Construction de la structure des Réseaux Bayésiens causaux appliqués au diagnostic

Thierno Diallo, Sébastien Henry, Yacine Ouzrout

Construction de la structure des réseaux Bayésiens causaux appliqués au diagnostic

Thierno M. L. DIALLO[1], Sébastien HENRY[1], Yacine OUZROUT[2]

[1] Laboratoire DISP, Université de Lyon, Université Lyon 1, France.
{Thierno.Diallo, Sebastien.Henry} @univ-lyon1.fr

[2] Laboratoire DISP, Université de Lyon, Université Lyon 2, France.
Yacine.Ouzrout@univ-lyon2.fr

Résumé— **Le formalisme des Réseaux Bayésiens possède plusieurs caractéristiques intéressantes pour l'analyse causale dans le domaine industriel. En effet, il permet de combiner des données certaines et incertaines d'une part et d'exploiter à la fois les données et l'expertise d'autre part. Mais la construction du réseau reste encore un défi dans les cas pratiques. Dans cet article, nous abordons cette problématique en proposant un algorithme de construction du réseau appliqué au diagnostic dans le domaine industriel. L'algorithme consiste, dans un premier temps, à exploiter les connaissances expertes et les propriétés du domaine d'application pour répartir les variables (ou paramètres ou nœuds) en différents niveaux de causalité. Cette première phase aboutit à une répartition en « cascade » des nœuds commençant par les causes racines pour aboutir aux effets ultimes en passant par un ou plusieurs niveaux intermédiaires. Puis dans une seconde phase, l'algorithme proposé permet de déterminer les liens de causalité existant entre les différents nœuds à partir des données de traçabilité unitaire historisées. L'étude comparative sur les données synthétiques avec les algorithmes d'apprentissage de structure parmi les plus performants actuellement a montré que notre algorithme est plus efficace en termes de capacité à reconstruire le vrai réseau.**

Mots-clés— **Réseau Bayésien, Algorithme de Construction, Diagnostic Industriel, Données de Traçabilité Unitaire**

I. INTRODUCTION

Grâce à l'instrumentation croissante des processus industriels et le développement des outils et techniques de collecte automatique des données (RFID, Data Matrix,…), les entreprises collectent de plus en plus de données. Le potentiel de ces données pour l'amélioration des performances industrielles n'est plus à démontrer. Nous considérons en particulier les données de traçabilité unitaire (produit et process). En général, l'unité de traçabilité est une agrégation de plusieurs articles (ex. palette ou lot). Ce niveau de détail peut être suffisant pour avoir une connaissance satisfaisante des conditions de fabrication dans le cas de la production par lot. Mais ce niveau de détails n'est pas suffisant pour les productions continues (flow shop) et discontinues multitâches-multiproduits (job shop). La traçabilité unitaire, contrairement à la traçabilité par lot, permet une sérialisation au niveau de l'article et permet de connaitre pour chaque article, les valeurs des paramètres process de sa fabrication. Les paramètres process à tracer sont les matières premières et autres ingrédients rentrant dans la composition du produit, l'historique des processus de transformation et de distribution et la localisation du produit après sa livraison. En plus des données process, les caractéristiques du produit doivent être enregistrées. En observant les pratiques industrielles, nous avons remarqué que la traçabilité réalisée en interne est gérée par différents services (production, maintenance, achat, R&D,…) et les données ne sont pas rapprochées/réconciliées dans la majorité des cas. Nous avons constaté aussi qu'il n'y a pas de règles générales sur les données à collecter et comment les conserver surtout en ce qui concerne la traçabilité interne. La traçabilité externe, quant à elle, présente moins de difficulté et les standards EPC Global proposent un modèle de données couvrant la chaine logistique de bout en bout. Dans l'une de nos publications [1], nous avons proposé un modèle de données pour la traçabilité interne qui permet de faire un lien avec la traçabilité externe et obtenir ainsi une traçabilité complète. Dans ce modèle, les données sont agrégées par ordre de fabrication. Le processus de fabrication est divisé en segments process et pour chaque segment, les données relatives au process et au produit sont collectées. Grâce à ce modèle, nous pouvons connaitre les paramètres process associés à chaque article. Les données ainsi collectées pourront servir au développement d'applications de diagnostic, à l'optimisation du processus de fabrication et de distribution, à la gestion du cycle de vie du produit, etc. Le développement d'outils et méthodes permettant l'exploitation de cette grande quantité de données (Big Data) constitue de nos jours un défi auquel font face universitaires et industriels. Les Réseaux Bayésiens (RBs) constituent l'un des outils adoptés pour exploiter ces données. Parmi les avantages des RBs par rapport aux autres outils d'intelligence artificielle statistiques (la maîtrise statistique des procédés, l'analyse en composantes principales/ régression des moindres carrés partiels, etc.) ou non (réseaux de neurone, etc.), on peut citer le fait qu'ils soient capables de combiner données certaines et incertaines et d'exploiter à la fois données et expertise. En particulier, plusieurs études ont montré que les RB offrent plus de performance dans les applications de supervision que les réseaux de neurones (RNs). De plus, les RNs n'ont pas de représentation sémantique ou de raisonnement symbolique, rendant difficile l'explication des conclusions [2]. Les résultats présentés dans cet article sont une partie de nos travaux sur le développement d'une approche de diagnostic à base des RBs. Notre approche [3] consiste à utiliser les RB pour déterminer les causes racines (paramètres process) à l'origine d'un défaut de qualité constaté sur un produit. Cette approche diffère des autres approches utilisant les RBs pour le diagnostic qui procèdent par classification (classificateur naïf, le TAN (Tree Augmented Naïve Bayes, etc.) comme dans [4, 5]. Pour mettre

en place un modèle Bayésien, deux éléments sont à déterminer : la structure du réseau bayésien (définir les nœuds et les arcs) et les paramètres (définir les tables de probabilité conditionnelle des nœuds). La détermination de la structure par apprentissage, dans le cas d'un nombre important de nœuds, est de loin la tâche la plus difficile [6, 7]. En général, il existe trois approches pour apprendre la structure d'un RB à partir des données : l'approche à base de contrainte, l'approche à base de score et l'approche hybride. La première approche utilise les relations de dépendance/indépendance apprises à partir des données pour guider la construction du réseau. L'approche à base de score consiste à identifier parmi tous les réseaux possibles, le réseau qui maximise un certain score mesurant l'adéquation du réseau aux données. L'approche hybride combine les principes des deux précédentes. En faisant une revue de littérature, nous avons constaté que la construction du réseau pour les applications réelles reste encore un défi. En effet, les algorithmes de construction existants se heurtent à l'explosion du nombre de variables. L'apprentissage d'un RB à partir des données est un problème NP-difficile [8, 9]. La plupart des heuristiques proposées ne sont pas assez efficaces sur les données de grande dimension avec une taille d'échantillon limitée. Les exemples d'application des réseaux bayésiens publiés dans la littérature portent généralement sur une dizaine de variables. La difficulté liée à la construction du réseau pour les applications industrielles dans le cas d'un grand nombre de variables est rarement abordée. Or dans les applications industrielles, les paramètres à considérer se comptent en centaines voire en milliers et les enregistrements sont en millions. Dès lors, les algorithmes dont la complexité est exponentielle par rapport au nombre de données ne sont pas applicables. Dans ce papier, nous proposons un algorithme de construction d'un réseau bayésien appliqué aux processus industriels. Cet algorithme que nous avons appelé CBNB (Causal Bayesien Network Building) comporte deux phases : la phase de répartition des variables en différents niveaux de causalité et la phase d'apprentissage. La phase d'apprentissage prend en entrées les variables (ou paramètres ou nœuds) réparties en différents niveaux de causalité définis par les experts du domaine. Cette phase d'apprentissage détermine les relations de causalité existantes entre les variables à partir des données de traçabilité unitaire.

La suite de ce papier est organisée comme suit : les concepts et résultats théoriques nécessaires à la compréhension de ce papier sont présentés à la section 2. Notre algorithme est présenté à la section 3 suivie par la description du protocole expérimental et les résultats de l'étude comparative à la section 4. Enfin, la section 5 résume nos contributions et annonce les perspectives.

II. RAPPELS THEORIQUES

Un réseau bayésien est un graphe orienté acyclique (Directed Acyclic Graph-DAG) G représenté par le couple (V, E) où V est un ensemble de sommets qui « encodent » une distribution de probabilité conjointe et E un ensemble d'arcs reliant les sommets [10]. Les nœuds représentent les variables « aléatoires » (discrètes ou continues) et les arcs représentent les relations (si possibles causales) entre ces variables. On associe à chaque nœud la table de distribution de probabilité marginale ou conditionnelle de la variable correspondante. L'ensemble des probabilités du réseau est noté P.

Terminologie. Lorsque deux nœuds sont reliés directement par un arc, le nœud à l'origine de l'arc est dit **parent** du nœud à l'extrémité de l'arc. Inversement, le nœud à l'extrémité de l'arc est dit **fils** du nœud à l'origine. Si les deux nœuds sont reliés par plus d'un arc (une chaine), le nœud à l'origine de la chaine est dit **ancêtre** du nœud à l'extrémité de la chaine. Ce dernier est dit **descendant** du nœud à l'origine de la chaine. Un nœud sans parents est appelé nœud **racine** et un nœud sans fils est dit nœud **feuille**. Tout nœud qui n'est ni racine ni feuille est dit nœud **intermédiaire**.

On dit que le couple (G, P) est un Réseau Bayésien, avec G = (V, E) un DAG, s'il vérifie la condition de Markov.

Condition de Markov. Un DAG G sur V et une distribution de probabilité P (V) satisfont la condition de Markov ssi $\forall W \in V$, W est indépendant de $V \backslash (Descendant(W) \cup Parent(W))$ $sanchant$ $Parent(W)$.

Cette condition de Markov établit un ensemble de relations d'indépendance sur le graphe G. En effet, la condition de Markov assure que toutes les dépendances directes dans le système modélisé sont explicitement exprimées à travers les arcs [11]. Lorsque deux nœuds ne sont pas connectés par un arc, alors ils sont conditionnellement indépendants.

Notion de D-séparabilité (définition formelle). Soit un graphe G, X et Y des sommets de G avec $X \neq Y$, et W un ensemble de sommets de G ne contenant pas X ou Y. On dit que X et Y sont **d-déparés** sachant **W** dans G ssi il n'existe pas un chemin non dirigé U entre X et Y tel que (i) chaque « collisionneur» sur U ait un descendant dans W et (ii) aucun autre sommet de U ne soit dans **W**.

On dit que X et Y sont **d-connectés** sachant W ssi ils ne sont pas **d-séparés** sachant **W**[12].

III. L'ALGORITHME CBNB (CAUSAL BAYESIAN NETWORKS BUILDING)

L'algorithme CBNB proposé est un algorithme mixte basé à la fois sur l'expertise et les données. Il s'applique à la construction des modèles causaux en particulier les modèles de diagnostic en milieu industriel. L'algorithme CBNB comporte deux phases : la phase d'allocation des variables et la phase d'apprentissage des relations causales. Dans un premier temps, en se basant sur les connaissances expertes, les variables du système sont réparties entre les différents niveaux de causalité prédéfinis. Puis dans un second temps, les relations causales existantes entre les variables sont déterminées par apprentissage à partir des données.

III.1. Algorithme CBNB: l'allocation des variables du système

Les algorithmes proposés dans la littérature n'intègrent pas ou peu de connaissances expertes. Or dans plusieurs domaines (diagnostic industriel ou médical par exemple), ces connaissances sont suffisamment importantes pour ne pas être ignorées. C'est notamment le cas lorsque le système possède un nombre de variables important et sachant que ces algorithmes sont souvent exponentiels par rapport au nombre de variables. Des informations utiles permettant de faciliter l'apprentissage que les experts peuvent fournir sont [6, 13]: l'identification de nœuds racines, l'identification de nœuds feuilles, affirmation de l'existence (ou absence) de relation

causale entre deux nœuds et la définition d'un ordre (partiel ou complet) sur les variables

Dans l'algorithme CBNB, les variables sont affectées à différents niveaux de causalité. Les différents groupes formés par ces niveaux de causalité doivent ensuite être ordonnés du niveau racine (constitué des nœuds racines) au niveau feuille (composé de nœuds feuilles). Il peut y avoir un ou plusieurs niveaux intermédiaires selon les choix de modélisation. Les variables appartenant au niveau racine constituent les causes racines du modèle et les celles appartenant au niveau feuille sont considérées comme les effets ultimes. Une variable appartenant à un niveau donné est susceptible d'influencer directement une ou plusieurs variables appartenant au niveau inférieur consécutif. Mais les liens directs ne sont pas possibles entre des nœuds appartenant à des niveaux non consécutifs. Les niveaux doivent être homogènes en matière de causalité de telle sorte qu'une variable ne puisse pas influencer une variable avec laquelle elle partage le même niveau de causalité. Nous désignons cette configuration particulière des nœuds d'un réseau bayésien par la « répartition en cascade » (cf. Fig. 1).

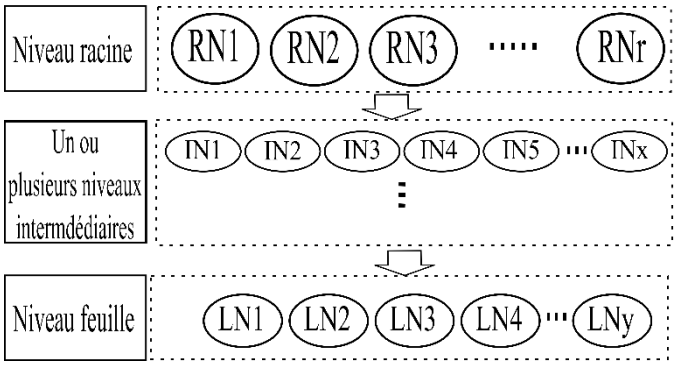

Fig. 1. Répartition en cascade des variables

Pour les réseaux bayésiens causaux appliqués au diagnostic industriel, la répartition de variables en niveaux de causalité homogènes et la hiérarchisation de ces niveaux peuvent se faire de façon intuitive: les paramètres machine et process constitueront le niveau racine, les modes de fonctionnement et de défaillance formeront les niveaux intermédiaires et enfin les symptômes et défauts produit seront considérés comme nœuds racines. Le nombre de couches et de nœuds de chaque couche sont des données d'entrées pour notre algorithme.

Les réseaux bayésiens causaux appliqués au diagnostic industriel que nous avons pu consulter dans la littérature ont pour la plupart cette répartition en cascade. Des exemples de ces modèles causaux peuvent être consultés dans [7], [14], [15], [16] et [17]. L'une des constatations qu'on peut faire est que ces modèles sont généralement des réseaux simples avec quelques dizaine de nœuds. Ces modèles de diagnostic sont appliqués à un équipement ou à un ensemble limité d'équipements. Les approches de diagnostic que nous visons avec notre algorithme sont des approches globales portant sur l'ensemble du processus de fabrication d'un produit. Il s'agit d'analyser les données process et produit sur l'ensemble de la chaine de valeur pour identifier les causes éventuelles d'un défaut constaté sur un produit ou d'une dégradation des performances. Dans ce type d'approche, le nombre de variables se compte en centaines. Les connaissances expertes seules sont insuffisantes pour définir complètement le réseau. D'où la nécessité de recourir aux algorithmes d'apprentissage structurel. La performance des algorithmes existants reste faible sur les données avec beaucoup de variables et un nombre relativement limité d'observations. Nous proposons un algorithme permettant d'améliorer cette performance grâce à l'incorporation des connaissances sur le système qui ont permis d'aboutir à la répartition en cascade.

III.2. Algorithme CBNB: l'apprentissage des relations causales

III.2.1. Justifications théoriques

Nous partons des hypothèses suivantes :

H1 : Le modèle causal à construire est un Réseau Bayésien

H2 : Ce Réseau Bayésien possède la configuration en cascade décrite dans la section III.1.

La première hypothèse implique que le graphe G (V, E) à construire est un graphe orienté acyclique et le couple (G, P), avec P l'ensemble des probabilités du réseau, respecte la condition de Markov.

La condition de Markov assure que toute indépendance conditionnelle qu'entraine le graphe G est aussi présente dans la distribution de probabilité P. C'est-à-dire que s'il n'y pas de lien entre 2 nœuds (liaison par un arc ou une chaine), alors ces deux nœuds sont conditionnellement indépendants. Mais l'inverse n'est pas vrai en général. L'indépendance conditionnelle de deux nœuds ne signifie pas l'absence d'arc ou de chaine de l'un vers l'autre (voir exemple dans [12]). Autrement dit, un lien peut exister entre deux nœuds sans qu'ils ne soient dépendants.

Condition de minimalité (ou de fidélité) [12] **:** Soient G un graphe acyclique orienté sur les sommets V et P une distribution de probabilité sur V, (G, P) satisfait la condition de minimalité ssi pour tout sous graphe H de G avec les sommets V, (H, P) ne satisfait pas la condition de Markov.

Pour tout réseau bayésien (G, P) remplissant les conditions de Markov et de minimalité, si les variables A et B sont statiquement dépendantes, alors nous avons l'un des trois cas suivant:

1. Il existe un chemin orienté dans G de A à B ;
2. Il existe un chemin orienté dans G de B à A ;
3. Il existe une variable C et des chemins orientés dans G de C à B et de C à A.

En considérant l'hypothèse H2 et seulement deux niveaux consécutifs à la fois, un seul des cas 1 et 2 peut être envisagé. De plus, sachant l'ordre des 2 niveaux, nous sommes en mesure de dire avec certitude s'il s'agit du cas 1 ou 2.

Deux nœuds du même niveau de la configuration en cascade peuvent être statistiquement dépendants dans 2 cas : lorsqu'ils ont un fils commun sachant ce fils ou lorsqu'ils partagent un même parent. Cette corrélation non nulle peut suggérer à tort de relier ces nœuds par un arc. Le théorème suivant permet de traiter ces 2 cas.

Théorème 1 **:** Dans la configuration en cascade, les nœuds appartenant au même niveau sont indépendants connaissant les nœuds du niveau supérieur.

Preuve 1: Soient 2 nœuds A ∈ Nx et B ∈Ny, Nx et Ny étant 2 niveaux consécutifs avec Nx >Ny (le niveau supérieur est le « niveau parent » par rapport au niveau inférieur). Supposons qu'une relation existe entre A et B, alors A est un parent de B. Soit C un autre parent de B. C ∈ Nx par construction. B est alors un collisionneur du chemin non dirigé A – B – C. Si Nx est le niveau racine, alors A et C sont indépendants. Ou bien, C n'étant pas un descendant de A et connaissant les parents de A, alors A et C sont indépendants d'après la condition de Markov. Soit D ∈Ny un autre fils de A. B et D sont indépendants connaissant A.

Théorème 2 **:** Soit G un réseau bayésien respectant la condition de minimalité ayant une configuration en cascade avec A ∈ Nx et B∈Ny deux nœuds. Nx et Ny étant deux niveaux consécutifs. Si A et B sont statistiquement dépendants, alors il existe un arc entre A et B. Cet arc est orienté du nœud appartenant au niveau supérieur vers le nœud du niveau inférieur.

Preuve 2: Supposons que A soit un parent de B. Le graphe étant causal, A et B sont alors dépendants. D'après la condition de Markov, cette dépendance directe est représentée par un arc dans le graphe. Par ailleurs, d'après la condition de minimalité, si un arc existe entre A et B alors ces 2 nœuds sont dépendants. Le réseau ayant la configuration en cascade, si nous supposons que A appartient au niveau supérieur, il est nécessairement un parent de B.

III.2.2. Phase d'apprentissage des relations causales

En plus des 2 précédentes hypothèses, nous formulons l'hypothèse suivante :

H3 : Les données D à partir desquelles les relations causales sont apprises sont fidèles au réseau bayésien recherché.

En considérant l'ensemble de ces 3 hypothèses et en se basant sur les 2 théorèmes précédents, nous avons développé la phase d'apprentissage des relations causales de l'algorithme CBNB (voir Fig. 2 ci-dessous)

Entrées:
- Un jeu de données D
- Le nombre de niveau L
- Le nombre de nœuds de chaque niveau {n_1, n_2,…, n_L}
- Le seuil α

Sortie: Un réseau bayésien causal (RBC)
0: Initialiser un RBC sans arc avec les variables de D
1: **pour** k =1 à L-1
2: **pour** i =1 à n_k
3: **pour** j=1 à n_{k+1}
4: Calculer Association ($\boldsymbol{X_i^k}$; $\boldsymbol{X_j^{k+1}}$)
5: **si** Association ($\boldsymbol{X_i^k}$; $\boldsymbol{X_j^{k+1}}$) > α **alors**
6: mettre un arc de $\boldsymbol{X_i^k}$ à $\boldsymbol{X_j^{k+1}}$
7: **fin pour**
8: **fin pour**
9: **fin pour**
10: **fin pour**
11: retourner RBC

Fig. 2. La phase d'apprentissage des relations causale de l'algorithme CBNB

La fonction Association (X, Y) (ligne 4). Cette fonction calcule le degré de dépendance entre les variables X et Y. Lorsque cette dépendance est jugée significative (ligne 5, voir Section IV.2. pour la fixation du seuil α), alors un arc est mis entre eux (ligne 6). Les tests d'indépendance conditionnelle tels que le Rapport de Vraisemblance Logarithmique G^2 (équivalent au test d'Information Mutuelle) ou le test Chi-carré de Pearson(X^2) peuvent permettre d'évaluer cette dépendance. Les 2 variables seront considérées dépendantes lorsque la valeur p du test est inférieure à un seuil donné. L'avantage de notre algorithme est qu'il ne teste pas la dépendance de chaque nœud avec l'ensemble du reste des nœuds (comme c'est le cas pour la plupart des autres algorithmes). Il teste la dépendance seulement avec un nombre limité de nœuds (ceux appartenant au niveau juste en dessous du niveau auquel appartient le nœud considéré).

Complexité. La complexité de l'algorithme dépend du nombre des 3 opérations élémentaires qu'il exécute : le calcul de l'association, le test du seuil et l'ajout d'arc (si nécessaire). Pour 2 niveaux k et k+1, $|n_k|*|n_{k+1}|$ calculs d'association sont exécutés. Le même nombre de tests de seuil sont réalisés. Enfin, au plus, ce même nombre d'ajouts d'arc sont réalisés. L'opération de calcul de l'association est de loin l'opération la plus coûteuse en ressource. Nous limitons donc l'évaluation de la complexité à cette opération. Soit $n = max\ (n_k,\ k = 1, \dots, L-1)$. Pour chaque niveau, l'algorithme réalise $O(n^2)$ calculs d'association. Pour l'ensemble du graphe, on aura $O((L-1)*n^2)$ calculs d'association, avec L-1 une constante. Ce qui correspond à un ordre de grandeur de $O(n^2)$ avec n<<N, où N est le nombre total de nœuds. Par comparaison, la complexité de l'algorithme IAMB[18] est de $O(N^2)$. Comme il existe un algorithme linéaire de calcul du degré de la dépendance, nous énonçons le théorème suivant :

Théorème 2 **:** L'algorithme CBNB est un algorithme d'apprentissage polynomiale $O(n^2)$ avec n<<N (N étant le nombre total de nœuds).

IV. ÉTUDES EXPERIMENTALES

L'algorithme CBNB a été implémenté en R en utilisant le package bnlearn. Toutes les simulations et les analyses expérimentales ont été réalisées à l'aide de ce package. L'analyse comporte 2 parties : l'analyse des performances en fonction du seuil α et l'étude comparative avec quelques algorithmes parmi les plus performants.

Nous avons sélectionné comme benchmark 3 réseaux (EARTHQUAKE, CANCER et SURVEY) fournis avec le package bnlearn et respectant la configuration définie dans ce papier. Nous commençons d'abord par générer des données à partir des tables de probabilités conditionnelles de chaque réseau puis nous utilisons l'algorithme testé pour reconstruire le réseau à partir des données générées. Pour toutes les expériences, dix tailles d'échantillon différentes ont été générées (100; 200; 300; 500; 800; 1000; 2000; 3000; 5000; 10000). Pour chaque taille d'échantillon, la génération des données et l'expérimentation ont été répétées 5 fois. La moyenne des valeurs des indicateurs obtenues pour les 5 répétitions a été considérée comme l'indicateur de cette taille d'échantillon.

Les tests ont été exécutés sur un PC Intel (R) Core (TM) i5-3340 M CPU @2.70GHz 2.7 GHz, 4Go de RAM sous Windows 7 Professional.

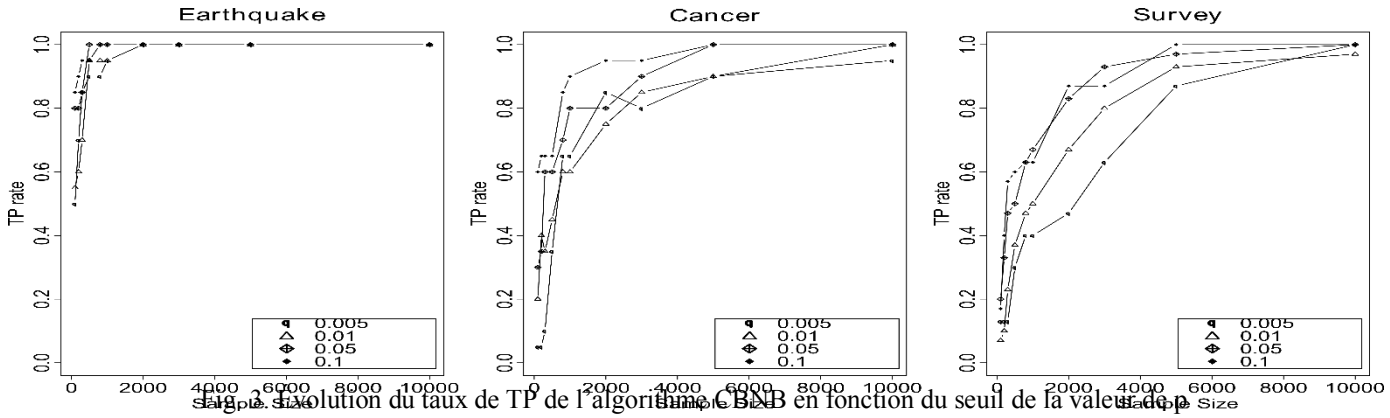

Fig. 3. Evolution du taux de TP de l'algorithme CBNB en fonction du seuil de la valeur de p

IV.1. Métriques de performance

Nous évaluons les performances en termes de qualité du réseau construit. Cette évaluation mesure la capacité de l'algorithme considéré à obtenir le vrai réseau. Plusieurs réseaux peuvent être formés à partir d'un ensemble de nœuds donné avec un même ensemble de dépendances et d'interdépendances et donc statistiquement indiscernables (Équivalence de Markov). Ces réseaux équivalents n'ont pas la même orientation pour certains arcs et n'ont pas le même score (BDe, AIC, BIC,…) par rapport aux données d'apprentissage. Il existe un réseau parmi ceux-ci qui offre un score optimal par rapport à un ensemble d'apprentissage donné [8, 12]. Le vrai réseau est celui qui est totalement orienté et qui représente le mieux (meilleur score) les dépendances et indépendances conditionnelles présentes dans les données d'apprentissage [19]. Deux catégories de critères de performance peuvent être distinguées. La première catégorie est celle qui mesure la probabilité à posteriori du réseau obtenu (BDeu, BIC,…) par rapport aux données d'apprentissage. Les scores calculés mesurent la concordance entre le réseau et les données. Cette catégorie est particulièrement utile lorsque le vrai réseau est inconnu. C'est le cas pour les applications réelles. La seconde catégorie évalue l'équivalence entre le réseau obtenu avec le vrai réseau. Les vrais réseaux étant connus dans notre cas (réseaux benchmark), nous employons cette seconde catégorie. Les critères de performance utilisés sont : TP (True Positive) et FP (False Positive). TP est le nombre d'arcs présents à la fois dans le réseau obtenu et le réseau réel et FP est le nombre d'arcs présents dans le réseau obtenu mais absents du vrai réseau. Pour faciliter la comparaison entre indicateurs provenant de réseaux différents nous avons calculé les taux en divisant les valeurs par le nombre total d'arcs du vrai réseau.

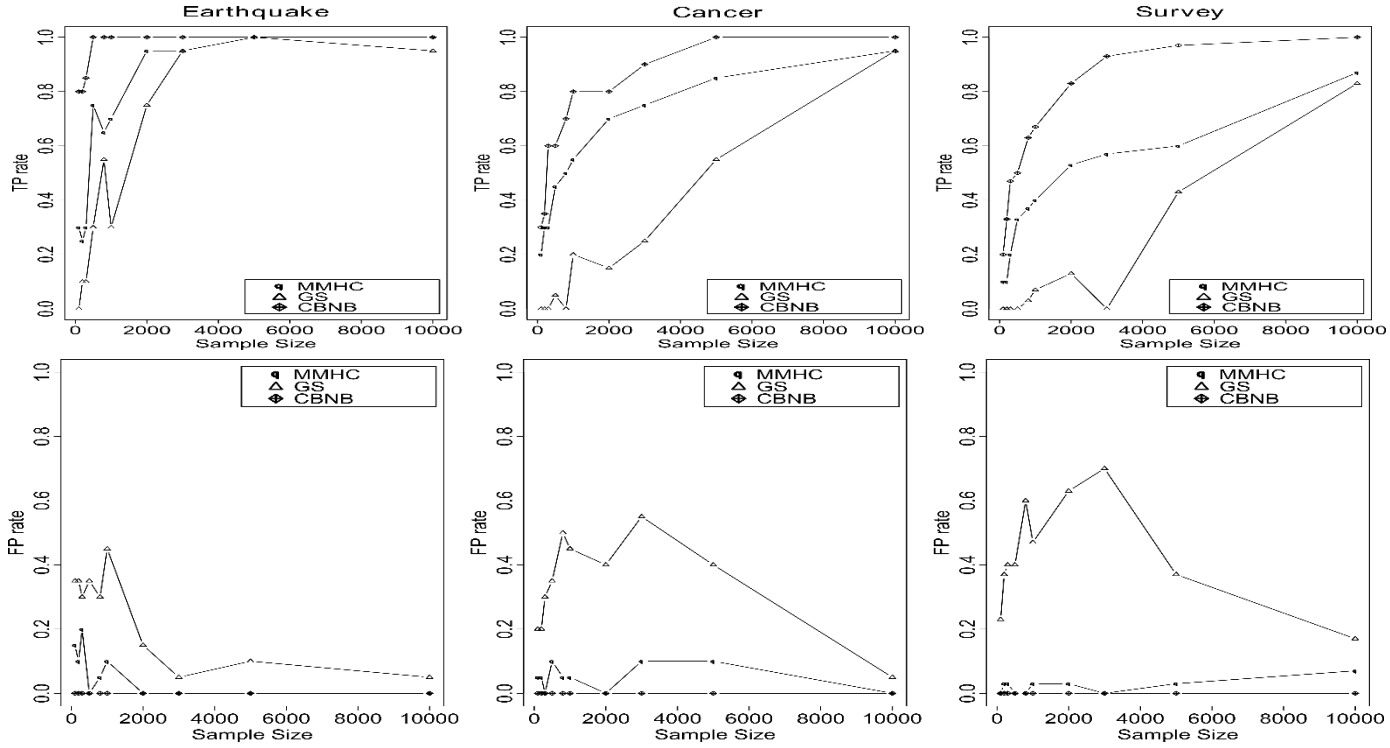

Fig. 4. Taux de TP et FP des algorithmes GS, MMHC et CBNB

IV.2. Évaluation des performances de CBNB

Dans l'implémentation de l'algorithme CBNB, le degré d'association entre deux nœuds est évalué par des tests d'indépendance. Ces tests d'indépendance sont réalisés avec la fonction *ci.test* de bnlearn. Le test d'information mutuel a été utilisé. Nous avons étudié les performances (TP, FP) de CBNB en fonction de la valeur p (l'erreur de type I) du test. Une grande

valeur de p signifie que la dépendance n'est pas significative. Pour décider si oui ou non il faut mettre un arc entre deux nœuds, un seuil de p est fixé. Lorsque la valeur de p est inférieure à ce seuil, alors la dépendance entre les deux nœuds est jugée significative. Les valeurs α= 0,01 ou 0,05 sont les valeurs les plus souvent considérées comme seuil pour le rejet de l'hypothèse nulle d'indépendance conditionnelle. α= 0,05 est la valeur par défaut du package bnlearn. Nous avons testé des valeurs autour de cette valeur par défaut. Pour chaque réseau, nous avons généré différentes tailles d'échantillon. Nous avons ensuite utilisé notre algorithme pour reconstruire le réseau à partir des données générées. Les résultats obtenus sur les 3 réseaux sont présentés à la figure 3.

En considérant chaque valeur de seuil et pour les 3 réseaux, nous observons que le taux de TP s'accroît lorsque la taille de l'échantillon augmente. Ce qui différencie les différentes valeurs de seuils est la taille de l'échantillon à partir de laquelle l'algorithme CBNB réalise 100% de taux de TP. Plus le seuil est petit, plus cette taille d'échantillon est grande. Mais augmenter la valeur du seuil signifie augmenter la probabilité de l'erreur de type I. Ce qui impliquerait une augmentation du taux de FP. Une grande valeur de seuil de p, généralement au-delà de 0,1, ne permet pas de décider quant au rejet de l'hypothèse nulle [20]. C'est pourquoi, nous suggérons de fixer le seuil à 0,01 si la taille de l'échantillon est importante relativement au nombre de nœuds m. Pour des réseaux différents, l'efficacité des algorithmes peut dépendre aussi de la densité du réseau. En général, le TP a un comportement hyperbolique convergeant asymptotiquement vers 1 pour $^{n}/_{m} > 1$ [19]. Lorsque la taille de l'échantillon est relativement faible, le seuil pourrait être fixé à 0,05. Pour toutes les valeurs de seuil testées, l'algorithme CBNB n'a pas produit de FP sur les 3 réseaux benchmark utilisés. Mais nous avons observé des FP sur d'autres benchmark.

IV.3. Résultats des études comparatives

Dans cette partie, nous comparons les performances de l'algorithme CBNB avec celles obtenues avec les algorithmes GS et MMHC. GS [21] est un exemple d'algorithme d'apprentissage à base de contrainte et MMHC [8] est un algorithme hybride. L'algorithme GES (Greedy Equivalent Search) est l'un des algorithmes de recherche les plus performants utilisés par les méthodes à base de score. Mais cet algorithme n'était pas implémenté dans le package bnlearn au moment de cette étude. Pour cette étude comparative, nous avons fixé le seuil des valeurs de p du test d'information mutuelle à 0,05. Nous avons conservé les valeurs par défaut pour les algorithmes GS et MMHC. Les résultats indiquent que le taux de TP des 3 algorithmes convergent vers 1. Le taux de FP est nulle pour CBNB et tends vers 0 pour GS et MMHC. Comme attendu, nous observons que MMHC est plus efficace que GS. Nous observons aussi que CBNB obtient de meilleurs résultats comparativement à GS et MMHC. CBNB est plus efficient car il obtient de bons résultats avec des tailles d'échantillon faibles et il effectue moins de calculs. Cette efficience est à mettre au compte des connaissances expertes incorporées.

En général, les performances des algorithmes d'apprentissage augmentent lorsque la taille de l'échantillon augmente. Comme mentionné précédemment, dans les applications réelles, le nombre de variables peut être très important et de différents types. Ces variables peuvent être les paramètres machines, les caractéristiques des matières premières, d'autres paramètres process, des défauts et des symptômes. Obtenir des données complètes pour l'apprentissage de la structure n'est pas aisé en pratique. De plus, fournir un échantillon de grande taille pour ces données n'est pas toujours possible. Un algorithme efficace sur les échantillons de taille relativement faible constitue donc un résultat utile.

V. CONCLUSION

Les réseaux bayésiens possèdent des caractéristiques intéressantes pour le développement des fonctions de diagnostic. Cependant, la construction du réseau à partir des données reste encore un défi. L'apprentissage des réseaux bayésiens à partir des données est un problème NP difficile et la performance des heuristiques existantes dépend de la nature et de la taille des données d'apprentissage. L'introduction des connaissances expertes permet de faciliter l'apprentissage de la structure.

Dans ce papier, nous avons présenté l'algorithme CBNB pour la construction des réseaux bayésiens causaux. Cet algorithme s'applique aux réseaux ayant la configuration que nous avons appelée la «répartition en cascade ». L'algorithme se déroule en 2 phases. La première phase consiste à répartir les variables en différents niveaux de causalité et à ordonner ces niveaux. Dans la seconde phase, basées sur les données de traçabilité unitaire, l'existence de dépendance entre nœuds est évaluée. Nous avons réalisé des études comparatives sur des données synthétiques générées à partir des réseaux bayésiens benchmark. Ces études ont impliqué les algorithmes GS et MMHC. Les résultats de ces études montrent que l'algorithme CBNB proposé est plus efficace et donne de meilleurs résultats. Mais ces performances sont obtenues en contreparties de l'incorporation d'importantes connaissances expertes. Les résultats obtenus sur ces données synthétiques doivent être confirmées sur des données réelles fournies par nos partenaires du projet FUI Traçaverre.

Remerciements

Ce travail s'inscrit dans le cadre du projet FUI Traçaverre financé en partie par la Bpifrance.

RÉFÉRENCES

1. Diallo, T.M.L., S. Henry, and Y. Ouzrout, *Using Unitary Traceability for an Optimal Product Recall*, in *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World*, B. Grabot, et al., Editors. 2014, Springer Berlin Heidelberg. p. 159-166.
2. Correa, M., C. Bielza, and J. Pamies-Teixeira, *Comparison of Bayesian networks and artificial neural networks for quality detection in a machining process.* Expert Syst. Appl., 2009. **36**(3): p. 7270-7279.

3. Diallo, T.L., S. Henry, and Y. Ouzrout, *Using Unitary Traceability for an Optimal Product Recall*, in *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World*, B. Grabot, et al., Editors. 2014, Springer Berlin Heidelberg. p. 159-166.
4. Verron, S., T. Tiplica, and A. Kobi, *Fault diagnosis of industrial systems by conditional Gaussian network including a distance rejection criterion.* Engineering Applications of Artificial Intelligence, 2010. **23**(7): p. 1229-1235.
5. Verron, S., J. Li, and T. Tiplica, *Fault detection and isolation of faults in a multivariate process with Bayesian network.* Journal of Process Control, 2010. **20**(8): p. 902-911.
6. Cheng, J., et al., *Learning Bayesian networks from data: An information-theory based approach.* Artificial Intelligence, 2002. **137**(1–2): p. 43-90.
7. Ramirez, V.J.C. and A.S. Piqueras. *Learning Bayesian Networks for Systems Diagnosis*. in *Electronics, Robotics and Automotive Mechanics Conference, 2006*. 2006.
8. Tsamardinos, I., L. Brown, and C. Aliferis, *The max-min hill-climbing Bayesian network structure learning algorithm.* Machine Learning, 2006. **65**(1): p. 31-78.
9. Friedman, N., et al., *Learning bayesian network structure from massive datasets: the «sparse candidate« algorithm*, in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. 1999, Morgan Kaufmann Publishers Inc.: Stockholm, Sweden. p. 206-215.
10. Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. 1988: Morgan Kaufmann Publishers.
11. Korb, K.B. and A.E. Nicholson, *Bayesian Artificial Intelligence*. 2003: Taylor & Francis.
12. Spirtes, P., C.N. Glymour, and R. Scheines, *Causation, Prediction, and Search*. 2000: MIT Press.
13. Riascos, L.A.M., M.G. Simoes, and P.E. Miyagi, *A Bayesian network fault diagnostic system for proton exchange membrane fuel cells.* Journal of Power Sources, 2007. **165**(1): p. 267-278.
14. Chen, B., et al., *Bayesian network for wind turbine fault diagnosis*, in *EWEA 2012*. 2012, European Wind Energy Association: Copenhagen, Denmark.
15. Weidl, G., A.L. Madsen, and S. Israelson, *Applications of object-oriented Bayesian networks for condition monitoring, root cause analysis and decision support on operation of complex continuous processes.* Computers & Chemical Engineering, 2005. **29**(9): p. 1996-2009.
16. Dey, S. and J.A. Stori, *A Bayesian network approach to root cause diagnosis of process variations.* International Journal of Machine Tools and Manufacture, 2005. **45**(1): p. 75-91.
17. Przytula, K.W. and D. Thompson. *Construction of Bayesian networks for diagnostics*. in *Aerospace Conference Proceedings, 2000 IEEE*. 2000.
18. Tsamardinos, I., C. Aliferis, and E. Statnikov. *Algorithms for Large Scale Markov Blanket Discovery*. in *In The 16th International FLAIRS Conference, St*. 2003.
19. Scutari, M. and R. Nagarajan, *Identifying significant edges in graphical models of molecular networks.* Artificial Intelligence in Medicine, 2013. **57**(3): p. 207-217.
20. Wasserman, L., *All of Statistics: A Concise Course in Statistical Inference*. 2004: Springer.
21. Margaritis, D., *Learning Bayesian Network Model Structure from Data*, in *School of Computer Science*. 2003, Carnegie Mellon University. p. 126.