Département Génie Industriel et Informatique

promotion 2016 4ème année

RAPPORT DE STAGE

Réseaux bayésiens à temps continu

Présenté par :
Bernardin HOUESSOU

Sous l'encadrement de l'Enseignant-Chercheur :
Christophe GONZALES

1er juin 2015 — 7 août 2015

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et à la rédaction de ce rapport.

Tout d'abord, j'adresse mes remerciements à mon professeur M. Marc Le Goc de l'Université Polytech Marseille qui m'a beaucoup aidé dans ma recherche de stage d'initiation à la recherche en me permettant de postuler dans ce laboratoire.

Je tiens à remercier vivement mon tuteur de stage, professeur M. Christophe GONZALES, membre de l'équipe "Décision" du département "DESIR" au sein du Laboratoire d'Informatique de Paris 6 - LIP6, pour son accueil et le partage de son expertise au quotidien.

Enfin, Je remercie également Mme. Ghislaine Mary, toute l'équipe de "Décision", ainsi que les stagiaires pour leur accueil et leur esprit d'équipe.

Sommaire

Glossaire

Cette page permet de définir certains mots ou termes utilisés dans ce rapport.

aGrUM : aGrUM is a C++ library designed for easily building applications using graphical models such as Bayesian networks, influence diagrams, decision trees, GAI networks or Markov decision processes.

aGrUM est une bibliothèque C ++ développée pour faciliter la création d'applications utilisant des modèles graphiques tels que les réseaux bayésiens , les diagrammes d'influence , les arbres de décision, les réseaux GAI ou des processus de décision de Markov.

Amalgamation :C'est une opération de «multiplication» réalisée sur les MICs et qui permet d'obtenir à partir de la combinaison de deux MICs une seule et plus grande MIC.

CNRS : Centre National de la Recherche Scientifique.

DESIR : Décision, Systèmes Intelligents et Recherche opérationnelle.

Dp : La densité de probabilité est une fonction qui permet de représenter une loi de probabilité sous forme d'intégrales. $f(t) = \lambda e^{-\lambda t} \quad \forall t \geq 0.$

ExPM : L'exponentielle d'une matrice l'exponentielle d'une matrice est une fonction généralisant la fonction exponentielle aux matrices et aux endomorphismes par le calcul fonctionnel.(Cf.Moler and Loan (2003))

FctR : Fonction de répartition ou CDF (Cumulative Distribution Function) est la loi de probabilité d'une variable aléatoire réelle. $F(t) = 1 - e^{-\lambda t} \quad \forall t \geq 0.$

Inférence bayésienne :L'inférence bayésienne est une méthode d'inférence permettant de déduire la probabilité d'un événement à partir de celles d'autres événements déjà évalués. Elle s'appuie principalement sur le théorème de Bayes.

LIP6 : Laboratoire d'Informatique de Paris 6 - 4 place Jussieu 75005 Paris

Marginalization : Opération sur des matrices d'intensité qui supprime une variable X à partir d'une matrice d'intensité afin d'obtenir une distribution sur une autre variable en utilisant une matrice d'intensité plus simple.

MIC : Matrice d'intensité Conditionnelle contenant les intensités de transitions d'une variable aléatoire.

MIJ : Matrice d'intensité Jointe ou Joint Intensity Matrix.

Moralisation de graphe : les nœuds possédant plusieurs parents se marient et il y a liaison des parents deux à deux avec des arcs supplémentaires puis récupération du squelette du graphe ainsi obtenu. **PMH** : Processus de Markov homogènes modélise la dynamique d'un système au cours du temps en respectant
$P[X(t+s) = j/X(t) = i] = P[X(t+s) = j/X(t) = i] = pij(s)avecX(t), t > 0.$
Ces processus possèdent un mécanisme de transition qui ne change pas au cours du temps.

PMTC : Le processus de Markov à temps continu est une variante à temps continu du processus de Markov qui est un modèle mathématique où le temps passé dans chacun des états est une variable aléatoire réelle positive suivant une loi exponentielle.

PS : La probabilité stationnaire est une mesure stationnaire qui remplit les conditions supplémentaires-particulières.

RBs : Réseaux bayésiens (RB) et **RBDs** : Réseaux bayésiens dynamiques(RBD).

RBTCs : Les réseaux bayésiens à temps continu (RBTC) permettent de modéliser des processus stochastiques structurés avec un nombre fini d'états qui évoluent continuement dans le temps. Ces modèles ont été introduits par Nodelman Horvitz (2003).

UPMC : Université Pierre et Marie Curie.

Introduction

Le cursus professionnel de la 4ème année d'ingénieur en génie Industriel et Informatique à Polytech Marseille, composante de l'université d'Aix-Marseille, inclut la réalisation d'un stage d'initiation à la recherche d'une durée minimum de 8 semaines en laboratoire de recherche.

J'ai effectué mon stage au sein de l'équipe *Décision* dans le département DESIR du Laboratoire d'Informatique de Paris 6 (LIP6), une unité Mixte de Recherche de l'Université Pierre et Marie Curie et du Centre National de la Recherche Scientifique . L'équipe *Décision* entreprend d'une part des travaux de modélisation visant à produire des représentations formelles de situations décisionnelles complexes et d'autre part des travaux de nature algorithmique visant à résoudre les problèmes formels posés et déterminer efficacement les solutions optimales.

Ce stage a été pour moi une excellente introduction au monde de la recherche et du travail en laboratoire. Il m'a été demandé, pendant mon stage, d'effectuer une analyse de piste sur les Réseaux bayésiens à temps continu (RBTCs) afin d'implanter des algorithmes qui permettront d'une part d'apprendre la structure et les paramètres de ces réseaux à partir d'une base de données d'événements et d'autre part de réaliser des inférences probabilistes. Je vous présenterai donc, dans ce rapport, le travail qui m'a été confié puis l'étude documentaire réalisée. Ensuite nous verrons les outils utilisés pour la réalisation de ce travail et nous suivrons enfin la présentation du travail effectué.

I . Travail à effectuer

Cette partie a pour objectif de présenter le travail qui m'a été demandé pendant ce stage. Après cette présentation nous verrons quelles métriques ont été utilisées pour ce travail.

Pour atteindre notre objectif principal qui est de réaliser des programmes permettant d'implanter des *algorithmes* offrant la possibilité d'apprendre *la structure et les paramètres* des RBTCs ou CTBNs à partir d'une base de données d'événements d'une part et d'autre part de réaliser des inférences probabilistes à l'aide de modèle graphique nous avons suivi tout d'abord un cours sur les réseaux bayésiens et les réseaux bayésiens dynamiques afin de nous familiariser avec les concepts clés de ces modèles probabilistes.

Ensuite, nous avons étudié deux articles de *Uri Nodelman, Christian Shelton et Daphne Koller* sur les RBTCs ou CTBNs parus en 2002 et 2003. À l'issue de cette étude, nous devrions développer en langage orientée objet, C++ des programmes qui permettront non seulement d'obtenir des résultats identiques que ceux des articles étudiés mais également délivrant de grandes performances selon les opérations complexes qu'elles devront effectuer tout en tenant compte de la machine sur laquelle elles seront exécutées. Aussi nous devrions utiliser certains outils développés et mis à notre disposition par le Laboratoire d'Informatique de Paris 6 (LIP6) au niveau de leur bibliothèque aGrUM dans la réalisation de nos tâches.

Il est important de préciser que ces programmes seront utilisés pour faire de l'inférence au niveau des RBTCs ou CTBNs.

Enfin, nous devrions terminer par la réalisation de tests sur des données synthétiques afin de tester l'efficacité de ces programmes.

II . Etude bibliographique

Cette section sera séparéé en trois sous-parties.

Dans un premier temps nous verrons quelques notions sur les RBs et RBDs puis, nous aborderons les articles que j'ai eu à utiliser afin de m'informer sur l'existence de travaux anciens ou récents relatifs aux RBTCs tout en mettant en exergue les solutions proposées ainsi que leurs avantages et défauts.

Enfin nous nous pencherons sur les articles directement liés aux tâches qui nous ont été confiées, afin de voir quels sont les méthodes et algorithmes utilisés au niveau des RBTCs.

II .1 RBs et RBDs

Dans cette partie nous parlerons des notions de bases à connaître avant d'entamer nos recherches et nos travaux sur les RBTCs.

II .1.1 Les réseaux bayésiens (RBs)

Les réseaux bayésiens(RBs) (qui sont appelés réseaux probabilistes, ou encore, en anglais, "Bayesian networks" ou" probabilistic networks") sont des graphes où les nœuds représentent des variables dites aléatoires et dont les arcs expriment des interdépendances, des influences entre ces variables. Chacun de ces graphes modélise donc de manière qualitative des influences entre ces dites variables.

Dans un (RB), l'absence d'arc entre deux variables entraîne l'absence d'une influence directe entre celles-ci d'où pour pouvoir déterminer l'action de ces influences il est nécessaire de joindre à chaque noeud sa loi de probabilité conditionnellement à ses parents dans le graphe. (Article [5] - (Réseaux bayésiens en modélisation d'utilisateurs) (Christophe Gonzales and Pierre-Henri Wuillemin, (1998)).

Ils sont aussi des outils permettant la réalisation de calculs de probabilités conditionnelles tout en représentant une base aux connaissances utilisées dans les Systèmes d'aide à la décision.

En d'autres termes, les (BNs) constituent un langage graphique et une méthodologie simples et corrects, pour exprimer pratiquement ce dont on est certain ou incertain. Ils reposent sur la formule de Bayes reliant des probabilités conditionnelles avec des probabilités jointes

La figure 1(cf. Décompositions fonctionnelles et structurelles dans les modèles graphiques probabilistes appliquées à la reconstruction d'haplotypes-M. Simon de Givry, M. Andrés Legarra 2011) représente un réseau bayésien contenant cinq variables. Il décrit par les saisons de l'année (X_1) si la pluie tombe (X_2), si l'arrosage est en marche (X_3), si le chemin est mouillé (X_4) et si le chemin est glissant (X_5). Toutes ces variables sont binaires. L'absence d'arc allant de X_1 à X_4

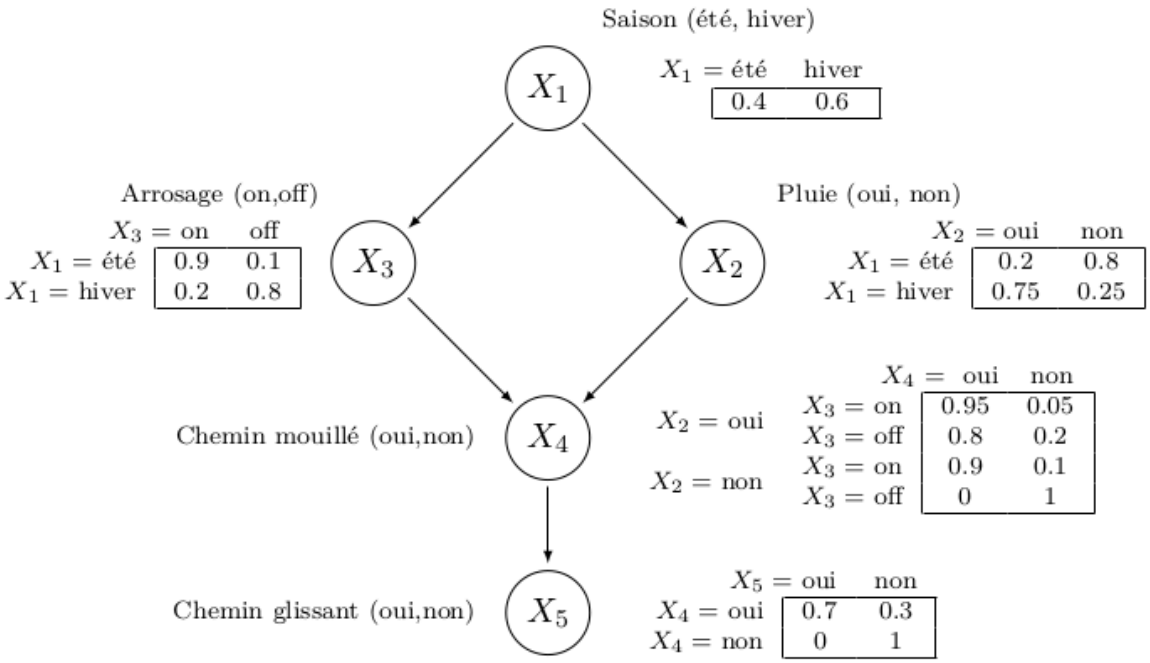

FIGURE .1 – Un réseau bayésien représentant les dépendances entre 5 variables.

II .1.1.1 La formule de Bayes

Soit deux événements A et B, qui ont des probabilités d'arriver P(A) et P(B).

P($\overline{A}$) est la probabilité que l'événement A n'arrive pas.

La formule de Bayes dit que :

P($A|B$) = $\frac{P(A,B)}{P(B)}$ (1) ou encore : P($A|B$) = $\frac{P(B|A)\cdot P(A)}{P(B)}$ (2)

Etant donné que : P(B) = P(B|A).P(A) + P(B| A).P(A) (3)

On écrit alors : P(A|B) = P(B|A).P(A)/(P(B|A).P(A) + P(B| P($\overline{A}$)).P(P($\overline{A}$))) (4)

La formule de Bayes peut être conditionnée par un événement X :

P(A|B,X) = P(B|A,X).P(A,X)/ P(B,X) (5)

Une évidence utile à préciser : P(A|B) + P(P($\overline{A}$)|B) = 1 (6)

II .1.1.2 Indépendance marginale

Deux variables aléatoires X et Y sont indépendantes (noté X $\perp\!\!\!\perp$ Y) *si et seulement si* :

$$\forall x \in D_X \ , \ y \in D_Y \quad \text{P(x, y)} = \text{P(x)} \cdot P(y)$$

Cependant deux variables aléatoires peuvent être indépendantes conditionnellement à d'autres,

c'est ce qu'on appelle simplement l'indépendance conditionnelle.

II .1.1.3 Indépendance conditionnelle

Soient trois variables aléatoires X, Y et Z. X est indépendant de Y conditionnellement à Z

(noté X $\perp\!\!\!\perp$ Y $|Z$) *si et seulement si* :

$$\forall x \in D_X \ , \ y \in D_Y \ , \ z \in D_Z \quad \text{P(x|y,z)} = \text{P(x|z)}$$

L'indépendance conditionnelle ne dépend pas de l'ordre. Ainsi nous avons également P(y | x, z) =P(y | z). L'indépendance marginale n'est autre qu'une indépendance conditionnelle où Z est l'ensemble vide.

Plusieurs propriétés découlent de ces indépendances.

II .1.2 Les réseaux bayésiens dynamiques (RBDs)

Les réseaux bayésiens dynamiques ou temporels (qui sont notés RBDs et appelés en anglais, “Dynamic Bayesian Networks(DBNs)”) sont des modèles statistiques et stochastiques qui étendent la notion des réseaux bayésiens. Cependant les réseaux bayésiens dynamiques permettent de représenter l'évolution des variables aléatoires en fonction d'une séquence discrète, par exemple des pas temporels. Ils sont aussi des modèles standards utilisés pour apprendre et raisonner sur les systèmes dynamiques. (Article [6] - Dynamic Bayesian networks (DBNs) (Dean Kanazawa, 1989))

II .2 Articles scientifiques relatifs aux RBTCs

Cette partie met en avant quelques travaux du monde de la recherche sur les RBTCs.

À partir de l'interprétation bayésienne des probabilités un agent peut ajuster les paramètres (la structure et les probabilités conditionnelles) des réseaux bayésiens de sorte à maximiser la vraisemblance du modèle par rapport à la connaissance a priori (par exemple présent dans une base de données). Ainsi, si la compatibilité de Markov n'est pas satisfaite, un réseau bayésien ne peut pas représenter correctement la distribution de probabilité qui a engendré les observations. Néanmoins, un réseau bayésien reste une bonne solution à adopter pour la modélisation d'un système lorsque la connaissance a priori que nous possèdons sur ce dernier satisfait la compatibilité de Markov.
L'article [1] nous parle de l'application des CTBNs à la dynamique de réseaux sociaux tout en introduisant un nouvel algorithme d'inférence appelé Markov chain Monte Carlo (MCMC). Cet algorithme peut être utilisé avec un grand nombre de variables voir 2500 variables et fournit de meilleurs résultats par rapport aux anciens algorithmes d'analyse de données des réseaux sociaux en offrant plus de choix sur le type de données sociales pouvant être analysé.

L'article [2] introduit une optimisation des résultats avec les méthodes utilisées pour l'apprentissage des RBTCs ou CTBNs de l'article [8] en tenant compte d'un plus grand nombre de cas voire des cas complexes.

L'inconvénient des réseaux bayésiens dynamiques classiques suppose un pas de temps discret ; ce qui peut être suffisant pour de nombreuses applications mais peut être une source d'handicap pour de nombreuses autres. Les réseaux bayésiens à temps continu permettent alors de modéliser des processus stochastiques structurés avec un nombre fini d'états qui évoluent continuellement dans le temps. Ces modèles ont été introduits par Nodelman Horvitz (2003). Certaines méthodes d'inférence spécifiques sont alors décrites dans Nodelman, Shelton Koller (2005). Ces modèles sont représentés par des graphes dirigés, pouvant posséder des circuits, car dans ce cas, les circuits sont vus de manière temporelle et ne représentent pas réellement une boucle vers la même variable, mais cette variable évaluée à un temps différent.

II .3 Articles scientifiques utilisés

Dans cette section nous aborderons deux articles sur les RBTCs de de *Uri Nodelman, Christian Shelton et Daphne Koller*

II .3.1 Premier article

Le premier article, [7] définit dans un premier temps quelques notions essentielles à connaître sur les RBTCs, puis met en avant leur syntaxe et sémantique afin d'aborder de façon brève une méthode d'inférence approximée.

Les mots et expressions clés à retenir au niveau de cet article sont : *CTBN ou RBTC , homogeneous Markov Processes ou les processus de Markov homogènes, "Continuous-Time Markov Chain" ou chaîne de Markov à temps continu ou processus de Markov à temps continu, "Intensity matrix" ou la matrice d'intensité, Cumulative Distribution Function ou la fonction de*

répartition, "stationary distribution" ou la probabilité stationnaire ou la loi stationnaire, "matrix exponential" ou exponentielle d'une matrice. "Amalgamation", "Marginalization", "Inference" ou Inférence , variables ou variables, number of states ou nombre d'états, Conditions 'Variables ou Parents (cf.Glossaire page 3).

Aussi il est important de savoir que chacune des variables peut être associée à une ou plusieurs MICs ou CIMs, un nombre d'états et de parents. Du coup nous pouvons observer des intensités de transition entre différentes variables. De plus, grâce au nombre de MICs que possède chaque variable nous pouvons déterminer le nombre de combinaisons possibles entre les états de cette variable. Chacune des MICs ou CIMs sont des matrices carrées ayant comme forme :

$$\mathrm{Q}_X = \begin{bmatrix} -q_1^x & q_{12}^x & \cdots & q_{1n}^x \\ -q_1^x & q_{12}^x & \cdots & q_{1n}^x \\ \vdots & \vdots & \ddots & \vdots \\ q_{n1}^x & q_{n2}^x & \cdots & -q_n^x \end{bmatrix}$$

où $\mathrm{q}_i^x = \sum j \neq \mathrm{i} \mathrm{q}_{ij}^x$, q_{ij}^x représente l'intensité de transition de X de l'état i à l'état j, (i ≠ j).
q_i^x est l'intensité de X quittant i . La durée où X reste à i est exponentiellement distribuée en $\lambda = \mathrm{q}_i$.

Exemple :

Variable	Nombre d'états	Parents(nombre de Parents)	nombre de CIMs
A	3	0	1
B	3	C	2
C	2	B	2
D	3	A,C,E	$3 \cdot 2 \cdot 3 = 18$
E	3	A,B	$3 \cdot 3 = 9$

NB : Nous pouvons remarquer que certaines des variables du tableau ci-dessus ne possèdent pas de parents tandis que d'autres en possèdent une ou plusieurs. Alors la synthaxe de leurs MICs ou CIMs différera selon leurs cas.
La variable A aura comme MICs ou CIMs : Q_A

La variable B aura comme MICs ou CIMs : $\mathrm{Q}_B|c_0 \quad et \quad Q_B|c_1$

La variable C aura comme MICs ou CIMs : $\mathrm{Q}_C|b_0 \quad et \quad Q_C|b_1$

Les variable D et E ont les plus grand nombre de CIMs.

Avec ou sans les états initiaux, effectuer une inférence exacte est un problème NP-difficile. Cela se justifie du faite que non seulement la multiplication matricielle n'est pas commutable mais également la dimension du réseau bayésien augmentant influant ainsi sur le temps de calcul. De plus si les tables de probabilités conditionnelles ne se pas exactes, l'interêt d'effectuer une inférence exacte avec ces valeurs approximatives n'est plus probant. Dans ce cas, il peut être intéressant d'effectuer une inférence approchée pour économiser le temps de calcul. L'inférence approchée se base sur l'algorithme de l'abre de jonction ou clique tree qui est une méthode divisée en cinq étapes : (moralisation du graphe, triangulation du graphe moral, construction de l'arbre de jonction, inférence dans l'arbre de jonction en utilisant l'algorithme des messages locaux et enfin la transformation des potentiels de clique en lois conditionnelles effectuée.)

II .3.2 Deuxième article

Le deuxième article, [8] présente les éléments statistiques nécessaires pour un RBTC en donnant les paramètres du maximum de vraisemblance et de posteriori d'apprentissage pour un grand nombre de jeux de tests avec des données complètes. Il met en avant un apprentissage de la structure, procédé basé sur une fonction de score bayésien. L'utilisation de la fonction de vraisemblance est un élément clé dans l'estimation de la densité d'une tâche. La fonction de vraisemblance et l' estimation du maximum de vraisemblance permettent respectivement de décrire les valeurs d'une loi statistique en fonction des paramètres supposés connus et d'inférer ainsi les paramètres de distribution de probabilté d'un échantilloné donné.

III . Outils utilisés

Afin de mener à bien les tâches qui m'ont été confiées, divers outils ont été utilisés. Ils seront donc présentés dans cette partie.

III .1 Spyder

Spyder est un environnement de développement intégré (EDI) multi-plateformes pour la programmation scientifique dans le langage Python.
En comparaison avec d'autres environnements de développement scientifique, Spyder dispose d'un ensemble de fonctions qui le rend unique. Il est disponible sur de multiples plateformes telles que Windows à travers WinPython et Python (x, y),Mac OS avec MacPorts et sur les principales distributions Linux.

J'ai été amené à utiliser cet outil afin de vérifier rapidement les résultats des articles de *Uri Nodelman, Christian Shelton et Daphne Koller* .

III .2 Netbeans

NetBeans est un environnement de développement intégré (EDI) moderne. Il supporte plusieurs langages tels que : C, C++, Java JavaScript, XML, Groovy, PHP et HTML de façon native ainsi que bien d'autres (comme Python ou Ruby) par l'ajout de greffons. Conçu en Java, NetBeans est disponible sous plusieurs systèmes d'exploitation à savoir : Windows, Linux, Solaris (sur x86 et SPARC). Cependant pour développer en Java un environnement Java Development Kit JDK est requis.

J'ai été amené à utiliser cet outil afin de créer mes différents programmes en c++ car je l'avais déjà utilisé préalablement pour d'autres projets. Il ne demande pas l'utilisation d'un compilateur particulier et permet un développement aisé avec la création automatique de fichiers "CMakeLists" pour le moteur de production CMake.

III .3 Armadillo

Armadillo est une suite de bibliothèques d'algèbre linéaire pour le langage de programmation C ++. Elle permet d'obtenir des calculs de base simples et efficaces, tout en ayant en même temps une interface simple et facile à utiliser. Elle vise une catégorie d'utilisateurs tels que les scientifiques et les ingénieurs tout en prenant en charge les nombres entiers, décimaux, complexes avec un sous-ensemble de fonctions trigonométriques et statistiques. Diverses décompositions matricielles sont fournies grâce à l'intégration en option de la suite d'algèbre linéaire (LAPACK) et du logiciel contenant les bibliothèques d'algèbre linéaire (Atlas) pour obtenir de grandes performances. Aussi Armadillo est liée à la bibliothèque Boost Basic Linear Algebra Subprograms (uBLAS) qui utilise également la métaprogrammation avec des patrons. Elle permet de fournir, grâce à ses fonctions des optimisations avancées en tenant compte de la machine de utilisateur.

Elle est une suite de bibliothèques libre distribuée sous la licence publique Mozilla, la rendant utilisable pour le développement à la fois "open source" et les logiciels propriétaires.

J'ai été amené à utiliser cette suite de bibliothèques car elle est facile à utiliser et génère plus de performances que d'autres bibliothèques testées. *(cf. http://scicomp.stackexchange.com/questions/351/recommendations-for-a-usable-fast-c-matrix-library* et *http://arma.sourceforge.net/)*

III .4 Autres outils

Cette section regroupe les outils secondaires utilisés lors de mon stage.

Système d'exploitation : Linux

Linux est le nom couramment donné à tout système d'exploitation libre fonctionnant avec le noyau Linux. Ce système est né de la rencontre entre le mouvement du logiciel libre et le modèle de développement collaboratif et décentralisé via Internet. Son nom vient du créateur du noyau Linux, Linus Torvalds

J'ai opté pour l'utilisation du Système d'exploitation Linux car il a pour réputation d'être stable, efficace en offrant plus d'avantages pour la programmation en c++ comme l'installation rapide de bibliothèques et proposant de nombreux environnements de développement intégré.

Internet et articles scientifiques

J'ai adopté l'utilisation d'Internet pour la recherche d'informations et de documents électroniques relatifs aux RBs et RBTCs.

IV . Travail effectué

Suite à l'étude des deux articles scientifiques, nous avons développé en nous basant sur l'article [7],3 programmes en utilisant la bibliothèque C++ Armadillo . Le premier détermine *la fonction de distribution (phase distribution)* d'un sous-système S, le second effectue l'opération de *"multiplication"* appelée amalgamation, puis le dernier réalise le processus d'approximation marginale (Marginalization). Aussi chacun de ces programmes possède un fichier "Readme.txt" qui sert de tutoriel à son utilisation. Nous avons opté comme langue pour les commentaires l'anglais.

IV .1 1er programme : "ctbnphasedistribution"

Ce programme détermine la fonction de distribution d'un sous-système S à partir de données d'entrée telles qu'une distribution d'entrée P_S^0, une matrice U_S d'intensité de transitions de S et un vecteur unitaire e en se basant sur la formule :
$F(t) = 1 - P_S^0 exp(U_S t)e$.
Ce programme permet en outre d'obtenir des informations sur les vecteurs et valeurs propres, l'exponentielle de la matrice U_S et de déterminer la distribution sur le temps où l'on reste dans le sous-système S en utilisant la formule :
$P^0(-U_S)^{-1}e$. Les résultats de ces opérations sont sauvegardés dans un fichier texte afin d'être récupérés ou consultés ultérieurement. (Voir Annexe-captures d'écran et codes).
Aussi ce programme propose deux options pour les données d'entrée dans les différentes matrices selon l'opération à réaliser, un ensemble de jeu de tests pour un essai rapide. Il est important de faire attention à la synthaxe de P_s^0 qui est
$P_s^0 = [0.xxxx0.xxxx...]$ en évitant d'utiliser des valeurs qui sont des fractions tout en arrondissant à 4 chiffres après la virgule. Par exemple pour :$P_s^0 = [ab]$ avec les valeurs telles que a=2/3 et b=a=1/3 il est conseillé d'utiliser a=0.6667 pour a=2/3 et b=0.3333 pour b=1/3. Il propose également une génération automatique de la matrice d'entrée "P0s" à condition de modifier le code source "ctbnphasedistribution.cpp" en suivant les instructions du tutoriel (fichier "Readme.txt").

Cette solution ne marche qu'avec les cas binaires rencontrés dans l'article [7].

(Voir répertoire du code source)

IV .2 2^{e} programme : "ctbnamalgamation"

Ce programme effectue l'opération de "multiplication" à partir de données d'entrée telles que des matrices d'intensité de transitions. Les résultats de ces opérations sont sauvegardés dans un fichier texte afin d'être récupérés ou consultés ultérieurement. (Voir Annexe-captures d'écran et codes).

Cette solution prend en compte l'évaluation des cas binaires comme ceux rencontrés dans l'article [7] mais aussi des cas complexes. Cependant quand aux cas complexes le nombre de variables est limité à 7 en tenant compte de leurs CIMs et des relations entre les variables.

(Voir répertoire du code source)

IV .3 3e programme : "ctbnmarginalization"

Ce programme effectue l'opération de "marginalization" à partir des matrices d'intensité de transitions et des données recupérées de l'opération d'"amalgamation". Les résultats de ces opérations sont sauvegardés dans un fichier texte afin d'être récupérés ou consultés ultérieurement. (Voir Annexe-captures d'écran et codes).

Cette solution prend en compte l'évaluation des cas binaires comme ceux rencontrés dans l'article [7].

(Voir répertoire du code source)

Conclusion

Ce stage effectué au sein de l'équipe *Décision* dans le Laboratoire d'Informatique de Paris 6 (LIP6) a été pour moi une excellente initiation au monde de la recherche. Il m'a permis d'approfondir mes connaissances dans le domaine de l'apprentissage statistique, en programmation objet C++ et de découvrir les modèles graphiques utilisés dans le domaine de l'intelligence artificielle comme les réseaux bayésiens dont je n'avais que de vagues notions avant ce stage. J'ai également acquis de nouvelles compétences avec l'utilisation de l'outil LaTeX pour rédiger mon rapport de stage.

Cependant j'ai rencontré quelques difficultés dues notamment à la courte durée du stage eu égard aux tâches à réaliser. En effet, la vérification des résultats obtenus selon les articles, le choix de la bibliothèque adéquate parmi les nombreuses disponibles en C++, celui de la structure appropriée afin d'élaborer des programmes satisfaisant les contraintes des algorithmes relatifs aux articles étudiés ont occupé une bonne période du stage.

Ainsi mon stage se termine avec des programmes fonctionnels pour des cas binaires et toutefois limités pour des cas complexes. Cette limitation pour les cas complexes n'a pas permis d'entamer la phase qui consistait à utiliser certains outils de la bibliothèque aGrUM.

Je souhaiterais à l'avenir une meilleure répartition des missions en rapport avec la durée du stage afin d'obtenir des résultats plus performants.

Bibliographie

[1] Yu Fan and Christian R. Shelton (2009). Learning Continuous-Time Social Network Dynamics. *Proceedings of the Twenty-Fifth International Conference on Uncertainty in Artificial Intelligence.*

[2] Uri Nodelman, Christian R. Shelton, and Daphne Koller (2005). Expectation Maximization and Complex Duration Distributions for Continuous Time Bayesian Networks. *Proceedings of the Twenty-First International Conference on Uncertainty in Artificial Intelligence*(pp. 421-430).

[3] Ralf Herbrich, Thore Graepel, and Brendan Murphy (2007). Structure from Failure. *SYSML'07: Proceedings of the 2nd USENIX workshop on tackling computer system problems with machine learning*(pp. 1-6.).

[4] Tal El-Hay, Nir Friedman, Daphne Koller and Raz Kupferman (2006). Continuous Time Markov Networks. PhD thesis, Stanford University. *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*(pp. 155-164).

[5] Christophe Gonzales and Pierre-Henri Wuillemin (1998).Réseaux Bayésiens en Modélisation Utilisateur. PhD thesis, Stanford University. *Revue Sciences et Techniques Educatives, Vol. 5, N. 2,*(pp. 173-198).

[6] Dean, T., Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence, 5*(pp. 142–150).

[7] Uri Nodelman, Christian R. Shelton, and Daphne Koller (2002). Continuous time Bayesian networks. *Proceedings of the Eighteenth Con-ference on Uncertainty in Artificial Intelligence*(pp. 378–387).

[8] Uri Nodelman, Christian R. Shelton, and Daphne Koller (2003). Learning Continuous Time Bayesian Networks. *Proceedings of the Nineteenth International Conference on Uncertainty in Artificial Intelligence*(pp. 451-458).

[9] Uri Nodelman (2007). Continuous Time Bayesian Networks. PhD thesis, Stanford University. *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*(pp. 2115-2140.).

[10] Yu Fan, Jing Xu, Christian R. Shelton (2010).Importance Sampling for Continuous Time Bayesian Networks. PhD thesis, Stanford University. *Journal of Machine Learning Research*(pp. 155-164).

[11] Moler and Loan (2003).Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later.SIAM Review, 45, 3–49.

[12] Saria, S., Nodelman, U., Koller, D. (2007). Reasoning at the right time granularitty. Proceedings of the Twenty-third Conference on Uncertainty in Artificial Intelligence.

[13] Tawfik, A. Y., Neufeld, E. M. (1994). Temporal Bayesian networks. Proceedings of the First International Workshop on Temporal Representation and Reasoning. Tawfik, A. Y., Neufeld, E. M. (2000). Temporal reasoning and bayesian networks. Computational Intelligence, 16, 349–377.

[14] Friedman, N. (1997). Learning belief networks in the presence of missing values and hidden variables. ICML '97 : Proceedings of the Fourteenth International Conference on Machine Learning (pp. 125–133). Morgan Kaufmann Publishers Inc.

[15] Friedman, N., Kupferman, R. (2006). Dimension reduction in singularly perturbed continuous-time Bayesian networks. Proceedings of the Twenty-second Conference on Uncertainty in AI (UAI). Boston, Massachussetts.

[16] Geiger, D., Heckerman, D. (1995). A characterization of the dirichlet distribution with application to learning Bayesian networks (Technical Report MSR-TR-94-16). Microsoft Research.

[17] Gihman, I. I., Skorohod, A. V. (1973). The theory of stochastic processes II. New York : Springer-Verlag.

[18] Gill, R. D., Johansen, S. (1990). A survey of product-integration with a view towards applications in survival analysis. The Annals of Statistics, 18, 1501–1555.

[19] Gopalratnam, K., Kautz, H., Weld, D. S. (2005). Extending continuous time Bayesian networks. AAAI05 : Proceedings of the Twentieth National Conference on Artificial Intelligence.

[20] Gross, D., Harris, C. M. (1998). Fundamentals of queueing theory. New York : John Wiley and Sons, Inc. Third edition.

[21] Hanks, S., Madigan, D., Gavrin, J. (1995). Probabilistic temporal reasoning with endogenous change. Proceedings of the Eleventh Conference on Uncertainty in Ar- tificial Intelligence.

[22] Heckerman, D. (1995). A tutorial on learning with Bayesian networks (Technical Report MSR-TR-95-06). Microsoft Research.

[23] Heskes, T., Zoeter, O. (2002). Expectation propagation for approximate infer- ence in dynamic Bayesian networks. Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence.

[24] Hjort, N. L. (1986). Bayes estimators and asymptotic efficiency in parametric counting process models. Scandinavian Journal of Statistics, 13, 63–85.

[25] Holmes, I., Rubin, G. M. (2002). An expectation maximization algorithm for training hidden substitution models. J. Mol. Biol., 317, 753–764.

[26] Howard, R. (1971). Dynamic probabilistic systems, vol. I II. New York : John Wiley and Sons, Inc.

[27] Gonzales, C. Jouve, N. (2006). Learning bayesian networks structure using markov network. Dans Studený, M. Vomel, J. (Eds.), Proceedings of the Third European Workshop on Proba- bilistic Graphical Models (PGM'06), number ISBN : 80-86742-14-8, (pp. 147–154)., Prague, Czech Republic.

www.wikipedia.com

www.stackoverflow.com

www.arma.sourceforge.net/docs.html

www.cse.sc.edu/mgv/BNSeminar/Spring2004/Continuous Time Bayesian Networks.pdf

www-desir.lip6.fr/gonzales/teaching/mgde/

Annexes

Captures d'écran

```
1 ********************************************************************************
2 WE WOULD LIKE TO OBTAIN :
3 (distribution function /phase distribution) with :Fs(t)=1-POS*(exp[(Us)t])*e
4 ********************************************************************************
5 Us:-0.1000   0.0500
6     0.2000  -0.2100
7 Unit vector 'e':   1.0000
8                    1.0000
9 ********************************************************************************
10 POs:   0.5000   0.5000
11 ********************************************************************************
12 Expected time in 'minutes' to remain within the subsystem:
13          Esptime = POs*[(-Us)^-1]*e =   25.4545
14 ********************************************************************************
15 Eigen vectors or  Matrix P:
16     (+6.457e-01,+0.000e+00)    (-2.835e-01,+0.000e+00)
17     (+7.636e-01,+0.000e+00)    (+9.590e-01,+0.000e+00)
18 Eigen values or Matrix diagonal D:
19     (-4.087e-02,+0.000e+00)
20     (-2.691e-01,+0.000e+00)
21 The inverse of P which is P^(-1) is:
22     (+1.148e+00,+0.000e+00)    (+3.392e-01,-0.000e+00)
23     (-9.137e-01,+0.000e+00)    (+7.727e-01,+0.000e+00)
24 Exponential de matrix D :exp(D)=exp(eigval):
25     (+9.600e-01,+0.000e+00)
26     (+7.640e-01,+0.000e+00)
27
28 POs.dot(P): (+7.046e-01,+0.000e+00)    (+3.377e-01,+0.000e+00)
29
30 P^(-1)*e=:    (+1.487e+00,+0.000e+00)
31               (-1.410e-01,+0.000e+00)
32 ********************************************************************************
33 !!! NB: Check product of sign(-) with  sign (-/+) before and inside A ,B,C & Z
34  following Syntax=1-POS*(exp[(Us)t])*e =1-A.(d^t)-B.(e^t)-C.(f^t)-Z.(w^t)...
35 !!!
36 Fs(t=1)= 1 -{+(+1.048e+00,+0.000e+00).(+9.600e-01,+0.000e+00)^1
37 +(-4.763e-02,+0.000e+00).(+7.640e-01,+0.000e+00)^1}
38 ********************************************************************************
39 Fs(t) BY INCLUDING VALUE OF ^(t=1) :
40
41 Fs(t=1)= 1-{+(+1.048e+00,+0.000e+00).(+9.600e-01,+0.000e+00)
42 +(+1.006e+00,+0.000e+00)}
```

1er programme : "ctbnphasedistribution"

```
1 ca|N°1|
2 KEY:  Var:     nbstate:   parents:     Matrixvalue:
3
4 0  :   w          2        z0          -1,1,2,-2
5 1  :   w          2        z1          -3,3,4,-4
6 2  :   z          2        w0          -5,5,6,-6
7 3  :   z          2        w1          -7,7,8,-8
8
9 Resources/Matrices/CIMs/Qw|z0.txt
10
11   -1.0000   1.0000   2.0000  -2.0000
12 Resources/Matrices/CIMs/Qw|z1.txt
13
14   -3.0000   3.0000   4.0000  -4.0000
15 Resources/Matrices/CIMs/Qz|w0.txt
16
17   -5.0000   5.0000   6.0000  -6.0000
18 Resources/Matrices/CIMs/Qz|w1.txt
19
20   -7.0000   7.0000   8.0000  -8.0000
21
22 Cartésien Product of states variables:π*card(Var): 4
23 --Offset or Weight of Variable: z is:1 with state: 2
24 --Offset or Weight of Variable: w is:2 with state: 2
25 ********************************************************
26
27 Resources/Result/Q_JIM.txt
28
29    -6.0000    1.0000    5.0000         0
30     2.0000   -9.0000         0    7.0000
31     6.0000         0   -9.0000    3.0000
32          0    8.0000    4.0000  -12.0000
33
34 ********************************************************
```

2e programme : "ctbnamalgamation

```
1 Resources/Result/Q_JIM.txt
2    -4.0000    1.0000    3.0000         0
3     2.0000   -7.0000         0    5.0000
4    15.0000         0  -16.0000    1.0000
5          0    4.0000    2.0000   -6.0000
6 ****************************************
7 Resources/Matrices/Distribution/P0y|0.txt
8    0.3000   0.7000
9
10 Resources/Matrices/Distribution/P0z|y1.txt
11    0.7000   0.3000
12
13 Resources/Matrices/Distribution/P0z|y2.txt
14    0.3000   0.7000
15
16 Concatenate distributions with parents
17
18    0.7000   0.3000
19    0.3000   0.7000
20
21 Matrix-before colonns elements sum
22
23    0.2100   0.2100
24    0.0900   0.4900
25
26 Matrix-after colonns elements sum
27
28    0.4200
29    0.5800
30 ************************************
31 Result of the Marginalization is:
32
33 Marg P0:
34   -4.0000   4.0000
35    5.7069  -5.7069
```

3e programme : "ctbnmarginalization"

Codes

(Voir fichiers compressés ou répertoires des codes sources).