

Réseaux bayésiens – aGrUM

OpenTurns 2010

Pierre-Henri Wuillemin

Décision – DESIR –  – 

Modèle probabiliste complexe

La représentation probabiliste d'un système est caractérisé par un univers Ω où chaque $\omega \in \Omega$ est un état du système.

Un **système complexe** est caractérisé par un univers Ω de grande taille.

➡ Définition (Modèle factorisé)

- Une *modèle probabiliste (sur Ω)* est **factorisé** lorsqu'il existe une famille $\mathfrak{X} = (X_i)_{i < n}$ de variables aléatoires sur Ω telle que chaque $\omega \in \Omega$ est caractérisé de manière unique par les valeurs $(X_i(\omega))_{i < n}$.
- Dans un modèle factorisé, une probabilité sur Ω sera donc représentée par une loi **jointe** des variables de \mathfrak{X} .

$$\forall \omega \in \Omega, p(\omega) = p(X_1 = X_1(\omega), X_2 = X_2(\omega), \dots, X_n = X_n(\omega))$$



Explosion combinatoire : Si toutes les variables sont binaires, un système factorisé en n variables nécessitent $\approx 2^n$ valeurs !

La factorisation peut-elle permettre d'améliorer la compacité? Grâce à l'**indépendance conditionnelle** !!

$$2^3 \quad p(X, Y, Z) = p(X) \cdot p(Y | X) \cdot p(Z | X, Y)$$

$$2 + 2^2 + 2^3$$

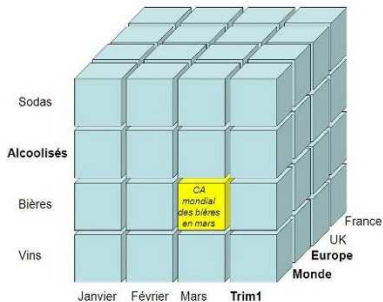
Avec $X \perp\!\!\!\perp Y$ et $Z \perp\!\!\!\perp X, Y$:

$$2^3 \quad p(X, Y, Z) = p(X) \cdot p(Y) \cdot p(Z)$$

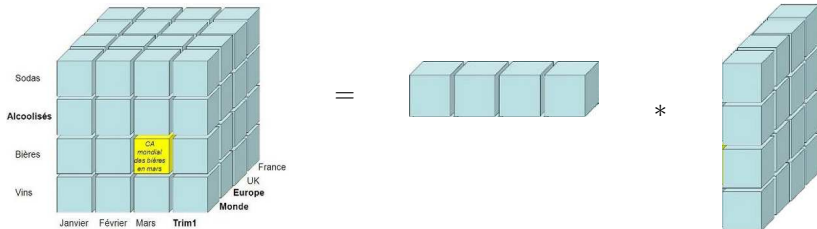
$$2 + 2 + 2$$

Modèles complexes décomposable

| | | Janvier | Février | Mars | Trim1 |
|---------------------|--------------|------------|------------|------------|-------------|
| France | Bières | 70 | 70 | 80 | 220 |
| | Vins | 100 | 110 | 90 | 300 |
| | Total | 170 | 180 | 170 | 520 |
| UK | Bières | 250 | 220 | 240 | 710 |
| | Vins | 50 | 40 | 60 | 150 |
| | Total | 300 | 260 | 300 | 860 |
| Total Europe | | 470 | 440 | 470 | 1380 |



Comment voir dans ce modèle que *Mois* $\perp\!\!\!\perp$ $\{Pays, Boisson\}$?



Modèle d'indépendances

Il serait intéressant de fournir un outil basé sur les variables aléatoires du modèle, qui permettrait de manipuler les indépendances conditionnelles de manière plus naturelle qu'indirectement dans l'hyper-cube de la loi jointe.

Quel objet manipule-t-on lorsqu'on parle de l'**ensemble** des indépendances conditionnelles de $p()$?

➡ Définition (Modèle d'indépendances – Séparabilité)

Soit $\mathfrak{I} \subset \mathcal{P}(\mathfrak{X}) \times \mathcal{P}(\mathfrak{X}) \times \mathcal{P}(\mathfrak{X})$. On nomme \mathfrak{I} un *modèle d'indépendance*.

$\forall U, V, W \subset \mathfrak{X}$, on dit que U et V sont séparés par W ($\ll U \diamond V \mid W \gg_{\mathfrak{I}}$) si et seulement si $(U, V, W) \in \mathfrak{I}$.

Relation entre \mathfrak{I} et p

L'ensemble $\mathfrak{I}_p = \{(U, V, W) \in \mathcal{P}(\mathfrak{X}) \times \mathcal{P}(\mathfrak{X}) \times \mathcal{P}(\mathfrak{X}), U \perp\!\!\!\perp V \mid W\}$ est un modèle d'indépendance.

$$U \perp\!\!\!\perp V \mid W \iff \ll U \diamond V \mid W \gg_{\mathfrak{I}_p}$$

Semi-graphoïde et graphoïde - définitions

➤ Définition (semi-graphoïde)

Un modèle d'indépendance \mathfrak{I} est un semi-graphoïde s'il satisfait $\forall A, B, S, P \subset \mathfrak{X}$:

- 1 Indépendance triviale $\lll A \diamond \emptyset \mid S \ggg_{\mathfrak{I}}$
- 2 Symétrie $\lll A \diamond B \mid S \ggg_{\mathfrak{I}} \Rightarrow \lll B \diamond A \mid S \ggg_{\mathfrak{I}}$
- 3 Décomposition $\lll A \diamond (B \cup P) \mid S \ggg_{\mathfrak{I}} \Rightarrow \lll A \diamond B \mid S \ggg_{\mathfrak{I}}$
- 4 Union faible $\lll A \diamond (B \cup P) \mid S \ggg_{\mathfrak{I}} \Rightarrow \lll A \diamond B \mid (S \cup P) \ggg_{\mathfrak{I}}$
- 5 Contraction $\left\{ \begin{array}{l} \lll A \diamond B \mid (S \cup P) \ggg_{\mathfrak{I}} \\ \lll A \diamond P \mid S \ggg_{\mathfrak{I}} \end{array} \right\} \Rightarrow \lll A \diamond (B \cup P) \mid S \ggg_{\mathfrak{I}}$

➤ Définition (graphoïde)

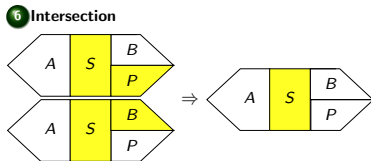
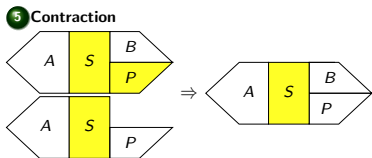
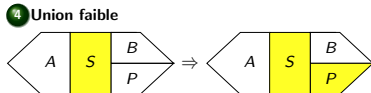
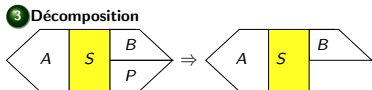
Un modèle d'indépendance \mathfrak{I} est un graphoïde s'il satisfait $\forall A, B, S, P \subset \mathfrak{X}$:

\mathfrak{I} est un semi-graphoïde

- 6 Intersection $\left\{ \begin{array}{l} \lll A \diamond B \mid (S \cup P) \ggg_{\mathfrak{I}} \\ \lll A \diamond P \mid (S \cup B) \ggg_{\mathfrak{I}} \end{array} \right\} \Rightarrow \lll A \diamond (B \cup P) \mid S \ggg_{\mathfrak{I}}$

Semi-graphoïde et graphoïde - représentation des axiomes

3 Décomposition $\ll A \diamond (B \cup P) \mid S \gg_{\mathcal{J}} \Rightarrow \ll A \diamond B \mid S \gg_{\mathcal{J}}$
4 Union faible $\ll A \diamond (B \cup P) \mid S \gg_{\mathcal{J}} \Rightarrow \ll A \diamond B \mid (S \cup P) \gg_{\mathcal{J}}$
5 Contraction $\left\{ \begin{array}{l} \ll A \diamond B \mid (S \cup P) \gg_{\mathcal{J}} \\ \ll A \diamond P \mid S \gg_{\mathcal{J}} \end{array} \right\} \Rightarrow \ll A \diamond (B \cup P) \mid S \gg_{\mathcal{J}}$
6 Intersection $\left\{ \begin{array}{l} \ll A \diamond B \mid (S \cup P) \gg_{\mathcal{J}} \\ \ll A \diamond P \mid (S \cup B) \gg_{\mathcal{J}} \end{array} \right\} \Rightarrow \ll A \diamond (B \cup P) \mid S \gg_{\mathcal{J}}$

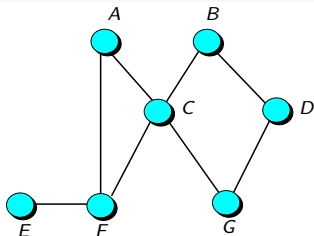


Semi-graphoïde et graphoïde - Utilisation

Théorème (loi de probabilité et graphoïde)

\mathfrak{I}_p possède une structure de semi-graphoïde.

Si $p > 0$ alors \mathfrak{I}_p possède une structure de graphoïde.



Théorème (graphe non orienté et graphoïde)

Soit un graphe $G = (X, \mathfrak{E})$,

$\forall U, V, W \subset X, \langle U | W | V \rangle_G$ indique que toute chaîne d'un nœud de U vers un nœud de V contient forcément un nœud de W .

Alors $\{\langle U | W | V \rangle_G, U, V, W \subset X\}$ possède une structure de graphoïde.

Et bien voilà !!! On va utiliser un graphe pour représenter les indépendances conditionnelles de p !!!

Modèle graphique

➤ Définition (Modèle graphique)

Un modèle graphique est un modèle probabiliste factorisé qui se sert d'un graphe entre les variables aléatoires pour représenter des indépendances conditionnelles.

Est-ce que ça se passe bien ? Peut-on toujours avoir $(X \perp\!\!\!\perp Y | Z)_p \Leftrightarrow \langle X | Z | Y \rangle_G$?

➤ Définition (I-map, D-map, P-map, graphe-isomorphisme)

soit $G = (\mathcal{X}, \mathcal{E})$ un graphe et une loi de probabilité p .

G est une **D-dependency-map** de p ssi $(X \perp\!\!\!\perp Y | Z)_p \Rightarrow \langle X | Z | Y \rangle_G$.

G est une **I-independency-map** de p ssi $(X \perp\!\!\!\perp Y | Z)_p \Leftarrow \langle X | Z | Y \rangle_G$.

G est une **P-perfect-map** de p ssi $(X \perp\!\!\!\perp Y | Z)_p \Leftrightarrow \langle X | Z | Y \rangle_G$.

Un loi de probabilité p est dite **graphe-isomorphe** si et seulement s'il existe un graphe G qui soit une **P-map** de p .

- Le graphe vide, sans arc est une **D-map** de toute distribution p .
- Le graphe complet est une **I-map** de toute distribution p .

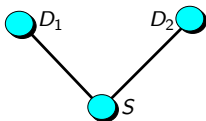
Modèle graphique non orienté : exemple 1

exemple 1

Soit un modèle probabiliste du système composé de 3 variables aléatoires : le tirage de deux dé D_1 , D_2 et $S = D_1 + D_2$ qui est la somme des tirages des 2 dés.

indépendances et dépendances du modèle de l'exemple 1

- $D_1 \not\perp S$ et $D_2 \not\perp S$
- $D_1 \perp D_2$ mais $D_1 \not\perp D_2 | S$



Modèle graphique non orienté : exemple 2

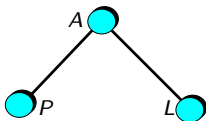
exemple 2

Dans un sondage, on s'aperçoit qu'il y a une forte corrélation entre l'aptitude à lire d'un individu et sa pointure...

On s'aperçoit rapidement que l'âge de l'individu est la variable qui explique cette corrélation bizarre.

indépendances et dépendances du modèle de l'exemple 2

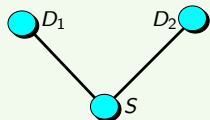
- $L \not\perp A$ et $P \not\perp A$
- $L \not\perp P$ mais $L \perp P | A$



Modèle graphique orienté

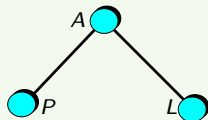
exemple 1

$D_1 \perp\!\!\!\perp D_2$ mais $D_1 \not\perp\!\!\!\perp D_2 | S$



exemple 2

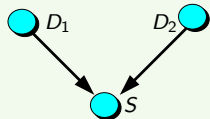
$L \not\perp\!\!\!\perp P$ mais $L \perp\!\!\!\perp P | A$



Lever l'ambiguïté en ajoutant de l'information qualitative sur les arcs : **l'orientation**.

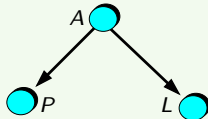
exemple 1 orienté

$D_1 \perp\!\!\!\perp D_2$ mais $D_1 \not\perp\!\!\!\perp D_2 | S$



exemple 2 orienté

$L \not\perp\!\!\!\perp P$ mais $L \perp\!\!\!\perp P | A$



Reste à donner un critère de séparation sur les graphes orientés : **la d-séparation**.

Réseau bayésien et propriétés de Markov

➤ Définition (réseau bayésien)

Soit un graphe \vec{G} , muni de la d -séparation. Si \vec{G} est I-map d'une loi p alors \vec{G} est un **réseau bayésien** pour p .

➤ Définition (Propriété de Markov globale)

\vec{G} vérifie la PMG pour $p \Leftrightarrow \forall A, B, S \subset \mathfrak{X}, \langle A | S | B \rangle_{\vec{G}} \Rightarrow A \perp\!\!\!\perp B | S$.

i.e. \vec{G} est une I-map pour p .

➤ Définition (Propriété de Markov locale)

\vec{G} vérifie la PML pour $p \Leftrightarrow \forall x \in \mathfrak{X}, \{x\} \perp\!\!\!\perp \text{nd}(x) | \Pi_x$.

où $\text{nd}(x)$ représente les nœuds non descendants de x et Π_x ses parents.

Réseau bayésien et propriétés de Markov (2)

Théorème

$$PMG \iff PML$$

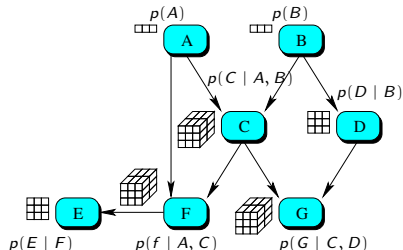
Un graphe est un réseau bayésien pour p si et seulement si chaque nœud est indépendant de ses non-descendants, conditionnellement à ses parents pour p .

Théorème (Factorisation récursive)

Soit $\vec{G} = (\mathfrak{X}, \mathfrak{E})$ un réseau bayésien pour p , alors :

$$p(\mathfrak{X}) = \prod_{X \in \mathfrak{X}} p(X \mid \Pi_X)$$

réseau bayésien : exemple et définition



$$p(A, B, C, D, E, F, G,) = ?$$
$$p(A) \cdot p(B)$$
$$\cdot p(C | A, B) \cdot p(D | B) \cdot p(F | A, C)$$
$$\cdot p(E | F) \cdot p(G | C, D)$$

Tout se passe comme si l'information était localisée dans les nœuds !

$P(A, B, C, D, E, F, G) : 3^7 = 2187$ paramètres vs **105 paramètres dans le BN !**

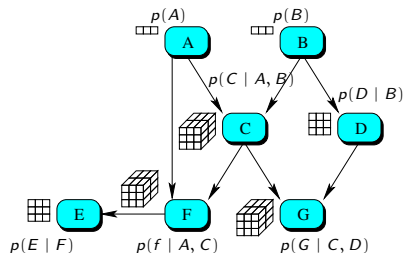
➡ Définition (Réseau bayésien (BN))

Un réseau bayésien est une représentation compacte d'une distribution de probabilité sur un ensemble de variables aléatoires. Il s'appuie sur un graphe orienté sans circuit (DAG) pour représenter son modèle d'indépendance.

La décomposition de la loi jointe suivant le graphe s'écrit :

$$P(\mathfrak{X}) = \prod_i P(X_i | \Pi_i)$$

Utilisations des BNs : inférences probabilistes



diagnostic : $P(A | F)$

- *diagnostic de panne*
- *sûreté de fonctionnement*
- *filtrage de spams*

prédiction $P(E | B, A)$

- *Simulation de process (industriels)*
- *prévisions boursières*
- *modélisation de joueurs*

Autres tâches

- *Cas le plus probable (MPE) : $\arg \max P(\mathcal{X} | D)$*
- *Analyse de sensibilité, information mutuelle, etc.*
- *Troubleshooting : $\arg \max \frac{P(\cdot)}{C(\cdot)}$*

Application : classification bayésienne

Soient deux v.a. X (de dimension d) discrète et Y (de dimension 1) discrète (*pas forcément binaire*).

Sur une base d'apprentissage (supervisé) Π_a , on peut estimer les probabilités par des fréquences pour $P(X, Y)$.

Classification

Pour une instantiation x de X , on cherche à prédire sa classe (valeur de Y) : \hat{y} .

1 Maximum de vraisemblance (ML)

$$\hat{y} = \arg \max_{y_i} P(x | y_i)$$

2 Maximum a posteriori (MAP)

$$\hat{y} = \arg \max_{y_i} P(y_i | x) = \arg \max_{y_i} P(y_i) \cdot P(x | y_i)$$

D'après la règle de Bayes, $P(Y | X) \propto P(X | Y) \cdot P(Y)$, on comprend que l'intérêt du MAP est de prendre en compte un *a priori* sur la fréquence de chaque classe.



Il peut être difficile d'obtenir ces distributions.

Particulièrement : $P(X | Y)$ peut demander beaucoup d'observation !!

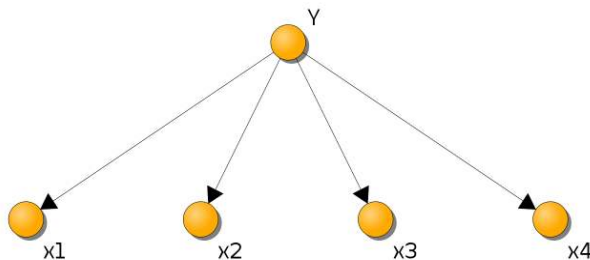
classification bayésienne (2) : Naïf Bayes

Comment calculer $P(X | Y)$?

Classifieur bayésien naïf

$$\forall k \neq l, X^k \perp\!\!\!\perp X^l | Y \quad \text{et} \quad P(x, y) = P(y) \cdot \prod_{k=1}^d P(x^k | y)$$

Cette hypothèse est très forte. Elle a peu de chance de s'avérer exacte dans un cas réel. Néanmoins cette approximation donne des résultats souvent satisfaisants.

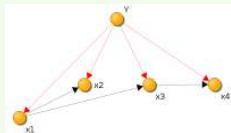


- Estimation des paramètres : trivial (si Π_a sans valeurs manquantes)
- ML : $\prod_{k=1}^d P(x^k | y) \dots$
- MAP : $P(y | x_1, \dots, x_d)$: **inférence dans le BN !**

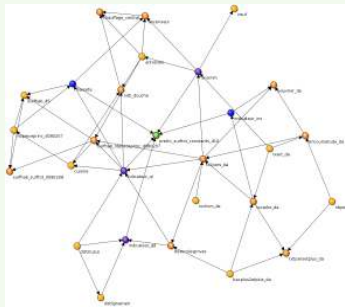
Classification bayésienne (3) : modèles plus complexes

Tree-Augmented Naive Models

Toute variable X_i peut avoir un parent autre que Y (mais un seul!).



Réseau bayésien complet



Dans un BN composé de Y et (X_i) , calculer $P(Y | X_1, \dots, X_n)$.

Note : on n'a pas besoin de tous les X_i : **Markov Blanket $MB(.)$** .

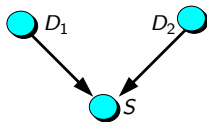
$$P(Y | X) = P(Y | MB(Y))$$

Autres modèles

Modèle d'indépendance

- $D_1 \perp\!\!\!\perp D_2$

(donc $D_1 \not\perp\!\!\!\perp D_2 | S$)

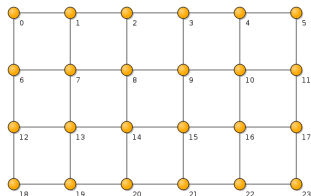


Modèle d'indépendance

- $A \perp\!\!\!\perp B | \{C, D\}$
- $C \perp\!\!\!\perp D | \{A, B\}$

?

Réseaux de Markov



- Séparation dans les réseaux de Markov :
séparation dans les graphes non-orientés.
- Propriété de Markov locale dans les réseaux de Markov :
 $X \perp\!\!\!\perp \text{non-voisin}(X) \mid \text{voisin}(X)$

➔ Définition (Réseau de Markov)

*Un réseau de Markov est une représentation compacte d'une distribution de probabilité sur un ensemble de variables aléatoires. Il s'appuie sur un **graphe non-orienté** pour représenter son modèle d'indépendance.*

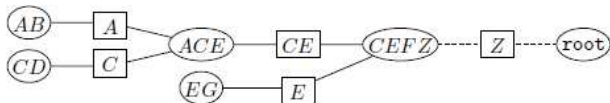
La décomposition de la loi jointe suivant le graphe s'écrit :

$$P(\mathfrak{X}) = \frac{1}{Z} \cdot \prod_{C \in \text{clique}(G)} \Phi(C)$$



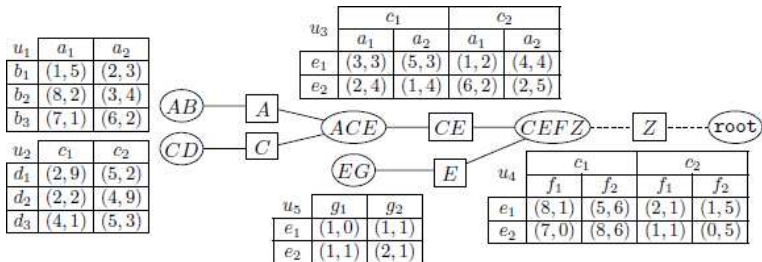
aGrUM : a library for decision
graphical models

Modèle graphiques

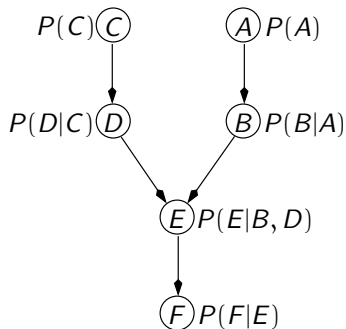


Formalisme permettant de manipuler :

- des lois de probabilité : réseaux bayésiens
- des utilités : réseaux GAI
- des contraintes : CSP



Motivations pour aGrUM



- gestion des graphes
(orienté, non-orienté, mixte)
- algorithme efficace sur les graphes
(e.g., triangulations)
- gestion des hyper-matrices $M(X_1, \dots, X_n)$
(quelque soit le type des paramètres)
- algorithme efficace sur les hyper-matrices
(e.g., produit, projection, normalisation, etc.)

librairie aGrUM

aGrUM

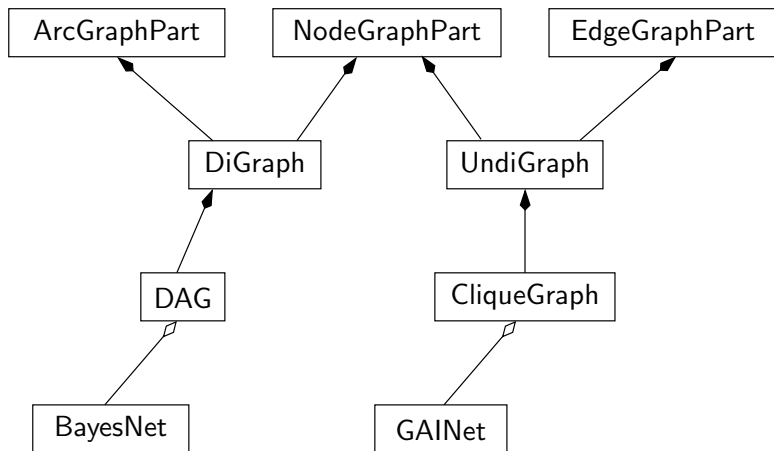
- aGrUM = a Graphical Universal Model
- aGrUM = librairie C++ pour des applications incluant des modèles graphiques
- License : (L)GPL
- publique : <http://agrums.lip6.fr>
- cross-plateforme (**linux**/mac/windows)

- structures de données de base adaptées et sécurisées
- gestion légère et efficace des graphes
- gestion efficace des hyper-matrices
- implémentation générique mais aussi implémentation efficace
- modèle graphique mais aucune interaction/dépendance avec une GUI
- Outils d'aide à la recherche
(e.g. générateurs aléatoire uniforme de BNs, introspection des classes, etc.)
- mais aussi outils d'aide à l'intégration
(e.g. *listener*, *pyAgrum*, BIF reader, etc.)

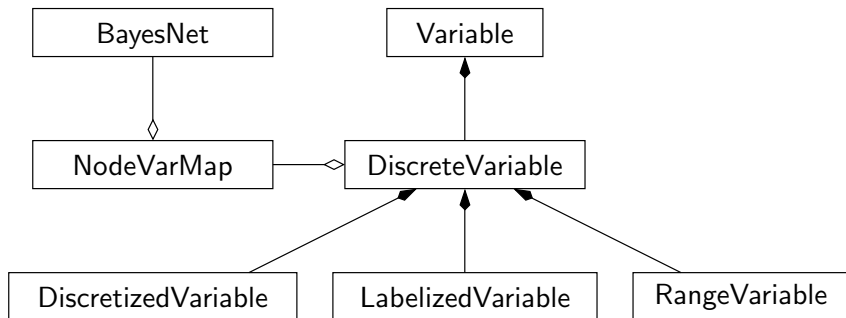
aGrUM pour les BNs

- Plusieurs types de variables aléatoires discrètes
- Plusieurs type de CPTs (array, parse, noisyOR, aggregator, etc.)
- Plusieurs types d'inférences probabilistes
 - Shafer-Shenoy
 - Value Elimination
 - Lazy propagation
 - Recursive conditioning (en cours d'intégration)
 - Gibbs sampling
 - GBP (en cours)
- Plusieurs type de triangulations / constructions de l'arbre de jonction, etc.
- Apprentissages structures / paramètres (en cours d'intégration).

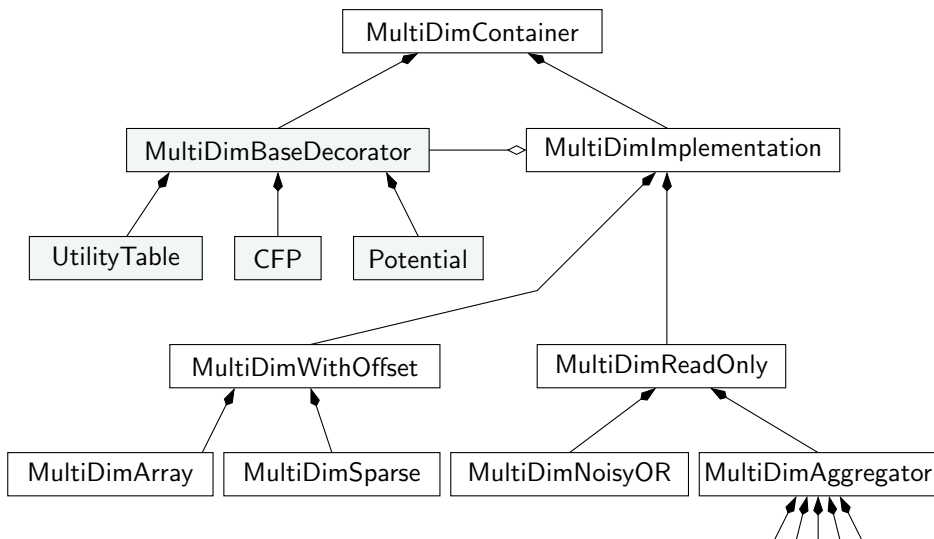
Hiérarchie pour les graphes



Hiérarchie pour les variables aléatoires



Hiérarchie pour les hyper-matrices (*multidim*)



Projects around aGrUM (1/2)

- **aGrUM** = C++ library \implies usable in any program C++
- **pyAgrum** (M. France, P.-H. Wuillemin)
Interface for aGrUM library in python (intégré dans aGrUM)
- **Squeezer** (M. Chopin, P.-H. Wuillemin)
 - Inferences in Dynamic Bayes Nets (DBN).
 - Inputs in XML, outputs in CSV, XML, DOT, etc
- **SKOOB** (L.Torti, PH Wuillemin)
 - Modélisations et inferences dans les VLBN (Very Large BNs) : PRM
- **GAI Networks** (J-P. Dubus, C. Gonzales)
- **Qlementine** (T. Porez, V. Renaudineau, P.-H. Wuillemin)
Qt4 GUI on top of aGrUM library
- **Lemon** (C. Gonzales)
Gnome GUI library for aGrUM

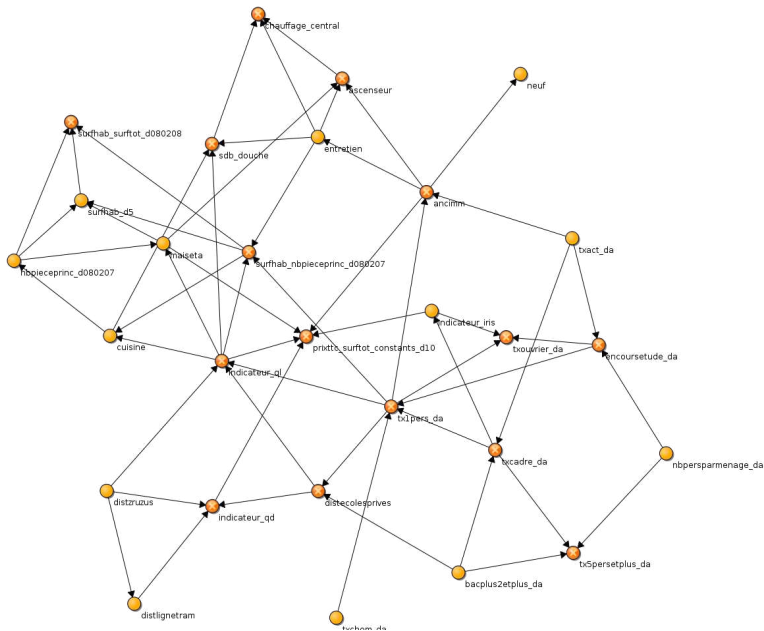
aGrUM en application

Nancy - le problème

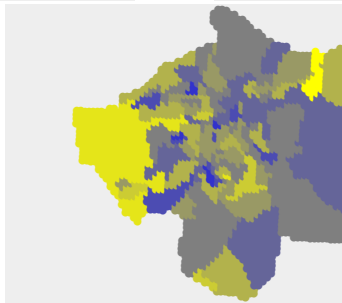
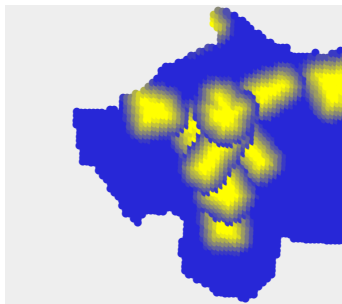
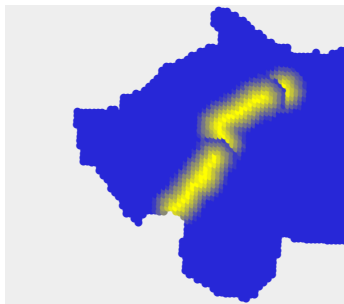


- Base de 30 000 transactions immobilières ;
- Question : explication du prix immobiliers à partir des différents facteurs connus dans la base ;
- **vers un outil de simulation d'impacts de décisions d'urbanisation.**

Nancy - le modèle



Nancy - le résultat



Projects already using aGrUM : Incalin (1/2)



But : Modélisation et contrôle de l'affinage du fromage (type camembert) à l'aide de réseaux bayésiens dynamiques.

Points forts

- Modélisation du process
- Apprentissage des paramètres
- contraintes temporelles : **dynamic** Bayes net

INCALIN : ANR 2007-2009



| | | |
|--|---|--|
| Espace multidimensionnel - mesures organoleptiques, mesures physiques - Instrumentation faible ou coût élevé | → | 27 mesures sensorielles, une dizaine de mesures physicochimiques |
| Connaissances des phénomènes partielles – différentes échelles | → | Comportement microbologique en environnement non stérile mal maîtrisé |
| Nombre de mesures et d'expériences significatives limitées | → | Temps d'affinage non négligeable |
| Variabilité des matières premières | → | Lait d'hiver, lait d'été |
| Contrainte de portabilité et de convivialité | → | Langages opérateurs |

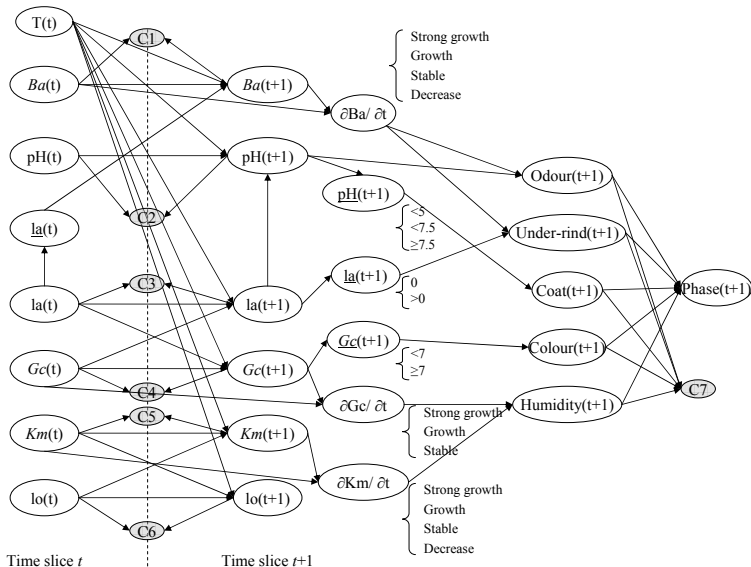


Connaissance très parcellaire sur différentes échelles

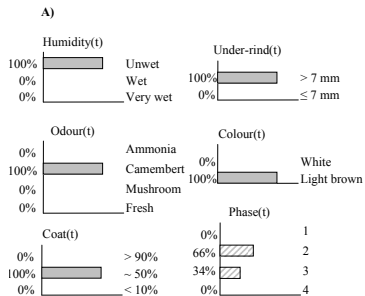


Quantité et qualité des données

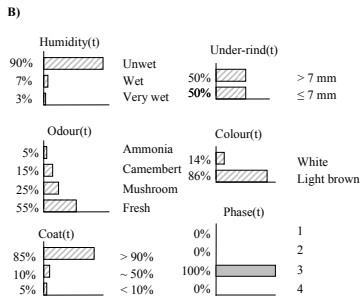
Modélisation : réseau bayésien dynamique et classification



Résultats : inférence et simulation



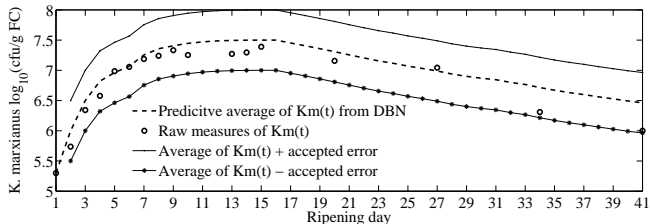
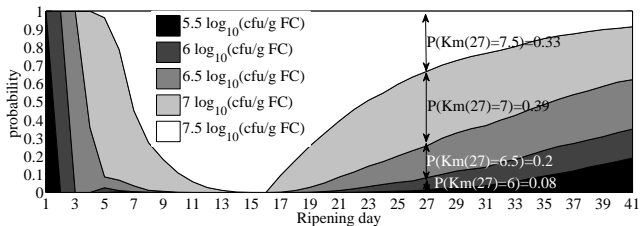
$P(\Phi \mid \text{Sensoriel})$



$P(\text{Sensoriel} \mid \Phi)$

Résultats : simulation

DBN results versus raw data for *K. marxianus* at RH=98% and T=8°C



Résultats académiques (M.Chopin, PH.Wuillemin)

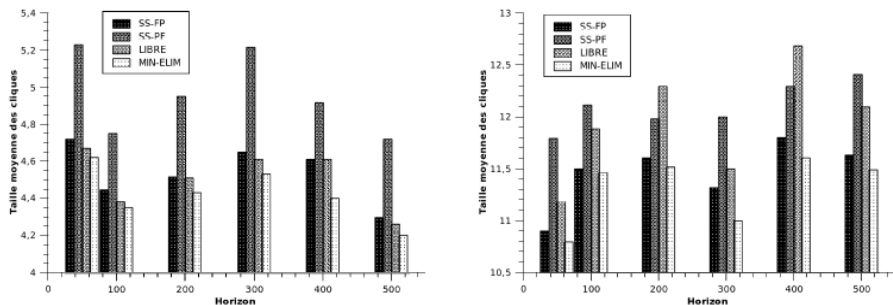
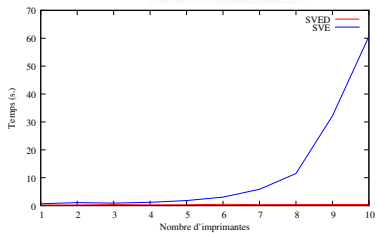
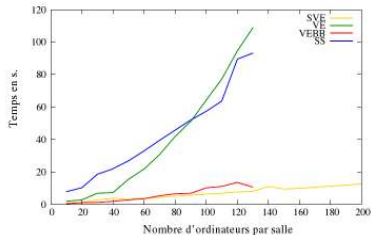
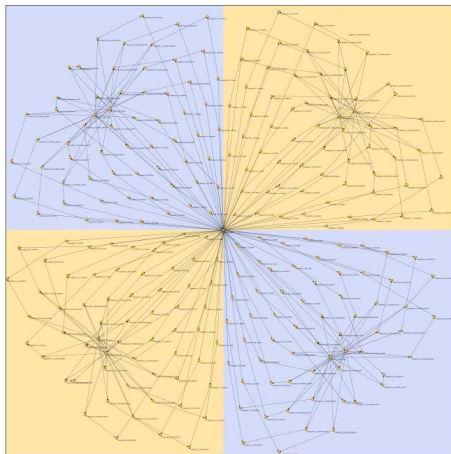


FIG. 11 – Histogramme représentant la taille moyenne des cliques en fonction de l'horizon obtenue sur une trentaine de dBN générés aléatoirement avec cinq variables par *slices* (à gauche) puis dix variables par *slices* (à droite).

SKOOB : ANR 2008-2010

Représentation des réseaux bayésiens de très grandes tailles



Conclusion

aGrUM

- (L)GPL
- C++ (rapide, de haut niveau)
- Bibliothèque riche (non limitée aux BNs)
- Facile à étendre
- Aucune boîte noire

pyAgrum

- Utilisation en prototypage rapide de fonctionnalités
- Utilisation en test rapide de modèles
- Grande facilité pour la création de petites applications incluant un BN (tableaux de bord, etc.)