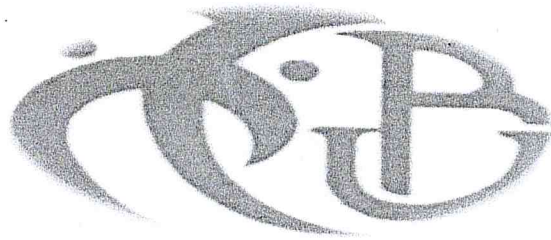


REPUBLICUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE MINISTRE DE
L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE UNIVERSITE
SAAD DAHLEB - BLIDA 1 FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE



*Mémoire de fin d'études proposé pour l'obtention d'un master en
Informatique
Option : Ingénierie de logiciel*

***Thème : Imputation en temps réel de données
manquantes dans un contexte de Big Data***

Promotrice : PhD Zahra Fatma Zohra

Présenté par :

Benachour abderrahim
Azzouz nourelhouda

Devant le jury composé de

: Présidente : mme oukid Salika
: Examinatrice
: Examinatrice mme ELghes

MA-004-388-1

Blida, Septembre 2017

Remerciements



Tout d'abord nous tenant à remercier notre bon dieu le tout puissant et miséricordieux de nous avoir donné la force et le courage de faire ce modeste travail.

A la clôture de ce travail nous tenons à exprimer dans un premier temps, nos remerciements les plus sincères...

A Notre Promotrice, M^{lle} ZAHRA
pour ces précieux conseils, sa disponibilité, la confiance qu'il nous a toujours témoigné et la sollicitude dont il nous a entouré, et ce tout au long de l'élaboration du présent travail.

Nous n'oublions pas non plus Nos Enseignants, qui tout au long du cycle d'études à Université Saâd Dahleb, Blida 1 Département d'Informatique, nous ont transmis leur savoir.

Finalement nous adressons nos plus sincères remerciements à tous nos proches et amis qui nous ont toujours soutenu et encouragés au cours de la réalisation de ce projet ainsi qu'à nos professeurs qui ont contribué à la réussite de cette formidable année universitaire.

Mille Merci.

Dédicaces

Je dédie ce travail :

A mes chers parents, pour tous leurs sacrifices, leur amour,

leur tendresse, leur soutien et leurs prières tout au long de mes chères frères mes études

A mes chère frère : souhaib et lokame

A ma sœur Soumia

A mes oncles et mes tantes

A tous mes cousins et cousines

A tous ceux qui me sont chers et proches

Dédicaces

Je dédie ce modeste travail comme un témoignage d'affection, de respect et d'admiration :

Au bon dieu

À mes très chers parents,

Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous

Avez fait pour moi, Je ferai de mon mieux pour rester un

Sujet de fierté à vos yeux avec l'espoir de ne jamais vous décevoir.

À mes frères Mohamed, chemseddine, abdalwahab et Abderrahmane

Et ma sœur wissam, mes belles sœurs ma cousine nassiba

Mes chers neveux Amir  et Rassim

À ma tante et toute ma famille et à tous ma chers amis Khadidja

Qu'ils trouvent dans ce travail l'expression de mon profond respect et reconnaissance.

NOUR

Résumé

L'imputation des données est une technique utilisée en fouille de données, elle s'appuie sur des principes simples relativement simple. Son objectif est de remplacer les valeurs incomplètes par les valeurs les plus probables. Les algorithmes classiques de l'imputation devenus inadéquats avec l'émergence de big data (données massives ou de masse). Ainsi notre travail consiste à appliquer une méthode intelligence, distribuée, parallèle sur réseaux bayésien dans le but d'estimation des valeurs complètes dans le contexte big data.

Mot clé : l'imputation des données, données manquantes, fouille de données, données de masse, réseaux bayésien, apprentissage incrémental, inférence parallèle.

Abstract

Data imputation is a technique used in data mining, it relies on simple principles relatively simple. Its objective is to replace the incomplete values with the most probable values. The classical algorithms of imputation have become inadequate in the emergence of big data (mass or mass data). Thus our work consists in applying intelligence, distributed, parallel method on Bayesian networks in order to estimate the values complete in context big data.

Keyword: imputation of data, missing data, data mining, big data, Bayesian networks, incremental learning, parallel inference.

ملخص

احتساب البيانات هي تقنية تستخدم في استخراج البيانات، فإنه تعتمد على مبادئ بسيطة نسبياً. هدفها هو استبدال القيم غير المكتملة مع القيم الأكثر احتمالاً. فالخوارزميات الكلاسيكية للاعتماد تصبح غير كافية في ظهور البيانات الضخمة (الكتلة أو البيانات الجماعية)، وبالتالي يتكون عملنا من تطبيق أسلوب استخباراتي موزع وموزع على شبكات بايزي من أجل تقدير القيم كاملة في سياق البيانات الكبيرة.

الكلمة الرئيسية: حساب البيانات، البيانات المفقودة، استخراج البيانات، البيانات الكبيرة، شبكات بايزي، التعلم التدريجي، الاستدلال الموازي.

Sommaire

Sommaire

Introduction Générale	1
1. Contexte et problématique	1
2. Objectif	1
3. Organisation de mémoire	2
Chapitre I :	3
1. Introduction :	4
2. Théorie sur les données manquantes	4
3. Type des données manquantes :	5
3.1 Missing completely at random (MCAR)	5
3.2 Missing at random (MAR)	5
3.3 Missing not at random (MNAR)	5
4. Méthodes de traitement des données manquantes	6
4.1 Procédures de suppression	7
4.1.1 Étude des cas complets (Listwise deletion)	7
4.1.2 Étude des cas disponibles (Pairwise deletion)	8
4.2 Procédures de remplacement	8
4.2.1 Les méthodes d'imputation	8
4.2.1.1 L'imputation simple	9
4.2.1.2 L'imputation multiple	10
4.2.2 Méthodes supervisées	14
4.2.2.1 Réseaux de neurones	14
4.2.2.2 Clustering des données incomplètes	15
4.2.2.3 Règles d'association, valeurs manquantes et complétion	16
4.2.2.4 Réseau bayésien	16
5. Conclusion :	17
Chapitre II : Les réseaux Bayésiens	18
1. Introduction :	19
2. Représentation de réseau bayésien	20
2.1 Définition :	20
2.2 Théorème de Bayes	21

Sommaire

2.1 Cloudera.....	58
2.2 Eclipse.....	58
2.3 Hadoop (version v2.7) :.....	58
2.4 MapReduce.....	59
3. Teste de résultat :	60
a) Interface pour faire appel à l'administrateur que les données ont atteint le nombre ...	60
Pour lancer l'algorithme incrémentale.....	60
3.1 Apprentissage incrémentale des réseaux bayésiens.....	61
3.2 Inférence parallèle :	65
4. Conclusion.....	67
conclusion générale	69

Liste des figures

Figure 1 : représentation graphique des mécanismes	7
Figure 2 : Les grandes catégories des méthodes pour le traitement des données	8
Figure 3 Exemple de réseau bayésien.	21
Figure 4 : les dimensions de bigdata.....	35
Figure 5 : Architecture de Big Data	37
Figure 6 : Les solutions de stockage	39
Figure 7 : Architecture hiérarchique de bibliothèque logicielle du HADOOP.	40
Figure 8 : L'architecture d'HDFS	41
Figure 9 : Processus d'un traitement MapReduce,	42
Figure 10 : Schéma proposé de framework.....	47
Figure 11 : signal à l'administrateur de lancer l'algorithme incrémental.....	61
Figure 12 : calculer le score	62
Figure 13 : Calculer le score max des candidats.....	63
Figure 14 : la structure optimale de RB	65
Figure 15 : inférence parallèle	66
Figure 17 : pourcentage de données incomplètes	67

Sommaire

3. Apprentissage des Réseaux bayésiens	21
3.1 Apprentissage de paramètres	22
3.2 Apprentissage de la structure.....	22
4. Inférence :.....	29
4.1 Les méthodes d'inférence exactes	29
4.2 Les méthodes approchées :	30
5 Conclusion.....	31
Chapitre III: les technologies de	33
Big data	33
1. Introduction :	34
2. Présentation de Big Data.....	34
2.1 Définition de Big Data.....	34
2.2 Caractéristiques du Big Data	34
2.3 Architecture Big Data	36
3. Différence entre BI (Business intelligence) et Big Data.....	37
4. Les principales technologies de Big Data :	38
4.1 Intelligence d'affaires (BI : Business intelligence) :	38
4.2 Bases No SQL.....	38
4.4 Cloud Computing.....	39
4.5 Hadoop :	39
4.6 MapReduce :	42
4.7 Spark :	43
5. Conclusion.....	44
Chapitre IV : imputation des données manquante en temps réel	44
1. Introduction :	45
2. Nettoyage des données manquantes en temps réel basé sur réseau bayésien :	45
2.1 Algorithme K2 (version classique):.....	47
2.2 L'algorithme K2 distribuée incrémentale pour l'apprentissage de structure des RBs :	49
3. Inférence parallèle:.....	54
3.1 Inférence exacte.....	55
4. conclusion.....	56
Chapitre V : Implémentation.....	57
1. Introduction :	58
2. L'environnement de travail :	58

Introduction Générale

1. Contexte et problématique

L'analyse de données est confrontée régulièrement à des données que l'on peut qualifier d'incomplètes ou <Missing data> comme le monde anglophone les appelle. Les problématiques de données manquantes restent très répandues dans les problèmes statistiques et nécessitent une approche particulière malgré la quantité croissante de données disponibles et l'émergence du Big Data. En effet dans les applications, on est très souvent en présence d'observations pour lesquelles on ne dispose pas de l'ensemble des valeurs des variables descriptives, et ceci se produit pour de nombreuses raisons : erreurs de saisie, rubriques non renseignées dans des enquêtes, valeurs aberrantes qu'on préfère supprimer, données recueillies difficilement, statistiques officielles non disponibles, etc.

L'analyse des valeurs manquantes aide à aborder de nombreux problèmes causés par des données incomplètes. Si des observations avec valeurs manquantes sont systématiquement différentes d'observations sans valeurs manquantes, cela peut aboutir à des résultats erronés. De même, les données manquantes peuvent réduire la précision des statistiques calculées car l'information disponible est inférieure à celle initialement prévue. La présence de ces données entraîne un dysfonctionnement du processus de traitement de données. Un autre problème est que les hypothèses effectuées en aval de nombreuses procédures statistiques sont basées sur des observations complètes et que les valeurs manquantes peuvent compliquer la théorie requise.

La plupart des logiciels statistiques suppriment purement et simplement les observations incomplètes, mais si cela n'a pas de conséquences pratiques lorsqu'on dispose de données très nombreuses, cela peut supprimer tout intérêt à l'étude si le nombre de données restantes est trop faible. Pour éviter de supprimer ainsi les données, Plusieurs outils informatiques ont été proposés pour imputer les données manquantes en utilisant plusieurs technologies de modélisation.

2. Objectif

C'est dans ce cadre s'inscrit notre travail. Plus précisément, Le but de notre travail est de proposer une méthode d'imputation de données manquantes dès leur arrivée et dans des environnements de Big Data.

Introduction Générale

Les réseaux Bayésiens sont des outils privilégiés pour les problèmes d'imputation, très attractifs car ils peuvent modéliser efficacement la structure du domaine à étudier. La compréhension facile des réseaux Bayésiens est l'un des plus grands avantages de cette représentation comparées par exemple à celle obtenue par les réseaux neuronaux qui sont plus difficiles à comprendre.

En effet, nous proposons une méthode d'imputation des valeurs manquantes basée sur les réseaux bayésiens. Le but de cette méthode est d'utiliser le réseau bayésien pour imputer les valeurs manquantes qui se base sur le principe de la recherche des valeurs les plus probables des données manquantes. Elle consiste à utiliser les techniques d'apprentissage incrémental développées grâce à ces techniques, Et en utilisant l'inférence parallèle des réseaux Bayésiens, on estime les valeurs à imputées. L'injection par la suite de ces valeurs dans l'ensemble de données incomplètes et la génération par conséquent d'un ensemble de données complètes mais incertaines.

Notre travail doit prendre en considération les caractéristiques des Big Data, de plus l'imputation doit être en temps réel, à l'arrivée des données, Nous nous sommes intéressés au Framework Hadoop. La partie applicative consistait à un déploiement d'Hadoop avec ses composants MapReduce et HDFS dans un environnement distribué à base de la distribution Cloudera.

3. Organisation de mémoire

Ce mémoire a été repartie en 5 chapitres

- Chapitre 1 : consacré au traitement de données incomplètes qui contiendra les différentes méthodes de traitement de données incomplètes. Cette étude nous permettra de faire le choix de notre méthode de traitement..
- Chapitre 2 : est présentera les réseaux bayésiens et leurs différentes méthodes d'inférence et d'apprentissage de la structure et des paramètres
- Chapitre.3 : Ce chapitre présente big data et ses technologies
- Chapitre 4 : dédié à l'approche proposée de notre problème.
- Chapitre 5 : Implémentation. Il expose les méthodes, les outils de développement et les résultats des tests

Et enfin nous terminerons par une conclusion générale références bibliographiques.

Chapitre I :

Nettoyage des données incomplètes

Chapitre 1 : Nettoyage des données incomplètes

1. Introduction :

Les problèmes liés au nettoyage de données sont apparues au début des années 2000 avec l'explosion d'Internet et des entrepôts de données. De nombreuses techniques de traitement des données manquantes ont été développées sans prétendre être exhaustifs, en identifiaient déjà plus d'une vingtaine, pour la plupart issues des recherches en statistique. Depuis, les chercheurs en intelligence artificielle et fouille de données (Data mining), se sont mis à étudier la question et à développer de nouvelles techniques

L'appréhension des données manquantes est un problème difficile. Non pas à cause de sa gestion informatique qui est relativement simple, il suffit de signaler les valeurs manquantes par un code spécifique, mais plutôt à cause des conséquences de leur traitement.

il y a trois stratégies possibles pour traiter des données manquantes :

- (a) Utilisation des procédures de suppression
- (b) Utilisation des procédures de remplacement, (substitution) les données manquantes par les valeurs présentes
- (c) Utilisation des procédures de modélisation de la distribution des données manquantes et

Les estimés par certains paramètres.

2. Théorie sur les données manquantes

Une donnée manquante peut être définie comme une donnée qui était visée par le processus de collecte, mais qui n'a pu être obtenue. Différents éléments peuvent être à l'origine de données manquantes, ils peuvent être associés aux participants, au plan de l'étude ou à une interaction entre les deux. Une donnée peut être manquante au moment du recrutement, de la collecte ou du traitement des données. La donnée manquante peut être un participant, un item d'un questionnaire ou une vague de collecte [2] Toutes ces caractéristiques permettent de décrire les données manquantes pour chacune des analyses menées. Les données manquantes qui nécessitent d'être traitées sont celles qui sont incluses dans le modèle statistique utilisé. Le principe est d'assurer que les relations entre les variables ne soient pas affectées par les données manquantes et que les tests d'hypothèses reflètent bien

Chapitre 1 : Nettoyage des données incomplètes

l'incertitude inhérente aux données manquantes. Souvent, le producteur des données et l'utilisateur final ne sont pas les mêmes personnes. [1]

3. Type des données manquantes :

En plus de définir le type de données manquantes auquel un chercheur fait face, celui-ci doit aussi comprendre le mécanisme qui cause ce type de données. L'efficacité des méthodes de traitement des données manquantes étant directement reliée à ce mécanisme [4]. Les mécanismes expliquant la présence de données manquantes ont été définis la première fois par Rubin en 1976.

3.1 Missing completely at random (MCAR) : Une donnée est MCAR, c'est à dire manquante de façon complètement aléatoire si la probabilité d'absence est la même pour toutes les observations. Cette probabilité ne dépend donc que de paramètres extérieurs indépendants de cette variable. Par exemple : "si chaque participant à un sondage décide de répondre à la question du revenu en lançant un dé et en refusant de répondre si la face 6 apparaît". A noter que si la quantité de données MCAR n'est pas trop importante, ignorer les cas avec des données manquantes ne biaisera pas l'analyse. Une perte de précision dans les résultats est toutefois à prévoir.

3.2 Missing at random (MAR) : Le cas des données MCAR est tout de même peu courant. Il arrive souvent que les données ne manquent pas de façon complètement aléatoire. Si la probabilité d'absence est liée à une ou plusieurs autres variables observées, on parle de missingness at random (MAR). Il existe des méthodes statistiques appropriées qui permettront d'éviter de biaiser l'analyse

3.3 Missing not at random (MNAR) : La donnée est manquante de façon non aléatoire (MNAR) si la probabilité d'absence dépend de la variable en question. Un exemple répandu est le cas où des personnes avec un revenu important refusent de le dévoiler. Les données MNAR induisent une perte de précision (inhérente à tout cas de données manquantes) mais aussi un biais qui nécessite le recours à une analyse de sensibilité. [2]

mecanisme des donnees manquantes

Chapitre 1 : Nettoyage des données incomplètes

Mécanisme	Représentation théorique
MCAR	$P(r_i x_i, y_i, \varphi) = P(r_i \varphi)$
MAR	$P(r_i x_i, y_i, \varphi) = P(r_i x_i, \varphi)$
MNAR	$P(r_i x_i, y_i, \varphi) = P(r_i x_i, y_i^{(0)}, y_i^{(m)}, \varphi)$

Tableau 1: mécanisme des données manquantes [1]

Ou

r_i = indicateur binaire [0.1] de la présence/absence d'une observation

x_i = *covariable(s)*

y_i = Variable d'intérêt (dépendante)

φ = paramètre associé à la probabilité de r_i

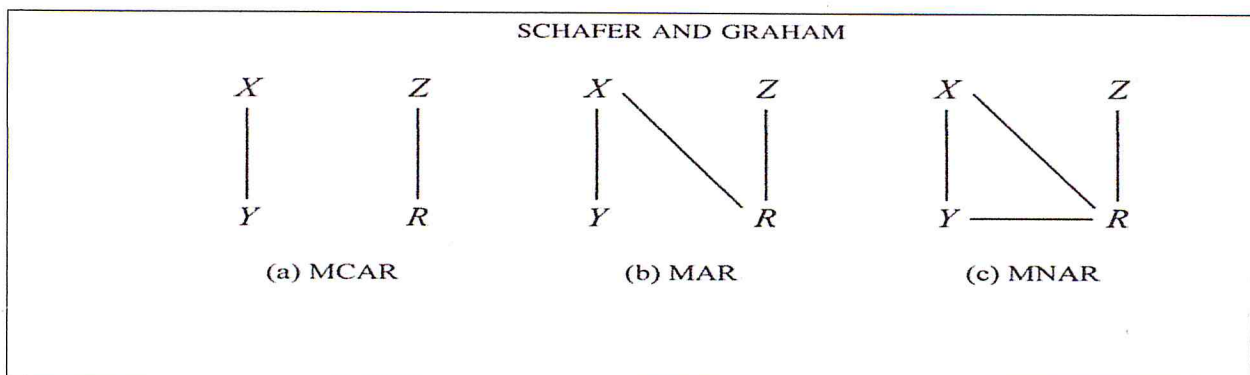


Figure 1: représentation graphique des mécanismes [10].

4. Méthodes de traitement des données manquantes

Les méthodes de traitement des données manquantes se distinguent selon deux approches, les méthodes supprimant les données manquantes et les méthodes utilisant toute l'information disponible. Dans la première catégorie, l'on retrouve les techniques connues sous l'appellation analyse des cas complets (listwise deletion) et analyse des cas complets par paires (pairwise deletion). Certains auteurs suggèrent également que le choix de ne pas inclure dans les analyses les variables qui comportent des données manquantes constitue une forme de suppression des données. Parmi les méthodes utilisant toute l'information disponible, notons l'ajustement par variable binaire, toutes les variantes de l'imputation, le maximum de vraisemblance, l'algorithme EM, le Markov Chain Monte Carlo (MCMC), la pondération

Chapitre 1 : Nettoyage des données incomplètes

ainsi que l'imputation multiple. Il est à noter que cette liste de méthodes n'est pas exhaustive. Les méthodes utilisant toute l'information disponible sont généralement préférées. [1]

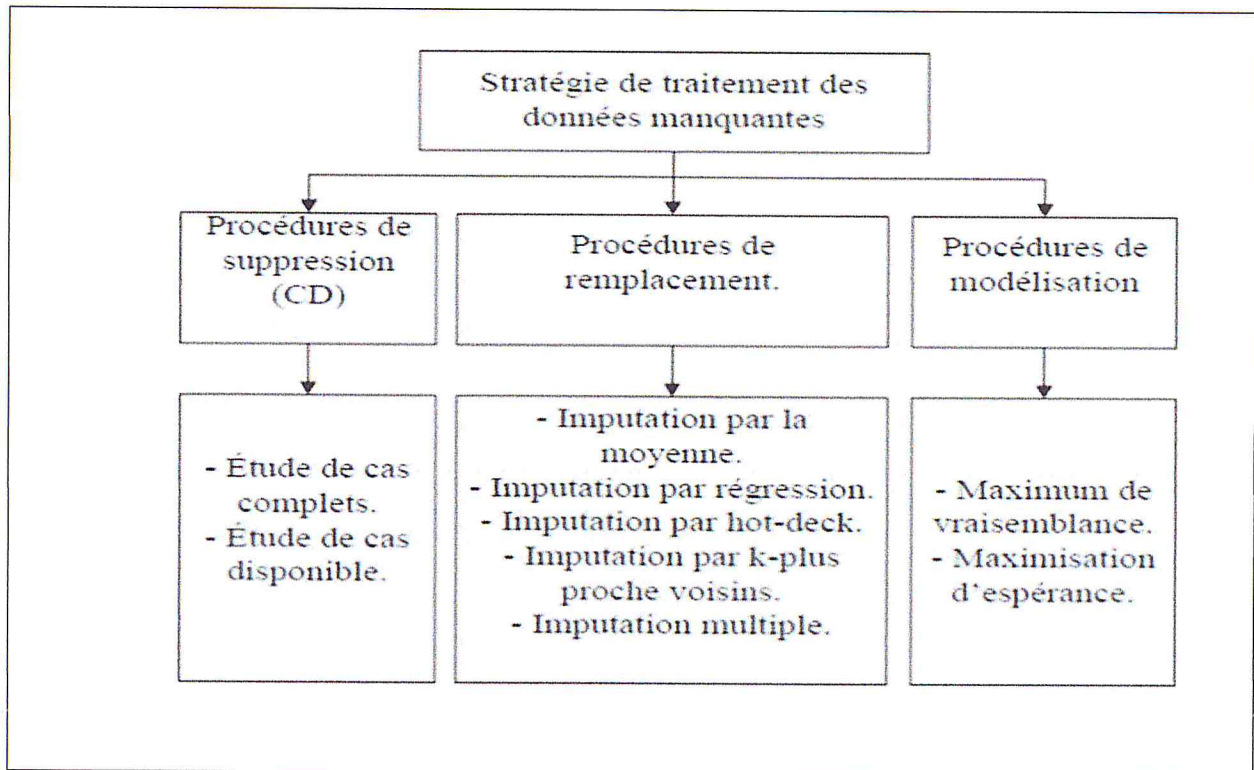


Figure 2: Les grandes catégories des méthodes pour le traitement des données [11].

4.1 Procédures de suppression

4.1.1 Étude des cas complets (Listwise deletion) : Cette méthode par les logiciels d'analyse statistique, omet des analyses tout sujet pour qui au moins une valeur est manquante sur une des variables incluses dans l'analyse demandée. Par exemple, pour une régression multiple avec cinq variables indépendantes, les sujets présentant des valeurs manquantes pour la variable dépendante ou pour une des variables indépendantes seront retirés de l'analyse. L'avantage principal de cette méthode réside dans sa simplicité d'application. Par contre, un des désavantages réside dans la difficulté à comparer les résultats de certaines analyses si les sujets inclus ne sont pas les mêmes d'une analyse à l'autre. Ainsi, suite à une première analyse dans l'exemple de la régression, s'il est décidé de retirer une variable indépendante, car elle n'explique aucune variance supplémentaire, lorsque la régression est effectuée avec quatre prédicteurs au lieu des cinq, le nombre de sujets risque d'être différent pour ces deux analyses. Les résultats d'une analyse de régression à l'autre ne sont alors plus parfaitement comparables.

Chapitre 1 : Nettoyage des données incomplètes

En plus de ce désavantage, les résultats obtenus à l'aide de cette méthode risquent d'être biaisés si les données manquantes ne respectent pas le postulat des données manquantes selon un mécanisme complètement aléatoire. [3]

4.1.2 Étude des cas disponibles (Pairwise deletion) : a pour but de créer une matrice de corrélation ou de variance/covariance pour laquelle chaque couple de variables est pris deux à deux et la corrélation ou variance/covariance est ainsi calculée à partir des données qui sont disponibles. En d'autres mots, le nombre de sujets par coefficient de corrélation va varier selon la distribution des valeurs manquantes dans la matrice de données. L'analyse statistique sera effectuée sur cette matrice plutôt que sur les données brutes. Tout comme pour la méthode précédente, l'avantage principal de cette méthode est sa simplicité d'utilisation. Par contre, une limite importante à la méthode *pairwise deletion* réside dans le fait que les risques sont grands d'obtenir une matrice qui ne soit pas positive et définie [6], empêchant de procéder aux analyses désirées. De plus, si les données manquantes ne sont pas manquantes selon un mécanisme complètement aléatoire, les résultats obtenus sont également biaisés comme dans le cas de la méthode *listwise deletion* [3].

4.2 Procédures de remplacement

4.2.1 Les méthodes d'imputation : L'imputation est une méthode où les données manquantes sont remplacées par des valeurs probables. L'identification des valeurs probables à utiliser est faite selon une panoplie de méthodes. Les méthodes d'imputation sont soit l'imputation simple ou l'imputation multiple. L'imputation simple remplace les données manquantes par une seule valeur [1], Ces méthodes tentent de prédire une valeur pour celles qui sont manquantes, permettant d'effectuer les analyses statistiques sur l'ensemble des sujets inclus dans l'échantillon. Diverses méthodes d'imputation ont été suggérées, par exemple l'imputation par la moyenne de la variable, imputation par régression, l'imputation hotdeck ou l'imputation multiple. L'intérêt majeur de l'imputation multiple qui génère plusieurs matrices de données plutôt qu'une, réside dans le fait qu'elle permet de tenir compte d'une certaine incertitude face aux valeurs imputées. Pour cette raison, [5] déconseillent l'utilisation de méthodes d'imputation simple plutôt que l'imputation multiple. Selon ces auteurs,

Chapitre 1 : Nettoyage des données incomplètes

il est trop facile pour un chercheur « d'oublier » qu'une partie des données utilisées pour ses analyses sont en fait des données non-observées et qu'il réside un certain niveau d'incertitude quant aux données sur lesquelles portent les analyses effectuées.

4.2.1.1 L'imputation simple

Les méthodes d'imputation simple sont généralement celles générant les moins bons résultats. Bien qu'un nombre important de méthodes d'imputation simple aient été développées, l'imputation multiple procure des résultats généralement plus adéquats. Les méthodes d'imputation simple peuvent être regroupées en trois catégories, les méthodes d'imputation par une constante, les méthodes d'imputation par une valeur aléatoire et les méthodes d'imputation par une valeur non aléatoire [1].

a) Le remplacement par la moyenne de la variable

Avec cette procédure, les valeurs manquantes sont remplacées par la moyenne des valeurs observées pour cette même variable. Cette manière de procéder serait encore moins recommandable que l'utilisation de la méthode *listwise deletion* [7]. Il est attendu que, même si les données sont manquantes selon un mécanisme complètement aléatoire, l'estimé de la moyenne de la distribution sera valide, mais, par contre, l'estimé de l'erreur type s'en trouve automatiquement biaisé [5]. En remplaçant les valeurs manquantes par une valeur constante, la variance de cette variable s'en trouve inévitablement réduite. Il s'en dégage, que la valeur de l'erreur type, diminuée par la réduction de la variance et par l'augmentation de la taille de l'échantillon, sera artificiellement plus petite que ce qui aurait dû être observé [7].

b) Le remplacement par équation de régression

Les données manquantes sont imputées à partir des valeurs prédites obtenues à l'aide d'une analyse de régression basée sur les sujets pour lesquels les données sont complètes. Cette méthode présente les mêmes biais que le remplacement par la moyenne de la variable. Dans

ce cas-ci, elle sous-estime la variance parce qu'elle ne prend pas en considération des erreurs résiduelles. Certaines procédures tenant compte d'une part d'erreur résiduelle ont par contre été proposées. Elles seraient beaucoup plus efficaces, mais exclusivement dans le cas où le modèle de régression est adéquat [8]. Si les données manquantes sont manquantes selon un mécanisme non aléatoire, les estimés des paramètres ne sont pas valides. Dans cette

Chapitre 1 : Nettoyage des données incomplètes

situation, le remplacement des valeurs manquantes par des valeurs prédites basées sur un modèle biaisé ne constitue pas un traitement approprié. De plus, en matière d'utilisation de l'information, ces méthodes seraient peu efficaces puisque seulement une partie des données est utilisée pour effectuer l'analyse de régression [9].

c) Le remplacement de type Hot-Deck

Chaque donnée manquante est remplacée par une valeur observée choisie aléatoirement à partir des sujets similaires à celui pour lequel on désire imputer une valeur. Les sujets similaires sont définis comme étant ceux pour lesquels les réponses sont les plus identiques à celles de l'individu présentant une donnée manquante pour un ensemble de covariables. Même si ce type de méthodes préserve les distributions marginales, elles risquent d'altérer les relations entre les variables [8].

d) Méthode des plus proches voisins (KNN) :

La complétion par k plus proches voisins (k-nearest neighbors ou KNN) consiste à suivre l'algorithme suivant :

Algorithme des k plus proches voisins (k-nn)

1. Choix d'un entier k : $1 \leq k \leq n$.
2. Calculer les distances $d(y_{i*} ; y_i)$; $i = 1, \dots, n$
3. Retenir les k observations $y(i_1), \dots, y(i_k)$ pour lesquelles ces distances sont les plus petites.
4. Affecter aux valeurs manquantes la moyenne des valeurs des k voisins :
$$(y_{ij})_{mis} = y_{i*j*} = 1/k(y(i_1) + \dots + y(i_k))$$

Comme pour la classification par KNN, la méthode des plus proches voisins nécessite le choix du paramètre k par optimisation d'un critère. De plus, la notion de distance entre les individus doit être choisie avec précaution. On considérera usuellement la distance Euclidienne ou de Mahalanobis. [2]

4.2.1.2 L'imputation multiple :

L'imputation multiple est une technique d'imputation visant à corriger la sous-estimation de la variance qui est caractéristique des méthodes d'imputation simple. C'est une méthode de plus en plus utilisée dans une variété de disciplines puisqu'elle génère des estimations sans biais lorsque l'analyse est *MAR* et que le modèle d'imputation tient compte de la cause des données manquantes. Les méthodes d'imputation multiple ajoutent un facteur

Chapitre 1 : Nettoyage des données incomplètes

de correction à la variance des estimations afin de tenir compte de l'incertitude de la bonne valeur à imputer. Ce facteur de correction est calculé à partir de la variance inter-imputation et appliqué à l'étape de l'agrégation des paramètres. L'imputation multiple est une expression générale qui comprend plusieurs manières de procéder. Généralement tous les processus d'imputation multiple ont en commun quatre étapes :

1. L'imputation (rappelons que différentes méthodes peuvent être appliquées à cette étape) ;
2. Les analyses
3. L'agrégation des paramètres estimés;
4. Le calcul du taux d'information manquante [1].

De manière générale, l'imputation multiple partage des similitudes avec les méthodes d'imputation simple, telle que l'imputation par régression. Cette dernière méthode remplace une valeur manquante par une valeur prédite à partir d'un modèle spécifique. Comme pour tout modèle de régression, il est constitué d'une part d'erreur. C'est-à-dire qu'il n'est pas assuré que la valeur prédite, et donc imputée, soit bien celle qui est attendue. L'imputation multiple permet de tenir compte de cette incertitude lorsque des valeurs sont imputées à partir d'un modèle théorique de la distribution des données [10].

Le tableau suivant montre la Taxinomie des techniques d'imputation des valeurs manquantes

Technique	Description	Champ d'application	Avantage	Inconvénient
Étude des cas disponibles (Pairwise deletion)	Crée une matrice de corrélation avec les valeurs disponibles (chaque couple de variables est pris deux à deux)	Lorsque les données sont relativement faibles (moins de 10 %)	Préserve davantage les données et est plus précise que la suppression listwise	Corrélation s ou covariances biaisées

Chapitre 1 : Nettoyage des données incomplètes

	deux)			
Imputation par la moyenne totale (Total mean substitution)	Remplacer par la moyenne des valeurs disponibles de la variable, toutes les valeurs manquantes pour la même variable.	Lorsque les corrélations entre les variables sont faibles ($r < .20 $) et le taux des valeurs manquantes moins que 10 %	Préserve la taille de la base de données et la rend facile à utiliser.	La sous-estimation de la variance et de biaiser la corrélation entre les variables (la distribution des données est loin d'être préservée).
Imputation par régression (Régression imputation)	On utilise les valeurs disponibles pour estimer les paramètres d'un modèle de régression, puis utiliser ces paramètres pour estimer la valeur manquante.	Lorsque plus de 20 % des données sont manquantes et les variables sont fortement corrélées.	Préserver l'écart des données estimées par rapport à la moyenne et la forme de la distribution	Distorsions des degrés de liberté et augmentation artificielle des relations entre les variables
Imputation multiple	D'abord, estimation de $m > 1$ ensembles de valeurs plausibles pour	Sous l'hypothèse que les valeurs manquantes sont aléatoires.	L'induction statistique (écart type, p-values, etc.) qui découle de l'IM est	La complexité de calcul des matrices (contraintes

Chapitre 1 : Nettoyage des données incomplètes

	les données manquantes sont créés. Chacun de ces ensembles est utilisé pour remplir les données manquantes et ainsi créer m ensembles complets de données, et elles sont combinées dans une unique base de données récapitulative		généralement valide car elle incorpore l'incertitude engendrée par les données manquantes.	d'espace mémoire et de temps de traitement). Ne permet pas de seulement compléter une base de données... mais oblige à réaliser une analyse statistique.
L'Imputation hot deck (Hot-deck imputation)	Remplace une valeur manquante par la valeur de la même variable à partir d'un cas similaire dans l'ensemble de données	Lorsque la similarité entre les cas est facile à déduire.	Préserve les distributions des variables	Risquent d'altérer les relations entre les variables
Imputation par k plus proches voisins (k-nearest)	Remplace les valeurs manquantes par la valeur du k	Lorsque la mesure de distance entre les k plus proches	Il ne fait aucune supposition quant à la	La difficulté réside dans le choix du paramètre k.

Chapitre 1 : Nettoyage des données incomplètes

neighbors imputation)	plus proche voisin dans l'ensemble de données.	voisins est facile à déduire, et les données sont chronologique.	distribution des données, et de prendre en considération la corrélation entre variables.	
--------------------------	---	---	---	--

4.2.2 Méthodes supervisées :

4.2.2.1 Réseaux de neurones : est constitué d'un graphe pondéré orienté dont les nœuds symbolisent les neurones. Ces neurones possèdent une fonction d'activation (fonction signe, fonction sigmoïde, ...) qui permet d'influencer les autres neurones du réseau. Les liens synaptiques sont les connexions entre les neurones, ces liens propagent l'activité de neurones avec une pondération caractéristique de la connexion (poids synaptique) [35].

Les réseaux de neurones artificiels pour imputer des valeurs appartiennent à la troisième catégorie. Dans ce cas, une valeur manquante est remplacée par la sortie calculée par le réseau. Les autres valeurs corrélées décrivant l'observation sont fournies en entrée du réseau. Dans un premier temps, on apprend au réseau à reconnaître la variable de sortie. Le réseau apprend donc les liens implicites reliant l'ensemble des variables d'entrée à cette variable de sortie. Puis on effectue un test pour vérifier l'apprentissage en définissant au hasard un échantillon de la population des sujets. On vérifie ainsi que le réseau a bien codé les liens et qu'il est capable d'effectuer la reconnaissance. On en déduit alors que cette variable de sortie est bien liée aux autres variables d'entrée, qu'on peut retrouver sa valeur en cas d'absence en utilisant le codage interne au réseau. [14]

- **Procédure d'imputation par réseau de neurones**

En pratique, la méthode d'imputation des données manquantes par réseaux de neurones consiste à réaliser les étapes suivantes [14]:

Chapitre 1 : Nettoyage des données incomplètes

1. Vérifier l'adéquation de la méthode aux besoins. Cette méthode n'est applicable que dans le cas où l'on dispose de peu de sujets présentant des non-réponses à certains items alors que l'on veut disposer de données complètes avec un fort taux de certitude.

2. On sépare un échantillon présentant des sujets avec des données complètes.

3. On identifie l'item présentant des données manquantes et on le qualifie comme variables de sortie.

4. On construit un réseau de type perceptron auquel on apprend par apprentissage supervisé à reconnaître les valeurs de la variable de sortie.

Cet apprentissage nécessite la plupart du temps l'utilisation d'une retro propagation des erreurs. Il convient de faire attention au sur apprentissage.

Dans certains cas en effet, les taux chutent brutalement au cours d'une opération de retro propagation des erreurs. Il faut alors retrouver la configuration du réseau précédent qui devient alors le modèle optimal. On le sauvegarde alors soigneusement. On calcule alors le taux de réussite en apprentissage. En entrée de ce même réseau, on présente ensuite les réponses aux items d'un sujet présentant une non-réponse sur.

4.2.2.2 Clustering des données incomplètes : Le *clustering* est un outil important pour l'analyse de données. Il vise à trouver les structures intrinsèques des données en les organisant en groupes homogènes et distincts (les *clusters*). Les objets dans un même *cluster* doivent être similaires entre eux et différents des objets des autres clusters.

Clustering des données incomplètes Selon. L'analyse de données incomplètes peut se faire grâce au clustering flou. Cette méthode nécessite toutefois de traiter les données incomplètes différemment selon l'origine des valeurs manquantes. La première étape consiste donc à analyser pour un ensemble de données les raisons de la présence de valeurs manquantes. Dans un deuxième temps, on recherche les corrélations dans la base. Comme pour le clustering. L'objectif du clustering flou est de diviser un ensemble de données en un ensemble de clusters tels que la similarité intra-classes est nettement supérieure à la similarité interclasses. Cependant, le but est de pouvoir traiter les données qui pourraient appartenir à plusieurs groupes en même temps. On introduit un degré d'appartenance aux différents clusters calculé en fonction de la distance entre cette donnée et le cluster. Chaque cluster peut

Chapitre 1 : Nettoyage des données incomplètes

donc être considéré comme un sous-ensemble flou Cette méthode de traitement des données incomplètes par clustering flou permet de compléter des valeurs manquantes à chaque itération de l'algorithme de clustering, sur le même principe que la complétion proposée par l'algorithme Expectation-Maximization. De plus, pour tenir compte de la différence entre données complètes et incomplètes, on réduit le degré d'appartenance des données incomplètes.[23]

4.2.2.3 Règles d'association, valeurs manquantes et complétion :

Règles d'association, valeurs manquantes et complétion Dans le cadre des techniques de description, des travaux ont été proposés pour la recherche de règles d'association, présentent un algorithme afin de prendre en compte les données incomplètes lors de l'extraction des règles, par omission partielle et temporaire de ces enregistrements. Ces règles peuvent ensuite être utilisées afin de compléter les valeurs manquantes. mettent en œuvre un système d'approximation probabiliste dans lequel une valeur manquante peut prendre plusieurs valeurs lors de la découverte des règles. Ces méthodes approximatives permettent d'extraire des règles proches de celles qui devraient être obtenues sur la base complète, tandis que extrait des représentations condensée exactes. Enfin d'autres méthodes utilisent les règles d'association et certains indices de confiance afin de compléter les valeurs manquantes. [23]

4.2.2.4 Réseau bayésien :

Les réseaux bayésiens constituent l'un des formalismes les plus complets et les plus cohérents pour l'acquisition, la représentation et l'utilisation de connaissances par des ordinateurs. Encore du domaine de la recherche au début des années 90 L'utilisation essentielle des réseaux bayésiens est de calculer des probabilités conditionnelles d'événements relies les uns aux autres, Alors qu'il est possible de faire de l'apprentissage de paramètres de réseaux bayésiens à partir de données incomplètes et que l'inférence dans les réseaux bayésiens est possible même lorsque peu d'attributs sont observés ,les algorithmes d'apprentissage de structure avec des données incomplètes restent rares.

Lorsque les données sont incomplètes, il est possible de déterminer les paramètres et la structure du réseau bayésien à partir des entrées complètes de la base. Comme les données manquantes sont supposées l'être aléatoirement, nous construisons ainsi un estimateur sans biais. Néanmoins, dans l'exemple d'une base de 2000 cas sur 20 attributs, avec une probabilité de 20% qu'une mesure soit manquante, nous ne disposerons en moyenne que de

Chapitre 1 : Nettoyage des données incomplètes

23 cas complets. Les autres données à notre disposition ne sont donc pas négligeables et il serait donc préférable de faire l'apprentissage en utilisant toute l'information à laquelle nous avons accès. Un avantage des réseaux bayésiens est qu'il suffit que seules les variables X_i et $Pa(X_i)$ soient observés pour estimer la table de probabilité conditionnelle correspondante. Dans ce cas, il est alors possible d'utiliser tous les exemples (même incomplets) où ces variables sont observées (dans l'exemple précédent, en supposant que X_i possède trois parents, nous aurions 819 exemples en moyenne pour estimer les paramètres correspondants).[37]

La représentation des connaissances utilisées dans les réseaux Bayésiens est la plus intuitive possible, il s'agit simplement de relier des causes et des effets par des flèches. Cette représentation des connaissances par liens entre causes et effets est souvent plus naturelle que la représentation par règles de production. Ainsi, il est plus facile pour un expert de formaliser ses connaissances sous forme de graphe causal que sous forme de système à base de règles.

5. Conclusion :

Dans ce chapitre, nous avons présenté un état de l'art des principales méthodes dédiées au traitement des valeurs manquantes. Nous nous sommes d'abord intéressés aux modèles d'imputation simple et d'imputation multiple. Ensuite, nous avons présenté des travaux faisant partie du domaine de bases de données, des ensembles approximatifs ainsi que de l'apprentissage supervisé. Dans le chapitre prochain on va voir les réseaux bayésiens et comment les construire.

Chapitre II :

Les réseaux

Bayésiens

Chapitre II : Les réseaux Bayésiens

1. Introduction :

Les Réseaux Bayésiens sont des modèles graphiques qui représentent les relations probabilisées entre un ensemble des variables. Pendant quelques années récentes les Réseaux Bayésiens deviennent un outil populaire pour représenter et manipuler des connaissances expertes dans un système expert.

Premièrement, les Réseaux Bayésiens peuvent représenter intuitivement un domaine de connaissance, beaucoup d'expériences montrent qu'il est souvent plus facile à formaliser les connaissances sous forme d'un graphe causal que sous forme d'un système basé sur les règles.

Deuxièmement, les Réseaux Bayésiens peuvent gérer l'ensemble des données incomplètes. Et

de plus les Réseaux Bayésiens permettent d'apprendre la relation causale qui peut nous aider de faire des décisions. Par exemple un commerçant veut savoir s'il faut faire plus de la publicité pour obtenir plus de profits. En utilisant les Réseaux Bayésiens on peut facilement répondre à cette question.

Selon le type d'application, l'utilisation pratique des réseaux bayésiens peut être envisagée au même titre que celle d'autres modèles comme les réseaux de neurones, les systèmes experts, les arbres de décision, etc. Mais pour quelques aspects, les réseaux bayésiens, dans de nombreux cas, sont préférables à d'autres modèles. On peut citer, l'acquisition des connaissances c'est à dire les réseaux bayésiens donnent la possibilité de rassembler et de fusionner des connaissances de diverses natures dans un même modèle, ils permettent de faire un retour d'expérience (données historiques ou empiriques), une expertise (exprimée sous forme de règles logiques, d'équations, de statistiques ou de probabilités subjectives), des observations. Dans le monde industriel, chacune de ces sources d'information est souvent insuffisante individuellement pour fournir une représentation précise et réaliste du système analysé.

Les réseaux bayésiens permettent la représentation des connaissances puisque la représentation graphique d'un réseau bayésien est explicite, intuitive et compréhensible par un non-spécialiste, ce qui facilite à la fois la validation du modèle, ses évolutions éventuelles et surtout son utilisation. Typiquement, un décideur est beaucoup plus enclin à s'appuyer sur un modèle dont il comprend le fonctionnement qu'à faire confiance à une «boîte noire».

Chapitre II : Les réseaux Bayésiens

2. Représentation de réseau bayésien

2.1 Définition :

Formellement, un réseau bayésien (qui est aussi appelé réseau probabiliste, ou encore, en anglais, “Bayésien network”, BBN : “Bayésien Belief Network”, “probabilistic network”) est un graphe dont les nœuds représentent des variables aléatoires et dont les arcs expriment des interdépendances, des influences entre ces variables. Le graphe modélise donc qualitativement des influences entre variables aléatoires. C’est en fait l’absence d’arc entre deux variables qui certifie l’absence d’influence directe entre elles. Pour quantifier la “force” de ces influences, on adjoint à chaque nœud sa loi de probabilité conditionnellement à ses parents dans le graphe [15] Par exemple, le RB de la figure 1, emprunté à [16], modélise le comportement.

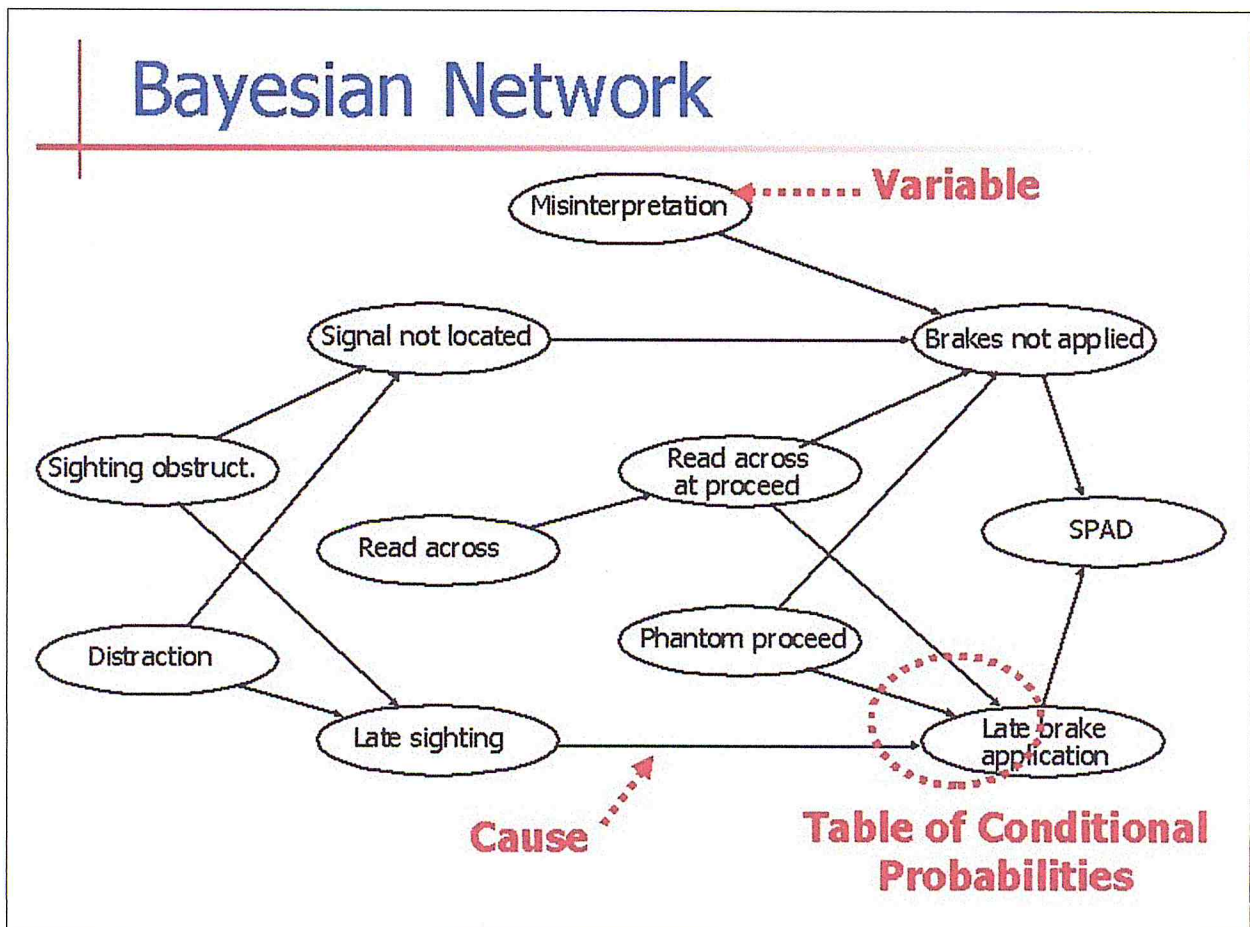


Figure 3: Exemple de réseau bayésien. [16]

Chapitre II : Les réseaux Bayésiens

La théorie des réseaux bayésien résulte d'une fusion entre la théorie des probabilités et la théorie des graphes [17]. On définit classiquement un réseau bayésien comme un graphe acyclique dirigé. Il est formé d'un ensemble de variables et d'un ensemble d'arcs entre les variables. Chaque variable correspond à un nœud du réseau. A chaque variable X_i ayant

pour parents l'ensemble : $Pa(X_i)$, on associe une probabilité conditionnelle $P(X_{ij} | Pa(X_i))$.

Dans les réseaux bayésien, la probabilité jointe du modèle s'écrit ainsi :

$$P(N_1; \dots N_n) = \prod_{1..n} (P(X_{ij} | pa(X_i))) \text{ [18].}$$

2.2 Théorème de Bayes

Le théorème de Bayes permet de calculer la probabilité d'une variable sachant la valeur d'une autre variable de la manier suivante :

$$P(A / B) = \frac{P(A \cap B)}{P(B)}$$

C'est la loi de bayes qui confère une caractéristique symétrique aux probabilités conditionnelles (possibilité d'inversion), ce qui permet de calculer les probabilités a posteriori probabilités a priori de certaine autres variables.

3. Apprentissage des Réseaux bayésiens

Pour définir un réseau Bayésien, il faut définir la structure et les paramètres du réseau. L'apprentissage de structure consiste à trouver le meilleur graphe représentant le problème à résoudre. Dans certains cas, la structure du réseau Bayésien est fournie a priori par un expert. Par contre, la détermination de cette structure à partir de données est une problématique NP-difficile pour laquelle de nombreuses méthodes d'apprentissage automatique ont été proposées. Ces méthodes peuvent être classées en deux familles. Le principe de la première famille est la détermination, à partir des données, des relations d'indépendance conditionnelle entre les variables en utilisant des tests statistiques. Le nombre des tests statistiques croît exponentiellement avec le nombre de nœuds. On préfère donc maximiser des fonctions de score, ce qui est le but de la deuxième famille. L'apprentissage des paramètres est une étape primordiale dans les réseaux Bayésiens. Elle consiste à supposer que la structure du réseau a été fixé et à estimer les distributions de probabilités (ou les lois des paramètres

Chapitre II : Les réseaux Bayésiens

correspondants) à partir d'une base d'exemples. Elle peut être effectuée avec un simple apprentissage statistique ou bayésien si la base est complète, et avec l'algorithme EM dans le cas de données manquantes [19].

3.1 Apprentissage de paramètres :

On entend par paramètres, les probabilités conditionnelles ou la partie quantitative du réseau Bayésien. Ce type d'apprentissage consiste à estimer les distributions de probabilités à partir de données disponibles complètes ou incomplètes : Cas de données complètes : Apprentissage statistique : Cette méthode repose sur l'estimation d'un événement par sa fréquence d'apparition dans la base de données. Apprentissage Bayésien : Cette méthode consiste à trouver les paramètres θ les plus probables en utilisant des a priori sur ces paramètres. Cas de données incomplètes : Généralement, les données sont souvent incomplètes. Certaines variables ne sont jamais observées ou ne le sont que partiellement. Les méthodes d'estimation les plus utilisées sont fondées sur l'algorithme itératif Expectation-Maximisation (EM). Apprentissage statistique : L'algorithme EM s'applique à la recherche des paramètres en répétant jusqu'à convergence de l'espérance et la maximisation. Apprentissage Bayésien : Il suffit de remplacer le maximum de vraisemblance de l'étape de maximisation par un maximum (ou une espérance) a posteriori. [20]

3.2 Apprentissage de la structure

les relations entre les variables, c'est-à-dire la topologie du RB, étaient considérées comme connues, spécifiées par un expert ou par le bon sens. Pour au moins certaines parties du réseau, en Modélisation de l'Utilisateur, cet a priori semble cohérent. En effet, les notions caractérisant l'utilisateur dans le RB et leurs relations sont des constructions de l'enseignant ou du domaine, rarement des données objectives. Par exemple, les concepts de "connaissance du domaine" et de "connaissance de sous-domaine" ainsi que leurs relations (supposées souvent hiérarchiques) sont créés par l'enseignant. On peut toutefois considérer que les relations entre variables dans un sous-système observable du RB ne sont pas parfaitement définies et méritent d'être apprises automatiquement (surtout si elles sont susceptibles de changer).[15]

Chapitre II : Les réseaux Bayésiens

Depuis plus de 20 ans de nombreuses stratégies ont été mises en place afin d'approcher au plus près la solution optimale, celles-ci sont généralement classées en deux catégories.

a. Recherche d'indépendances conditionnelles

Cette première série d'approches d'apprentissage de structure, souvent appelée recherche sous contraintes, est issue des travaux de deux équipes « concurrentes », Pearl et Verma d'une part avec les algorithmes IC et IC*, Spirtes, Glymour et Scheines de l'autre avec les algorithmes SGS, PC, CI, FCI. Citons de même, plus récemment, l'algorithme BN-PC de Cheng et al. Ces algorithmes sont tous basés sur un principe identique :

- construire un graphe non dirigé contenant les relations entre les variables, à partir de tests d'indépendance conditionnelle,
- détecter les V-structures (en utilisant aussi des tests d'indépendance conditionnelle),
- « propager » les orientations de certains arcs,
- prendre éventuellement en compte les causes artificielles dues à des variables Latentes [21]

• Algorithmes PC et IC

Une des premières méthodes de recherche d'indépendances conditionnelles efficace proposée fût celle de l'algorithme PC (pour Peter and Clark, les prénoms des inventeurs de la méthode). Elle a été introduite par Spirtes, Glymour & Scheines . Elle utilise un test statistique (X^2 ou g^2 originellement) pour évaluer s'il y a indépendance conditionnelle entre deux variables. Il est alors possible de reconstruire la structure du réseau bayésien à partir de l'ensemble des relations d'indépendances conditionnelles découvertes. En pratique, un graphe complètement connecté sert de point de départ, et lorsqu'une indépendance conditionnelle est détectée à l'aide d'un test statistique, l'arc correspondant est retiré. Les tests statistiques sont effectués suivant un ordre donné préalablement aux variables. Dash & Druzdzel ont alors montré que cet ordre avait une grande influence sur le résultat rendu par cette méthode. [22]

Chapitre II : Les réseaux Bayésiens

Algorithme PC

- Construction d'un graphe non orienté

Soit $\mathcal{G}$ le graphe reliant complètement tous les nœuds $\mathcal{X}$

$i \leftarrow 0$

Répéter

Recherche des indépendances cond. d'ordre i

$\forall \{X_A, X_B\} \in \mathcal{X}^2$ tels que $X_A - X_B$ et $Card(Adj(\mathcal{G}, X_A, X_B)) \geq i$

$\forall S \subset Adj(\mathcal{G}, X_A, X_B)$ tel que $Card(S) = i$

si $X_A \perp X_B \mid S$ alors

- suppression de l'arête $X_A - X_B$ dans $\mathcal{G}$
- $SepSet(X_A, X_B) \leftarrow SepSet(X_A, X_B) \cup S$
- $SepSet(X_B, X_A) \leftarrow SepSet(X_B, X_A) \cup S$

$i \leftarrow i + 1$

Jusqu'à $Card(Adj(\mathcal{G}, X_A, X_B)) < i, \forall \{X_A, X_B\} \in \mathcal{X}^2$

- Recherche des V-structures

$\forall \{X_A, X_B, X_C\} \in \mathcal{X}^3$ tels que $\neg \overline{X_A X_B}$ et $X_A - X_C - X_B$,

si $X_C \notin SepSet(X_A, X_B)$ alors on crée une V-structure :

$X_A \rightarrow X_C \leftarrow X_B$

- Ajout récursif de $\rightarrow$

Répéter

$\forall \{X_A, X_B\} \in \mathcal{X}^2$,

si $X_A - X_B$ et $X_A \rightsquigarrow X_B$, alors ajout d'une flèche à X_B :

$X_A \rightarrow X_B$

si $\neg \overline{X_A X_B}$, $\forall X_C$ tel que $X_A \rightarrow X_C$ et $X_C - X_B$ alors $X_C \rightarrow X_B$

Tant qu'il est possible d'orienter des arêtes

Notations :

$\mathcal{X}$	ensemble de tous les nœuds
$Adj(\mathcal{G}, X_A)$	ensemble des nœuds adjacents à X_A dans $\mathcal{G}$
$Adj(\mathcal{G}, X_A, X_B)$	$Adj(\mathcal{G}, X_A) \setminus \{X_B\}$
$X_A - X_B$	il existe une arête entre X_A et X_B
$X_A \rightarrow X_B$	il existe un arc de X_A vers X_B
$\overline{X_A X_B}$	X_A et X_B adjacents $X_A - X_B$, $X_A \rightarrow X_B$ ou $X_B \rightarrow X_A$
$X_A \rightsquigarrow X_B$	il existe un chemin dirigé reliant X_A et X_B

Algorithme de pc [21].

Chapitre II : Les réseaux Bayésiens

La première étape de l'algorithme PC (recherche d'indépendances conditionnelles) est l'étape la plus coûteuse de l'algorithme. Spirtes et al. ont suggéré plusieurs simplifications ou heuristiques afin de diminuer cette complexité.[21]

Dash & Druzdzel introduisent une méthode qu'ils nomment PC*, dont la procédure de recherche de structure est la même que celle de l'algorithme PC. Par contre, les tests statistiques sont basés sur une méthode originale nommée Hybrid Independence Test qui évalue les statistiques essentielles de la base d'exemples et qui utilise alors ces statistiques essentielles pour effectuer les tests statistiques dont a besoin

une méthode comme PC.[21]

– Dans l'algorithme PC*, ils proposent de ne plus parcourir tous les S possibles, mais seulement les ensembles de variables adjacentes à X_A ou X_B qui sont sur un chemin entre X_A et X_B . Cette solution est malheureusement inutilisable avec un trop grand nombre de variables puisqu'elle équivaut à stocker tous les chemins possibles dans le graphe.

– Trois heuristiques permettent d'accélérer l'algorithme PC en choisissant judicieusement les noeuds X_A et X_B et l'ensemble S :

– PC-1 : les couples de variables $\{X_A X_B\}$ et les ensembles S possibles sont parcourus dans l'ordre lexicographique.

– PC-2 : les couples de variables $\{X_A X_B\}$ sont testés dans l'ordre croissant de la statistique utilisée pour le test d'indépendance (des moins dépendants aux plus dépendants). Les ensembles S sont parcourus dans l'ordre lexicographique.

– PC-3 : pour une variable X_A fixée, sont testés d'abord les X_B les moins dépendants à X_A conditionnellement aux ensembles S les plus dépendants à X_A . [5]

• **L'algorithme IC** : Un algorithme de principe similaire (IC, pour Inductive Causation) a été introduit à la même époque par Pearl & Verma. La principale différence entre ces deux algorithmes est que l'algorithme PC est initialisé avec un graphe complètement connectée raisonne par cartes d'indépendances successives jusqu'à obtenir une I-map minimale tandis que l'algorithme IC est initialisé avec un graphe vide et raisonne par cartes de dépendances successives jusqu'à obtenir une D-map maximale [22].

- Etape 0 : Génération du graphe non orienté reliant tous les nœuds :

- Etape 1a : Suppression des indépendances conditionnelles d'ordre 0 :

Chapitre II : Les réseaux Bayésiens

Etape 1b : Suppression des indépendances conditionnelles d'ordre 1

- Etape 1c : Suppression des indépendances conditionnelles d'ordre 2

- Etape 2 : Recherche des V-structures

- Etape 3 : Orientation récursive de certaines arêtes

- Etape 4 : Orientation des arcs restants [21]

- **L'algorithme BN-PC** : Cette méthode de recherche de causalité nommée BN-PC-B reprend le principe de l'algorithme PC en essayant de diminuer le nombre de tests d'indépendance conditionnelle. Elle est constituée de trois phases : Rechercher une structure initiale arborescente Rechercher des ensembles de conditionnement entre les variables pour décider s'il faut relier les nœuds correspondants Essayer de retirer des arcs superflus Le graphe partiellement dirigé obtenu à la fin de la 3ème phase est alors orienté complètement de la même manière que l'algorithme PC.[20]

b. Algorithmes basés sur un score

Contrairement à la première famille de méthodes qui tentait de retrouver des indépendances conditionnelles entre les variables, les approches suivantes vont soit chercher la structure qui maximise un certain score, soit chercher les meilleures structures et combiner leurs résultats.

Pour que ces approches à base de score soient réalisables en pratique, nous verrons que le score doit être décomposable localement, c'est-à-dire s'exprimer comme la somme de scores locaux au niveau de chaque nœud. Se pose aussi le problème de parcours de l'espace B des réseaux bayésiens à la recherche de la meilleure structure. Comme une recherche exhaustive est impossible, les algorithmes proposés travaillent sur un espace réduit (espace des arbres, ordonnancement des nœuds), ou effectuent une recherche gloutonne dans cet espace. [21]

- **Le score BIC**

Le score BIC est un score qui tient compte de la vraisemblance maximale et de la complexité

du modèle. Il a été proposé par Schwartz en 1979

$$\text{BIC}(\beta, D) = \log P(D | \beta, \theta) - \frac{1}{2} \text{Dim}(\beta) \log N$$

Chapitre II : Les réseaux Bayésiens

Où D est notre base d'exemples, θ est la distribution des paramètres obtenue par maximum de vraisemblance pour le réseau β , et où $\text{Dim}(\beta)$ est la dimension du réseau bayésien.

- **Le score MDL**

Le principe MDL (Minimum Description Length), proposé initialement par Rissanen [24], consiste à minimiser la somme de la longueur de codage du modèle et la longueur de codage des données lorsqu'on utilise le modèle pour les représenter. Plusieurs approches sont proposées après

Le score MDL est représenté par :

$$S_{MDL}(B, D) = \log L(D | \theta, \beta) - |A_B| \log N - c \cdot \text{Dim}(B)$$

Avec $|A_B|$ est le nombre d'arcs dans le graphe B et c est le nombre de bits utilisés pour stocker chaque paramètre numérique.

Le score BD (Bayesian Dirichlet) :

En 1992, Cooper & Herskovits [25] ont proposé une approche bayésienne décrite par :

$$\begin{aligned} S_{BD}(B, D) &= P(B, D) = \int_{\theta} L(D | \theta, B) P(\theta | B) p(B) d\theta \\ &= P(B) \int_{\theta} L(D | \theta, B) P(\theta | B) d\theta \end{aligned}$$

Après l'application des hypothèses classiques d'indépendance des exemples, et en utilisant la distribution **a priori** de Dirichlet sur les paramètres, le score BD est :

$$S_{BD}(B, D) = \propto \prod_{i=1}^n \left(\prod_{j=1}^{q_i} \frac{\tau(\alpha_{ij})}{\tau(N_{ij} + \alpha_{ij})} \prod_{k=1}^{r_i} \frac{(N_{ij} + \alpha_{ij})}{\tau(\alpha_{ij})} \right)$$

Le score BDe

Heckerman [26] a montré que le score bayésien n'était pas un score équivalent et en a proposé une variante, **BDe = BD + score équivalent**, qui corrige cela en utilisant une

distribution **a priori** spécifique sur les paramètres du réseau bayésien :

Chapitre II : Les réseaux Bayésiens

$$\alpha_{ijk} = N' * P(X_i = x_k, \text{parent}(X_i) = x_j | B_c)$$

Où c B est un graphe complètement relié (n'encodant aucune indépendance conditionnelle)

et N' est un nombre d'exemples "équivalent" défini par l'utilisateur.

Si chaque instance de X_i est équiprobable (cond. à B_c), on obtient :

$$\alpha_{ijk} = \frac{N'}{r_i q_i}$$

Le score BDe utilisant ces α_{ijk} est souvent appelé score BDeu.

- **Le score MCMC**

L'algorithme Markov Chain Monte Carlo (MCMC) génère les événements en faisant un changement aléatoire à l'événement précédent. L'algorithme maintient donc un état courant où toutes les variables ont une valeur.

$$P(X_i | X_{\{i \neq j\}})$$

Pour générer le prochain état doit:

- Choisir une variable qui n'est pas une variable d'évidence.
 - La distribution de cette variable dépend des valeurs des variables dans son Markov Blanket. Cette méthode n'est pas limitée aux distributions exponentielles conjuguées. Pour les modèles graphiques, ces distributions conditionnelles ne dépendent que des variables de couverture de Markov [27].
- **L'algorithme K2** : Parmi les algorithmes les plus utilisés et qui ont prouvé leur efficacité, on trouve l'algorithme K2 proposé par COOPER et HERSOVITS [25]. L'idée de cet algorithme est de maximiser la probabilité de la structure sachant les données dans l'espace de DAG respectant un ordre d'énumération de variables donné en entrée. La réduction qui découle de cette connaissance permet de limiter la recherche à un espace de $2^{c_n^2}$ structures. De même, l'espace à explorer est aussi limité en faisant l'hypothèse qu'un nœud ne peut avoir plus d'un certain nombre de parents. [28]

Cooper & Herskovits [25] ont imposé un ordre sur les nœuds, pour simplifier l'espace de recherche, de manière à ce qu'un nœud ne puisse être parent d'un autre que s'il précède

Chapitre II : Les réseaux Bayésiens

celui ci dans l'ordre fixé. Il est alors possible de réduire l'espace de recherche, qui devient alors l'ensemble des réseaux bayésiens respectant cet ordre d'énumération. [27]

L'algorithme K2 teste alors l'ajout de parents de la manière suivante : le premier nœud ne peut pas posséder de parents et pour les nœuds suivants, l'ensemble des parents choisi est le sous-ensemble de nœuds qui augmente le plus le score parmi l'ensemble des nœuds le précédant dans l'ordre d'énumération. L'espace de recherche est ainsi réduit à l'ensemble des DAG respectant cet ordre. [28]

4. Inférence :

L'utilisation essentielle des réseaux Bayésiens est le calcul des probabilités conditionnelles d'événements reliés les uns aux autres par des relations de cause à effet. Cette utilisation s'appelle inférence. Vu la correspondance entre la structure graphique et la structure probabiliste, les problèmes de l'inférence sont ramenés à des problèmes de la théorie des graphes. On l'utilise pour calculer des probabilités marginales sur les nœuds en présence ou absence de variables observées aussi appelées « évidence ». De point de vue mathématique, il s'agit de calculer $P(X|e)$. Où X désigne l'ensemble des variables aléatoires et e une instanciation possible de tout ou partie de l'ensemble de X . Plusieurs approches ont été développées pour résoudre le problème de l'inférence et on distingue deux classes d'algorithmes d'inférence [19].

Il existe plusieurs algorithmes d'inférence dans les réseaux Bayésiens. Ils sont classés en deux groupes :

4.1 Les méthodes d'inférence exactes : qui exploitent les indépendances conditionnelles contenues dans les réseaux et donnent à chaque inférence les probabilités *a posteriori* exactes. Par exemple l'algorithme Clustering effectue l'inférence en transformant le réseau en un arbre pour lequel chaque nœud regroupe plusieurs nœuds du réseau initial. [20]

4.1.1 Messages locaux : Les premiers algorithmes d'inférence exacte pour les réseaux bayésiens sont basés sur une architecture à passage de messages et ils étaient limités aux arbres. Dans cette technique, à chaque nœud est associé un processeur qui peut envoyer des messages de façon asynchrone à ses voisins jusqu'à ce qu'un équilibre soit atteint, en un nombre fini d'étapes. Cette méthode a été depuis étendue aux réseaux quelconques pour donner l'algorithme de l'arbre de jonction. [16]

4.1.2 L'arbre de jonction : La méthode de l'arbre de jonction est une technique d'inférence qui peut se diviser en deux parties. La première consiste à transformer le réseau bayésien pour

Chapitre II : Les réseaux Bayésiens

obtenir un arbre de jonction. L'idée consiste à dissimuler chaque cycle existant sous un ou plusieurs nœuds de l'arbre de jonction. La seconde étape de l'algorithme effectue le calcul d'inférence sur l'arbre, en utilisant des raisonnements similaires à l'inférence sur un réseau bayésien-arbre (JT). [30] Cette algorithme s'applique en 5 étapes de transformation du graphe :

Moralisation du graphe, Triangulation, Construire l'Arbre de Jonction. inférence dans l'arbre de jonction en utilisant l'algorithme des messages locaux, Et une étape transformation des potentiels de clique en lois conditionnelles mises à jour Le coût de l'algorithme JT est déterminé par la taille de la plus grande clique et est exponentiel en espace, pour les graphes densément connectés l'inférence peut être impraticable [28].

Une autre méthode s'appelle le cut-set conditioning : elle consiste à instancier un certain nombre de variables de manière à ce que le graphe restant forme un arbre. On procède à une propagation par messages sur cet arbre. Puis une nouvelle instantiation est choisie. On réitère ce processus jusqu'à ce que toutes les instantiations possibles aient été utilisées. On fait alors la moyenne des résultats. L'avantage de cette méthode est une complexité en temps linéaire sur la taille du réseau, mais le calcul des probabilités conditionnelles est en général impraticable pour les réseaux assez grands car ayant une complexité dans le pire des cas exponentielle dans le nombre de variables et aussi à cause du problème des boucles dans le réseau [29].

4.2 Les méthodes approchées : qui estiment les probabilités *a posteriori*. Pour ces méthodes, deux exécutions d'une inférence peuvent donner des probabilités *a posteriori* différentes. [20] , Il existe trois approches pour réaliser des inférence approchées, faire comme si le graphe était un arbre : « loopy belief propagation », Markov chain Monte Carlo (e.g . échantillonnage de Gibbs), Inférence variationnelle.

4.2.1 Les méthodes MCMC (pour Markov Chains Monte Carlo): permettent d'échantillonner des variables aléatoires en construisant une chaîne de Markov. Les deux algorithmes à base de MCMC pour faire de l'estimation de densité les plus répandus, sont l'algorithme de Metropolis- Hastings et l'échantillonneur de Gibbs. Ces méthodes statistiques approchées ne calculent pas exactement les lois marginales, mais en donnent une estimation. Elles permettent donc de traiter des applications de grande taille en temps raisonnable, ce qui n'est pas le cas des méthodes exactes [28].

4.2.2 L'inférence variationnelle : est une méthode de plus en plus utilisée, elle est une sorte d'adaptation de l'algorithme EM (Expectation-Maximization)[29] . Dans l'état actuel des

Chapitre II : Les réseaux Bayésiens

choses, l'inférence dans les modèles graphiques probabilistes n'est possible que lorsque les distributions de probabilités des variables continues sont gaussiennes et n'ont pas de variables filles discrètes. Lorsque ce cas se présente, il faut avoir recours à des méthodes d'inférence approchée. Une technique pour cela serait de discrétiser toutes les variables, ou encore d'utiliser les méthodes variationnelles. Néanmoins, même lorsque nous ne sommes pas dans ce cas, ces méthodes ont l'avantage d'être peu complexes, et peuvent alors se révéler utiles. Les méthodes variationnelles ont été introduites pour approcher l'apprentissage par maximum de vraisemblance en présence d'une base d'exemples incomplète. Elles sont alors utilisées pour approcher les intégrales nécessaires pour l'inférence Bayésienne [28].

4.2.3 Loopy belief propagation :

L'algorithme des messages locaux de Pearl ne fonctionne de manière exacte que sur des structures en forme d'arbre. Cependant cette méthode a été généralisée dans pour effectuer de l'inférence approchée avec une structure de DAG quelconque proposent une étude empirique de la convergence de cet algorithme en présence de cycles, et parfois, certains résultats très mauvais peuvent être obtenus avec cette méthode [22] cet algorithme permet d'obtenir, le plus souvent, de très bonnes approximations des marginales sans réelles justifications théoriques . Plus précisément, on ne sait pas prédire s'il convergera, ni dire, dans le cas où il convergerait, si les marginales approximées sont correctes.

5 Conclusion :

Les réseaux bayésiens sont donc un outil de choix dans la représentation de connaissances et dans l'exploitation de celles-ci. Nous l'avons vu, beaucoup de domaines sont intéressés par ce type de représentation. Comme on a pu le voir, l'inférence sur les réseaux bayésiens est un problème NP-difficile, c'est pourquoi il était convenable de le voir de façon complète pour des instances réalisables et incomplète dans les autres cas.

Chapitre III: les technologies de Big data

Chapitre III: les technologies de big data

1. Introduction :

De grandes quantités d'informations sont mises en ligne sur le web par des milliers d'entreprises, d'organisations et d'individus, la charge ainsi que le volume de données à gérer ont crû de façon exponentielle, pour cela les sociétés ont recouru au Datawarehouse ou entrepôt de données pour l'analyse et le stockage de données. Généralement le Datawarehouse est centralisé dans un serveur connecté à une baie de stockage, cette solution est difficilement scalable (ajout de puissance à la demande) en plus du fait qu'elle ne gère que les données structurées dans des SGBD. Pour faire face à l'explosion du volume des données, on parle actuellement de pétaoctet (billiard d'octets) voir de zettaoctet (trilliard d'octets) et aussi face à la grande variété des données (image, texte, web, etc.) un nouveau domaine technologique a vu le jour : le Big Data inventé par les géants du web, au premier rang comme Yahoo, Google et Facebook, qui ont été les tous premiers à déployer ce type de technologie. Ce concept apporte une architecture distribuée et scalable pour le traitement et le stockage de données. Ce nouveau paradigme a pour principal objectif l'amélioration des performances et l'augmentation de la vitesse d'exécution des requêtes et des traitements.

2. Présentation de Big Data

2.1 Définition de Big Data : exploration de très vastes ensembles de données pour obtenir des renseignements utilisables. Le terme Big Data se réfère aux technologies qui permettent aux entreprises d'analyser rapidement un volume de données très important et d'obtenir une vue synoptique. En mixant intégration de stockage, analyse prédictive et applications, le Big Data permet de gagner en temps, en efficacité et en qualité dans l'interprétation de données. Les objectifs de ces solutions d'intégration et de traitements des données sont de traiter un volume très important de données aussi bien structurées que non structurées, se trouvant sur des terminaux variés (PC, smartphones, tablettes, objets communicants...), produites ou non en temps réel depuis n'importe quelle zone géographique dans le monde [31].

Chapitre III: les technologies de big data

2.2 Caractéristiques du Big Data : les grandes caractéristiques qui englobent les problématiques auxquelles le Big Data répond sont le Volume des données, la Vitesse d'acquisition et de traitement des données et la Variété des types de données : plus simplement, on parle des trois «V» du Big Data

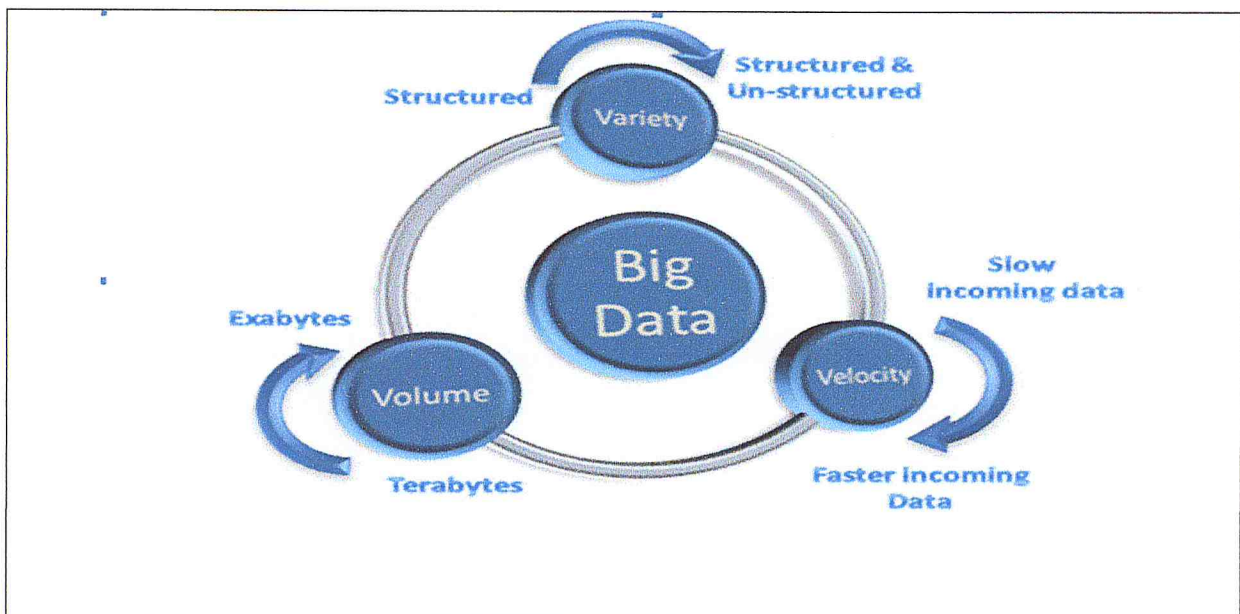


Figure 4: les dimensions de big data

Volume : Le Big Data est associé à un volume de données vertigineux, se situant actuellement entre quelques dizaines de téraoctets et plusieurs péta-octets en un seul jeu de données. Les entreprises et tous les secteurs d'activités confondus, devront trouver des moyens pour gérer le volume de données en constante augmentation qui est créé quotidiennement. Les catalogues de plus de 10 millions de produits sont devenus la règle plutôt que l'exception. Voici quelques chiffres pour illustrer ce phénomène :

- 90% des données actuelles ont été créées dans les deux dernières années
- seulement ; Twitter comme exemple, génère 7 To de données chaque jour.

Vitesse : La vitesse décrit la fréquence à laquelle les données sont générées, capturées et partagées. Les entreprises doivent appréhender la vitesse non seulement en termes de création de données, mais aussi sur le plan de leur traitement, de leur analyse et de leur restitution à l'utilisateur en respectant les exigences des applications en temps réel.

Chapitre III: les technologies de big data

Variété : La croissance de la variété des données est la conséquence des nouvelles données multi structurelles et de l'expansion des types de données provenant de différentes sources hétérogènes. Aujourd'hui, on trouve des capteurs d'informations aussi bien dans les appareils électroménagers, les trains, les automobiles ou les avions, qui produisent des informations très variées [32].

Ces nouvelles données dites non-structurées sont variées :

- Des photos ;
- Des mails (avec l'analyse sémantique de leur contenu)
- Les données issues des réseaux sociaux (commentaires et avis des internautes sur Facebook ou Twitter par exemple) ;

Ces trois caractéristiques illustrées par les trois « V », sont les principes définissant le Big Data. Avant tout, il s'agit d'un changement d'orientation sur l'utilisation de la donnée. En somme, le point clé du Big Data est de donner un sens à ces grosses données et pour cela, il faut les analyser.

2.3 Architecture Big Data : On distingue principalement les couches suivantes :

- Couche matériel (infrastructure Layer) : peut-être des serveurs virtuels VMware, ou des serveurs lame blade ;
- Couche stockage (Storage layer) : les données seront stockées soit dans une base NoSQL, ou bien directement dans le système de fichier distribué ou les Datawarehouse
- Couche management et traitement : on trouve dans cette couche les outils de traitement et analyse des données comme MapReduce ou Pig ;
- Couche visualisation : pour la visualisation du résultat du traitement [33].

Chapitre III: les technologies de big data

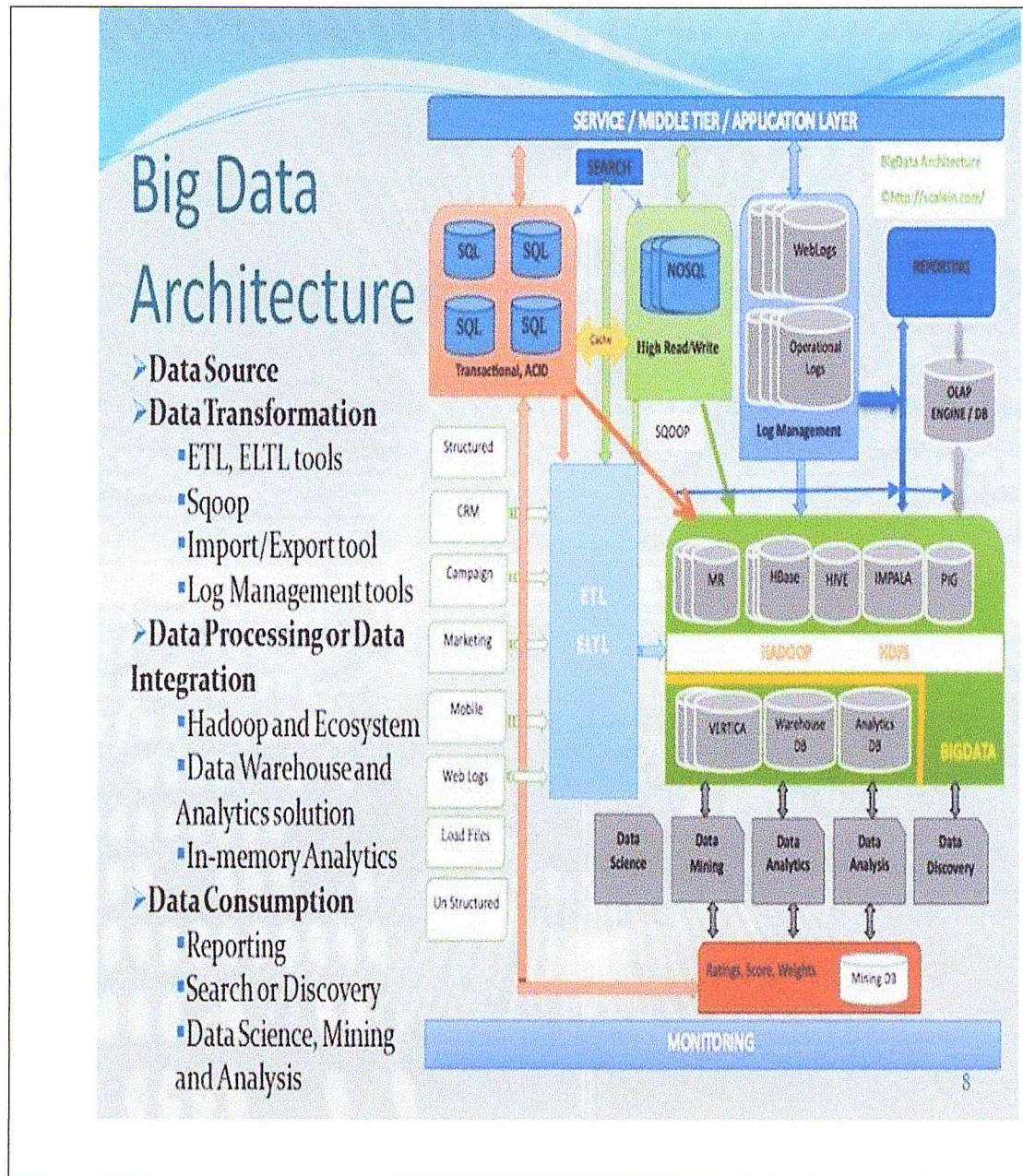


Figure: 5 Architecture de Big Data [w8].

3. **Différence entre BI (Business intelligence) et Big Data** : La méthodologie BI traditionnel fonctionne sur le principe de regrouper toutes les données de l'entreprise dans un serveur central (Datawarehouse ou entrepôt de données). Les données sont généralement analyse en mode déconnecté. Les données sont généralement structurées en SGBDR avec très peu de données non structurées. [33] Une solution Big Data, est différente d'une BI traditionnel dans les aspects suivants :

Chapitre III: les technologies de big data

- Les données sont conservées dans un système de fichiers distribué et scalable plutôt que sur un serveur central ;
- Les données sont de formats différents, à la fois structurées ainsi que non structurées ;
- Les données sont analysées en temps réel ;
- La technologie Big Data s'appuie sur un traitement massivement parallèle (concept MPP). [34]

4. Les principales technologies de Big Data :

Elles sont nombreuses. Pour optimiser les temps de traitement sur des bases de données géantes, plusieurs solutions peuvent entrer en jeu :

4.1 Intelligence d'affaires (BI : Business intelligence) : Un type de logiciel d'application conçu pour rapporter, analyser et présenter des données : Les outils de BI sont souvent utilisés pour lire les données qui ont déjà été stockées dans un data mart d'un entrepôt de données. Les outils de BI peuvent également être utilisés pour créer des rapports standards qui sont générés de façon périodique, ou pour afficher des informations sur des tableaux de bord de gestion en temps réel, c.-à-d. des affichages intégrées des critères qui mesurent la performance d'un système.

4.2 Bases No SQL : NoSQL est l'acronyme qui a été trouvé depuis 2009 pour s'attaquer au phénomène Big Data. Pour certains il signifie « non SQL », pour d'autres, « Not only SQL », ce qui sous-entend que le langage n'est pas « que » du SQL. La principale problématique qui a conduit à inventer NoSQL a été de solutionner le fait qu'une même base de données sur un site Web, pouvait être utilisée en même temps dans le monde entier par des millions d'utilisateurs ; la problématique typique d'un *Amazon*... Ce que l'on cherche donc à réaliser avec NoSQL, c'est à réduire la complexité du langage de requêtes, à simplifier l'architecture de la base de données, et à trouver un moyen de stocker la base sur un maximum d'ordinateurs peu coûteux en fonction des besoins. Ainsi, une base NoSQL est une base de données distribuée pour répartir la charge de calcul et de données dynamiquement, non relationnelle, préférant la gestion d'une table gigantesque à celle de nombreuses tables interdépendantes. [w7]

Les bases de données relationnelles ont une philosophie d'organisation des données bien spécifiques, avec notamment le langage d'interrogation SQL, le principe d'intégrité des

Chapitre III: les technologies de big data

transactions (ACID), et les lois de normalisation. Bien utiles pour gérer les données qualifiées de l'entreprise, elles ne sont pas du tout adaptées au stockage de très grandes dimension et au traitement ultra rapide. Les bases NoSQL autorisent la redondance pour mieux servir les besoins en matière de flexibilité, de tolérance aux pannes et d'évolutivité.[w2]

4.3 Stockage "In-Memory" : pour évoquer les traitements qui sont effectués dans la mémoire vive de l'équipement informatique, plutôt que sur des serveurs externes. L'avantage du traitement in-memory est celui de la vitesse puisque les données sont immédiatement accessibles. En revanche, ces données ne sont pas stockées sur le long terme, ce qui peut poser des problèmes d'historisation. [w1]

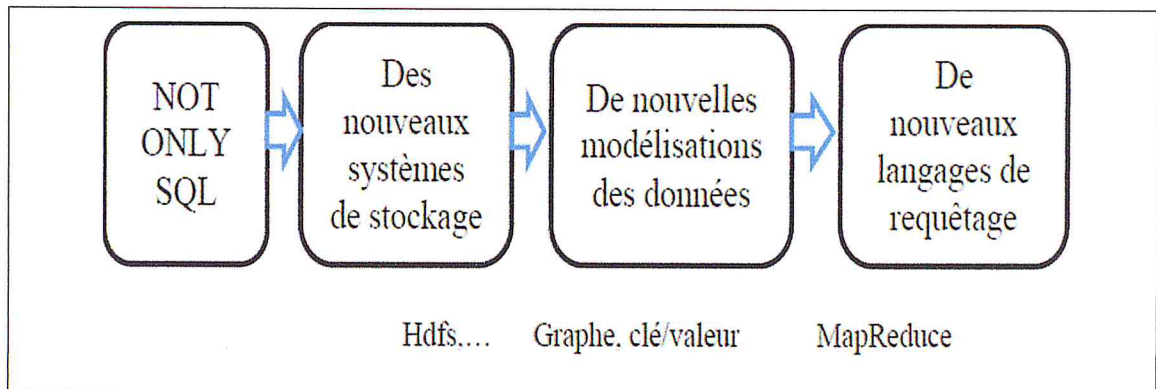


Figure 6: Les solutions de stockage [w1].

4.4 Cloud Computing

Le Big Data exige une capacité matérielle hors du commun, que ce soit pour le stockage comme pour les ressources processeurs nécessaires au traitement. Nul besoin de s'équiper outre mesure, le "Cloud" est là pour cela. Encore faut-il avoir bien compris le concept pour différencier, le cloud privé du cloud public, l'interne de l'externe et les hybrides combinant plusieurs types de solutions. Ensuite il est aussi prudent de différencier les niveaux de services de chacune des solutions : IAAS, PAAS, SAAS [w2].

4.5 Hadoop :

Le Big data regroupe plusieurs nouvelles technologies et d'outils pour répondre à une triple problématique : un Volume de données important à traiter, une grande Variété d'informations (structurées ou non structurées), et un certain niveau de Vitesse à atteindre.

Chapitre III: les technologies de big data

Pour répondre à ces besoins, il s'avère que l'écosystème Hadoop serait la solution open source par excellence.

a) Définition de Hadoop :

Utilisé par les géants du web comme Twitter, LinkedIn, Ebay et Amazon, Hadoop est un **framework libre et Open Source** écrit en langage Java dont le but est de faciliter la création d'applications distribuées et scalables. Hadoop est donc conçu pour réaliser des traitements sur des volumes de données importants, de l'ordre de plusieurs petaoctets. Il est donc aujourd'hui massivement utilisé pour stocker, analyser et manipuler d'immenses quantités de données: Hadoop est un standard pour le traitement **Big Data**. Hadoop se réfère à son écosystème et à l'ensemble des logiciels comme Apache Spark, Cloudera Impala, Sqoop, ... etc [w3].

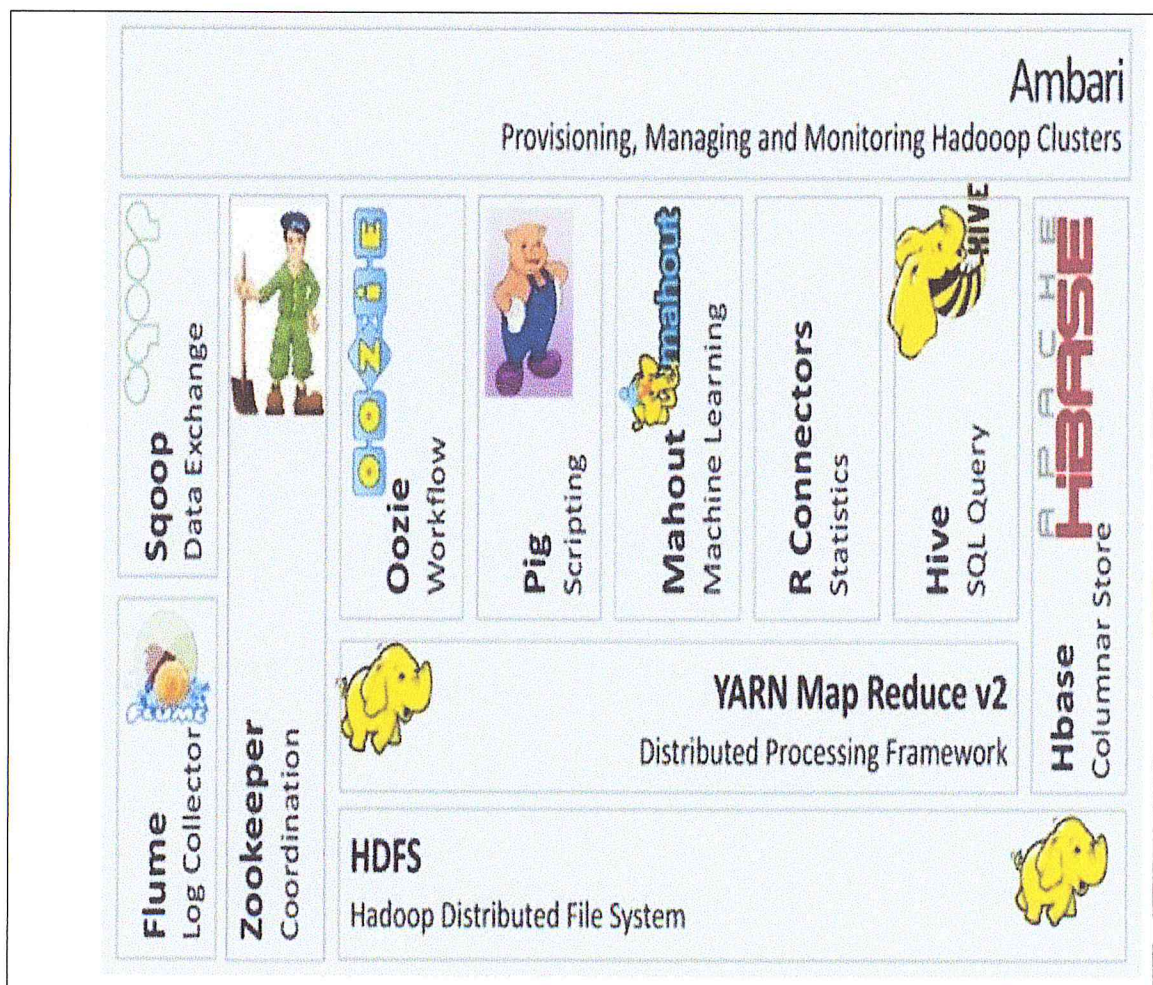


Figure 7 : Architecture hiérarchique de bibliothèque logicielle du Corp d'HADOOP [w8].

Chapitre III: les technologies de big data

b) Hadoop Distributed File System HDFS:

Hadoop HDFS est un fichier système distribué développé pour fonctionner sur du matériel de base qui fait référence à la conception GFS. HDFS est le substrat de stockage de données primaire des applications Hadoop. Un cluster HDFS se compose d'une seule *NameNode* qui gère les métadonnées du fichier système et les collections de *DataNodes* qui stockent les données réelles. Un fichier système est divisé en un ou plusieurs blocs, et ces blocs sont stockés dans un ensemble de *DataNodes*. Chaque bloc a plusieurs répétitions réparties dans différents *DataNodes* pour empêcher les données manquantes. Apache Hbase est un store en colonnes modélisée à partir de Bigtable de Google. Ainsi, Apache HBase fournit des capacités Bigtable comme discuté dans le dernier chapitre. HBase peut servir à la fois à l'entrée et à la sortie pour les fonctions MapReduce s'exécutant sur Hadoop et qui peuvent être consultées via Java API, REST, Avro ou Thrift API.[w4]

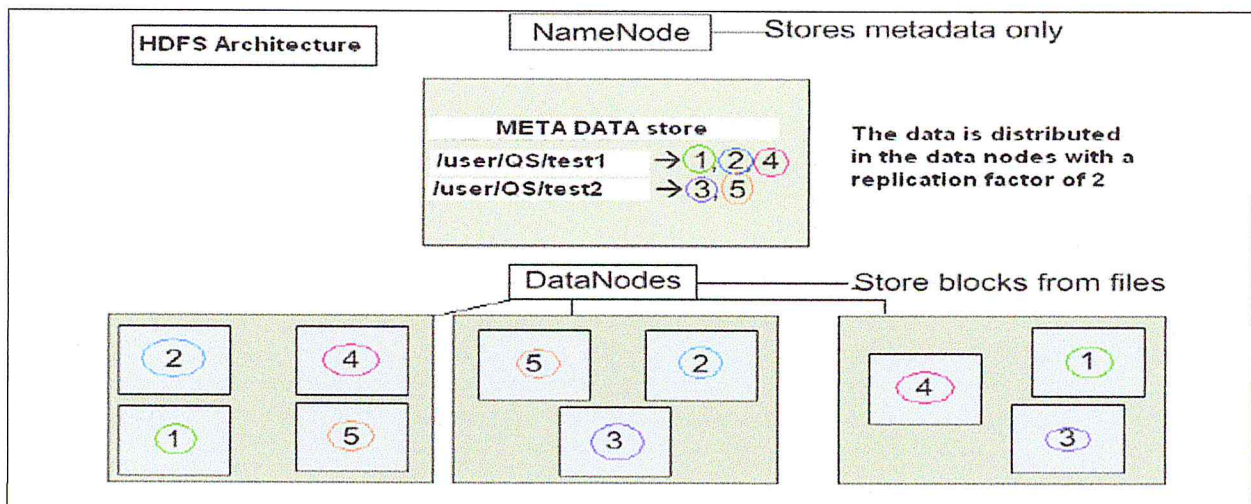


Figure 8:L'architecture d'HDFS [w5]

Hadoop Common: il contient les bibliothèques et les utilitaires nécessaires aux autres modules Hadoop.

Hadoop YARN (Yet Another Resource Negotiator): cette plateforme permet de gérer les ressources informatiques du cluster et les utilise pour la planification des applications des utilisateurs.[w3]

Chapitre III: les technologies de big data

4.6 MapReduce :

MapReduce qui est le deuxième composant du noyau Hadoop permet d'effectuer des traitements distribués sur les nœuds du cluster. Il décompose un job (unité de traitement mettant en œuvre un jeu de données en entrée, un programme MapReduce (packagé dans un JAR (Java Archive : fichier d'archive, utilisé pour distribuer un ensemble de classes Java)) et des éléments de configuration) en un ensemble de tâche plus petites qui vont produire chacune un sous ensemble du résultat final ; ce au moyen de la fonction **Map**. [w1]

Il s'agit de décomposer une tâche en tâches plus petites, ou plus précisément découper une tâche portant sur de très gros volumes de données en tâches identiques portant sur des sous-ensembles de ces données. Les tâches (et leurs données) sont ensuite dispatchées vers différents serveurs, puis les résultats sont récupérés et consolidés. La phase de décomposition des tâches est la partie Map, tandis que la phase de la consolidation des résultats est la partie Reduce. [22] Chaque phase manipule des paires (clé,valeur) en entrée et sortie. Un cluster MapReduce utilise une architecture de type Maître-esclave où un noeud maître dirige tous les nœuds esclaves.

Le schéma ci-dessous présente le processus d'un traitement MapReduce.

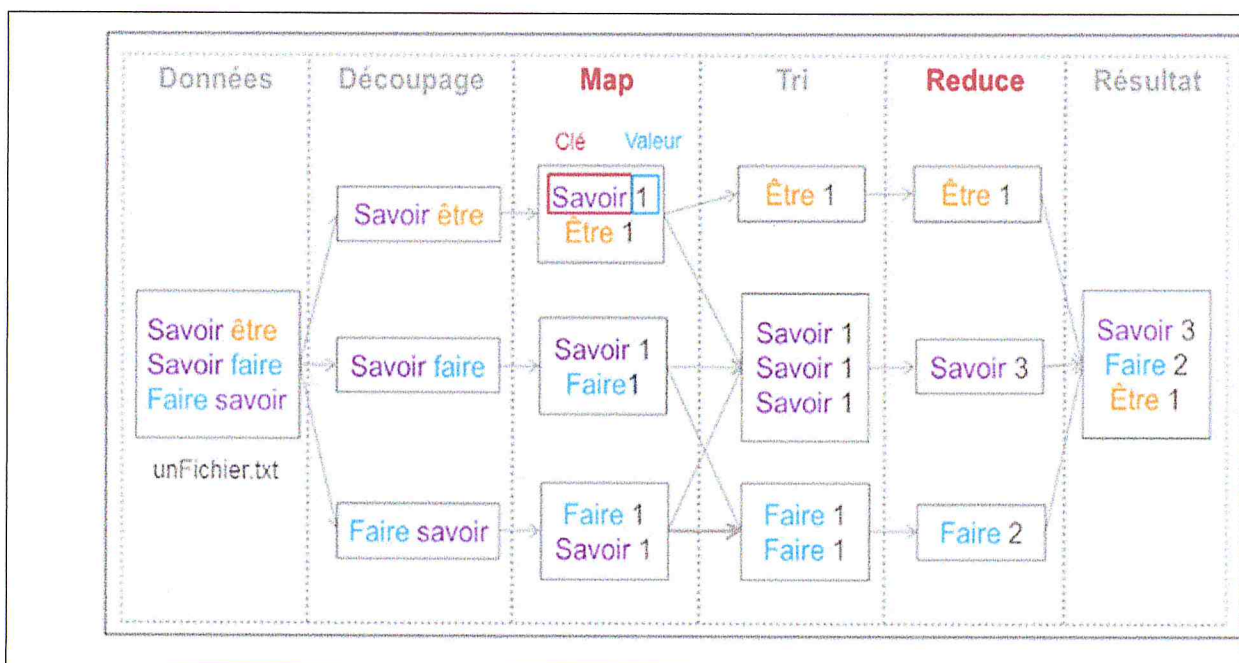


Figure 9 : Processus d'un traitement MapReduce, [w1]

Chapitre III: les technologies de big data

4.7 Spark :

Apache Spark est un framework de traitements Big Data open source construit pour effectuer des analyses sophistiquées et conçu pour la rapidité et la facilité d'utilisation. Celui-ci a originellement été développé par AMPLab, de l'Université UC Berkeley, en 2009 et passé open source sous forme de projet Apache en 2010.

Spark présente plusieurs avantages par rapport aux autres technologies big data et MapReduce comme Hadoop et Storm. D'abord, Spark propose un framework complet et unifié pour répondre aux besoins de traitements Big Data pour divers jeux de données, divers par leur nature (texte, graphe, etc.) aussi bien que par le type de source (batch ou flux temps-réel). Ensuite, Spark permet à des applications sur clusters Hadoop d'être exécutées jusqu'à 100 fois plus vite en mémoire, 10 fois plus vite sur disque. Il vous permet d'écrire rapidement des applications en Java, Scala ou Python et inclut un jeu de plus de 80 opérateurs haut-niveau. De plus, il est possible de l'utiliser de façon interactive pour requêter les données depuis un shell.[w6]

a) Hadoop et Spark

Hadoop est positionné en tant que technologie de traitement de données depuis 10 ans et a prouvé être la solution de choix pour le traitement de gros volumes de données. MapReduce est une très bonne solution pour les traitements à passe unique mais n'est pas la plus efficace pour les cas d'utilisation nécessitant des traitements et algorithmes à plusieurs passes. Chaque étape d'un workflow de traitement étant constituée d'une phase de Map et d'une phase de Reduce, il est nécessaire d'exprimer tous les cas d'utilisation sous forme de patterns MapReduce pour tirer profit de cette solution. Les données en sortie de l'exécution de chaque étape doivent être stockées sur système de fichier distribué avant que l'étape suivante commence. Cette approche a tendance à être peu rapide à cause de la réplication et du stockage sur disque.

De plus, les solutions Hadoop s'appuient généralement sur des clusters, qui sont difficiles à mettre en place et à administrer. Elles nécessitent aussi l'intégration de plusieurs outils pour les différents cas d'utilisation big data (comme Mahout pour le Machine Learning et Storm pour le traitement par flux).

Spark permet de développer des pipelines de traitement de données complexes, à plusieurs étapes, en s'appuyant sur des graphes orientés acycliques (DAG). Spark permet de

Chapitre III: les technologies de big data

partager les données en mémoire entre les graphes, de façon à ce que plusieurs jobs puissent travailler sur le même jeu de données. Spark s'exécute sur des infrastructures Hadoop Distributed File System (HDFS) et propose des fonctionnalités supplémentaires. Il est possible de déployer des applications Spark sur un cluster Hadoop v1 existant (avec SIMR – Spark-Inside-MapReduce), sur un cluster Hadoop v2 YARN ou même sur Apache Mesos. Plutôt que de voir en Spark un remplaçant d'Hadoop, il est plus correct de le voir comme une alternative au MapReduce d'Hadoop. Spark n'a pas été prévu pour remplacer Hadoop mais pour mettre à disposition une solution complète et unifiée permettant de prendre en charge différents cas d'utilisation et besoins dans le cadre des traitements big Data. [w6]

5. Conclusion

L'augmentation du volume et du détail des informations saisies par les entreprises, la montée du multimédia, les médias sociaux et l'Internet sont des choses qui vont alimenter la croissance exponentielle des données pour l'avenir prévisible. Le traitement de telles données nécessitera alors de nouvelles capacités logicielles et matérielles plus efficaces et intelligentes.

Chapitre IV :

imputation des données manquante en temps réel

Chapitre 4 : imputation des données manquante en temps réel

1. Introduction :

Après avoir fait une étude sur les réseaux bayésiens, ces principales méthodes d'apprentissage de la structure, des paramètres et de l'inférence afin de connaître ces étapes pour le traitement des données manquantes. Dans ce chapitre on va expliquer l'approche basée par RB inspirés de [38-39] qui impliquent les distributions de probabilité de toutes les valeurs possibles pour les données élémentaires absentes et puis complètent les valeurs absentes par ces distributions.

Généralement nos contributions principales peuvent être récapitulées comment utilise les réseaux bayésien pour imputation des données manquantes en temps réel

2. Nettoyage des données manquantes en temps réel basé sur réseau bayésien :

Notre approche d'imputation des données manquantes en temps réel de données consiste à (voir le schéma 1) :

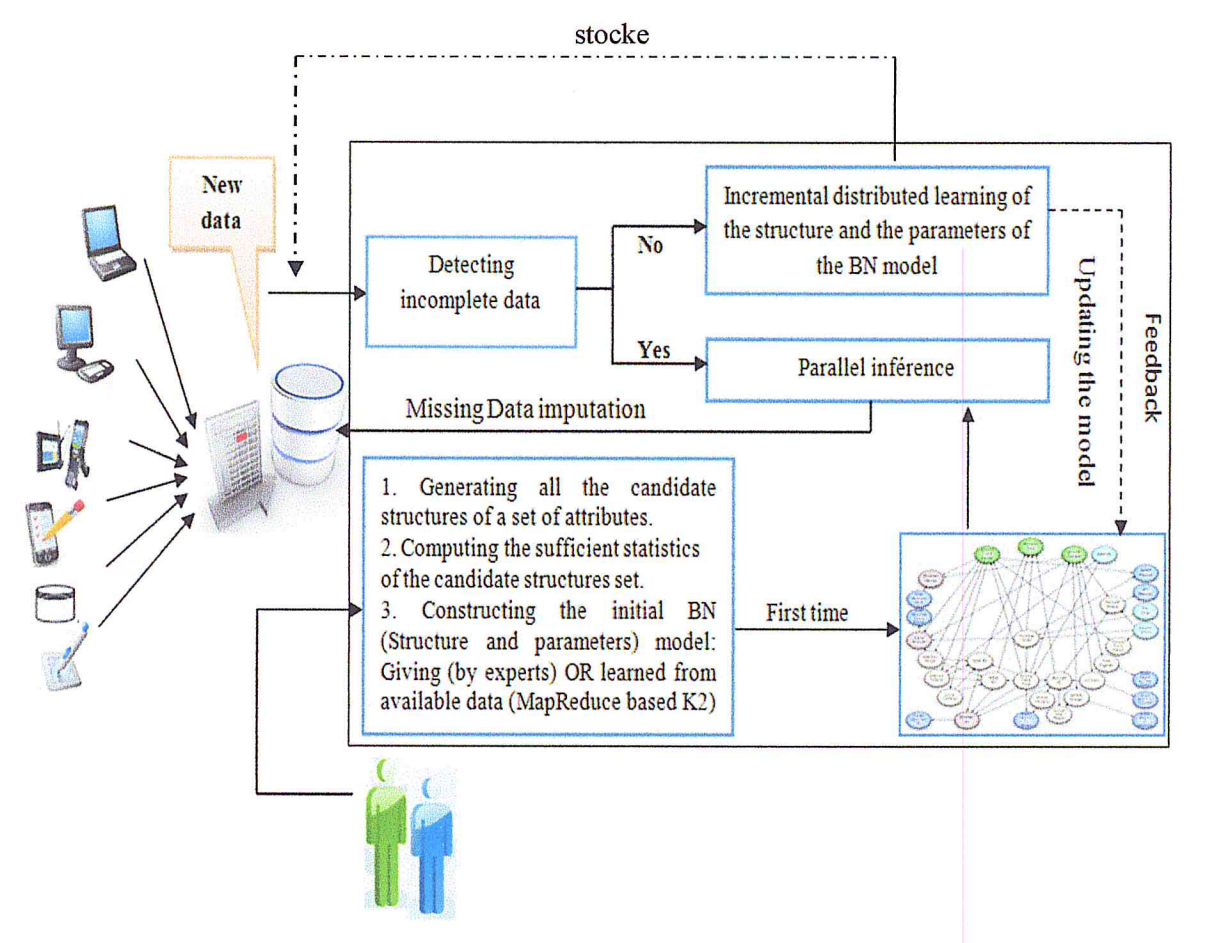


Figure 10 : Schéma proposé de framework.

À la réception d'un tuple incomplet t , nous en déduire les distributions de probabilité pour les valeurs possibles de l'attribut sur lequel la valeur de t est manquant. En utilisant la procédure d'inférence parallèle, les distributions de probabilité de toutes les valeurs possibles sont calculées. Cette distribution de probabilité représente exactement des tuples de données incertains. Nous stockons alors la distribution de probabilité des valeurs possibles pour chaque élément de données manquantes comme x -tuple avec une clef étrangère liée à la table incomplète d'origine. Enfin, nous pouvons remplir la valeur manquante en utilisant la plus probable dans les x -tuples correspondants (c.-à-d., l'alternative avec la plus grande probabilité).

Cette méthode d'imputation des données passe par des phases :

Chapitre 4 : imputation des données manquante en temps réel

- **1^{er} Phase** : réception un signal indiquant que les données ont atteint le nombre spécifié pour lance l'algorithme incrémental.
 - **2^{eme} pahse** : Génération de structures candidates :
 - **3^{eme} Phase**: la construction du RB basé sur le modèle d'imputation des données manquantes. Lorsque nous construisons un RB, nous comptons sur deux sources d'information: entrée d'experts du domaine si ce là n'est aucune donnée disponible, et des données statistiques
- Par des experts du domaine : Structure et paramètres peuvent être fournis par des experts qui vont souvent surestimer la probabilité de réussite d'un projet le concernant, l'étape consiste à fournir à l'expert des outils associant des notions qualitatives et quantitatives, pour qu'il puisse associer une probabilité aux différents événements.
 - Ou, pour obtenir un modèle RB implique l'utilisation d'algorithmes d'apprentissage en utilisant l'algorithme K2 pour construire le R_0 (réseau bayésien initial)
L'algorithme de K2 classique est utilisé comme algorithme de base pour la création de la structure RB set l'estimation de leurs paramètres.

2.1 Algorithme K2 (version classique):

Cooper et Hersovits [25] ont proposé une méthode efficace largement utilisé pour l'apprentissage de la structure RB à partir des données complètes. Le but de l'algorithme de K2 est de maximiser la probabilité de la structure donnée des données dans l'espace DAG respectant une contrainte de l'ordre d'énumération (ordre variable). Cette contrainte permet de limiter l'espace de recherche. Cependant, cette contrainte a un inconvénient. Trouvez le meilleur ordre est pas évident.

Comme nous venons de dire, l'algorithme K2 exige un ordre total a priori sur les nœuds pour s'assurer que le graphique résultant est acyclique et recherche de manière heuristique la structure RB la plus probable qui maximise la fonction de score g.

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!$$

Chapitre 4 : imputation des données manquante en temps réel

De là où le $\emptyset_i$ est la liste de toutes les instanciations possibles x_i dans D , $q_i = |\emptyset_i|$, et le r_i le nombre de toutes les valeurs possibles du XI variable, X_i , N_{ijk} est le nombre de cas (c.-à-d. exemples) dans D dans lequel l'attribut XI est instancié avec sa valeur de k^{ieme} , et les parents x_i dans π_i sont instanciés avec l'instanciation de j^{ieme} dans le $\emptyset_i$, et $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

Algorithme 1. K2 *algorithme (version classique)*

1. {Enter: 1. un ensemble de n nœud, une séquence de nœud
 2. Une limite supérieure u le nombre de nœud parent
 3. une base de données D contenant m enregistrement}
1. {Output: pour chaque nœud x_i , parents de ce nœud.}
3. for $i := 1$ to n do
4. $\pi_i := \emptyset$
5. $P_{old} := g(i, \pi_i)$; {This function is computed using Equation 2.}
6. OKToProceed := true;
7. While OKToProceed and $|\pi_i| < u$ do
8. let z be the node in $Pred(x_i) - \pi$ that maximizes $g(i, \pi_i \cup \{z_i\})$;
9. $P_{new} := g(i, \pi_i \cup \{z_i\})$;
10. if $P_{new} > P_{old}$ then
11. $P_{old} := P_{new}$;
12. $\pi_i := \pi_i \cup \{z\}$;
13. else OKToProceed := false;
14. end {while};
15. write('Node: ', x_i , ' Parent of x_i : ' π_i);

Chapitre 4 : imputation des données manquante en temps réel

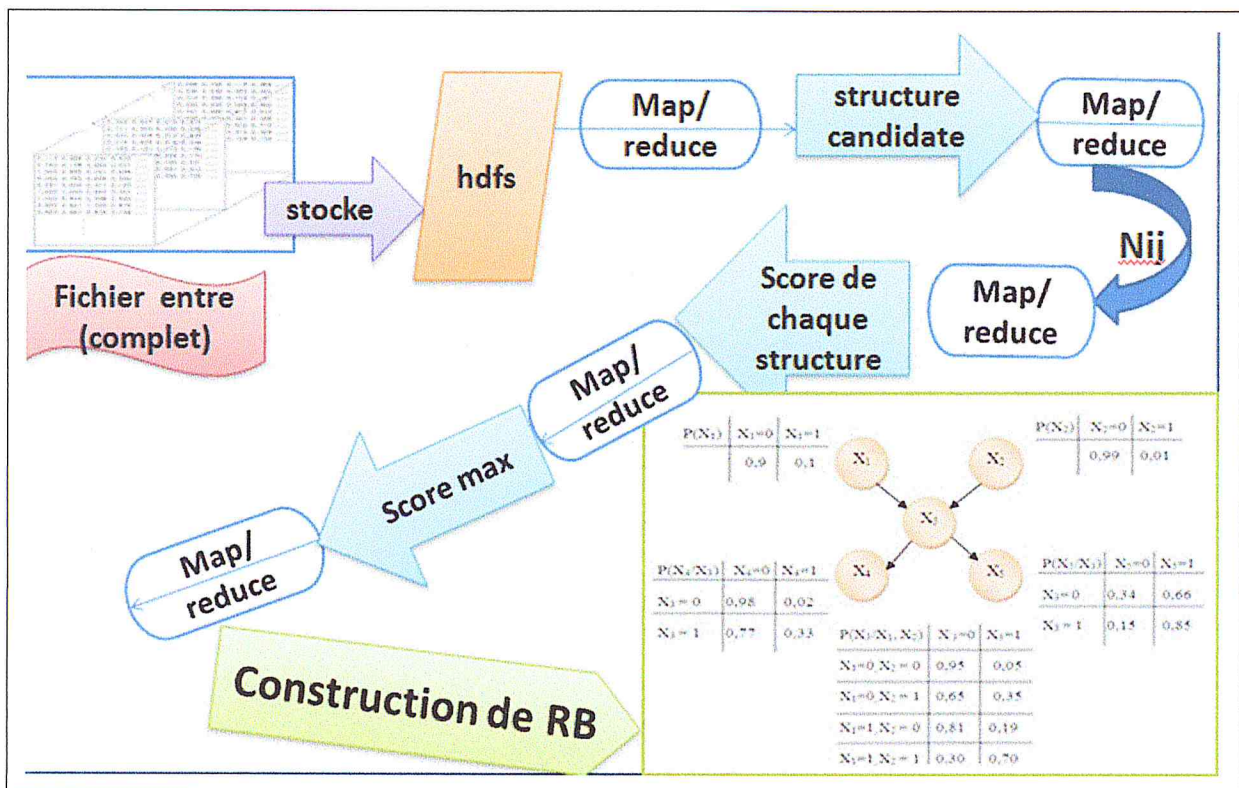
16. end {for};

17. end {K2};

- Et algorithme K2 incrémentale de MapReduce pour construire la structure du RB et pour calculer les paramètres dans le cas où les données sont disponibles.

2.2 L'algorithme K2 distribuée incrémentale pour l'apprentissage de structure des RBs :

La version classique de l'algorithme K2 présentée ci-dessus est algorithme en lots. On a proposé dedans une version incrémentale de K2 [26]. Dans le cadre de notre travail, nous avons besoin d'une procédure d'apprentissage progressif rapide pour apprendre le modèle BN. Par conséquent, nous avons proposé une version distribuée incrémentale de l'algorithme K2



Fang et al [43] ont proposé un K2 basé MapReduce dans lequel le marquage et le processus de recherche peuvent être exécutés en parallèle. Ils ont mis en place une fonction de score basée sur MapReduce, qui peut être utilisé pour obtenir tous les paramètres en parallèle.

Chapitre 4 : imputation des données manquante en temps réel

En processus de Map, les paramètres de chaque enregistrement peuvent être obtenus par les statistiques. Dans le processus Reduce, Les paramètres obtenus par le processus de la carte seront additionnés pour obtenir les besoins .Une fonction de recherche basée sur MapReduce a été proposée. Dans le processus de Map, les structures locales de candidat sont se produisent et ont marqué en parallèle. Dans le processus Reduce, toutes les structures optimales locales sont fusionnées pour générer une optimale globale.

Inspiré par cette version distribuée de l'algorithme K2 présenté ci-dessus, nous avons proposé une version incrémentale distribuée de l'algorithme K2 (voir l'algorithme 2). Cette version a été développée pour obtenir réaction (feedback) et une amélioration continue du modèle d'imputation de données manquantes.

Algorithme 2. Distribuée incrémentale de l'algorithme K2

1. Input:

- {
- $\{CS_1, \dots, CS_n\}$ (Ensemble de toutes les structures candidates de chaque noeud)
- $SS(D)$ (statistiques suffisantes des données précédentes)
- D_n (nouvel ensemble de données)
- }

2. Output:

- {
- Un graphique acyclique dirigé pour le réseau bayésien selon le précédent et les nouvelles données.
- Les nouvelles statistiques suffisamment mises à jour, $SS(D)$
- }

3. Compute $SS(D_n)$

4. $SS(D) = SS(D) \cup SS(D_n)$



Chapitre 4 : imputation des données manquante en temps réel

2.2.1 La génération des structures candidates basé MapReduce :

La génération des structures candidates est une tâche à forte intensité de calcul. Une approche exhaustive est impraticable en raison de la taille de l'espace de recherche. En effet, le nombre de structures possibles à partir de n nœuds est super-exponentiel.

Donner une séquence de nœuds $\{X_1, \dots, X_n\}$, qui représente une liste ordonnée de nœuds et u le nombre maximum d'enfants pour un nœud, nous proposons l'algorithme 3 basé sur le modèle MapReduce pour générer des structures candidates pour chaque nœud x_i .

Pour obtenir les successeurs nous développons l'algorithme 3, y compris le processus Map et le processus reduce respectivement, pour chaque Map nécessite un nœud père et une séquence de nœud et le nombre maximum de fils, et pour reduce nous donne tous les structures candidates

Algorithme 3. génération structure Candidate

1. entrer:

```
{  
    •  $\{X_1, \dots, X_n\}$  (une liste ordonnée de nœuds)  
    •  $X_i$  (le nœud parent)  
    •  $u$  (le nombre maximum d'enfants d'un nœud)  
}
```

2. sortie : Toutes les structures candidates d'un nœud parent

Algorithme 3.1: Map (String key, String value)

```
1. // key: clé d'enregistrement de données, la valeur: le contenu du fichier de données  
2. noeud_pere seq_noeud[]  
3. taille_seq_noeud:=n  
4. pos_pere :=m
```

Chapitre 4 : imputation des données manquante en temps réel

```
5. for i=1 to n do
6.   clef:= noeud_pere&&sec_noeud[i]
7.   valeur:=1
8.   strtct:=clef
9.   Insert.fichier_noeud_per(strtct)
10.  Generate <key, value>
11. end for
12. lire.fichier_noeud_pe(key)
13. while key < > null
14.  if key (dernier_position = seq_neoud(last))
15.    // Passez à la structure suivante
16.  else
17.    int k:=postion_dernier_noeud_cle
18.    for i=k to n do
19.      clef=clef+seq_neoud[i];
20.      valeur=1
21.      if clef.lentgh<=max
22.        String strtct=cle;
23.        Insert.fichier_noeud_per(strtct)
24.        Generate <key, value>
25.      else
```


Chapitre 4 : imputation des données manquante en temps réel

// Passez à la structure suivante

26. *end if*

27. *end for*

28. *end if*

29. *end while*

30. *end*

Algorithme 3.2: *Reduce (String key, String value)*

// key: Structure des candidats d'un noeud parent

key=structure_candidate;

value=L;

Generate<key, value>

Calcule nombre occurrence N_{ij} et N_{ijk}

En générale R_i dans l'équation (3,1) est une constance, que nous ne calculons pas en parallèle. Supposons qu'il existe un ensemble de donnés d'échantillons fixés comme le tableau 3.1 dans laquelle chaque colonne représente une variable unique, et chaque enregistrement représente un cas sur les variables x_1 , x_2 et x_3 qui dénotent le tabagisme, le cancer et la tuberculose, respectivement pour plus de simplicité, 0,1 sont utilisés pour désigner absent et présent respectivement. Dans cet exemple R_i est 2

Cas	Les valeurs des variables pour chaque cas		
1	Présent	absent	absent
2	Présent	Présent	Présent
3	absent	absent	Présent
4	Présent	Présent	Présent
5	absent	absent	absent
6	absent	Présent	Présent
7	Présent	Présent	Présent
8	absent	absent	absent
9	Présent	Présent	Présent

Chapitre 4 : imputation des données manquante en temps réel

10	absent	absent	absent
----	--------	--------	--------

Algorithme 3.3

Entrer :

- 1) M enregistrement sur n variable
- 2) Enregistre L_X ($1 \leq x \leq m$) y compris les valeurs de $\{s_1, s_2 \dots s_n\}$ ou s_i est un 1 ou 0 désigne le i éme variable est présent ou absent respectivement
- 3) La structure candidate (structure $(x_1 \dots x_n)$ avec ($1 \leq i \leq n$))

Sortie : paire $\langle \text{clef} / \text{valeur} \rangle$ de " N_{ij} " ou (" N_{ijk} ") et la valeur de " N_{ij} " ou (" N_{ijk} ")

Algorithme 3.3.1 map (string cle , string valeur)

//clef : clef d'enregistrement de données, valeur : contenu du dossier de données calculer N_{ij} et N_{ijk} de chaque structure à partir de L_X ($1 \leq i \leq n, 1 \leq j \leq q_i, 1 \leq k \leq r_i$)

Génère (structure $(x_0 \dots x_n)$ && L_X ($1 \leq i \leq n, 1 \leq j \leq q_i, 1 \leq k \leq r_i$) && $N_{ijk}, 1$)

Génère (structure $(x_0 \dots x_n)$ && L_X ($1 \leq j \leq q_i, 1$) && $N_{ij}, 1$)

Fin

Algorithme 3.3.2: reduce (string clef , iterator valeur)

//clef $\langle$ (structure $(x_0 \dots x_n)$ && L_X ($1 \leq i \leq n, 1 \leq j \leq q_i, 1 \leq k \leq r_i$) && $N_{ijk}, 1$) $\rangle$

ou et $\langle$ (structure $(x_0 \dots x_n)$ && L_X ($1 \leq j \leq q_i, 1$) && $N_{ij}, 1$) $\rangle$ valeur dans l'itérateur qui ont la même clef de la paire $\langle \text{clef} / \text{valeur} \rangle$

Result := 0

Pour chaque paire clef/valeur faire

Valeur := iterator.next ()

Fin pour

Sortie $\langle \text{clef}, \text{résultat} \rangle$

Calcule du score basé sur MapReduce

Chapitre 4 : imputation des données manquante en temps réel

pour tous les $\text{succ}(x_1)=\{x_1, x_2\}$, $\text{succ}(x_2)=\{x_3\}$, $\text{succ}(x_3)=\{\}$ et $\{\}$, $\text{succs}=\{x_1, x_2, x_3\}$, le nœud x_1 , il a 3 structures candidates $(x_1 \rightarrow x_2)$, $(x_1 \rightarrow x_3)$, $(x_1 \rightarrow x_2, x_2 \rightarrow x_3)$ pour calculer le score de chaque structure candidate nous développons l'algorithme 3.4

Algorithme 3.4 : fonction de score basée sur MapReduce

Entrer : structure candidate locale de chaque nœud père avec N_{ij} ou N_{ijk}

Sortie : score de la candidate locale

Algorithme 3.4.1 map(string clef , string valeur)

//clef : structure $(x_0 \dots x_n)$, valeur : N_{ij} ou N_{ijk}

If (valeur = N_{ijk}) factorielle = $N_{ijk}!$

Else $c = 1/(N_{ij} + 1)!$

Génère $\langle \text{clef}, \text{valeur} \rangle$ paire $\langle \text{structure}(x_0 \dots x_n), \text{factorielle} \rangle$

//clef se compose du structure $(x_0 \dots x_n)$, et valeur est factorielle de N_{ij} ou N_{ijk}

Le résultat intermédiaire générer par le processus Map sont formées en tant que $\langle \text{clé}/\text{valeur} \rangle$ paire $\langle \text{Structure}(x_0 \dots x_n) \text{factorielle} \rangle$ dans chaque processus reduce (3.4.2) produit toute les valeurs qui ont les mêmes clefs .et on obtient à la fin du processus le score de chaque structure locale

Algorithme 3.4.2 Reduce(string clef , string valeur)

//cle : structure $(x_0 \dots x_n)$, valeur : factorielle

Score :=1

Pour chaque structure $(x_0 \dots x_n)$

Score =score* factorielle

Fin pour

Génère le paire clé $\langle \text{structure}(x_0 \dots x_n), \text{score} \rangle$

Et les résultat génère par l'algorithme 3.4 de la forme $\langle \text{structure}(x_0 \dots x_n), \text{score} \rangle$ sont utilise dans l'algorithme 3.5 pour la détermination du score le plus élevé pour chaque nœuds pères

Algorithme 3.5 : maximum score pour les nœuds père base sur MapReduce

Chapitre 4 : imputation des données manquante en temps réel

Entrer : structure $(x_0 \dots x_n)$, et son score

Sortie : le score maximum de chaque nœud père

Algorithme 5.3.1 map(string cle ,string valeur)

//clef : structure $(x_0 \dots x_n)$, valeur : score

Nœud_ père =l'extraction de nœud père dans la structure locale

Génère clef/valeur pair < Nœud_ père ,score >

//clef se compose du Nœud_ père et valeur est le score

Le résultat intermédiaire générer par le processus Map sont formées en tant que <clé/valeur> paire<Structure $(x_0 \dots x_n)$, score > et dans chaque processus reduce (algorithme 3.5.2) calcule maximum du score pour le même clé Nœud_ père

Algorithme 3.5.2 reduce (string clef, string valeur)

//cle: Nœud_ père , valeur : score

Maximum = valeur

Iterateur =valeur

Minimun =valeur ;

If (minimun <=maximum) maximum = maximum

Else maximum =minimum ;

Endif

Gènève le pair cle < Nœud_ père ; maximum>

Fin

Maintenant on prend en charge les résultats de l'algorithme 3.4 et 3.5 ; et pour chaque Nœud_ père x_i et son score on le compare en parallèle avec les résultats de algorithme 3.4 la structure $(x_0 \dots x_n)$ et son score qui ont le meme Nœud_ père x_i et le ,e,e score ; Prendre en compte cette structure $(x_0 \dots x_n)$ locale comme une structure optimale locale pour le Nœud_ père (x_i)

Toute la structure optimale locale disponible reste que les fusionnent des structures optimales locales pour obtenir la structure optimale globale

Exemple

Chapitre 4 : imputation des données manquante en temps réel

$x1 \rightarrow x2$ c'est la structure optimale locale pour le nœud $x1$

$x2 \rightarrow x3$ c'est la structure optimale locale pour le nœud $x2$

par fusionnement on obtient comme une structure optimale globale

Extraire le N_{ij} et N_{ijk} pour chaque motif

L'Algorithme 3.6 calcule les paramètres N_{ij} et N_{ijk} seront obtenus comme résultat intermédiaire

Algorithme 3.6

Entrer : les motifs, le fichier de l'algorithme 3.4

Sortie : N_{ijk} pour chaque motif, fichier N_{ij} pour chaque motif

Algorithme 3.6.1 map(string clef , string valeur)

Clef : clef d'enregistrement de données , valeur : contenu du dossier de données

```
If ( motif == structure ) {  
    If ( structure .contient (  $N_{ijk}$  ) ) {  
        Clef = structure ;  
        Valeur = la _valeur _  $N_{ijk}$  //pour cette structure  
        Génère <clef,valeur> ;  
    }  
}
```

Else {

Insert .fichier N_{ij} (structure &&valeur (n_{ij})

}

fin

algorithme 3.6.2 reduce (string clef, int valeur)

Enter clef : structure

Valeur : la valeur de N_{ijk} pour structure

Chapitre 4 : imputation des données manquante en temps réel

Sortie : toutes les structure avec valeur de N_{ijk} correspondant a la structure

Génère <clef, valeur >

fin

Calcule les paramètres pour chaque motif :

On prend on charge les résultats de algorithme 3.6 pour calculer table de probabilité conditionnelle de chaque motif

Algorithme 3.7

entre : fichier de N_{ij} pour le fichier de l'algorithme 3.6

sortie : table de probabilité de chaque motif

algorithme 3.7.1 Map (string clef , double valeur)

entrer :

1.clef : clef d'enregistrement de données

2. valeur : contenu du dossier de données

Sortie : probabilité de chaque structure

Structure =valeur

r = lire.fichier_ N_{ij}

while (r! = null){

 structure_ n_{ij} =r

 if structure = structure_ n_{ij}){

 int n_{ij} = valeur_ n_{ijk} _structure

 int n_{ij} = valeur_ n_{ij} _structure_ n_{ij} ;

 Double d= n_{ijk}/n_{ij} ;

 Clef=structure ;

 Valeur = d

 Génère <clef, valeur>

Chapitre 4 : imputation des données manquante en temps réel

}

}

Algorithme 3.7.2 reduce (string clef, double valeur)

Entrer : clef, la structure

Valeur : probabilité

Sortie : la table de probabilité pour chaque motif

Clef =structure

2.2.2 Recherche distribuée des structures optimales :

Friedman et Goldszmith [41] introduisent le concept de statistiques suffisantes pour la structure RBs, pour extraire ses paramètres de distribution de probabilité. Ce concept est défini comme suit: supposons que X représente un vecteur de variables et $N_X^D(\mathbf{x})$ le nombre d'enregistrements dans le l'ensemble de données D tel $X = \mathbf{x}$. Le vecteur $\hat{N}_X^D(\mathbf{x})$ contenant $N_X^D(\mathbf{x})$ pour toutes les valeurs possibles de X , on appelle les statistiques suffisantes de X . Maintenant, en utilisant cette propriété de concept et décomposabilité de score attribuée à une structure BN d'un ensemble de données D , nous avons seulement besoin de garder les statistiques suffisantes de chaque X_i nœud et son parent possible fixe $Pa(X_i)$, $N_{X_i, Pa(X_i)}^D(\mathbf{x})$, de chaque structure candidate pour calculer sa fonction de score pour sélectionner la structure optimale de chaque nœud, et enfin la création de la structure BN en fusionnant toutes les structures locales optimales de toute la liste des nœuds. Nous devons aussi les statistiques suffisantes pour le calcul des paramètres du BN résultant en utilisant l'équation 3.

$$\hat{P}(X_i = x_k | Pa(X_i = x_j)) = \hat{\theta}_{i,j,k}^{MV} = \frac{N_{ijk}}{\sum_k N_{ijk}}$$

Algorithme 4. Recherche distribuée des structures optimales (DistribOptimBN)

1. Input:

{

- $\{X_1, \dots, X_n\}$ (Une liste ordonnée de nœuds)

Chapitre 4 : imputation des données manquante en temps réel

- $\{CS_1, \dots, CS_n\}$ (Ensemble de toutes les structures candidates de chaque nœud)
- $SS(D)$ (Statistiques suffisantes)

}

2. Output:

{

Toutes les structures optimales (en les fusionnant, nous obtenons le nouveau BN)

}

Avec la version classique de MapReduce, la répétition du calcul complet de l'ensemble de données entiers est réalisable avec des Frameworks de l'informatique distribués tels que Hadoop, il est évidemment ressources inefficaces et gaspille. Pour mettre en œuvre notre modèle d'apprentissage BN incrémental distribué, nous avons besoin d'une version incrémentale distribuée de MapReduce. Quelques travaux dans ce contexte ont été proposées dans la littérature comme [44][45].

IncMR [46] est un framework amélioré pour le Traitement incrémental à grande échelle des données. Ce framework hérite de la simplicité du modèle MapReduce d'origine. Il ne modifie pas HDFS et utilise toujours les mêmes API de MapReduce. Tous les algorithmes ou programmes peuvent compléter le traitement des données supplémentaires sans aucune modification. Plus précisément, il se traduit répond à nos objectifs. En effet, nous avons utilisé ce framework pour mettre en œuvre notre méthode d'apprentissage incrémental distribué de la BNE.

➤ 3^{eme}Phase : Vérification des données à la réception :

- Si les données sont complètes, alors elles seront utilisées pour améliorer le modèle d'imputation. En effet, la procédure d'apprentissage incrémental distribué sera lancé et donc le modèle sera mis à jour.
- Si les données sont incomplètes, les éléments manquants sont détectés et la procédure d'inférence parallèle est lancée pour estimer les valeurs les plus probables à imputer.

Chapitre 4 : imputation des données manquante en temps réel

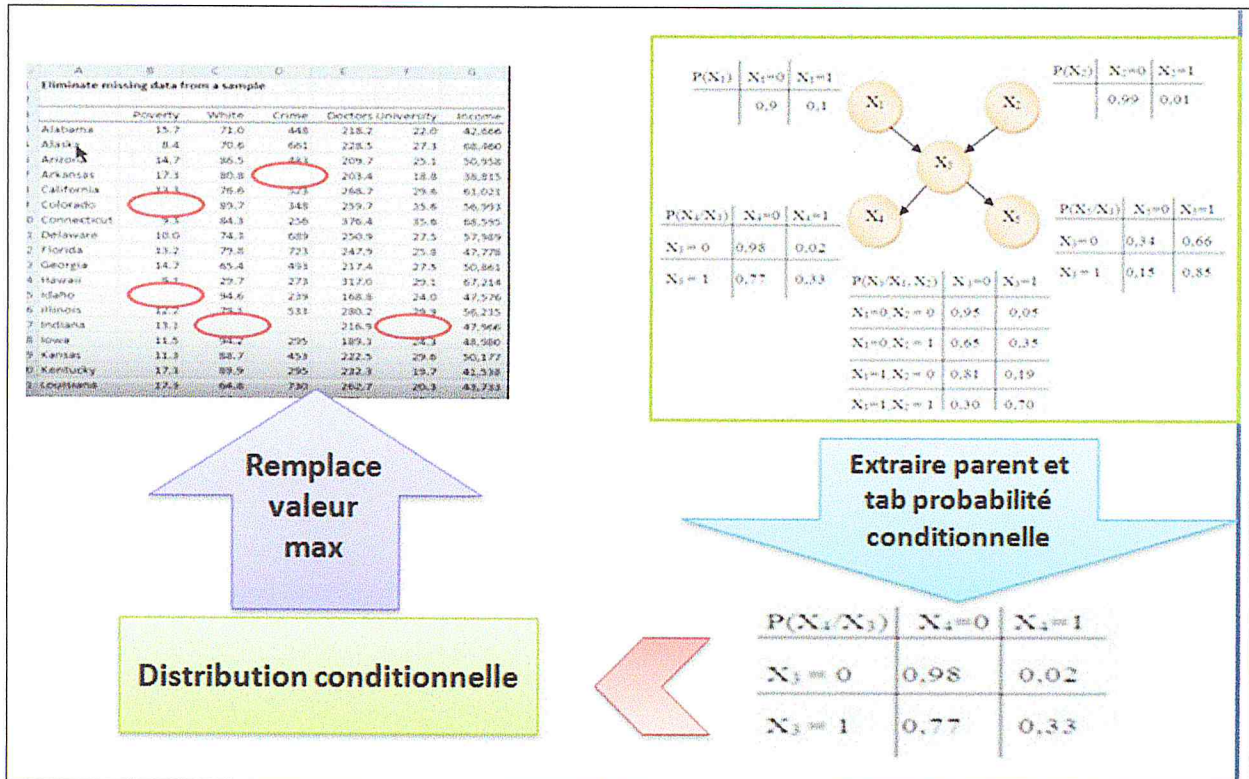
3. Inférence parallèle:

L'inférence dans un BN est le calcul de la probabilité conditionnelle des variables de requête, compte tenu d'un ensemble de variables de preuve comme connaissance du réseau. L'inférence dans un réseau bayésien peut être exacte ou approximative. En général, l'inférence exacte est NP difficile. Dans notre approche, l'inférence doit être faite en temps réel. En Inférence en temps réel, l'exactitude d'un calcul dépend non seulement de sa précision mais aussi de sa rapidité. Les algorithmes d'inférence approximatifs comme les algorithmes d'échantillonnage stochastiques peuvent faire une inférence en temps réel [47]. Cependant, dans notre approche, nous avons utilisé la version MapReduce de l'algorithme d'inférence exact largement utilisé, la propagation des preuves des arbres de jonction proposée par Ma & al [48].

3.1 Inférence exacte: inférence exacte dans un réseau bayésien Implique la détermination des probabilités de la requête Variables, compte tenu de l'état exact des variables de la preuve. Lorsque nous recevons de nouvelles informations sur les variables Réseau, nous mettons à jour les tableaux de probabilité conditionnelle Réfléchissez cette nouvelle information. Cette mise à jour est connue sous le nom de Propagation de la preuve [49]. Une fois que toutes les croyances sont mises à jour, Les tableaux de probabilité conditionnelle contiennent le plus récent Croyances dans n'importe quelle variable et peuvent être interrogées comme une simple Base de données pour évaluer les probabilités.

Compte tenu de la distribution commune des probabilités d'un arbitraire Réseau bayésien, on peut effectuer une inférence exacte sur le réseau.

Chapitre 4 : imputation des données manquante en temps réel



Algorithme 5 : (inférence parallèle)

Entre : données incomplet

Sortie : Données complet

Algorithme 5.1 : ThreadInference (File File) Implement Thread

{

Constructor(){

File =File }

Functionrun (){

$\forall x \in \text{Donnes_incomplete dans File}$

If $x \neq \text{null}$

$\forall \text{noued } y \in \text{cherche_Parent}(\text{noued } x) // \text{recherché dans la structure RB actuelle}$

If $y \neq \text{null}$

Valeur = get proba conditionnelle (noued y , noued x , valeur observe Parent)

Chapitre 4 : imputation des données manquante en temps réel

```
Valeur = valeur.Split
Maximum = valeur
Iterateur :valeur
Mimimun = valeur
if (Mimimun <= maximum) Maximum = maximum
Else maximum = minimum ;
value=RepalceValue_Incompet(Maximum)
Synchronise () //ordonner les threads le moment lire le fichier
Write_File(value) ;
}
ThreadInference1.start(Partie1_fichier) ; // lancer les thread
ThreadInference2.start((Partie2_fichier) ;
ThreadInference3.start((Partie3_fichier) ;
ThreadsInferenceN.start((PartieN_fichier) ;
}
}
```

4. Conclusion:

Dans ce chapitre, nous avons proposé une approche des données manquantes sur le nettoyage en temps réel. En nous concentrant sur les associations entre les attributs, nous avons proposé d'apprendre de manière décalée progressive un modèle RB à partir des données complètes et en utilisant l'inférence RB afin d'estimer les valeurs les plus adéquates des éléments manquants de données incomplètes en temps réel. Le modèle d'imputation de données manquantes basé sur RB est mis à jour de manière continue au moyen de la version distribuée incrémentale de K2 proposée dans ce travail. Nous avons présenté une implémentation d'un algorithme parallèle pour l'inférence base sur une méthode thread.

Chapitre V :

Implémentation

Chapitre V : Implémentation

1. Introduction :

Dans ce chapitre, nous allons aborder la partie pratique de notre projet qui consiste à décrire l'architecture, la conception technique et la mise en œuvre complète de l'environnement Hadoop .avec présentation de quelque teste de résultat

2. L'environnement de travail :

Dans la section suivante, on va décrire les outils et systèmes composant notre environnement de travail.

2.1 Cloudera : Cloudera a été formée en 2008 pour aider les entreprises à utiliser Hadoop pour obtenir plus de valeur sur ensemble de leur données.Cloudera fournit une plate-forme évolutive, flexible et intégrée qui facilite la gestion rapide et rapide des volumes et des variétés de données dans votre entreprise. Les produits et solutions Cloudera vous permettent de déployer et gérer Apache Hadoop et des projets connexes, de manipuler et d'analyser vos données et de garder ces données sécurisées et protégées.

2.2 Eclipse : est un environnement de développement intégré (IDE) utilisé dans la programmation informatique et est l'IDE Java le plus utilisé. ^[6] Il contient un espace de travail de base et un système extensible de plug-in pour personnaliser l'environnement. Eclipse est écrit principalement en Java et son utilisation principale est pour le développement d'applications Java, mais elle peut également être utilisée pour développer des applications dans d'autres langages de programmation via des plug-ins

2.3 Hadoop (version v2.7) :

Le logiciel Apache Hadoop, framework open-source complet pour le Big Data, représente l'une des meilleures approches pour traiter d'importants ensembles de données variés. Le principe de fonctionnement du framework Hadoop consiste à placer les ressources de calcul à proximité des données , Le framework Hadoop fournit un modèle de programmation simple pour le traitement distribué d'ensembles de données volumineux. Il inclut le système de fichiers distribué Apache* Hadoop (HDFS*), une structure de planification des tâches appelée Apache Hadoop YARN et une

Chapitre V : Implémentation

structure de traitement parallèle appelée Apache Hadoop MapReduce. Plusieurs composants supportent l'ingestion de données (service Apache Flume*), les requêtes et analyses (logiciels Apache Pig*, Apache Hive* et Apache HBase*), la coordination des workflows (Apache Oozie), ainsi que la gestion et la surveillance du cluster de serveurs sous-jacent (logiciel Apache Ambari*). Ensemble, les composants logiciels Apache créent un puissant framework pour traiter et analyser en lots des données dispersées pour des analyses historiques [w9]

Une architecture de machine HDFS repose sur 2 types :

NameNode (nœud de nom) : ce composant gère l'espace de nom, l'arborescence du système de fichier et les métadonnées des fichiers et des répertoires. Il centralise la localisation des blocs de données répartis dans le cluster. Il est unique mais dispose d'une instance secondaire qui gère l'historique des modifications dans le système de fichiers (rôle de backup). Ce NameNode secondaire permet la continuité du fonctionnement du cluster Hadoop en cas de panne du NameNode d'origine. **DataNode** (nœud de données) : ce composant stocke et restitue les blocs de données. Lors du processus de lecture d'un fichier, le NameNode est interrogé pour localiser l'ensemble des blocs de données. Pour chacun d'entre-eux, le NameNode renvoie l'adresse du DataNode le plus accessible, c'est-à-dire le DataNode qui dispose de la plus grande bande passante. Les DataNodes communiquent de manière périodique au NameNode la liste des blocs de données qu'ils hébergent. Si certains de ces blocs ne sont pas assez répliqués dans le cluster, l'écriture de ces blocs s'effectue en cascade par copie sur d'autres.

2.4 MapReduce : est un modèle de programmation pour le traitement des grandes masses de données avec un algorithme parallèle et distribué dans un cluster. Ce modèle est en train de connaître un vif succès auprès des sociétés possédant d'importants datacenters telles que Amazon ou Facebook, il commence aussi à être utilisé au sein du Cloud computing. Google l'a utilisé pour régénérer tout son index du Word Wild Web.

Principe

Il s'agit de décomposer une tâche en tâches plus petites, ou plus précisément découper une tâche portant sur de très gros volumes de données en tâches identiques portant sur des sous-ensembles de ces données. Les tâches (et leurs données) sont ensuite dispatchées vers différents serveurs, puis les résultats sont récupérés et consolidés. La phase de décomposition

Chapitre V : Implémentation

des tâches est la partie Map, tandis que la phase de la consolidation des résultats est la partie Reduce. [22] Chaque phase manipule des paires (clé,valeur) en entrée et sortie. Un cluster MapReduce utilise une architecture de type Maître-esclave où un noeud maître dirige tous les noeuds esclaves.

Mapreduce et hadoop :

Hadoop, une implémentation de MapReduce, fournit un cadre pour la distribution des données et des emplois, MapReduce spécifie par l'utilisateur à travers un grand nombre de noeud de cluster ou de machines. Il est basé sur l'architecteur maître/esclave. Le seul serveur maître, appelé JobTracher, reçoit une affectation des tâches de l'utilisateur, distribue la Map et de réduire les tâches de noeuds esclaves (tasktrachers) et surveille leur progression. Stockage et la distribution des données aux noeuds esclaves sont traitées par le système de fichiers distribués Hadoop (HDFS). Dans le texte suivant, un noeud Hadoop peut désigner un tasktrachers ou machine JobTracher. La tâche Map décrit le travail exécuté par un mappeur sur une partition d'entrée. La tâche reduce traite les dossiers avec la même clé intermédiaire. Un mapper / réducteur peut avoir multiple tâche Map / Reduce

3. Teste de résultat :

- a) Interface pour faire appel à l'administrateur que les données ont atteint le nombre

Pour lancer l'algorithme incrémentale

Chapitre V : Implémentation

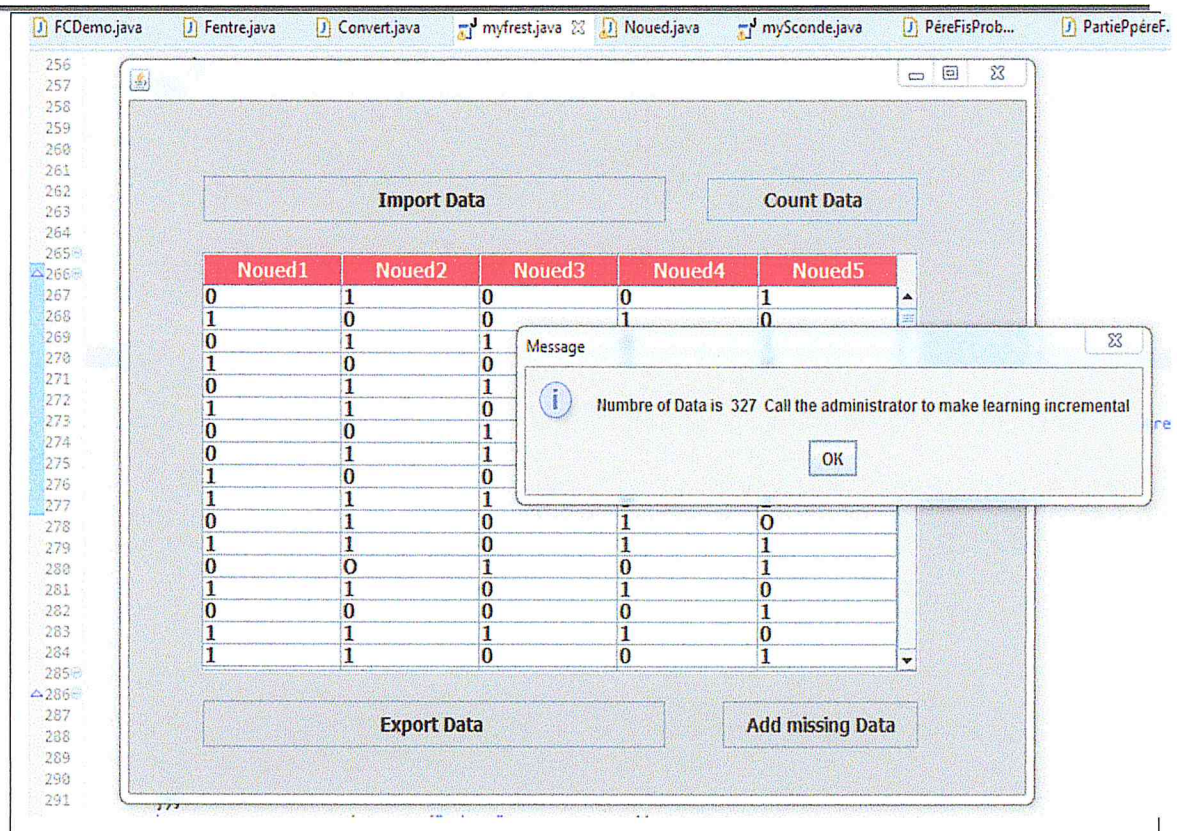


Figure 11 : signal a l'administrateur de lance algorithme incrémental

3.1 Apprentissage incrémentale des réseaux bayésiens

Dans ce qui suit nous allons représenter les teste de résultat et de algorithme k2 incrémentale

A) Calcule score : calcule score pour les candidats possibles à partir de chaque nœud

Chapitre V : Implémentation

```
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=120
File Output Format Counters
  Bytes Written=116
[cloudera@quickstart ~]$ hadoop fs -ls /calculeScore1
Found 2 items
-rw-r--r-- 1 cloudera supergroup      0 2017-09-05 20:49 /calculeScore1/_SUCCESS
-rw-r--r-- 1 cloudera supergroup    116 2017-09-05 20:49 /calculeScore1/part-000000
[cloudera@quickstart ~]$ hadoop fs -cat /calculeScore1
cat: '/calculeScore1': Is a directory
[cloudera@quickstart ~]$ hadoop fs -cat /calculeScore1/part-000000
x1x2    1.0
x1x3    0.25
x1x4    0.25
x1x5    0.5
x2x3    0.5
x2x4    0.5
x2x5    0.041666668
x3x4    0.5
x3x5    0.041666668
x4x5    0.041666668
[cloudera@quickstart ~]$
```

Figure 12: calcule score

- B) Calcule score max des candidate : calcule le score le plus élève incrémentale de chaque candidate pour la détermination des nœuds pères

Chapitre V : Implémentation

```
Combine output records=0
Reduce input groups=4
Reduce shuffle bytes=96
Reduce input records=10
Reduce output records=4
Spilled Records=20
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=151
CPU time spent (ms)=1410
Physical memory (bytes) snapshot=359084032
Virtual memory (bytes) snapshot=3007549440
Total committed heap usage (bytes)=242618368

Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0

File Input Format Counters
  Bytes Read=116
File Output Format Counters
  Bytes Written=36

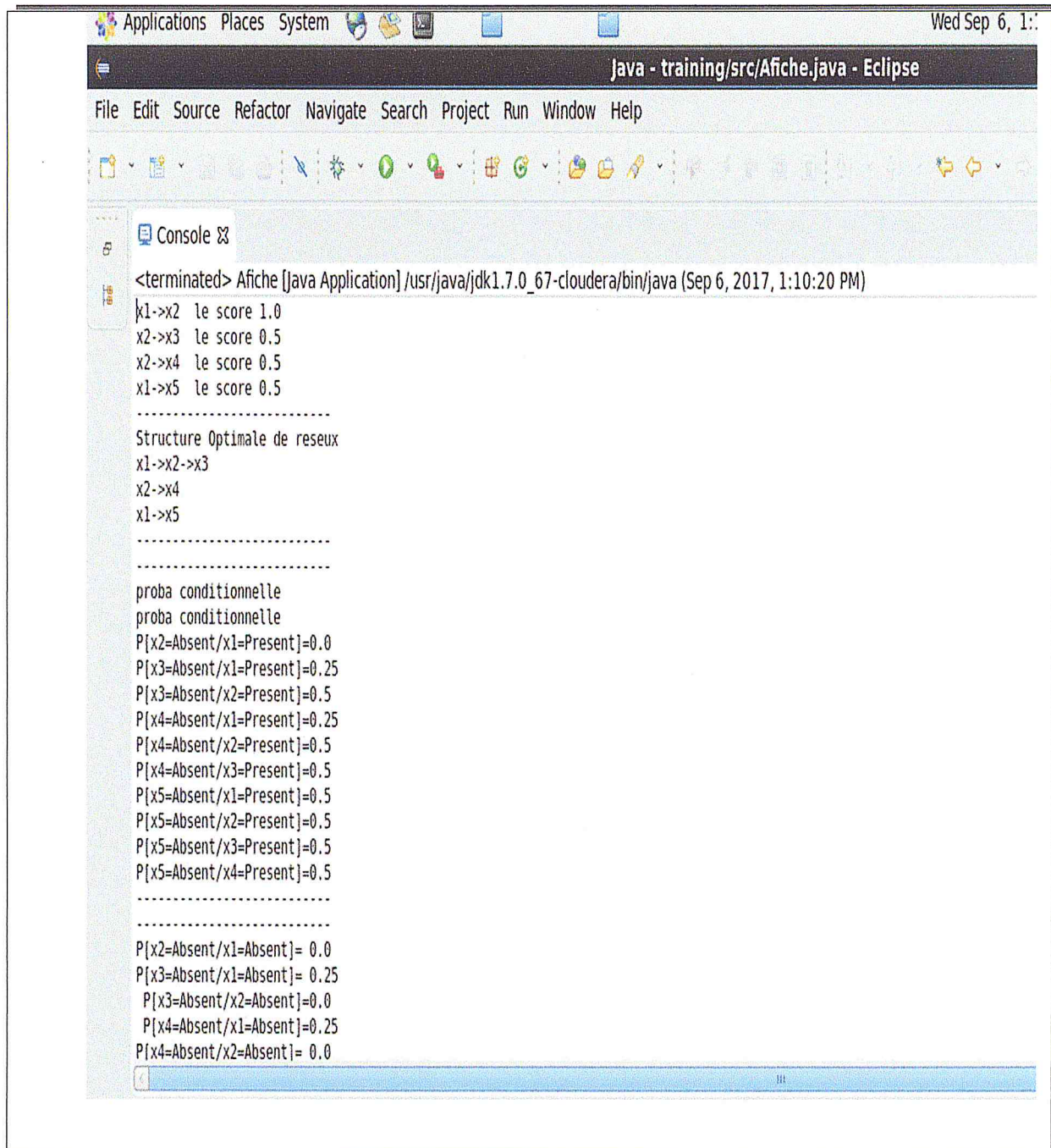
cloudera@quickstart ~]$ hadoop fs -ls /MaxScoreResult
Found 2 items
-rw-r--r--  1 cloudera supergroup          0 2017-09-06 12:18 /MaxScoreResult/_SUCCESS
-rw-r--r--  1 cloudera supergroup        36 2017-09-06 12:18 /MaxScoreResult/part-r-000000
cloudera@quickstart ~]$ hadoop fs -cat /MaxScoreResult/part-r-000000
1x2    1.0
2x3    0.5
2x4    0.5
1x5    0.5
cloudera@quickstart ~]$
```

[cloudera@quic...] [java - training/...] cloudera@quic... [cloudera@quic...] cloudera@quic...

Figure 13 : Calcule score max des candidate

d) Construire la structure RB globale : en utilisant algorithme K2 base sur MapReduce a partir du fichier sortie du traitement des nœuds, chaque nœud avec son père et sa table de probabilité conditionnel consiste a trouve la structure optimal après fusionner tous les structure optimal local.

Chapitre V : Implémentation



```
Applications Places System Wed Sep 6, 1:10:20 PM
Java - training/src/Afiche.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

Console
<terminated> Afiche [Java Application] /usr/java/jdk1.7.0_67-cloudera/bin/java (Sep 6, 2017, 1:10:20 PM)
x1->x2 le score 1.0
x2->x3 le score 0.5
x2->x4 le score 0.5
x1->x5 le score 0.5
.....
Structure Optimale de reseux
x1->x2->x3
x2->x4
x1->x5
.....
proba conditionnelle
proba conditionnelle
P{x2=Absent/x1=Present}=0.0
P{x3=Absent/x1=Present}=0.25
P{x3=Absent/x2=Present}=0.5
P{x4=Absent/x1=Present}=0.25
P{x4=Absent/x2=Present}=0.5
P{x4=Absent/x3=Present}=0.5
P{x5=Absent/x1=Present}=0.5
P{x5=Absent/x2=Present}=0.5
P{x5=Absent/x3=Present}=0.5
P{x5=Absent/x4=Present}=0.5
.....
P{x2=Absent/x1=Absent}= 0.0
P{x3=Absent/x1=Absent}= 0.25
P{x3=Absent/x2=Absent}=0.0
P{x4=Absent/x1=Absent}=0.25
P{x4=Absent/x2=Absent}= 0.0
```

Figure 14 : la structure optimale de RB

e) Interface des données incomplète : on a utilisé un programme qui vide quelque variable les rendre manquante pour être utilise a inférence parallèle et le teste notre application

ipse-worksp

Edit Sour

FCDem

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

Source

Import data

Noued1	Noued2	Noued3	Noued4	Noued5
0	1	0	1	1
1	1	0	1	0
0	1	1	1	0
1	0	0	0	1
0	0	1	0	1
1	1	0	1	0
0	0	1	0	0
0	1	1	0	0
1	0	0	1	0
1	0	1	0	1
0	0	0	1	0
1	0	0	1	0
0	0	1	0	1
1	1	0	1	0
0	1	0	0	1
0	0	1	1	1
1	0	0	0	1
1	1	0	1	0
0	1	0	1	1

Noued1	Noued2	Noued3	Noued4	Noued5
0	--	0	1	1
1	++	0	1	0
!	1	1	1	0
1	0	/	0	1
0	'	1	0	1
1	1	&	1	0
0	0	1	0	'
0	1	1	0	=
'	0	0	1	0
**	0	1	0	1
&	0	0	1	0
1	0	0	1	&
0	&	1	0	1
1	1	0	@	0
0	1	0	&	1
0	0	1	}}	1
1	0	0	++	1

Make missing Data

Percentage correction

Add RB

Estimate Data

3.2 Inférence parallèle :

72

Chapitre V : Implémentation

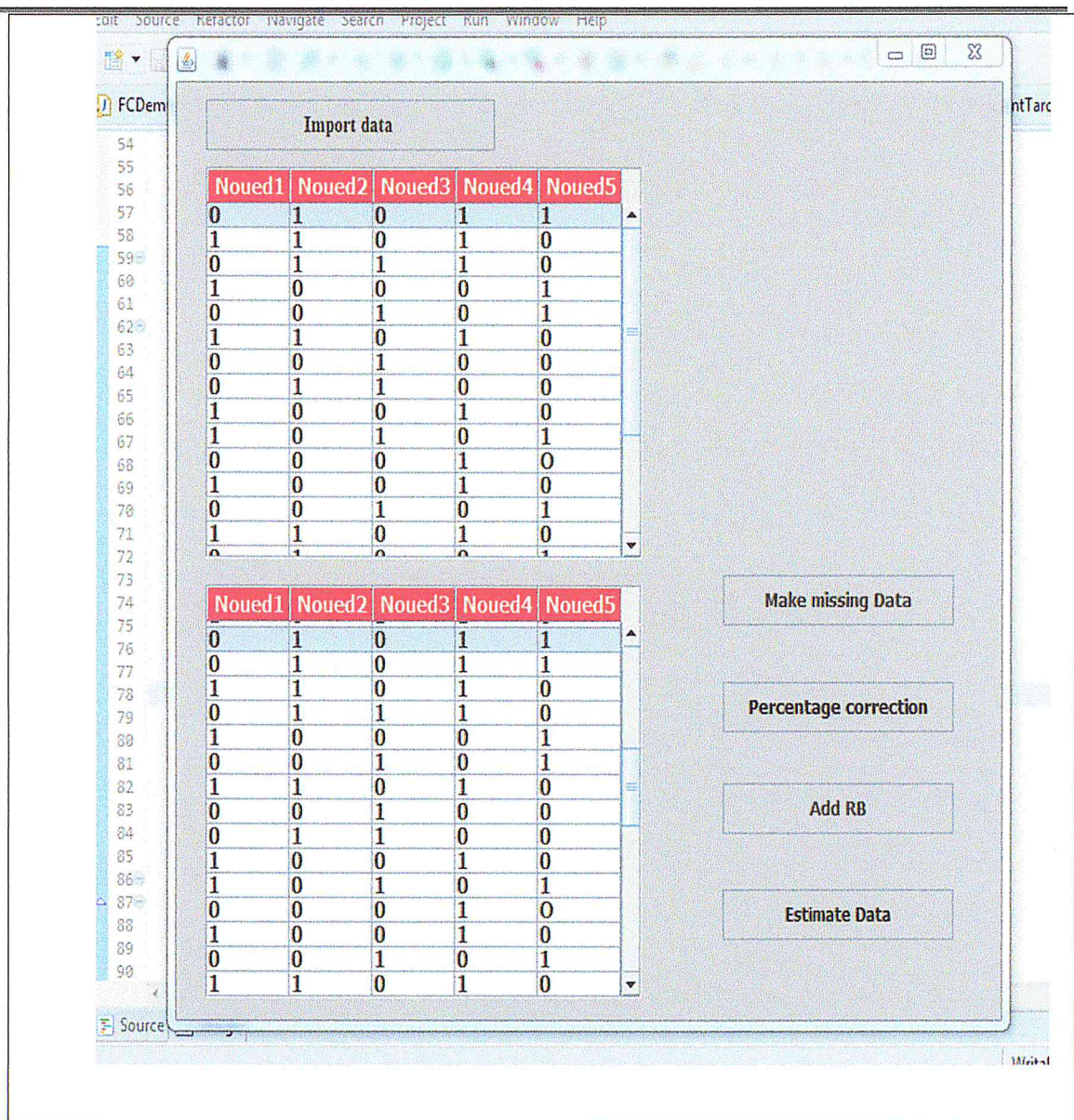


Figure 16 : Inférence parallèle

- b) pourcentage de données incomplètes : l'interface indique le pourcentage de données manquantes après l'imputation des données (inférence parallèle) cette interface teste l'efficacité de notre travail

Chapitre V : Implémentation

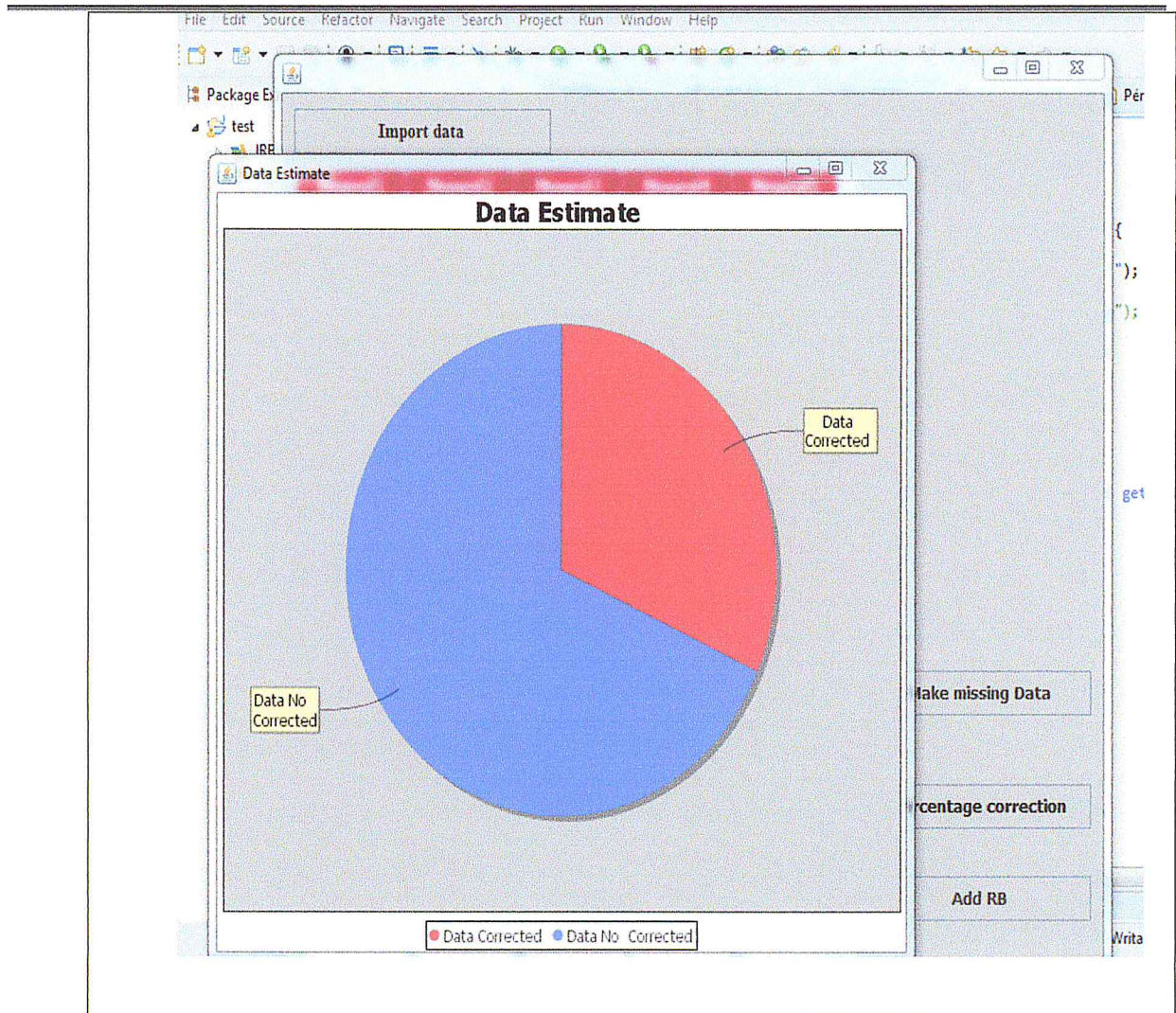


Figure 17 : pourcentage de données incomplet

4. **Conclusion :** Nous avons décrit dans ce chapitre la partie de l'implémentation ainsi que les différents outils utilisés pour la réalisation de notre projet de application après on a fait des teste qui été fait avec les technologies big data notamment MapReduce qui nous aide à faire parallélisme et améliore le temps d'exécution.

Conclusion

Conclusion général :

L'appréhension des données incomplète est un problème délicat. Non pas à cause de sa gestion informatique mais plutôt à cause des conséquences de leur traitement, Notre projet a fait l'objet d'une étude technologie imputation de données manquante basé sur l'utilisation des réseaux bayésiens pour la modélisation de systèmes d'imputation. Cette modélisation passe par la détermination de la structure et des paramètres du réseau Bayésien en premier lieu, par la suite l'utilisation de l'inférence pour déterminer les valeurs les plus probables des valeurs manquantes.

Au cours de la réalisation de ce mémoire, on est passé d'une généralisation à une spécialisation. En premier lieu nous avons étudié les différentes méthodes de traitement de données incomplètes et une présentation des différentes méthodes d'imputation qu'ils ont été comparés par leurs performances, selon notre problème d'initialisation de données manquantes.

Puis, nous avons étudié les différentes techniques d'apprentissage de structure de réseaux bayésiens à partir de données complètes et nous avons réalisé une étude des principaux algorithmes d'inférence et d'apprentissage les plus fréquents sur les Réseaux Bayésiens.

Notre Projet est basé sur le framework Hadoop, L'objectif assigné à notre présent travail était de mettre en œuvre une application pour l'imputation des données à partir des sources de données incomplètes dans un entrepôt de données, nous avons besoin d'une version incrémentale distribuée de MapReduce, Utilisant l'algorithme incrémental basé sur MapReduce, qui peut être utilisé pour obtenir tous les paramètres en parallèle. En processus de Map, les paramètres de chaque enregistrement peuvent être obtenus par les statistiques.

Le paradigme de calcul parallèle MapReduce pour traiter parallèlement, au sein d'un cluster de serveurs, des données massives, connues par Big Data.

La mise en œuvre d'un cluster Hadoop avec ses composants, l'exécution de plusieurs jobs MapReduce sur des fichiers HDFS dans des architectures à plusieurs nœuds et l'évaluation des différents statistiques des opérations effectuées nous ont permis d'approuver l'impact de la parallélisme sur l'efficacité des systèmes gérant de grands volumes de données en réduisant la latence et en garantissant une meilleure extensibilité.

Bibliographie

Bibliographie

Bibliographie

[1] : Stéphane , Paquin .Comparaison de quatre méthodes pour le traitement des données manquantes au sein d'un modèle multiniveau paramétrique visant l'estimation de l'effet d'un programme de prévention. Université de Montréal .2010.

[2] : PHILIPPE BESSE & BÉATRICE LAURENT Apprentissage Statistique modélisation, prévision, data mining cours Institut National des Sciences Appliquées de Toulouse

[3] : Rousseau, Michel .L'impact des méthodes de traitement des valeurs manquantes sur les qualités psychométriques d'échelles de mesure de type Likert Université Laval 2006

[4] : Little et Rubin Statistical Analysis with Missing Data, New York: Wiley , 2002

[5]: Roderick J. A. Little, Donald B. Rubin Statistical Analysis with Missing Data. September 2002

[6] : Enders CK. A Primer on Maximum Likelihood Algorithms Available for Use With Missing Data. STRUCTURAL EQUATION MODELING. 2001

[7] : Pigott td . Missing predictors in models of effect size.Loyola University 2001

[8] : Sinharay, Stern, et Russel, The use of multiple imputation for the analysis of missing data. Iowa State University 2001

[9] : Schafer, Analysis of Incomplete Multivariate Data1997

[10]: Schafer et Graham, Missing data: our view of the state of the art 2002

[11] :ABDERRAZAK , BENNANE .TRAITEMENT DES VALEURS MANQUANTES POUR L'APPLICATION DE L'ANALYSE LOGIQUE DES DONNEES À LA MAINTENANCE CONDITIONNELLE. UNIVERSITÉ DE MONTRÉAL.AOUT 2010.

[12] : Rosenbaum & Rubin, The Central Role of the Propensity Score in Observational Studies for Causal Effects 1983

[13]: D'Agostino & Rubin, Inverse probability weighting with error-prone covariates 2000

[14]: Benoît Virole RESEAUX DE NEURONES ET PSYCHOMETRIE, Centre de Psychologie Appliquée Juin 2001

[15] : Christophe Gonzales—Pierre-Henri Willemin Réseaux bayésiens en modélisation d'utilisateurs, Université Paris 6

[16] : ALBRECHT, D., ZUKERMAN, I., NICHOLSON, A., BUD, A., "Towards a Bayesian model for keyhole plan recognition in large domains, Paris.

Bibliographie

- [17]: F.V Jensen. Bayesian Networks and Decision Graphs. Springer, 2001.
- [18]: S. Baghdadi, G. Gravier, C.H. Demarty, et P. Gros. Structure learning in a Bayesian network-based video indexing. In Proc. IEEE ICME.
- [19] : Abdesslem Bouzaïeni. Réseaux Bayésiens et quelques applications en traitement d'images Université de Nancy 2
- [20] : Iounis.h Fondements des réseaux Bayésiens 2006.
- [21] : Philippe LERAY .Réseaux bayésiens : apprentissage et modélisation de systèmes complexes. Université de Rouen 2006
- [22]: François Olivier ,De l'identification de structure de réseaux Bayésiens `a la reconnaissance de formes `a partir d'informations complètes ou incomplètes. Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes.
- [23] : BRAHIMI Belgacem, Extraction de connaissances à partir de données incomplètes et imprécises, 2011
- [24]: Rissanen, J. Modeling by shortest data description. Automatica 1978.
- [25]: Cooper G. & Hersovits E. A bayesian method for the induction of probabilistic networks from data. *Maching Learning*, 1992 .
- [26]: Heckerman D. & Geiger D. & Chickering M. Learning Bayesian networks: *The combination of knowledge and statistical data*. In R. L. de Mantaras & D. Poole, Eds, Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence, 1994
- [27]: Fatiha NAOUAR , ALGORITHME APPROXIMATIF POUR LE DIAGNOSTIC MÉDICAL BASÉ SUR UN RÉSEAU BAYÉSIEN POSSIBILISTE GUIDÉ PAR ONTOLOGIE FLOUE , Université de Sousse , 2007
- [28]: ZAHRA FATMA ZOHRA, UTILISATION DES RESEAUX BAYESIENS POUR CALCULER LA FIABILITE DES SYSTEMES, UNIVERSITE SAAD DAHLAB DE BLIDA , 2012
- [29]: Olivier PARENT Julien EUSTACHE Les Réseaux Bayésiens A la recherche de la vérité Université Claude Bernard Lyon, 2007
- [30] : Marie-Pierre Béal , Nicolas Sendrier .Théorie de l'information et du codage . Institut Gaspard Monge
- [31]: SH. Lazare / F. Barthélemy, Introduction Big Data , AXIODIS 2005.
- [32]: M.CORINUS, T.Derey, J.Marguerie, W.Techer, N.Vic, Rapport d'étude sur le Big Data, octobre 2012

Bibliographie

- [33] : himanshu et soumendra mohanty Big data application and architecteur ;
- [34]: Mekideche Mounir, Conception et implémentation d'un moteur de recherche à base d'une architecture Hadoop (Big Data), Avril 2015
- [35] : C.Z. Ye, J. Yang, D.Y. Geng: Fuzzy rules to predict degree of malignancy in brain glioma. Med Biol Eng Comput,2000
- [36] : Bermond, A. *Big Data*. (2013).
- [37]: Olivier François, Quand devons-nous utiliser l'algorithme EM pour effectuer un apprentissage de structure de RB ?,2008
- [38]: Nakai, M., Ke, W.: Review of the Methods for Handling Missing Data in Longitudinal Data Analysis. In: Journal of Math. Analysis, Volume 5, no. 1. (2011)
- [39]: Chu, F., Wang, Y., Parker, D.S, Zaniolo, C.: Data Cleaning Using Belief Propagation, In: Proceedings of *IQIS*. (2005)
- [40]: Uffe, B., Kjbrulff-Anders, L.: Bayesian Networks and Influence Diagrams A Guide to Construction and Analysis. In: Springer Science + Business Media, LLC (2008).
- [41]: Friedman, N., Goldszmidt, M.: Sequential update of Bayesian network structure. In: Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI'97), Providence, Rhode Island, USA. (1997).
- [42]: Saeed, S., Miri, A., Granger, E.: Incremental learning of privacy-preserving Bayesian networks. In: Elsevier journal of Applied Soft Computing 13. (2013).
- [43]: Fang, Q., Yue, K., Fu, X., Wu, H., Liu, W.: A Mapreduce-based method for learning Bayesian network from Massive Data. In: APWeb 2013, LNCS 7808. (2013).
- [44]: Böse, J.H., Andrzejak, A., Höggqvist, M.: Beyond Online Aggregation: Parallel and Incremental Data Mining with Online Map-Reduce (DRAFT). In: DAC '10, April 26, 2010 Raleigh, NC, USA. (2010).
- [45]: Daewoo, L., Jin-Soo, K., Maeng, S.: Large-scale incremental processing with MapReduce. In: Elsevier Future Generation Computer Systems 36 (2014).
- [46]: H. Guo and William H. Hsu: A Survey of Algorithms for Real-Time Bayesian Network Inference. AAAI/KDD/UAI-2002 Joint Workshop on Real-Time Decision Support and Diagnosis Systems. Edmonton, 29 July 2002.

Bibliographie

- [47]: Ma, M., Xia, Y., Prasanna, FK.: Parallel Exact Inference on Multicore Using MapReduce. In: IEEE 24th International Symposium on Computer Architecture and High Performance Computing. (2012).
- [48] : S. L. Lauritzen and D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," in J. of the Royal Statistical Society, 1988,
- [w1]: <http://www.journaldunet.com/solutions/analytics/big-data>
- [w2]: <http://www.piloter.org/business-intelligence/technologie-big-data.htm>
- [w3] : <https://www.saagie.com/fr/blog/hadoop-et-big-data>
- [w4] : <http://mbaron.developpez.com/tutoriels/bigdata/hadoop/introduction-hdfs-map-reduce/>
- [w5] : <http://www.devx.com/opensource/exploring-the-hadoop-distributed-file-system-hdfs.html>
- [w6] : <https://www.infoq.com/fr/articles/apache-spark-introduction>
- [w7] : <http://blog.ontrack.fr/big-data-pourquoi-nosql>
- [w8] : <https://www.linkedin.com/pulse/-big-data-with-sql>
- [w9] : Guide de planification Se lancer dans le Big Data, intel.fr.juin 2014

