rapport entre B et $\overline{B}$ sachant le même A.

  - $Odds(A \Rightarrow B) = x > 1$: il est x fois plus vraisemblable que $A \Rightarrow B$ que $A \Rightarrow \overline{B}$
  - $x < 1$: l'inverse

Comparaison au Complémentaire d'une règle (suite)

- N.B. : en probabilité conditionnelle de Bayes, $Odd - Ratio(A, B)$ est le rapport

$$\frac{Odds(A)}{Odds(B)} = \frac{\frac{P(A)}{1-P(A)}}{\frac{P(B)}{1-P(B)}}$$

  - Pour la table précédent, on a $Odd - Ratio(\text{Café}, \text{Thé}) = \frac{\frac{P(Caf\acute{e})}{A-P(Caf\acute{e}}}{\frac{Th\acute{e}}{1-Th\acute{e}}} = 36$

➤ Vraisemblablement, le café est 36 fois plus demandé / consommé que le Thé.

  - Effet / importance comparé du Café par rapport au Thé.

➤ Un conso. qui a soif a 36 fois plus de chance de prendre 1 Café qu'un Thé.

- N.B. : le (prior) *Bayes Odd* pour une variable B : $Odds(B) = \frac{P(B)}{P(\overline{B})} = \frac{P(B)}{1-P(B)}$

Et on peut calculer : $Odds(A \Rightarrow B) = Odds(B|A) = \frac{P(B|A)}{P(\overline{B}|A)} = \frac{P(A|B)}{P(A|\overline{B})} \times Odds(B)$

➤ Posterior Odds($A \Rightarrow B$) = Ratio de vraisemblance (cf. BD.) × prior Odds(B)

- Le rapport $\frac{P(A|B)}{P(A|\overline{B})}$ (ci-dessus) est appelé Bayes Factor (BF)

  - $BF = \frac{P(A|B)}{P(A|\overline{B})} > 1$: il est BF fois plus vraisemblable que $B \Rightarrow A$ que $\overline{B} \Rightarrow A$

  - $BF = \frac{P(A|B)}{P(A|\overline{B})} < 1$: l'inverse

Lift vs. Conf

- On peut comparer *Lift* vs. *Confiance* à l'aide de la table de contingence.

Conso.	Café	¬ café	
Thé	15	5	20
¬ Thé	75	5	80
	90	10	100

Soit la règle d'association : Thé ⇒ Café.

➞ confiance = P(Café | Thé) =0.75 Or, $P(Café) = 0.9$

- ***Lift* permet de corriger l'insuffisance de la *confiance* :**
  - $Lift_{(Café \Rightarrow \neg Thé)} = \frac{0.75}{0.9} = 0.8333$ (< 1 ➞ négativement corrélées).

    ☞ L'itemset $\{Café, Thé\}$ ne donne pas de règle intéressante !

  - Mais on avait P(Café | ¬ Thé) = 0.9375

    ➞ $Lift_{(\neg Thé \Rightarrow Café)} = Lift_{(Café \Rightarrow \neg Thé)} = \frac{75}{72}$ (> 1 ➞ *corrélation positive*).

  ➢ Cet itemset donnera une meilleure règle.

Lift vs. Conf (suite)

Résumé pour l'itemset {Thé,Café} :

- $Lift_{(A,B)}$ ne distingue pas les règles $A \Rightarrow B$ et $B \Rightarrow A$
- $Odd_{(A,B)}$ apporte un complément (pour les distinguer).
- On a $Odd_{A \Rightarrow B} = \frac{P(A\ B)}{P(A\ \overline{B})}$ directement de la table.

Conso.	Café	¬ café	
Thé	15	5	20
¬ Thé	75	5	80
	90	10	100

① Règle $Thé \Rightarrow Café$:

➙ $Conf_{(Thé \Rightarrow Café)} = 0.75$, $Odd_{(Thé \Rightarrow Café)} = \frac{Café \Rightarrow Thé}{Café \Rightarrow \neg Thé} = 3$

➙ $Lift_{(Café, Thé)} = \frac{0.75}{0.9} = 0.8333 < 1$ ➙ corrélation négative.

② Pour Café $\Rightarrow$ Thé, Conf=0.16, Odd=0.2 et $Lift = 0.8333$ (corr. négative).

③ Règle $\overline{Thé} \Rightarrow Café$ (sera plus **intéressante**) :

➙ $Conf_{(\overline{Thé} \Rightarrow Café)} = 0.9375$, Odds $= 15$

➙ $Lift_{(\overline{Thé}, Café)} = 1.06$ ➙ corrélation positive.

④ Pour Café $\Rightarrow \overline{Thé}$, Conf=0.83, Odd=5 et $Lift = 1.06$ (positive).

Lift vs. Conf (suite)

Un bémol : *Lift* n'est pas toujours la panacée à toute épreuve !

➙ Cela dépend des données (la BD.)

Un Contre exemple : soient les deux tables des contingence (pour 2 paires de mots en apprentissage *TextMining*) :

➙ Par exemple : X=*compilation* et Y=*mining* (table gauche)

vs. X=*data* et Y=*mining* (table droite)

	Y	$\bar{Y}$	
X	10	0	10
$\bar{X}$	0	90	90
	10	90	100

	Y	$\bar{Y}$	
X	90	0	90
$\bar{X}$	0	10	10
	90	10	100

../..

Lift vs. Conf (suite)

X=compilation et *Y=mining*

	Y	$\overline{Y}$	
X	10	0	10
$\overline{X}$	0	90	90
	10	90	100

X=data et *Y=mining*

	Y	$\overline{Y}$	
X	90	0	90
$\overline{X}$	0	10	10
	90	10	100

De la table gauche : $Lift_{X \Rightarrow Y} = \dfrac{P(X,Y)}{P(X).P(Y)} = \dfrac{0.1}{0.1 \times 0.1} = 10$

De la table droit : $Lift_{X \Rightarrow Y} = \dfrac{P(X,Y)}{P(X).P(Y)} = \dfrac{0.9}{0.9 \times 0.9} = 1.11$

- A gauche : très forte corrélation par Lift mais $P(X,Y)$ faible
  Mais $P(\neg X \wedge \neg Y)$ fort
- A droite (pour 2 autres mots) : faible corrélation (≈ 1)
  mais $P(X,Y)$ fort.

➛ Dans ces cas, la mesure **confiance** est <u>mieux adaptée</u>.

Autre mesure : Chi-2

- Une autre mesure d'intérêt d'association : la mesure χ^2 de *Pearson*.
  - ➙ Permet de valider l'hypothèse d'indépendance entre 2 variables.
  - ➙ Elle est ici étendue au cas d'association multivariée :

$$\boxed{\chi^2_{(A\Rightarrow B)} = \frac{(F(AB)-F(A).F(B))^2}{F(A).F(B)} = \chi^2_{(B\Rightarrow A)}}$$

  - ◦ χ^2 mesure **une** distance entre le couple (A, B) et l'indépendance de A, B.
    - ➙ Elle donne l'écart entre l'observé $A \cup B$ et une référence si A et B sont indép.

  Ex. : Si $\chi^2_{(A\Rightarrow B)} = \chi^2_{(B\Rightarrow A)} = 0$ alors A, B indépendants :
    - ➙ χ^2 est l'écart à l'hypothèse nulle : $P(AB) \stackrel{?}{=} P(A).P(B)$?

- **Exemple** : pour la règle R_1 : *Humidité=normale* $\Rightarrow$ *jouer=oui*
  - ➙ $\chi^2(R_1) = 0.37$ — Pas d'indépendance ; mesure à comparer avec le χ^2 d'autres règles pour conserver les meilleures.

Si *Humidité=normale* et *jouer=oui* sont indép., $\chi^2(R_1) = 0$ ../..

Autre mesure : Chi-2 (suite)

Remarques sur χ^2

- χ^2 Permet de conserver les meilleurs itemsets
  ➙ On a : $\chi^2_{(A \Rightarrow B)} = \chi^2_{(B \Rightarrow A)}$.

- χ^2 peut servir à imposer un point de départ de la mesure d'indépendance en faisant appel au seuil de prédiction (basé sur la distribution χ^2)

N.B. : χ^2 a une distribution de proba asymptotique (théorique)

➙ Peut évaluer de manière inductive le pouvoir (et le seuil) d'inférence

➙ Comme Lift, elle examine le degré de dépendance dans un couple d'itemsets (règle, causalité).

➙ Les règles avec un χ^2 plus élevée sont supposées être meilleures.

Addendum : A propos de χ^2

• En statistiques, le test du χ^2 permet de mesurer l'écart entre une situation observée et une situation théorique et d'en déduire l'existence et l'intensité d'une dépendance.

Exemple : il y a la même chance (théorique) d'obtenir "pile" que de "face" au lancer d'une pièce ; mais en pratique, les choses sont différentes.

➙ Le test χ^2 mesure alors l'écart entre la distribution théorique (une chance sur 2) et celle observée à la suite de lancements successifs.

• On vérifie un effet du hasard ou une coïncidence.

• Plus l'observé est proche de la théorique, plus grande sera l'indépendance (de l'observation) permettant d'écarter les coïncidences d'observations.

$$\chi^2 = \sum \frac{Ob - Th}{Th}$$

où Ob=observé et Th= la théorique (connaissances du domaine).

➙ Pour $\chi^2 = 0$, on dira que le site d'observation est indépendant de la théorie.
➙ Par contre, si $\chi^2 = 4$ avec un indice de confiance de 97% de signifiance, alors l'indépendance sera rejetée

• Le test χ^2 doit être fait en disposant des connaissances a priori du domaine (une distribution, une expertise, une statistique connue, etc).

Compléments des mesures usuelles

Autres mesures pour le choix des meilleurs itemsets :

- Il est possible de calculer une **distance** entre les itemsets
  ➙ Permet de faire des groupes similaires d'itemsets (quand il y en a bcp.).
- Une autre mesure pour les itemsets : **affinité**

$$aff_{\{A,B\}} = \frac{Supp(AB)}{Supp(A).Supp(B) - Supp(AB)}$$

- Bayes Facteur : BF($A \Rightarrow B$) : $\frac{\frac{P(B|A)}{P(\overline{B}|A)}}{\frac{P(B)}{P(\overline{B})}} = \frac{P(A|B)}{P(A|\overline{B})} = \frac{P(B \Rightarrow A)}{P(\overline{B} \Rightarrow A)}$ (Conséq. "A" fixe)

  ➛ Le Bayes factor est aussi appelé Le **Ratio de vraisemblance** (*Likelihood ratio*)
  ➛ De même : $Odds(B|A) = \frac{P(A|B)}{P(A|\overline{B})} \times Odds(B)$

$$Posterior\ Odds = Likelihood\ ratio \times Prior\ Odds$$

  ➛ est une mesure de qualité d'une règle $A \Rightarrow B$

- Rappel de Bayes Odd : $Odds(A \Rightarrow B) = \frac{P(B|A)}{P(\overline{B}|A)}$ (Prémisse "A" fixe).

Compléments des mesures usuelles (suite)

Autres mesures "populaires" pour les règles :

- $PS = P(X,Y) - P(X).P(Y)$ (mesure *Piatetsky - Shapiro*)
  ➙ Distance linéaire (comparer à χ^2)
- $\phi - coefficient = \frac{P(X,Y) - P(X) \times P(Y)}{\sqrt{P(X)[1-P(X)]\ P(Y)[1-P(Y)]}}$

  ➙ Pour un cas binaire : $\phi \equiv$ coef. de Corr. de Pearson
  ➙ Aussi : $\phi^2 = \frac{\chi 2}{n}$

Kappa (Chance corrected agreement)

- Le *bon agrément* (ou *value*) : similaire à un coefficient de corrélation ;
  - on l'utilise pour la similarité et la fiabilité des résultats d'un modèle.
  - ou pour comparer plusieurs modèles (ou entre les avis de deux experts)

$$Kappa = \frac{Observee - Attendue}{1 - Attendue}$$

- Kappa compare la **justesse observée** d'un modèle à la **justesse attendue** (le *hasard*) du modèle (en tenant compte de la chance ou des résultats aléatoires)

- Kappa permet de ne pas s'appuyer sur la seule justesse (*Accuracy*) d'un modèle (qu'il faudrait relativiser) :
  - P. Ex. une justesse observée de 80% est moins intéressante si on sait que la justesse attendue est de 75%

  ➛ La justesse hasardeuse est de 50% dans un cas binaire

- On étudie le Kappa dans les sous-ensembles de données (plus prometteurs)

Kappa (Chance corrected agreement) (suite)

Un exemple : soit la matrice de confusion

	chat	chien	Totaux (modèle)	
chat	10	7	10+17=17	← modèle
chien	5	8	5+8=13	← modèle
Totaux (BD)	10+5=15	7+8=15	30	
	↑ observation (BD)	↑ BD		

Les "prédicteurs" ci-dessous sont : le modèle / l'expert / la BD.
La justesse **observée** = accord modèle-vs-BD : $\frac{TP+TN}{total} = \frac{10+8}{30} = 0.6$
La justesse **attendue** (le hasard, si non fournie) : utiliser les pourcentages des chats et chiens.

Calcul de l'attendu :

1. Multiplier la marginale des chats pour un prédicteur par la marginale des chats de l'autre prédicteur puis diviser par le total des instances :
   - On avait $10+5=15$ *chats* selon la BD. et $10+7=17$ *chats* selon le modèle : $\frac{15*17}{30} = 8.5$
   - Et pour les *chiens* : $\frac{(10+5)*(5+8)}{30} = 6.5$
2. Puis additionner les deux divisé par le totale : $\frac{8.5+6.5}{30} = 0.5$

➙ D'où $Kappa = \frac{Observee - Attendue}{1 - Attendue} = \frac{0.6 - 0.5}{1 - 0.5} = 0.2$

☞ L'attendue sera toujours 0.5 si le nombre d'instances d'une classe dans la BD = le nombre d'instances de l'autre classe (ici 15 chats et 15 chiens dans la BD.).

☞ Si plus de 2 classes, faire de même avec les autres classes.

Kappa (Chance corrected agreement) (suite)

- Si la matrice de confusions était :

	chat	chien
chat	22	9
chien	7	13

➙ On aurait un $Kappa = 0.37$ à comparer ici à l'attendu $= 0.5$

En général :

☞ $kappa > 0$ veut dire : le modèle appris fait mieux que la chance (pile ou face)

$Kappa \leq 0$: désaccord (ou accord seulement au hasard)

$Kappa = 1$: max d'accord

➙ $Kappa > 0.6$ est préférable (d'autres statisticiens acceptent ≥ 0.5)

- **Contre-exemple** : dans la matrice (avec un $Kappa = 0.47$ (pas mauvais !))

	chat	chien
chat	60	125
chien	5	5000

Mais : env. 1/3 des chats (60/185) sont bien classés (le reste mal classés) et si la bonne classification des chats est important alors tout autre classifieur avec un kappa moindre mais une meilleur justesse est préférable.

N.B. : Plusieurs références (où hypothèses) sont possibles dans le calcul de Kappa dont les résultats dépendent de la prévalence et du biais (v. + loin)

Kappa : un exemple détaillé

Un exemple détaillé pour la table de contingence suivante : $K = \frac{(O_{ag} - E_{ag})}{(1 - E_{ag})}$

O_{ag} : *observed agreement* (diagonale / total) = **Accord** = **TP+TN**
E_{ag} : *expected agreement* (expected in diag / total) = **Hasard**
➙ Le **Hasard** = la probabilité d'un accord aléatoire

	ref std A	ref std B	ref std C	total
system A	13 (6.6)	4	6	23
system B	8	23 (13.1)	2	33
system C	5	9	21 (11.3)	35
Total	26	36	29	91

$O_{ag} = 57/91 = .63$ — $13 + 23 + 21 = 57$: les valeurs sur la diagonale
$E_{ag} = 31/91 = .34$ — $6.6 + 13.1 + 11.3 = 31$

31 est la somme des valeurs attendues entre parenthèse (expected) sur la diagonale

$K = (.63 - .34)/(1 - .34) = 0.43$

☞ Les détails de ces calculs sont donnés dans l'exemple suivant.

Kappa : un exemple détaillé (suite)

Un autre exemple : ici, les valeurs attendues ne sont pas données
→ on se réfère au calcul du *Hasard*.

	Ref. Std +	Ref. Std -	total
Système +	25	0	25
Système -	50	25	75
Total	75	25	100

- Calcul du Kappa :
  (1) Accord : proportion des données sur laquelle les deux (TP, TN) sont d'accord
  Accord $= O_{ag} =$ (TP+TN)/total BD   *c-à-d. : (la diagonale) / (taille BD)*

  (2) **Hasard** $= E_{ag} = \dfrac{\sum_i (total\ \ ligne\ \ i * total\ \ col\ \ i)}{(total\ \ du\ \ BD)^2}$ *si pas d'autre indication*

- Ici, on a :
  pour (1) : $\frac{25+25}{100} = 0.5$

  pour (2) : $\frac{(lig_1 * col_1) + (lig_2 * col_2)}{(total\ \ BD)^2} = \frac{(25*75)+(75*25)}{100*100} = 0.375$

→ Ce qui donne : $kappa = \frac{0.5-0.375}{1-0.375} = 0.2$

Kappa : un exemple détaillé (suite)

Remarque sur la première table précédente :

	ref std A	ref std B	ref std C	total
system A	13(6.6)	4	6	23
system B	8	23 (13.1)	2	33
system C	5	9	21 (11.3)	35
Total	26	36	29	91

<u>Rappel</u> : pour calculer E_{ag} (le Hasard), les valeurs entre parenthèses étaient données pour chaque i mais on peut les retrouver :

$$\text{Hasard} = E_{ag} = \frac{\sum_i (total\ \ ligne\ \ i * total\ \ col\ \ i)}{(total\ \ du\ \ BD)^2}$$

Ici : $\quad E_{ag} = \frac{(23*26)+(33*36)+(35*29)}{(91*91)} = 0.338 \sim 0.34$ (calculé ci-dessus)

Remarques sur Kappa

Notes sur la mesure Kappa pour 2 avis (Kappa de *Cohen*)

$$\frac{P(accord\ relatif\ des\ deux) - P(accord\ par\ hasard)}{1 - P(accord\ par\ hasard)}$$

➙ $1 - P(accord\ par\ hasard)$ représente le maximum d'accord possible.
➙ Kappa mesure l'accord entre ces deux modèles (entre 2 codeurs)

☞ Dans le cas des méthodes d'apprentissage, les 2 avis sont ceux exprimés dans la matrice de confusion où l'accord relatif des 2 modèles = $TP + TN$

- Kappa = (la différence entre l'avis d'un modèle et le hasard) divisée par (1- hasard).
- $Kappa = 1$: max d'accord
- Kappa ≤ 0 : désaccord (ou accord seulement au hasard)
- $Kappa \geq 0$: le modèle fait mieux que le hasard (pile ou face).
- Moins il y a des classes, plus *Kappa* sera grand
- Si plus de 2 avis : prendre Kappa de *Fleiss* (cf. cas général ci-dessus)

Kappa pour les règles d'association

- Pour une règle $A \Rightarrow B$ (tirée de l'itemset {A,B}), **Kappa** se décline par :

$$Kappa = \frac{P(AB)+P(\overline{AB})-P(A)P(B)-P(\overline{A})P(\overline{B})}{P(A)+P(B)-P(A)P(B)-P(\overline{A})P(\overline{B})}$$

- Sans tenir compte de $\overline{A}$ et $\overline{B}$, on simplifie par :

$$Kappa = \frac{P(AB)-P(A)P(B)}{P(A)+P(B)-2P(A)P(B)}$$

  - Où $P(A)P(B)$ est la probabilité théorique de A et de B en l'absence de toute hypothèse (de dépendance ou d'indépendance) inspirée d'un processus général de *Bernoulli*,
  - Ici $P(A)P(B)$ représente donc E_{ag} dans $Kappa = \frac{(O_{ag}-E_{ag})}{(1-E_{ag})}$

☞ Un cas particulier (où l'absence de $\overline{A}$ et $\overline{B}$ est justifié) est une BD où on a forcément A ou B dans chaque instance auquel cas $P(A) + P(B) - P(A)P(B) = 1$ et nous pouvons retrouver la forme originelle de Kappa.

Remarques sur Kappa

Kappa et la matrice de confusion :

- Soient les données d'un modèle :

Kappa statistic	*0.3108*
ROC	*0.709*

- Et la matrice de Confusion (tirée de WEKA) :

$$\begin{array}{llll} a & b & \leftarrow & \textit{classifié comme} \\ 59 & 2 & \mid & a = 0 \\ 27 & 12 & \mid & b = 1 \end{array}$$

- Ici, pour 100 instances, on a :
  $TP + TN = 59 + 12 = 71, \qquad FP + FN = 27 + 2 = 29.$

- $TP + TN$ = pourcentage de correctement classés = *justesse* simple (*accuracy*).

➙ Son inconvénient est que la valeur n'est pas pondérée par le hasard (*chance corrected*) et n'est pas **sensible à la distribution** des classes.

➙ Dans ce cas, l'aire ROC est un bon complément (ici, 0.709 pour les 2 classes).

Remarques sur Kappa (suite)

Autres remarques :

• Kappa est pondérée et mesure l'accord entre le modèle et la BD (d'apprentissage).

☞ En général, $Kappa \geq 0$: le modèle fait mieux que le hasard (pile ou face).

• En statistiques, certains considèrent que Kappa est exploitable seulement à partir de 0.6 (ou 0.7) ; en deçà, Kappa dit peu de chose.

Autres remarques :

• Les taux d'erreurs numériques (données par Weka) comme *Mean absolute error*, *Root mean squared error*, *Relative absolute error*, *Root relative squared error* sont plus utiles à la prédiction numérique qu'à la classification.

➙ Les prédictions numériques ne sont pas justes ou erronées mais leur erreur a une certaine magnitude reflétée par ces valeurs d'erreur.

☞ Voir aussi les annexes sur les compléments dans le BE1.

Remarques sur Kappa (suite)

☞ Pour calculer Kappa, on tient compte du nombre d'instances dans chaque classe pour avoir l'**attendu**.

- On compare **attendu** avec **observé**.
  ➛ On ne compare pas **Kappa** avec **Attendu** !.
  ➛ Donc, un kappa > 0 est déjà pas mal !

- Pour une même base de données, une comparaison des Kappa suiffit.

☞ On observe toujours kappa + d'autres mesures :
AUROC, matrice de confusion, différentes erreurs (cf. MSE), etc.

- On calcule souvent le *Bon Agrément* à (at) 0.4, 0.7 ou à 0.8 (on ordonne les résultats sur les probas. de succès puis on prend les meilleurs 40% 70% 80%, voir le *complément* plus loin.)

Ex : pour un modèle, on aurait at 0.4 : AUC = 0.89 ; sensitivity = specificity = 0.82
et at 0.7 : AUC = 0.96 ; sens=spec = 0.92
➛ Les premiers 40% (ou 70%) sont prometteurs (évite de prendre toute la BD.).

Les "meilleures" mesures

Tableaux des règles d'évaluation [Lenca & al - 2004]

Abrev.	Nom	Signification Probabiliste	vs. $A \Rightarrow B$
SUP	Support	$P(AB)$	
Conf	Confidence	$P(B\|A)$	$A \Rightarrow B$
BF	Bayes Factor	$\frac{P(A\|B)}{P(A\|\overline{B})}$	$\frac{B \Rightarrow A}{\overline{B} \Rightarrow A}$
CenConf	Centered Conf.	$P(B\|A) - P(B)$	
Lift	Lift	$\frac{P(B\|A)}{P(B)}$	
IG	Info. Gain	$log \frac{P(AB)}{P(A)P(B)}$	
Conv	Conviction	$\frac{P(A)P(\overline{B})}{P(A.\overline{B})}$	
ECR	Ex. & Contr. Ex ratio	$1 - \frac{P(A.\overline{B})}{P(AB)}$	
LC	Least Contradiction	$\frac{P(AB) - P(A\overline{B})}{P(B)}$	
R	Pearson Corr. Coef.	$(P(AB) - P(A)P(B))/\sqrt{P(A))P(\overline{A})P(B)P(\overline{B})}$	
SEB	Sebag & Shoenauer	$P(AB)/P(A\overline{B})$	
Kappa	Kappa Coef	$2\frac{P(AB) - P(A)P(B)}{P(A) + P(B) - 2P(A)P(B)}$	
LAP	Laplace	$P(B\|A) + \frac{1}{n.P(A)}/1 + \frac{1}{n.P(A)}$	
LOE	Loevinger	$\frac{P(B\|A) - P(B)}{1 - P(B)}$	
PS	Piatetsky-Shapiro	$n(P(AB) - P(A)P(B))$	
-ImpInd	Implication Index	$-\sqrt{n}[P(A).\overline{B} - P(A)P(\overline{B})]/\sqrt{P(A)P(\overline{B})}$	
Zhang	Zhang	$[P(AB) - P(A)P(B)]/max\{P(AB)P(\overline{B}); P(B)P(A\overline{B})\}$	

Table des mesures de qualité

Remarques :

- Beaucoup de mesures dans la littérature, adaptées selon les applications.
- Quel critère choisir pour qualifier une mesure ?
- Peut-on baser l'élagage dans *A Priori* sur ces mesures plutôt que sur la fréquence ?

#	Measure	Formula
1	ϕ-coefficient	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's (λ)	$\frac{\sum_j \max_k P(A_j,B_k)+\sum_k \max_j P(A_j,B_k)-\max_j P(A_j)-\max_k P(B_k)}{2-\max_j P(A_j)-\max_k P(B_k)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(A,\overline{B})P(\overline{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\overline{AB})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{AB})+P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{AB})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{AB})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
7	Mutual Information (M)	$\frac{\sum_i \sum_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i)\log P(A_i),-\sum_j P(B_j)\log P(B_j))}$
8	J-Measure (J)	$\max\Big(P(A,B)\log(\frac{P(B\|A)}{P(B)})+P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}),$ $P(A,B)\log(\frac{P(A\|B)}{P(A)})+P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})\Big)$
9	Gini index (G)	$\max\Big(P(A)[P(B\|A)^2+P(\overline{B}\|A)^2]+P(\overline{A})[P(B\|\overline{A})^2+P(\overline{B}\|\overline{A})^2]$ $-P(B)^2-P(\overline{B})^2,$ $P(B)[P(A\|B)^2+P(\overline{A}\|B)^2]+P(\overline{B})[P(A\|\overline{B})^2+P(\overline{A}\|\overline{B})^2]$ $-P(A)^2-P(\overline{A})^2\Big)$
10	Support (s)	$P(A,B)$
11	Confidence (c)	$\max(P(B\|A),P(A\|B))$
12	Laplace (L)	$\max\left(\frac{NP(A,B)+1}{NP(A)+2},\frac{NP(A,B)+1}{NP(B)+2}\right)$
13	Conviction (V)	$\max\left(\frac{P(A)P(\overline{B})}{P(A\overline{B})},\frac{P(B)P(\overline{A})}{P(B\overline{A})}\right)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A,B)-P(A)P(B)$
17	Certainty factor (F)	$\max\left(\frac{P(B\|A)-P(B)}{1-P(B)},\frac{P(A\|B)-P(A)}{1-P(A)}\right)$
18	Added Value (AV)	$\max(P(B\|A)-P(B),P(A\|B)-P(A))$
19	Collective strength (S)	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})}\times\frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)}\max(P(B\|A)-P(B),P(A\|B)-P(A))$

Complément tableau des mesures

- Expression des meures seulement en fonction des nombres d'occurrences des itemsets [Hiep Xuan Huynh1 & all, 2008-VNU].
  - ➙ Pour la règle $X \Rightarrow Y$, $n_X = card(X)$, $\overline{Y}$ = le complément de Y et n = taille BD.

N°	INTERESTINGNESS MEASURES	$f(n, n_X, n_Y, n_{X\overline{Y}})$
1	Causal Confidence	$1-\frac{1}{2}(\frac{1}{n_X}+\frac{1}{n_{\overline{Y}}})n_{X\overline{Y}}$
2	Causal Confirm	$\frac{n_X+n_{\overline{Y}}-4n_{X\overline{Y}}}{n}$
3	Causal Confirmed-Confidence	$1-\frac{1}{2}(\frac{3}{n_X}+\frac{1}{n_{\overline{Y}}})n_{X\overline{Y}}$
4	Causal Support	$\frac{n_X+n_{\overline{Y}}-2n_{X\overline{Y}}}{n}$
5	Collective Strength	$\frac{(n_X+n_{\overline{Y}}-2n_{X\overline{Y}})(n_X n_{\overline{Y}}+n_{\overline{X}}n_Y)}{(n_X n_Y+n_{\overline{X}}n_{\overline{Y}})(n_{\overline{X}Y}+n_{X\overline{Y}})}$
6	Confidence	$1-\frac{n_{X\overline{Y}}}{n_X}$
7	Conviction	$\frac{n_X n_{\overline{Y}}}{n n_{X\overline{Y}}}$

8	Cosine	$\frac{n_x - n_{x\bar{y}}}{\sqrt{n_x n_y}}$
9	Dependency	$\left\lvert \frac{n_{\bar{y}}}{n} - \frac{n_{x\bar{y}}}{n_x} \right\rvert$
10	Descriptive Confirm	$\frac{n_x - 2n_{x\bar{y}}}{n}$
11	Descriptive Confirmed-Confidence / Ganascia	$1 - 2\frac{n_{x\bar{y}}}{n_x}$
12	EII (α=1)	$\sqrt{\varphi \times I^{\frac{1}{2\alpha}}}$
13	EII (α=2)	$\sqrt{\varphi \times I^{\frac{1}{2\alpha}}}$
14	Example & Contra-Example	$1 - \frac{n_{x\bar{y}}}{n_x - n_{x\bar{y}}}$
15	F-measure	$\frac{2(n_x - n_{x\bar{y}})}{n_x + n_y}$
16	Gini-index	$\frac{(n_x - n_{x\bar{y}})^2 + n_{x\bar{y}}^2}{nn_x} + \frac{n_{\bar{x}y}^2 + (n_{\bar{y}} - n_{x\bar{y}})^2}{nn_{\bar{x}}} - \frac{n_y^2}{n^2} - \frac{n_{\bar{y}}^2}{n^2}$
17	II	$1 - \sum_{k=\max(0,n_x-n_y)}^{n_{x\bar{y}}} \frac{C_{n_y}^{n_x-k} C_{n_{\bar{y}}}^{k}}{C_n^{n_x}}$

18	Implication index	$\frac{n_{x\bar{y}} - \frac{n_x n_{\bar{y}}}{n}}{\sqrt{\frac{n_x n_{\bar{y}}}{n}}}$
19	IPEE	$1 - \frac{1}{2^{n_x}} \sum_{k=0}^{n_{x\bar{y}}} C_{n_x}^{k}$
20	Jaccard	$\frac{n_x - n_{x\bar{y}}}{n_y + n_{x\bar{y}}}$
21	J-measure	$\frac{n_x - n_{x\bar{y}}}{n} \log_2 \frac{n(n_x - n_{x\bar{y}})}{n_x n_y} + \frac{n_{x\bar{y}}}{n} \log_2 \frac{n n_{x\bar{y}}}{n_x n_{\bar{y}}}$
22	Kappa	$\frac{2(n_x n_{\bar{y}} - n n_{x\bar{y}})}{n_x n_{\bar{y}} + n_{\bar{x}} n_y}$
23	Klosgen	$\sqrt{\frac{n_x - n_{x\bar{y}}}{n}} (\frac{n_{\bar{y}}}{n} - \frac{n_{x\bar{y}}}{n_x})$
24	Laplace	$\frac{n_x + 1 - n_{x\bar{y}}}{n_x + 2}$
25	Least Contradiction	$\frac{n_x - 2n_{x\bar{y}}}{n_y}$
26	Lerman	$\frac{n_x - n_{x\bar{y}} - \frac{n_x n_y}{n}}{\sqrt{\frac{n_x n_y}{n}}}$
27	Lift / Interest factor	$\frac{n(n_x - n_{x\bar{y}})}{n_x n_y}$
28	Loevinger / Certainty factor	$1 - \frac{n n_{x\bar{y}}}{n_x n_{\bar{y}}}$

29	Mutual Information	$\dfrac{\frac{n_x - n_{x\bar{y}}}{n}\log(\frac{n(n_x - n_{x\bar{y}})}{n_x n_y}) + \frac{n_{x\bar{y}}}{n}\log(\frac{nn_{x\bar{y}}}{n_x n_{\bar{y}}}) + \frac{n_{\bar{x}y}}{n}\log(\frac{nn_{\bar{x}y}}{n_{\bar{x}} n_y}) + \frac{n_{\bar{x}\bar{y}}}{n}\log(\frac{nn_{\bar{x}\bar{y}}}{n_{\bar{x}} n_{\bar{y}}})}{\min(-(\frac{n_x}{n}\log(\frac{n_x}{n}) + \frac{n_{\bar{x}}}{n}\log(\frac{n_{\bar{x}}}{n})), -(\frac{n_y}{n}\log(\frac{n_y}{n}) + \frac{n_{\bar{y}}}{n}\log(\frac{n_{\bar{y}}}{n})))}$
30	Odd Multiplier	$\dfrac{(n_x - n_{x\bar{y}})n_{\bar{y}}}{n_y n_{x\bar{y}}}$
31	Odds Ratio	$\dfrac{(n_x - n_{x\bar{y}})(n_{\bar{y}} - n_{x\bar{y}})}{n_{x\bar{y}} n_{\bar{x}y}}$
32	Pavillon / Added Value	$\dfrac{n_{\bar{y}}}{n} - \dfrac{n_{x\bar{y}}}{n_x}$
33	Phi-Coefficient	$\dfrac{n_x n_{\bar{y}} - nn_{x\bar{y}}}{\sqrt{n_x n_y n_{\bar{x}} n_{\bar{y}}}}$
34	Putative Causal Dependency	$\dfrac{3}{2} + \dfrac{4n_x - 3n_y}{2n} - (\dfrac{3}{2n_x} + \dfrac{2}{n_{\bar{y}}})n_{x\bar{y}}$
35	Rule Interest	$\dfrac{n_x n_{\bar{y}}}{n} - n_{x\bar{y}}$
36	Sebag & Schoenauer	$\dfrac{n_x}{n_{x\bar{y}}} - 1$
37	Support	$\dfrac{n_x - n_{x\bar{y}}}{n}$
38	TIC	$\sqrt{TI(X \to Y) \times TI(\bar{Y} \to X)}$
39	Yule's Q	$\dfrac{n_x n_{\bar{y}} - nn_{x\bar{y}}}{n_x n_{\bar{y}} + (n_y - n_{\bar{y}} - 2n_x)n_{x\bar{y}} + 2n_{x\bar{y}}^2}$
40	Yule's Y	$\dfrac{\sqrt{(n_x - n_{x\bar{y}})(n_{\bar{y}} - n_{x\bar{y}})} - \sqrt{n_{x\bar{y}} n_{\bar{x}y}}}{\sqrt{(n_x - n_{x\bar{y}})(n_{\bar{y}} - n_{x\bar{y}})} + \sqrt{n_{x\bar{y}} n_{\bar{x}y}}}$

Addendum : Propriétés d'une bonne mesure

Un exemple d'application des mesures :

- On calcule toutes les mesures d'intérêt pour 10 exemples (la table de contingence).
- Voir le tableau suivant des mesures ../..

Example	f_{11}	f_{10}	f_{01}	f_{00}
E1	8123	83	424	1370
E2	8330	2	622	1046
E3	9481	94	127	298
E4	3954	3080	5	2961
E5	2886	1363	1320	4431
E6	1500	2000	500	6000
E7	4000	2000	1000	3000
E8	4000	2000	2000	2000
E9	1720	7121	5	1154
E10	61	2483	4	7452

../..

Addendum : Propriétés d'une bonne mesure (suite)

- Le tableau ci-dessous contient le rang (de 1 à 10, 1= meilleur). Si la valeur 1 dans une colonne, alors la mesure est la mieux adaptée à l'exemple ➙ (la valeur 10 = pire).

#	ϕ	λ	α	Q	Y	κ	M	J	G	s	c	L	V	I	IS	PS	F	AV	S	ζ	K
E1	1	1	3	3	3	1	2	2	1	3	5	5	4	6	2	2	4	6	1	2	5
E2	2	2	1	1	1	2	1	3	2	2	1	1	1	8	3	5	1	8	2	3	6
E3	3	3	4	4	4	3	3	8	7	1	4	4	6	10	1	8	6	10	3	1	10
E4	4	7	2	2	2	5	4	1	3	6	2	2	2	4	4	1	2	3	4	5	1
E5	5	4	8	8	8	4	7	5	4	7	9	9	9	3	6	3	9	4	5	6	3
E6	6	6	7	7	7	7	6	4	6	9	8	8	7	2	8	6	7	2	7	8	2
E7	7	5	9	9	9	6	8	6	5	4	7	7	8	5	5	4	8	5	6	4	4
E8	8	9	10	10	10	8	10	10	8	4	10	10	10	9	7	7	10	9	8	7	9
E9	9	9	5	5	5	9	9	7	9	8	3	3	3	7	9	9	3	7	9	9	8
E10	10	8	6	6	6	10	5	9	10	10	6	6	5	1	10	10	5	1	10	10	7

➙ Pas de mesure universelle mais la même doit être utilisée pour une même BD pour qualifier différentes modèles obtenus.

Addendum : Propriétés d'une bonne mesure (suite)

Constat : il n'y a pas UNE mesure adaptée à tout !

- Propriétés d'une bonne mesure M (selon *Piatetsky-Shapiro*) :
  - M(A,B)=0 si A et B sont statistiquement indépendantes
  - M(A,B) doit croître de façon monotone avec P(A,B) lorsque P(A) et P(B) restent inchangées (hors cas d'indépendance)
  - M(A,B) doit décroître de façon monotone avec P(A) (ou P(B)) lorsque P(A,B) et P(B) (ou P(A)) restent inchangées.
- Sur l'indépendance de A et de B dans la règle $A \Rightarrow B$:
  - ➙ On remarque le rôle de $P(A,B) - P(A).P(B)$ comme dans :

$$\phi - coefficient = \frac{P(X,Y) - P(X) \times P(Y)}{\sqrt{P(X)[1 - P(X)]\ P(Y)[1 - P(Y)]}}$$

Addendum : Propriétés d'une bonne mesure (suite)

Propriété d'une bonne mesure : permutation des variables

	B	$\bar{\mathbf{B}}$
A	p	q
$\bar{\mathbf{A}}$	r	s

	A	$\bar{\mathbf{A}}$
B	p	r
$\bar{\mathbf{B}}$	q	s

- Pour la mesure M choisie, est-ce que M(A,B) = M(B,A) ?
- La permutation est courante : observer *(A vs. B)* ou *(B vs. A)*.

  ➙ <u>Mesures symétriques :</u> (si oui)
  support (fréquence), lift, collective strength, cosinus, Jaccard, etc.
  ➙ Mesures asymétriques :
  confiance, conviction, Laplace, J-measure, etc.

(Voir le tableau précédent pour les mesures.)

Addendum : Propriétés d'une bonne mesure (suite)

Propriété d'une bonne mesure : échelle de ligne/colonne

- Ex taille / genre :

	Male	Femelle	
Grand	2	3	5
Petit	1	4	5
	3	7	10

	Male	Femelle	
Grand	4	30	34
Petit	2	40	42
	6	70	76

$\Downarrow$ $\Downarrow$

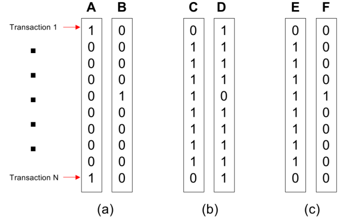
$2\times$ $10\times$

➙ Les (règles de) associations sous-jacentes doivent être indépendantes du nombre relatif de mâle/femelle dans la BD.

➙ Mais l'intervalle de confiance sera pas le même, même si les associations seront inchangées.

Addendum : Propriétés d'une bonne mesure (suite)

Propriété d'une bonne mesure : inversion



(b) est l'inverse de (a) : 0 ➙ 1 et 1 ➙ 0 ; (c) est l'inverse en nombre de (a)

- Propriété d'inversion si l'échange de f_{11} avec f_{00} et l'échange de f_{10} avec f_{01} ne modifent pas la mesure :
  - ➙ Exemple de ces mesures :
    ϕ -coefficient, kappa, collective strength, Odds ratio

Addendum : Propriétés d'une bonne mesure (suite)

Exemple de mesure invariante à l'inversion : ϕ-coefficient

Rappel : $\phi - coefficient = \dfrac{P(X,Y) - P(X) \times P(Y)}{\sqrt{P(X)[1-P(X)]\ P(Y)[1-P(Y)]}}$

- ϕ-coefficient est analogue au **coefficient de corrélation** pour les variables continues.

	Y	$\overline{\text{Y}}$	
X	60	10	70
$\overline{\text{X}}$	10	20	30
	70	30	100

	Y	$\overline{\text{Y}}$	
X	20	10	30
$\overline{\text{X}}$	10	60	70
	30	70	100

$$\phi = \frac{0.6 - 0.7 \times 0.7}{\sqrt{0.7 \times 0.3 \times 0.7 \times 0.3}} = 0.5238$$

$$\phi = \frac{0.26 - 0.3 \times 0.3}{\sqrt{0.7 \times 0.3 \times 0.7 \times 0.3}} = 0.5238$$

➙ Le ϕ-coefficient est identique pour les deux tables !

Addendum : Propriétés d'une bonne mesure (suite)

Propriété d'une bonne mesure : Addition Nulle

- Exemple en TextMining : une BD sur *data mining* (corrélation des mots *data* et *mining*) ;
  - ➞ On a ajoute des couples de mots sur la *peche* et *eaux-troubles* (en f_{00})

	B	$\overline{\mathbf{B}}$
A	p	q
$\overline{\mathbf{A}}$	r	s

	B	$\overline{\mathbf{B}}$
A	p	q
$\overline{\mathbf{A}}$	r	s + k

- Si la mesure ne change pas, on a la propriété *Addition Nulle*.
- C'est souvent le cas en analyse de documents et en analyse de caddie.
  - ➞ Sinon, l'ajout de beaucoup de faux documents**peut noyer les vrais** !
- Exemple de ces mesures : *Cosinus* et *Jaccard*.

Notes sur les mesures d'intérêt (qualité)

Il y a deux sortes de mesures :

Mesures Objectives :

Par exemple : construire les tables de contingence et calculer (toutes) les mesures pour les associations trouvées.

➙ On peut mener différentes stratégies d'apprentissage,
➙ Vérifier que la meilleure mesure reste meilleure dans les différents modèles d'apprentissages.

Mesures Subjectives :

Dépend de l'expertise de l'utilisateur, mesure la "nouveauté" p/r à l'expertise.
➙ Classer les motifs selon l'interprétation de l'utilisateur,
➙ *Un motif est subjectivement intéressant* s'il **contredit** l'attente (prévision) de l'utilisateur !

../..

L'attente de l'utilisateur

- Besoin de modéliser l'attente de l'utilisateur :

$+$: motifs que l'on croit fréquents
$-$: motifs que l'on croit NON fréquents
□ : motifs trouvés fréquents
◯ : motifs trouvés NON fréquents
⊞ ⊖ : motifs prévisibles
⊟ ⊕ : motifs NON prévisibles

➞ Besoin de combiner cette attente avec l'évidence tirée des données

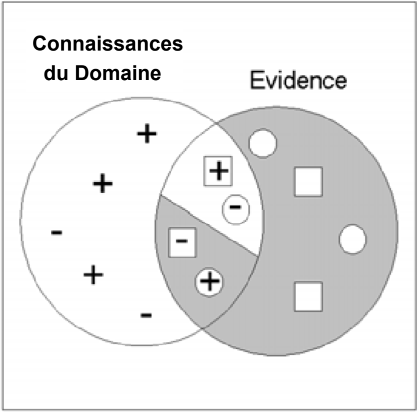


Table des matières

Table des matières (suite)

Table des matières (suite)