Université de Reims Champagne-Ardenne

École Doctorale Sciences du Numérique et de l'ingénieur

THÈSE

pour obtenir le grade de

Docteur de l'Université de Reims Champagne-Ardenne

Discipline : informatique

Présentée et soutenue publiquement le 22 novembre 2019 par

Joris Falip

Structuration de données multidimensionnelles : une approche basée instance pour l'exploration de données médicales

JURY

M. Pierre Gançarski	Professeur, Université de Strasbourg	*Président*
Mme. Anne Boyer	Professeur, Université de Nancy	*Rapporteur*
Mme. Marie-Jeanne Lesot	Maître de conférences HDR, Université Paris VI	*Rapporteur*
Mme. Zahia Guessoum	Maître de conférences HDR, Université de Reims	*Examinateur*
M. Michel Herbin	Professeur, Université de Reims	*Directeur*
M. Frédéric Blanchard	Maître de conférences, Université de Reims	*Co-encadrant*

Résumé

L'exploitation, a posteriori, des données médicales accumulées par les praticiens représente un enjeu majeur pour la recherche clinique comme pour le suivi personnalisé du patient. Toutefois les professionnels de santé manquent d'outils adaptés leur permettant d'explorer, comprendre et manipuler aisément leur données. Dans ce but, nous proposons un algorithme de structuration d'éléments par similarité et représentativité. Cette méthode permet de regrouper les individus d'un jeu de données autour de membres représentatifs et génériques aptes à subsumer les éléments et résumer les données. Cette méthode, procédant dimension par dimension avant d'agréger les résultats, est adaptée aux données en haute dimension et propose de plus des résultats transparents, interprétables et explicables. Les résultats obtenus favorisent l'analyse exploratoire et le raisonnement par analogie via une navigation de proche en proche : la structure obtenue est en effet similaire à l'organisation des connaissances utilisée par les experts lors du processus décisionnel qu'ils emploient. Nous proposons ensuite un algorithme de détection d'anomalies qui permet de détecter des anomalies complexes et en haute dimensionnalité en analysant des projections sur deux dimensions. Cette approche propose elle aussi des résultats interprétables. Nous évaluons ensuite ces deux algorithmes sur des données réelles et simulées dont les éléments sont décrits par de nombreuses variables : de quelques dizaines à plusieurs milliers. Nous évaluons les résultats obtenus sur ces données en grande dimension, en analysant particulièrement les propriétés du graphe résultant de la structuration des éléments. Nous décrivons par la suite un outil de prétraitement de données médicales ainsi qu'une plateforme web destinée aux médecins. Via cet outil à l'utilisation intuitif nous proposons de structurer de manière visuelle les éléments pour faciliter leur exploration. Ce prototype fournit une aide à la décision et au diagnostique médical en permettant au médecin de naviguer au sein des données et d'explorer des patients similaires. Cela peut aussi permettre de vérifier des hypothèses cliniques sur une cohorte de patients.

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué à ce travail de thèse, directement ou indirectement, et l'ont transformé en une expérience scientifique et humaine enrichissante.

Plus particulièrement, je souhaite remercier mon directeur, Michel Herbin, et mon co-encadrant, Frédéric Blanchard, pour m'avoir offert cette opportunité unique et m'avoir aidé tout au long du parcours de la thèse : leurs conseils ont été précieux et ont joué un rôle central dans le déroulement de ce travail. Leur vision et leur expérience ont façonné mon approche de la recherche et ont toujours confirmé ma volonté de poursuivre une carrière scientifique, et pour cela je les en remercie encore une fois.

Je remercie également Marie-Jeanne Lesot, Anne Boyer, Pierre Gançarski et Zahia Guessoum pour avoir accepté d'être membres de mon jury. Vos remarques ont contribué à la qualité du manuscrit et m'ont en plus permis de prendre du recul par rapport aux travaux effectués ainsi qu'à leurs implications.

J'adresse aussi mes remerciements aux collègues et amis du laboratoire CReSTIC qui ont contribué à mes recherches autant par leur points de vue critiques que par leur bonne humeur. Merci spécialement aux membres de l'équipe UTILE, aux doctorants avec qui j'ai eu le plaisir de partager ces trois ans, à Bernard Riera et Nicolas Passat pour leur implication sans faille dans la réussite des doctorants, et enfin à Marielle Tur et Ida Lenclume pour leur bienveillance et leur humour de toutes les circonstances.

Je tiens à remercier, aussi, ma famille et mes amis. Vous avez totalement transformé cette expérience en quelque chose d'incroyablement épanouissant. Merci entre autres à Valérian Delouche, Antoine Lambert et Geoffroy Thiebaud qui ont été présents depuis le début de l'aventure. Merci aussi à Quentin Czerwiec, Nicolas Courilleau, Ulysse Larvy Delariviere, Mohamed Bennai et Jonathan Sarton : plus qu'avec des collègues, c'est avec des amis que j'ai eu le plaisir de partager ces trois années de thèse.

Enfin, merci tout particulièrement à mes parents, à Emilie Huby et à Pierre Cettour-Janet. Votre soutien inconditionnel m'a porté toutes ces années et amené là où je suis. Merci.

Table des matières

1 Introduction **1**
- 1.1 Contexte général . . . 3
- 1.2 Objectifs scientifiques . . . 6
- 1.3 Contributions . . . 7
- 1.4 Plan du manuscrit . . . 9

2 État de l'art **11**
- 2.1 Analyse exploratoire . . . 13
- 2.2 Haute dimensionnalité . . . 17
- 2.3 Interprétabilité . . . 28
- 2.4 Processus décisionnel . . . 33

3 Algorithme de structuration **41**
- 3.1 Représentativité . . . 43
- 3.2 Structuration . . . 51
- 3.3 Propriétés . . . 58
- 3.4 Détection d'anomalies . . . 61

4 Expérimentations **65**
- 4.1 Évaluation . . . 67
- 4.2 Résultats . . . 75

5 Application aux données médicales **83**
- 5.1 Données médicales . . . 85
- 5.2 Pré-traitement . . . 86
- 5.3 Prototype d'exploration . . . 91

6 Conclusion **95**
- 6.1 Structuration par représentativité . . . 97
- 6.2 Prototype d'aide au diagnostic médical . . . 99
- 6.3 Perspectives . . . 100

A Annexes **103**

Colophon **107**

Bibliographie **111**

Table des figures

2.1 Analyse graphique univariée 16
2.2 Analyse graphique multivariée 16
2.3 Ratio de l'hypersphère par rapport à l'hypercube circonscrit 18
2.4 Pourcentage d'éléments en queue d'une distribution normale 20
2.5 Illustration de la concentration des distances 21
2.6 Concentration des distances euclidiennes 22
2.7 Distribution de la hubness sur des données uniformes 24
2.8 Distribution de la hubness sur des données normales 25
2.9 Distribution de la hubness sur des données réelles 25
2.10 Taxonomie de l'interprétabilité 29
2.11 Explications textuelles de résultats 32
2.12 Explication des résultats par visualisation 33
2.13 Explication des résultats par l'exemple 34
2.14 Création d'un prototype pour un concept 37
2.15 Comparaison prototypes et exemples 38

3.1 Fonctions de transformation en score 50
3.2 Évolution des composantes connexes sur Iris 55
3.3 Structuration de Iris avec k = 30 56
3.4 Structuration de Iris avec k = 75 56
3.5 Graphe des associations de Iris avec k = 75 57
3.6 Estimation de typicalité 59
3.7 Détails d'une association entre éléments 61
3.8 Anomalies et distributions uniformes 63

4.1 Comparaison de composantes connexes 69
4.2 Graphe Iris plus proche voisin Minkowski 77
4.3 Graphe Iris k45 78
4.4 Détail d'une composante connexe de Residentiel 80

5.1 Pré-traitement écran accueil 88
5.2 Pré-traitement et conversion des dates 89
5.3 Pré-traitement et agrégation des patients 89
5.4 Pré-traitement et règles d'imputation 90
5.5 Analyse de survenue des complications 91
5.6 Fiche détaillée d'un patient 92
5.7 Visualisation des patients par similarité 93

A.1 Traitement des colonnes par lot 103
A.2 Filtrage des patients étudiés grâce au prototype 104
A.3 Vérification de la sélection de patients 105

Liste des tableaux

2.1 Analyse non-graphique univariée 15
2.2 Formules des distances Minkowski, Manhattan et Euclidienne. 22

4.1 Description des jeux de données pour la structuration. 71
4.2 Description des jeux de données pour la détection d'anomalies. 72
4.3 Comparaison taille composantes 75
4.4 Comparaison diamètre composantes 76
4.5 Comparaison fmesure classification 79
4.6 Comparaison fmesure detection d'outliers 81

Chapitre 1

Introduction

Sommaire

1.1 Contexte général 3
1.1.1 Aide à la décision 3
1.1.2 L'impératif d'interprétabilité 4
1.1.3 Analyse de données médicales 5
1.2 Objectifs scientifiques 6
1.3 Contributions 7
1.3.1 Communications 8
1.4 Plan du manuscrit 9

1.1 Contexte général

Noyés dans l'océan digital

« Vous n'êtes pas le fric que vous avez à la banque, vous n'êtes pas votre voiture, vous n'êtes pas votre portefeuille, vous n'êtes pas la marque de vos jeans » [1] lorsqu'il a écrit cette réplique David Fincher n'avait probablement pas anticipé à quel point elle serait loin de la réalité vingt ans après. Pourtant, l'obsession créative de l'être humain s'est interrogée il y déjà longtemps sur la possibilité d'une société de contrôle omniprésente qui réduirait l'humain à des variables arbitraires. « Le télécran n'avait aucun moyen d'être éteint complètement. [. . .] Il n'y avait pas moyen de savoir si, à un moment donné, on était surveillé. » [2] nous disait déjà George Orwell en 1949 et depuis, il n'est même plus possible de savoir où sont les télécrans. Cette société d'omni-surveillance, théorisée en détail il y a pourtant plusieurs décenies [3], est revenue au centre de nombreuses oeuvres artistiques [4] [5].

Cette dystopie du quotidien, surnommée « société de traces » par Alain Damasio, est rendue possible par l'ubiquité des données : ces traces de nos interactions que nous laissons derrière nous à chaque rencontre avec une machine animée au silicium. C'est en centralisant ces données qu'il devient possible de mieux connaître les citoyens que l'on gouverne, les clients que l'on cible, les concurrents que l'on surveille. Au grand dam de Fincher, nous sommes devenus les produits que l'on consomme et les traces que l'on laisse. Cela à tel point que ces données sont souvent qualifiées de « pétrole du 21ème siècle » [6]. [7] Toutefois, l'analogie s'arrête ici car là où celui du 20ème siècle s'épuise, le pétrole du 21ème siècle est omniprésent et se démultiplie au point de devenir un enjeu majeur de notre époque dont l'exploitation est une préoccupation aussi bien scientifique qu'industrielle, économique ou sociétale (Mayer-Schönberger & Cukier, 2013). Le volume et la complexité des données disponibles ne cessent de s'accroître exponentiellement (Turner, Gantz, Reinseil, & Minton, 2014), les données sont plus volumineuses, fréquentes, hétérogènes que jamais. Générées à des rythmes soutenus, souvent en flux ininterrompus et permanents, ces données qui occupent des dizaines de zétaoctets revêtent toutes les formes, de l'image au son en passant par la vidéo, le texte, les informations statistiques, géographiques ou temporelles (Erevelles, Fukawa, & Swayne, 2016).

1.1.1 Aide à la décision

Si notre société de consommation est imbattable quand il s'agit de laisser des traces (McAfee & Brynjolfsson, 2012), c'est dans l'exploitation de ce capital économique sans bornes que de nombreuses difficultés se posent. Ainsi, dans ce panoptique où chacun alimente sans cesse un *alter ego* numérique qui peut être exploité à coups d'algorithmes, une fuite en avant s'est organisée autour de la capacité de l'humain à voir plus, plus vite

1. *Fight Club*, David Fincher (1999). Film.
2. *1984*, George Orwell (1949). Secker and Warburg
3. *Shadowrun*, (1989). FASA Corporation
4. *Black Mirror*, Charlie Brooker (2011)
5. *Mr Robot*, Sam Esmail (2015)
6. *Data is the new oil of the digital economy*, Wired (2014)
7. *The world's most valuable resource is no longer oil, but data*, The Economist (2017)

et mieux. Pour parvenir à cela, des domaines scientifiques et technologiques comme l'intelligence artificielle (incluant l'apprentissage automatique ou les systèmes d'aide à la décision) ont de nombreux challenges à relever car la croissance hors norme des volumes de données est reflétée par les opportunités décuplées de les exploiter [8]. C'est donc par l'importance et l'ubiquité des données qu'un effort global de perfectionnement des outils et algorithmes de *machine learning* a vu le jour afin de traiter et d'extraire des informations à partir de ces traces. Ce besoin d'outils adaptés a été accentué par les difficultés exposées précédemment (volume, variété, vélocité des données (Erevelles et al., 2016)), et la demande grandissante a donné lieu à l'émergence d'un domaine lui étant dédié. Car récolter les données et les agréger n'est en effet que la première étape du processus de compréhension et d'extraction d'information, il est ensuite nécessaire (tout comme le pétrole) de les raffiner pour passer des données aux connaissances qui, elles, peuvent fournir des informations sur le contexte dans lequel elles ont été récoltées et les objets qu'elles décrivent. L'objectif est de traiter les données non pas dans le but d'être utilisées par un autre programme ou algorithme, mais de les rendre utilisables directement pour l'être humain afin d'en extraire des tendances ou des explications par exemple. C'est le but des méthodes d'aide à la décision.

Assistant l'humain pour analyser, manipuler, explorer et visualiser les bases de données disponibles, les systèmes d'aide à la décision se sont popularisés en réaction à l'ubiquité des données dans tous les domaines professionnels. Le but de ces outils est de permettre la compréhension des jeux de données, afin d'en retenir de l'information pertinente. Cette information peut prendre plusieurs formes comme la prédiction d'évènements futurs, l'explication des mécanismes générant ces données, la présence de motifs ou de tendances dans leur génération, la découverte et la validation empirique de nouvelles hypothèses ou encore la recherche de similarités entre éléments ou situations.

En utilisant un système d'aide à la décision pour assister l'expert, il est possible d'évaluer les conclusions du premier grâce au jugement expérimenté du second. Ces outils utilisant des connaissances pour assister l'utilisateur ont commencé très tôt avec les systèmes experts basés sur les règles : un ensemble de règles est utilisé pour modéliser des connaissances sur un domaine, et les décisions découlent de ce modèle. Il s'agit donc de digitaliser l'expertise humaine pour pouvoir y recourir plus facilement et rapidement. D'autres outils comme les systèmes de recommandations permettent d'orienter l'utilisateur dans l'exploration de données en lui fournissant des éléments jugés comme pertinents. Ici l'objectif est d'épargner à l'utilisateur l'analyse exhaustive des données en l'aiguillant directement vers celles qui devraient l'intéresser.

1.1.2 L'impératif d'interprétabilité

Si les progrès dans l'aide à la décision se multiplient et trouvent rapidement leur chemin vers des innovations industrielles, ils ont aussi un écho considérable auprès du grand public dont ces outils complexes définissent le quotidien. Car ces systèmes permettant d'aider l'être humain dans sa consommation d'information, de divertissement ou dans son travail sont des systèmes souvent opaques pour les utilisateurs dont ils façonnent pourtant les interactions avec l'information et les médias par exemple. Parfois l'aide à la

8. *Rapport de synthèse France IA*, 2017

décision occupe même un rôle plus prévalant, guidant les décisions stratégiques de multinationales, l'admissibilité de la population à l'éducation ou encore les lois des gouvernements. C'est pour cette raison, et à cause de l'exploitation des traces à tous les niveaux de la société et dans tous les types d'industries, que de nouvelles considérations sont apparues [9]. Les notions de transparence et d'explicabilité sont devenues des facteurs essentiels aux libertés individuelles et à la vie privée : c'est par le spectre de ces deux qualités qu'il devient possible d'appliquer une législation et un contrôle rationels de ces outils complexes, et que le public peut être acteur de cette transformation plutôt que de subir les décisions prises ou suggérées par ces algorithmes. Et si les algorithmes deviennent de plus en plus des boîtes noires, c'est une interprétabilité accrue qui se pose comme solution pour des applications éthiques et éclairées. Dans des secteurs critiques comme la justice, la finance, le médical ou la sécurité par exemple, il est important de comprendre la provenance des résultats des méthodes de *machine learning* sous peine d'en être victime : « la technique, c'est l'ensemble de ce qu'il faut savoir pour échapper à la technique » [10]. En vérifiant comment et pourquoi ces résultats sont calculés, il devient possible de les questionner, de les vérifier, de s'assurer de l'absence de biais et d'accroître la confiance d'un utilisateur envers les systèmes qui façonnent son quotidien. Cela fait donc de l'interprétabilité un facteur important pour une utilisation éthique de ces technologies (Tran, Riveros, & Ravaud, 2019). Toutefois dans ces mêmes domaines critiques cités plus tôt, l'intelligence artificielle n'a pas vocation à remplacer l'être humain mais à l'épauler pour mieux valoriser les bases de données toujours plus nombreuses. Car les données peuvent être réutilisées sans limites, gagnant à chaque fois en intérêt et en valeur. L'objectif pour valoriser ces données est donc de les utiliser afin de produire des connaissances et, *in fine*, de la rationnalité.

1.1.3 Analyse de données médicales

Pour atteindre cet objectif de valorisation des données, l'apport de l'intelligence artificielle dans de nombreux domaines peut ainsi se situer à l'interface entre l'humain et la machine. C'est par exemple le cas dans le domaine médical [11] dans lequel s'inscrit le travail de cette thèse. Les données médicales sont produites par des sources toujours plus nombreuses et agrégées par des organismes privés comme publics [12]. Notre objectif est d'aider les professionnels de la santé à tirer parti du capital constitué par ces données en leur permettant de plus facilement les visualiser, les manipuler et les explorer. Actuellement et dans de nombreux cas, cela demande aux médecins d'avoir des connaissances en bases de données, de savoir formuler une requête dans un langage adapté et de pouvoir manipuler les outils statistiques et informatiques adéquats. Nous souhaitons proposer une structuration automatique de données médicales telles que les dossiers médicaux numériques des patients. Afin qu'elle soit générique et applicable à d'autres contextes, cette structuration ne doit pas requérir pas de connaissances *a priori*, cela facilite aussi la naissance de nouvelles idées et hypothèses cliniques avec une visualisation qui ne repose pas sur des présuppositions.

9. *Donner un sens à l'intelligence artificielle*, Cédric Villani, 2018
10. *Les Furtifs*, Alain Damasio (2019). La Volte
11. *Black Box, White Coat: Rise Of The Machines In Medicine*, Science Trends (2018)
12. *Health Data Hub*, Stéphanie Combes, 2018

L'outil que nous souhaitons proposer à ces professionnels de santé les assisterait en structurant par similarité et représentativité les patients présents dans leurs bases de données. Dans cette représentation, chaque patient serait lié à un autre patient similaire mais plus représentatif qui le subsumerait, avec l'idée qu'analyser les patients les plus représentatifs fournirait une vue condensée de la cohorte. Les relations entre individus, deux à deux, fourniraient aussi des informations décrivant la nature de leur similarité de sorte à pouvoir faciliter l'émergence et la vérification d'hypothèses. Cette approche basée sur des associations d'individus similaires et hierarchisés par représentativité trouve sa source dans le raisonnement utilisé par les experts médicaux [13] : l'analogie. Quand ils sont confrontés à un nouveau cas clinique, les médecins vont utiliser leur expérience professionnelle en plus de leurs connaissances métiers, en rapprochant le patient d'un ou plusieurs cas typiques rencontrés précédemment chez des patients similaires. Ce type de raisonnement permet à la fois des résultats efficaces et une prise de décision très rapide. Il se présente comme le mécanisme de décision utilisé dans de nombreux domaines où le temps est un facteur majeur dans la prise de décision.

Un système d'aide à la décision structurant sous forme de graphe les profils des patients permettrait ainsi de guider l'exploration des données en recommandant des cas typiques et similaires à un patient étudié. Dans ce système, les cas recommandés correspondraient ainsi aux cas analogques auxquels se réfèrent les professionnels lors de la prise de décision ; la démarche s'inspire donc des systèmes de recommandation même si le but final n'est pas la recommandation mais l'explicabilité à partir de cas. Il s'agit par exemple de suggérer des cas similaires mais assez typiques et représentatifs pour réduire la possibilité d'erreur de diagnostique par analogie et suggérer un éventail plus large de pathologies pouvant correspondre à un patient. Toutefois, pour qu'un tel outil de recommandation fonctionne efficacement dans le domaine clinique il est nécessaire qu'il soit transparent, explicable et interprétable. Les avatantages sont multiples car cela facilite l'adoption par le professionnel, l'acceptation par le patient de l'intelligence artificielle comme outil médical, la vérification de l'absence de biais ou encore la possiblité de remettre en cause des résultats en comprenant ce qui a motivé leur construction. Afin de fournir un outil d'exploration des données et d'aide à la décision qui soit adapté à ces spécificités, nous avons collaboré avec le Centre Universitaire Hospitalier de l'hôpital Robert Debré, à Reims. Plus particulièrement, ce sont les données et l'expertise fournies par le service Diabète-Endocrinologie-Nutrition qui ont été à la fois la source de nos réflexions scientifiques mais aussi la finalité de ce travail, en tant que domaine applicatif.

1.2 Objectifs scientifiques

Nous tentons dans ce manuscrit de contribuer au domaine de l'extraction automatique de connaissances, afin d'enrichir les données et d'augmenter la capacité de raisonnement à partir de celles-ci. Ce travail de thèse se concentre sur la structuration d'un ensemble d'éléments par similarité et représentativité. Cette structuration doit pouvoir permettre d'organiser, hiérarchiser et lier selon leur similarité des éléments décrits par de nombreuses variables. Cela de sorte à faciliter l'exploration et la visualisation de ces

13. *How Doctors Tink*, The New Yorker (2007)

éléments, afin qu'un expert puisse appréhender les informations contenues dans une base de données. Le but final est, par la création d'une méthode d'association appropriée, de proposer une approche intuitive pour appréhender l'information et vérifier des hypothèses sans effectuer de manipulation informatique complexe. Plus que sur la réalisation d'un outil médical, nous concentrons ce travail sur les problématiques soulevées par la création d'un tel outil utilisant des données multidimensionnelles : la haute-dimensionnalité et l'interprétabilité.

Haute-dimensionalité

Nous devons développer un indice de similarité et de représentativité efficace avec des objets définis en grande dimension. En effet, la haute dimensionnalité provoque des changements sur la distribution des éléments et les distances les séparant, ce qui nécessite des méthodes appropriées. Ce sujet d'étude revêt une importance par le fait que de très nombreux phénomènes réels ne peuvent être décrits précisément que par un très grand nombre de variables, ou descripteurs. Les méthodes de réduction de dimensionalité par sélection de variables ou par projection ne sont pas envisageables dans notre cas applicatif d'analyse exploratoire pour des raisons d'interprétabilité. Aussi, la résilience des associations face à la malédiction de la dimensionalité est un critère essentiel à l'adéquation de la méthode. Il est nécessaire de valider l'approche choisie sur différents types de données synthétiques et réelles, afin de s'assurer que la distribution des éléments ainsi que le bruit n'aient pas d'impact sur la qualité des associations.

Interprétabilité

Bien que les développements de l'intelligence artificielle soient présents dans tous types d'applications, ils sont peu souvent transparents et compréhensibles pour l'utilisateur final. Nous tentons donc de créer une approche aux résultats interprétables, qui puisse être utilisée dans des applications où la compréhension de l'algorithme est essentielle pour des raisons d'éthique, d'acceptation de la technologie ou encore de transparence de la procédure. Aucun prétraitement ne doit avoir lieu sur les éléments, de plus les variables responsables de chaque association doivent être identifiées de sorte qu'il soit possible d'extraire des explications depuis les résultats. En évitant les transformations, il est possible de proposer aux experts une solution fournissant des résultats qu'ils peuvent directement interpréter et replonger dans un contexte concret. Enfin, les généralisations sont à éviter afin de privilégier une approche basée sur les individus de manière à tirer parti de l'expérience et l'expertise de l'utilisateur en se rattachant à des cas réels.

1.3 Contributions

La contribution principale de ce manuscrit se présente sous la forme d'un algorithme de calcul d'associations entre éléments similaires, structurés selon leur représentativité. Cette similarité, appelée *Degré de Représentativité*, permet de trouver les éléments d'un jeu de données les plus adaptés pour représenter des sous-populations parmi les données. Ce calcul de la représentativité est adapté aux données en haute dimensionnalité, et propose des résultats qui répondent à plusieurs critères de l'interprétabilité, tels que la

transparence ou l'explicabilité. Nous utilisons les rangs des éléments les uns par rapport aux autres comme base afin de former des associations pertinentes entre éléments. De plus, l'approche proposée est entièrement basée sur les instances afin de permettre des résultats interprétables et concrets pour l'utilisateur. Cela nous permet, par cette structuration, de guider le processus exploratoire en favorisant le raisonnement par analogie. En plus de la proposition et de l'étude comparative de cet algorithme de calcul de représentativité, nous décrivons un prototype d'outil d'aide à l'anayse exploratoire des données médicales. Pour cela, nous nous appuyons sur une base de données médicales issue du suivi de patients diabétiques : le prototype construit favorise l'exploration de ces données médicales en les structurant autour de patients représentatifs. Enfin, nous proposons l'étude en haute dimensionnalité d'un algorithme de détection d'anomalies adapté aux *outliers* détectables seulement avec des projections spécifiques. Dans le contexte médical, cela correspond aux patients atypiques qui sont les plus difficiles à repérer. Nous évaluons cet algorithme sur plusieurs jeux de données réelles et le comparons à la détection d'anomalies fournie par le *Subspace Outlier Degree*.

1.3.1 Communications

Ces travaux ont pu être valorisés lors de plusieurs communications au sein de conférences dédiées à l'extraction de connaissances, à la visualisation et aux interfaces utilisateur :

Conférence internationale

— Falip, J., Aït-Younes, A., Blanchard, F., Delemer, B., Diallo, A., & Herbin, M. (2017). *Visual instance-based recommendation system for medical data mining.* Knowledge-Based and Intelligent Information & Engineering System, In Procedia computer science, 112.

Dans cette conférence internationale dédiée aux systèmes d'extraction de données et d'aide à la décision, nous avons présenté notre prototype visant à assister le diagnostique médical en proposant une structuration par représentativité des dossiers des patients.

Atelier de conférence internationale

— Falip, J., Blanchard, F., & Herbin, M. (2019, March). *Explainable structuring and discovery of relevant cases for exploration of high-dimensional data.* In Intelligent User Interfaces, ESIDA Workshop.

Nous avons présenté, dans une conférence dédiée aux interfaces utilisateurs et à l'explicabilité, la version finale à la fois de notre algorithme de structuration interprétable et du prototype d'aide à la décision destiné au personnel médical.

Ateliers nationaux

— Falip, J., & Blanchard, F. (2018). *Système de recommandation interactif pour l'analyse de données médicales.* In Rencontres R.

Nous avons détaillé, dans cet évènement dédié aux applications techniques liées au langage *R*, les briques ayant permis la construction du prototype d'outil de d'aide à la décision médicale.

— Falip, J., Blanchard, F., & Herbin, M. (2018) *Exploration et système de recommandation pour l'aide au raisonnement médical.* In Plate-forme Intelligence Artificielle.

À l'occasion de cette rencontre autour de l'informatique et de ses applications médicales, nous avons proposé des évolutions majeures par rapport à notre première version du calcul de représentativité et de son utilisation pour structurer des données.

— Falip, J., Blanchard, F., & Herbin, M. (2018). *Outlier detection in high-dimensional spaces using one-dimensional neighborhoods.* In Extraction et Gestion des Connaissances, FDC Workshop.

Durant cet atelier traitant de la fouille de données nous avons partagé nos premières réflexions autour d'une méthode de détection d'*outliers* basée instance et adaptée à la haute dimensionnalité, utilisant des mécanismes inspirés de notre algorithme de structuration.

— Falip, J., Aït-Younes, A., Blanchard, F., & Herbin, M. (2017). *Représentativité, généricité et singularité : augmentation de données pour l'exploration de dossiers médicaux.* In Extraction et Gestion des Connaissances, VIF Workshop.

Dans cet atelier dédié à la visulisation de données nous avons proposé une première réinvention du *Degré de Représentativité* et de ses applications à la fouille de données.

1.4 Plan du manuscrit

La suite de ce manuscrit est composée de cinq chapitres. Dans le chapitre deux, nous présentons les enjeux de l'analyse exploratoire avant de dresser un état de l'art autour des problématiques de haute-dimensionalité, d'interprétabilité et enfin de cognition et décision. Ces éléments ont trait à l'objectif de la méthode proposée, pour l'analyse exploratoire, et aux problèmes rencontrés par cette démarche, pour les points suivants. Le troisième chapitre détaille la méthode de structuration proposée, dont le calcul du *Degré de Représentativité* ainsi que son utilisation pour structurer des données. Une méthode de détection d'anomalie y est aussi proposée. Le quatrième chapitre compare les résultats de ces approches sur plusieurs jeux de données réelles et simulées. D'abord la structure fournie par le *Degré de Représentativité* ainsi que la pertinence des associations créées entre les patients sont étudiées. Ensuite, les résultats de la détection d'anomalie sont comparés à une autre méthode concurrente. Le chapitre cinq expose les travaux réalisés pour prototyper ce travail dans un outil utilisé sur des données médicales afin d'analyser une base de données de patients diabétiques et de pouvoir la structurer en associant les patients similaires et en déterminant les patients les plus représentatifs. Le sixième et dernier chapitre propose une conclusion résumant les contributions précédemment présentées puis propose des perspectives ouvertes par ce travail.

Chapitre 2

État de l'art

Sommaire

2.1 Analyse exploratoire 13
2.1.1 Principes de l'EDA 14
2.1.2 Fouille de données 15
2.2 Haute dimensionnalité 17
2.2.1 Espaces creux 18
2.2.2 Concentration des distances 19
2.2.3 Hubness 23
2.2.4 Réduction de la dimensionnalité 26
2.3 Interprétabilité 28
2.3.1 Objectifs 28
2.3.2 Approches 31
2.4 Processus décisionnel 33
2.4.1 Rule-based 34
2.4.2 Exemplar-based 35

Ce chapitre dresse un panorama des domaines et problématiques scientifiques abordés dans ce manuscrit de thèse. Tout d'abord nous décrivons le contexte d'analyse exploratoire dans lequel s'ancre ce travail. Nous y évoquons entre autre le but de l'analyse exploratoire et comment elle peut être mise en oeuvre. Dans la deuxième section nous exposons les problèmes inhérents aux espaces en grande dimension. Dans le cadre applicatif médical dans lequel nous nous situons, il est fréquent que des jeux de données soient décrit par plusieurs centaines ou milliers de variables : il s'agit donc de données en haute dimensionnalité. La haute dimensionnalité s'accompagne de plusieurs phénomènes qui lui sont liés, tels que la *sparsity* affectant la densité de l'espace étudié, la concentration des distances entre les éléments, ou la *hubness* qui se manifeste par l'apparition d'éléments faisant figure de « voisins populaires » au sein des données. Nous mentionnerons aussi les méthodes de réduction de dimmensionalité, bien qu'elles ne soient pas utilisées dans ce travail pour des raisons d'interprétabilité. Concernant l'interprétabilité justement, la troisième section détaille les différents critères d'interprétabilité ainsi que les objectifs nous poussant à privilégier une solution interprétable. La quatrième section, enfin, est consacrée à l'*exemplar theory* et aux processus décisionnels humains qui entrent en jeu lorsqu'un expert doit prendre une décision et catégoriser un stimuli. Ce point revêt une importance particulière car il est nécessaire de comprendre le raisonnement des professionnels de santé pour espérer proposer un outil adapté à leurs usages. En résumé, ce chapitre sera l'occasion d'expliquer le choix des méthodes proposées ainsi que du vocabulaire utilisé dans la suite de ce travail.

2.1 Analyse exploratoire

L'analyse exploratoire de données (ou *EDA* pour *Exploratory Data Analysis*) consiste à étudier un jeu de données, sans *a priori*, afin d'en extraire des hypothèses. Il ne s'agit pas d'un ensemble précis de techniques, mais plutôt d'une démarche spécifique dans la manière de mener l'analyse.

Proposée par Tukey (1977) comme une approche exploratoire à l'analyse de données, il est à noter qu'elle ne reposait initialement pas sur l'utilisation de l'informatique, étant donné la faible disponibilité de l'outil à cette époque. L'évolution de l'*EDA* est fortement liée à celle de la puissance de calcul des machines et à la popularisation de la visualisation de données (Tufte, 1983). La motivation principale mise en avant par Tukey dans ses travaux est de contrer le biais de confirmation en essayant de mettre en avant un panel d'hypothèses uniquement basées sur les données, sans certitude ou supposition préalable. Ainsi, cette approche très ouverte n'est pas à considérer comme autonome, mais plutôt comme une étape d'un raisonnement basé sur les faits : on la retrouve dans de nombreux domaines scientifiques où elle est fondamentale dans l'analyse de phénomènes naturels par exemple. L'*EDA* trouve sa place au début du processus scientifique où elle permet de découvrir par induction des hypothèses fiables, ancrées dans le réel et formées sur des phénomènes qui n'étaient pas forcément attendus *a priori* ; le processus scientifique continuant ensuite avec l'analyse confirmatoire, *a posteriori*, qui utilise le raisonnement déductif pour valider l'hypothèse. Il s'agit donc d'une synergie entre ces deux modes d'analyse qui sont complémentaires.

Dans cette section, nous mettons d'abord en avant les principes et spécificités de l'analyse exploratoire en tant qu'outil. La sous-section suivante se concentre sur la fouille de données, en définissant sa relation à l'*EDA* et en donnant un aperçu de l'éventail des méthodes disponibles.

2.1.1 Principes de l'EDA

Il existe de multiples approches de l'analyse exploratoire, mais des travaux comme ceux de Behrens (1997) fournissent une vue d'ensemble du domaine. Behrens apparente l'*EDA* à « l'écoute des données de toutes les manières possibles jusqu'à faire apparaître une "histoire" probable », il s'agit donc de combattre le biais de confirmation par une analyse pragmatique de l'information disponible (Einhorn & Hograth, 1978) pour découvrir des analogies susceptibles de guider le raisonnement. L'*EDA* vise ainsi à s'approprier les données tout en les enrichissant grâce à des analyses statistiques ou graphiques par exemple. Ces données enrichies peuvent entre autres être plus propices à la découverte de liens entre le modèle qui les a générées et les motifs que l'on observe, favorisant l'émergence d'hypothèses nouvelles. Attention, comme le précisaient Jebb, Parrigon, & Woo (2017), le but n'est toutefois pas « d'aller à la "pêche" aux hypothèses, ou de "torturer" les données jusqu'à pouvoir en déduire une hypothèse ». C'est une étape préliminaire dont les résultats devront être soumis ensuite à une analyse confirmatoire (ou *CDA* pour *Confirmatory Data Analysis*).

L'*EDA* se base donc sur la logique par abduction, qui consiste à trouver des causes probables à des effets observés. Ce type de raisonnement issu du pragmatisme a été mis en avant par Charles Sanders Peirce (Tiercelin, 2013) et parfois comparé au travail du détective qui va tenter de découvrir des indices lui permettant ensuite, par abduction, d'élucider l'enquête (Josephson & Josephson, 1996). Ce même raisonnement occupe une partie du processus scientifique, amenant à la formulation d'hypothèses ensuite vérifiées par induction. C'est aussi l'abduction qui est au cœur du diagnostic médical, où il est utilisé par exemple pour émettre un diagnostic à partir de symptômes (Mirza, Danesh, Noesgaard, Martin, & Staples, 2014).

Dans le domaine médical Komorowski, Marshall, Salciccioli, & Crutain (2016) définissent les objectifs de l'*EDA* de la manière suivante :

— Visualiser de potentielles relations
— Détecter les *outliers* et anomalies
— Inférer ou sélectionner des modèles adaptés
— Maximiser la compréhension des structures de données
— Extraire ou créer des variables cliniquement pertinentes

L'*EDA* est une méthodologie et non un ensemble défini d'outils, mais il existe un vaste choix d'approches permettant d'atteindre les objectifs mis en avant ici. Cette méthodologie s'incarne en la fouille de données, qui a pour but d'extraire de l'information à partir des données étudiées et propose des outils adaptés pour cela.

2.1.2 Fouille de données

Avec son approche *data-driven*, la fouille de données (ou *data mining*) peut être vue comme un vaste ensemble d'outils permettant d'extraire automatiquement des informations et hypothèses à partir des données (Liao, Chu, & Hsiao, 2012), la mettant ainsi en lien direct avec l'analyse exploratoire. Nous retrouvons dans la fouille de données de nombreux problèmes tels que l'extraction de règles, le partitionnement (Jain, Murty, & Flynn, 1999), la classification (Kotsiantis, 2007) ou la détection d'anomalies (Aggarwal, 2012). Cela est appliqué à de nombreux types de données comme l'image, le texte, l'audio ou les données géospatiales entre autres. Komorowski et al. (2016) proposent deux classifications pour les méthodes d'analyse exploratoire : elles peuvent être univariées ou multivariées, graphiques ou non. Dans les exemples qui suivent, les données illustrant ces méthodes d'analyse exploratoire sont les données *iris*, un jeu de données botaniques compilé par Anderson (1936) et popularisé la même année par Fisher (1936). Il comporte la longueur et la largeur (en centimètres) des pétales et sépales de 150 iris ainsi que l'espèce à laquelle appartient chaque individu recensé (*setosa*, *virginica*, *versicolor*). Même si elles ne contiennent que 4 dimensions quantitatives, ces données sont devenues au fil des années un cas d'école dans le domaine de l'apprentissage automatique.

— Les approches univariées consistent à étudier la distribution des individus selon une de leurs variables, de manière visuelle (figure 2.1) ou non-graphique (table 2.1)
— Les méthodes de fouille de données multivariées et graphiques, comme illustré par la figure 2.2 où des données en quatre dimensions sont présentées selon leurs deux composantes principales

Nous nous plaçons, dans ces travaux, dans le cadre de l'analyse graphique de données multivariées. Toutefois, les données issues du monde réelles sont très souvent décrites par plusieurs dizaines ou centaines de dimensions, ce qui pose des difficultés spécifiques aux méthodes de fouille de données et d'analyse exploratoire.

TABLE 2.1 – Ensemble d'analyses univariées sur chacune des dimensions du jeu de données iris. La répartition des classes est explicitée, et pour les attributs quantitatifs le minnimum, maximum, premier et troisième quartiles ainsi que la médianne et la moyenne sont indiquées.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
Min. :4.300	Min. :2.000	Min. :1.000	Min. :0.100	setosa :50
1st Qu. :5.100	1st Qu. :2.800	1st Qu. :1.600	1st Qu. :0.300	versicolor :50
Median :5.800	Median :3.000	Median :4.350	Median :1.300	virginica :50
Mean :5.843	Mean :3.057	Mean :3.758	Mean :1.199	NA
3rd Qu. :6.400	3rd Qu. :3.300	3rd Qu. :5.100	3rd Qu. :1.800	NA
Max. :7.900	Max. :4.400	Max. :6.900	Max. :2.500	NA

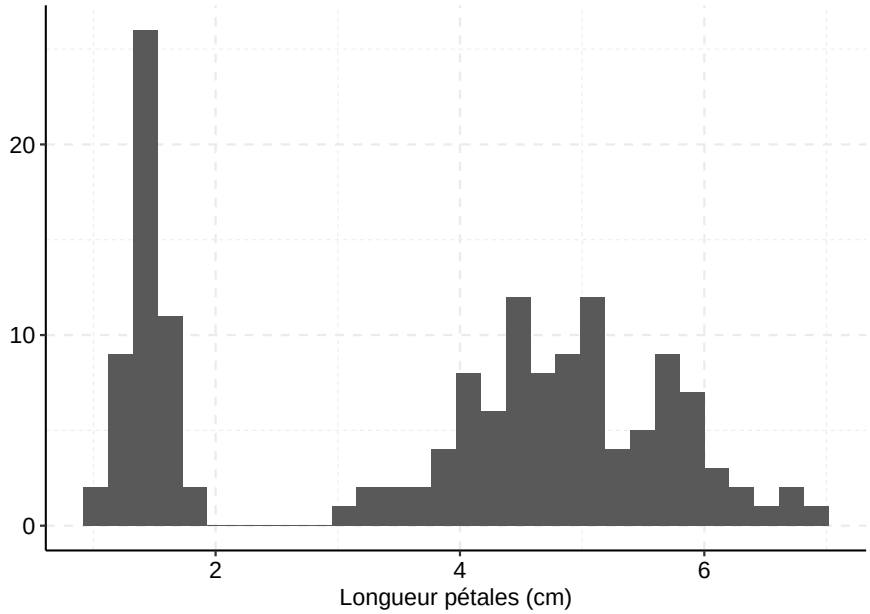


FIGURE 2.1 – Histogramme de la distribution de la longueur des pétales, en centimètres, sur le jeu de données iris. Cette analyse graphique univariée permet de discerner deux groupes.

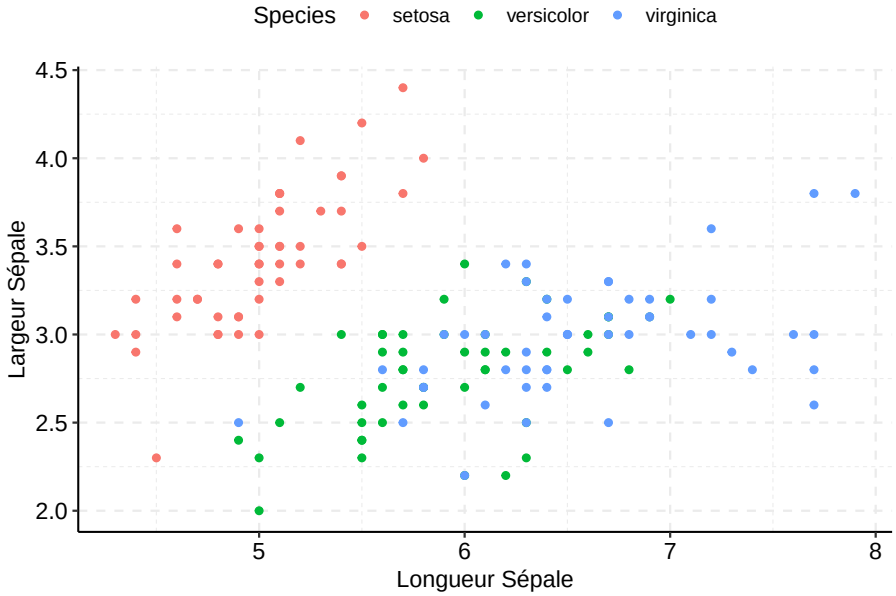


FIGURE 2.2 – Diagramme de dispersion illustrant une analyse graphique multivariée. Deux composantes du jeu de données iris sont visualisées, montrant la répartition des trois classes présentes au sein des données.

2.2 Haute dimensionnalité

Caractériser la ressemblance entre deux objets « complexes » est une préoccupation importante de l'intelligence artificielle. La notion de similarité est centrale dans de nombreux domaines du *machine learning* où on l'utilise entre autres sur des échantillons vocaux, documents textuels ou données multidimensionnelles. Les indices de similarité (ou de dissimilarité, comme la distance) choisis varient selon le domaine étudié, et le nombre d'indices disponibles ne cesse de croître car chaque application nécessite une méthode adaptée aux données qu'elle traite. Toutefois, des problèmes spécifiques sont rencontrés lorsqu'on tente de quantifier la similarité entre des éléments caractérisés par un grand nombre de variables (Hinneburg, Aggarwal, & Keim, 2000). Pour caractériser ces problèmes, nous utiliserons indifféremment les mots variables, attributs, *features* ou dimensions pour parler des descripteurs d'un objet multidimensionnel.

Ces données en grande dimension, bien qu'extrêmement courantes pour modéliser des problèmes issus du monde réel, posent de nombreuses difficultés. Tout d'abord, celui de la compléxité algorithmique qui leur est associée de par l'espace occupé par ces données. À cause d'une explosion combinatoire, même les algorithmes les plus efficaces deviennent coûteux en espace et en temps lorsqu'on les exécute sur des éléments définis par de nombreuses *features* (Indyk, 2004). Mais d'autres problèmes se présentent aussi, regroupés sous le terme de « malédiction de la dimensionnalité » (Bellman, 1961) : un phénomène affectant les métriques utilisées habituellement pour déterminer la similarité entre éléments. Cette « malédiction » rend peu performants des algorithmes qui, avec un faible nombre de dimensions, donnent pourtant des résultats pertinents. La répartition des éléments, leur densité ou la distribution des distances sont autant de variables qui évoluent de manière contre-intuitive quand la dimensionnalité croît (Donoho, 2000), comme nous allons l'illustrer dans cette section. Ces problèmes sont étudiés depuis de nombreuses années, la croissance de la taille des bases de données ayant fait de la « malédiction de la dimensionnalité » un obstacle pour de nombreuses méthodes d'analyse (Weber, Schek, & Blott, 1998) mais aussi un sujet de recherche en évolution permanente (Feldbauer & Flexer, 2018 ; Liu, Li, Li, & Zhang, 2017 ; Tomasev & Mladenic, 2011).

Les trois sous-sections suivantes détaillent et illustrent certains aspects de cette « malédiction » associée aux espaces en haute dimension : les espaces creux, la concentration des distances et la *hubness*. Une quatrième section détaille les méthodes habituelles pour traiter des données en haute dimension. Dans la suite, nous allons illustrer ces notions à l'aide de données simulées. Nous appellerons Ω l'ensemble composé de N éléments, ou individus, dans un espace multidimensionnel. Ces N éléments sont décrits par D caractéristiques, variables ou *features*, et correspondent donc à N points dans un espace à D dimensions. Ces objets multidimensionnels sont donc décrits par des variables qui induisent les dimensions d'un espace paramétrique dans lequel chaque objet est un point.

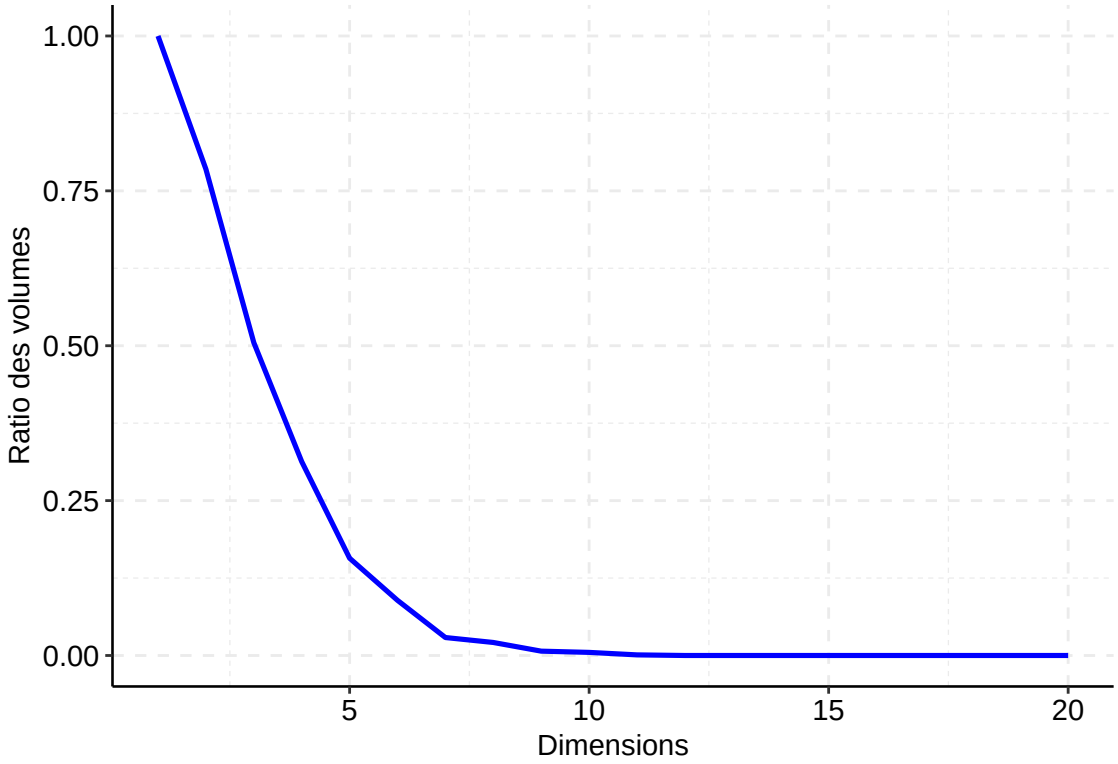


FIGURE 2.3 – Illustration de l'effet d'espace "creux". 1000 points sont générés aléatoirement selon une loi uniforme au sein d'un hypercube. La courbe affiche le ratio du nombre de points au sein d'une hypersphère inscrite dans cet hypercube, par rapport au nombre total de points dans l'hypercube, selon le nombre de dimensions.

2.2.1 Espaces creux

Le problème d'espace « creux », aussi appelé *sparsity*, est caractérisé par une raréfaction des données en tout point de l'espace à mesure que la dimensionnalité augmente. Lorsque le nombre D de dimensions augmente, la densité de données en chaque point de l'espace diminue, au point où dans certains contextes la *sparsity* devient un problème crucial dès la dizaine de dimensions (Charu C Aggarwal et al., 2001).

Nous pouvons illustrer ce problème de *sparsity* par une simulation, en considérant en distance Euclidienne une hypersphère de rayon r inscrite dans un hypercube d'arête $2r$. En générant aléatoirement un nombre fixé de points à l'intérieur de cet espace, nous pouvons ensuite étudier l'évolution du ratio du nombre de points compris à l'intérieur de l'hypersphère par rapport au nombre total de points générés dans l'hypercube circonscrit à la sphère, et cela en fonction du nombre de dimensions décrivant l'espace. Avec $r = 1$ en distance Euclidienne, tout point dont la distance à l'origine est supérieure à 1 sera en dehors de l'hypersphère. À partir d'un exemple composé de $n = 1000$ points générés aléatoirement selon une loi uniforme, nous constatons dans la figure 2.3 que le ratio du volume de l'hypersphère par rapport au volume de l'hypercube tend vers zéro. Cette observation se vérifie si l'on considère les expressions littérales des volumes, en dimension d, pour une hypersphère de rayon r et un hypercube d'arête $2r$.

Soit Γ la fonction gamma, le volume de l'hypersphère s'exprime de la façon suivante :

$$V^d_{Sphere}(r) = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)} r^d$$

et celui de l'hypercube par :

$$V^d_{Cube}(r) = (2r)^d$$

Avec V le volume des éléments, nous confirmons les observations issues de la simulation :

$$\lim_{d\to\infty} \frac{V^d_{Sphere}(r)}{V^d_{Cube}(r)} = 0$$

Une hypothèse courante concernant la densité est celle de la distribution normale des données. Il est en effet fréquent que l'observation d'un phénomène réel se traduise par un ensemble d'objets décrits par de multiples variables suivant chacune une loi normale. Dans des cas comme ceux-ci et en faible dimension, la densité est prévisible et permet notamment de retrouver les éléments en queue de distribution. Si l'on considère un ensemble d'éléments définis sur une dimension selon une loi normale centrée réduite, 95% des éléments se situeront par exemple à une distance inférieure à 1.96 du centre de la distribution. En créant un exemple avec $n = 1000$ éléments décrits en générant aléatoirement leurs d variables selon une loi normale centrée réduite, nous pouvons observer le pourcentage de points hors de l'hypersphère de rayon 1.96. La figure 2.4 nous indique l'évolution de ce ratio selon le nombre de dimensions d. Dès $d = 5$ la majorité des éléments est dans ce qui serait considéré comme la queue de distribution, en dehors de l'hypersphère. En doublant le nombre de dimensions le ratio avoisine un, indiquant que la quasi totalité des éléments est désormais à une distance supérieure à 1.96 de l'origine, l'espace devenant creux.

Ces illustrations soulignent à quel point une dimensionnalité même modérée crée des espaces extrêmement creux : pour éviter cette tendance il serait nécessaire d'agrandir exponentiellement la taille du jeu de données étudié pour chaque nouvelle dimension ajoutée. Ce problème de *sparsity* entraîne des difficultés pour les nombreuses méthodes basées sur des estimations de densité. La densité est par exemple au coeur d'un sous-ensemble d'algorithmes de détection d'*outliers* (Ester, Kriegel, Sander, & Xu, 1996) dont l'efficacité décroît lorsqu'ils sont appliqués à des espaces en grande dimension. Des adaptations ont été formulées par la suite pour réduire l'effet de la dimensionnalité sur ces approches (Kriegel, Kroger, Schubert, & Zimek, 2009).

2.2.2 Concentration des distances

En plus de l'émergence d'espaces creux, un autre phénomène impacte négativement les méthodes d'analyse basées sur la proximité selon une distance ou pseudo-distance, donc toute approche basée sur la géométrie Euclidienne. En haute dimensionalité on observe

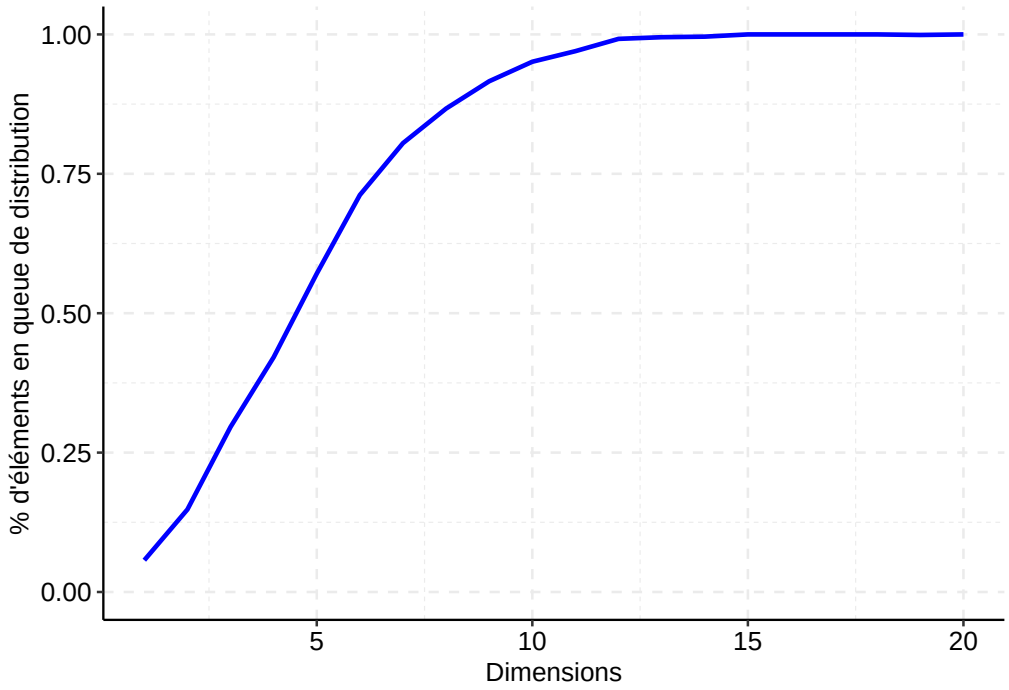


FIGURE 2.4 – Évolution du nombre d'éléments en queue de distribution, selon le nombre de dimensions. Les 1000 éléments sont générés selon une loi normale centrée réduite, et l'on étudie le pourcentage de point situés hors de l'hypersphère de rayon 1.96.

une concentration des distances classiques : les éléments tendent à devenir équidistants les uns des autres (Hinneburg et al., 2000). La distance devient donc une mesure de dissimilarité beaucoup moins discriminante (Jayaram & Klawonn, 2012) et de très nombreuses approches basées sur celles-ci sont inappropriées aux grandes dimensions.

Cette propriété peut elle aussi être illustrée par la simulation. Soit l'ensemble d'éléments Ω, définissons D_{max} la distance maximale entre un point et le centre de l'espace, et D_{min} la plus petite distance d'un point au centre. Nous pouvons étudier le contraste relatif défini par Beyer, Goldstein, Ramakrishnan, & Shaft (1999) comme :

$$C = \frac{D_{max} - D_{min}}{D_{min}}$$

La figure 2.5 étudie l'évolution du contraste relatif sur $n = 100$ données générées selon une loi uniforme tout d'abord, puis selon une loi normale. Nous considérons ici à but d'illustration que les variables sont indépendantes, hypothèse rarement exacte avec des données réelles. Quatre mesures de distance sont comparées pour le calcul du contraste, leur choix est expliqué dans le paragraphe suivant. Sur cette figure, nous pouvons voir qu'à partir de quelques dizaines de dimensions, le contraste relatif devient inférieur à un, et va tendre vers une valeur proche de zéro à mesure que la dimensionnalité augmente. Ce comportement du contraste relatif est révélateur de la concentration des distances. En comparant les deux graphiques de la figure, il est à noter que la distribution des données a un faible impact sur le contraste relatif, et que la distribution normale est tout autant sujette à ce problème.

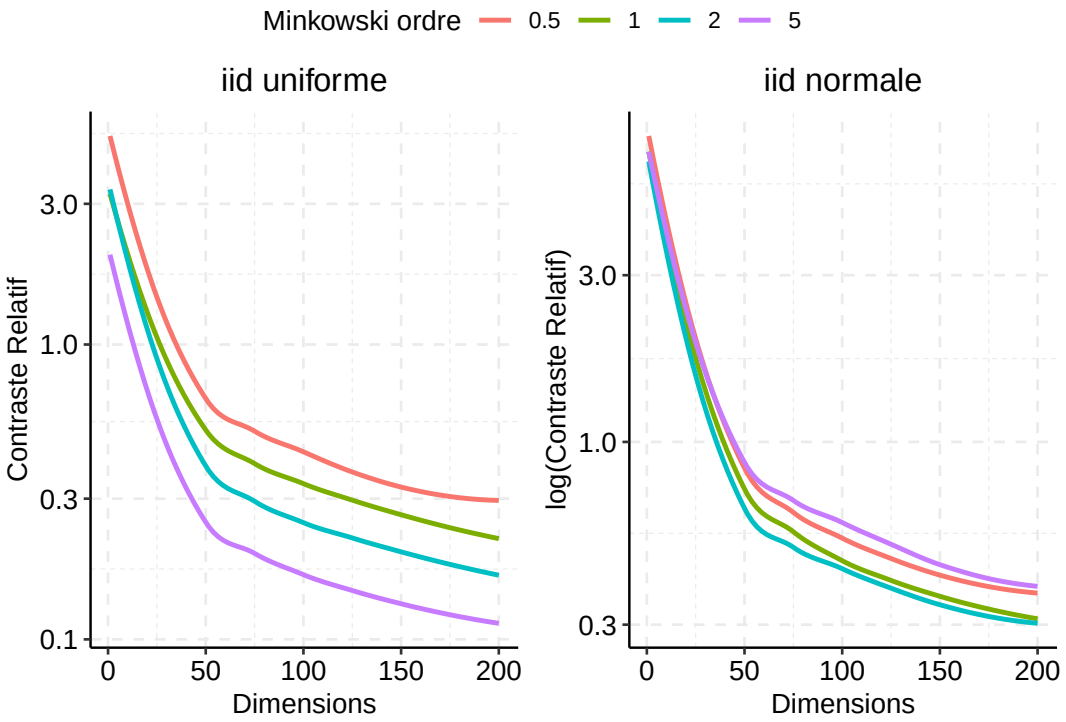


FIGURE 2.5 – Évolution du contraste relatif pour des données générées aléatoirement selon une distribution uniforme ou normale, selon le nombre de dimensions. Les distances de Minkowksi d'ordre 0.5, 1, 2 et 5 sont comparées.

Toutefois et comme illustré par la figure 2.5, la concentration des distances n'affecte pas de manière identique toutes les métriques. La distance Euclidienne, majoritairement utilisée, est parmi les plus impactées par l'augmentation du nombre de dimensions. On peut observer le fait que la distance de Manhattan lui est préférable dès que l'espace étudié comporte plusieurs dizaines de dimensions. Ces deux distances peuvent être généralisées grâce à la distance de Minkowski dont la formule est détaillée dans le tableau 2.2, où $x = (x_1, x_2, \ldots, x_d)$ et $y = (y_1, y_2, \ldots, y_d)$ sont deux points quelconques de Ω. Avec D_p la distance de Minkowski d'ordre p, les distances de Manhattan et Euclidienne peuvent être exprimées respectivement par D_1 et D_2. Les distances choisies pour la figure 2.5 sont donc $D_{0,5}$, D_1, D_2 [1] et D_5. Dès la cinquantaine de dimensions, nous pouvons constater que les distances de Minkowski d'ordre faible sont à privilégier si l'on souhaite discriminer au mieux les éléments, les meilleurs résultats étant obtenus avec des distances d'ordre $p < 1$, dites « fractionnelles » (Aggarwal et al., 2001). Mais l'utilisation de ce type de distance ne mitige que partiellement la concentration des distances, et avec des jeux de données qui atteignent couramment plusieurs centaines voire milliers de dimensions, cette option ne s'impose pas comme une solution à la « malédiction de la dimensionnalité ».

1. pour des raisons de lisibilité des graphes nous avons exclu la norme infini L_∞

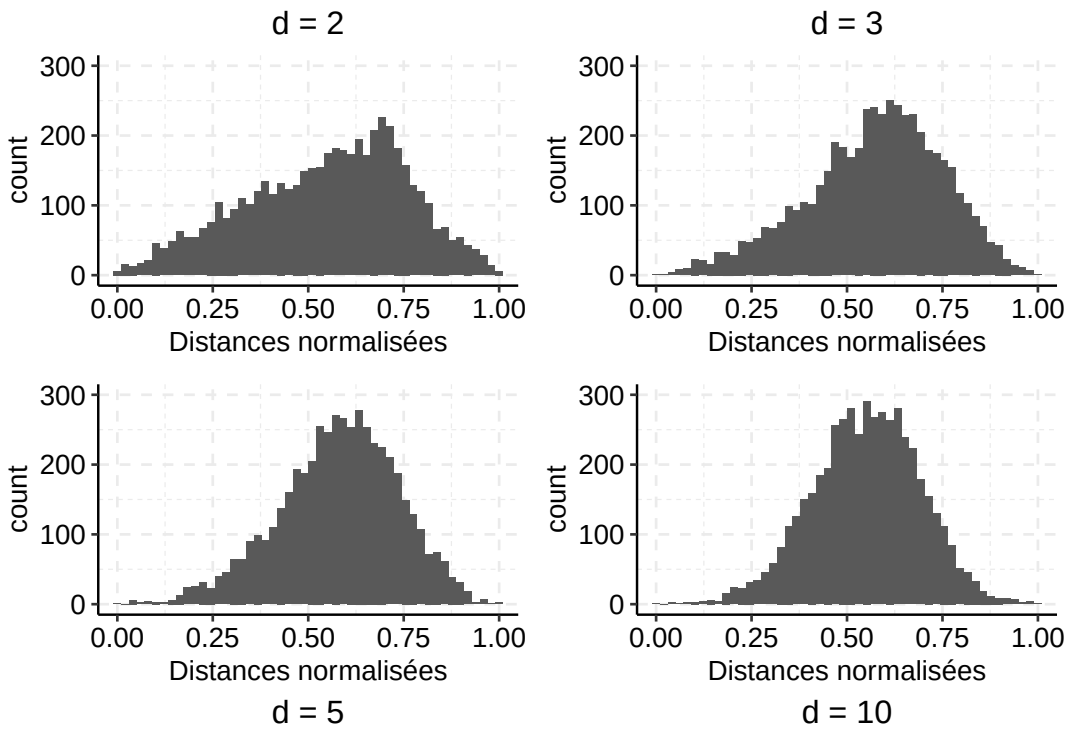


FIGURE 2.6 – Distribution des distances euclidiennes normalisées selon le nombre de dimensions. Les 5000 éléments sont générés aléatoirement selon une loi uniforme résultant en une concentration des distances quand la dimensionnalité augmente.

TABLE 2.2 – Formules des distances Minkowski, Manhattan et Euclidienne.

Nom de la métrique	Formule de distance	Ordre correspondant
Minkowski	$(\sum_{i=1}^{d} \lvert x_i - y_i \rvert^p)^{\frac{1}{p}}$	-
Manhattan	$\sum_{i=1}^{d} \lvert x_i - y_i \rvert$	Minkowski ordre 1
Euclidienne	$\sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$	Minkowski ordre 2

En plus de la mesure de contraste relatif, nous pouvons étudier la distribution des distances selon le nombre de dimensions. La figure 2.6 illustre la distribution des distances Euclidiennes au sein d'un jeu de données comportant $n = 5000$ éléments tirés de manière aléatoire selon une loi uniforme, avec d égal à 2, 3, 5 et 10 dimensions. Pour faciliter la comparaison, les distances y sont normalisées en 0 et 1. On peut observer une très nette concentration des distances dès la troisième dimension, et le phénomène s'accentue lorsque la dimensionnalité croît. Ces deux figures 2.5 et 2.6 illustrent, pour différentes métriques et dimensionalités, un même comportement de la distance qui devient une mesure de similarité moins efficace pour discerner les éléments à mesure que l'écart-type diminue.

Après avoir constaté l'émergence de ce phénomène en haute dimensionnalité, revenons à l'impact sur les algorithmes d'apprentissage automatique et d'extraction de données. Nombre d'approches répandues reposent sur la distance dans l'espace entier pour calculer la similarité entre éléments, mais lorsque la distance devient peu discriminante il a été démontré qu'un très large éventail de méthodes en est compromis (Weber et al., 1998). L'algorithme de *clustering* des *K-means* ainsi que toutes ses variantes, fréquemment utilisées de par leur efficacité en basse dimension, reposent sur le postulat que la distance dans l'espace multi-dimensionnel est une mesure de similarité adéquate. S'il ne s'agit pas de remettre totalement en cause ces algorithmes éprouvés, on peut cependant constater que le choix de la distance est critique vis-à-vis du sens à donner aux résultats (Jayaram & Klawonn, 2012). C'est aussi le cas pour les algorithmes basés sur des mesures comme les rangs : la *k-NN classification* ou le *s-NN clustering* en sont des exemples. Ces mesures alternatives dérivées de la distance ont vu le jour très tôt (Fix & Hodges, 1951 ; Jarvis & Patrick, 1973) et sont aussi appelées mesures secondaires, en cela qu'elles se basent sur les mesures primaires de distance dans l'espace entier (Houle, Kriegel, Kroger, Schubert, & Zimek, 2010) qu'elles tentent d'améliorer. Si les mesures primaires et secondaires sont toutes deux affectées par la dimensionnalité, les secondaires forment une meilleure alternative et sont plus adaptées lorsque le nombre de dimensions augmente. Le rang par exemple, qui est une mesure secondaire, est plus stable et reste plus discriminant quand la densité et les distances évoluent (Conover & Iman, 2012). L'utilisation de mesures secondaires fournit donc une alternative pouvant mitiger *a posteriori* l'effet de la dimensionnalité (Liu et al., 2017).

2.2.3 Hubness

Si l'on utilise un algorithme *instance-based* afin d'implémenter une approche basée sur des exemples plutôt que des prototypes, un troisième phénomène émerge en haute dimension. En plus de la concentration des distances et de l'apparition d'espaces creux, la *hubness* apparaît autour de certains éléments du jeu de données. Le concept de hubness décrit le fait que, lors de l'augmentation du nombre de dimensions, certains éléments vont se réveler être des *hubs* : des individus apparaissant plus fréquemment que les autres dans les plus proches voisins d'un grand nombre d'éléments. Les *hubs*, résultant eux aussi de la « malédiction de la dimensionnalité », peuvent être vus comme des « voisins populaires » au sein d'un ensemble de données. Cela accentue l'asymétrie entre les relations de voisinage (Feldbauer & Flexer, 2018).

Ainsi, il est possible d'observer ce phénomène par la simulation. Prenons un ensemble Ω composé de n éléments $x_1, \ldots, x_n$ dans un espace multidimensionnel. Avec D_p la distance de Minkowski d'ordre p, nous pouvons définir la fonction $V_{i,k}$ avec $i,k \in \{1,\ldots,n\}$ telle que :

$$V_{i,k}(y) = \begin{cases} 1, & \text{si } y \text{ figure parmi les } k \text{ plus proches voisins de } x_i \text{ selon } D_p \\ 0, & \text{autrement} \end{cases}$$

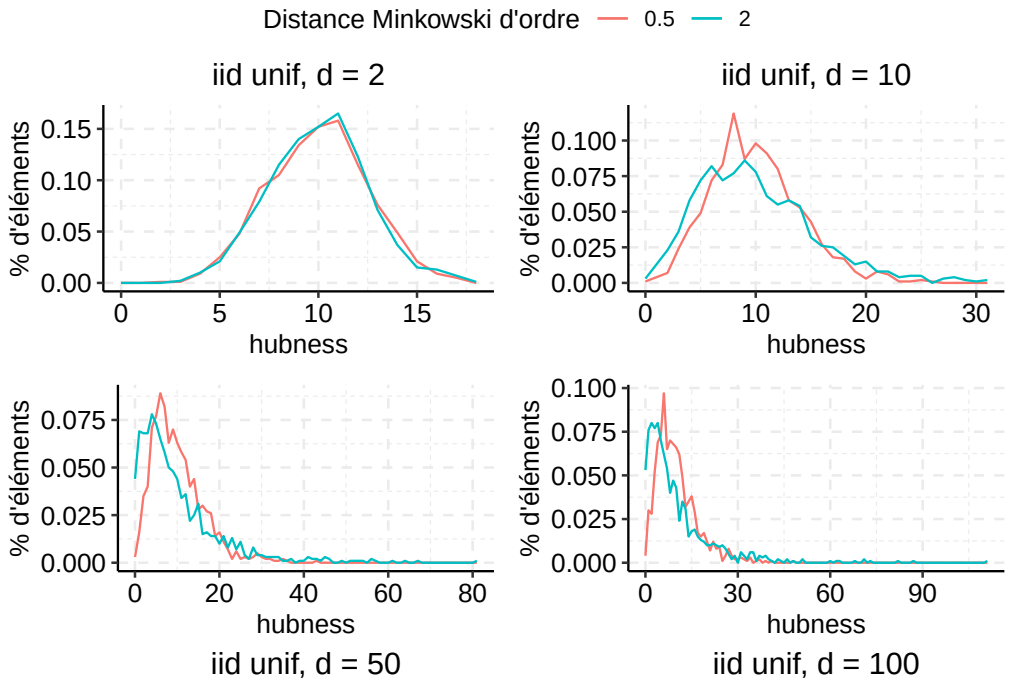


FIGURE 2.7 – Distribution de la hubness sur des données composées de 1000 éléments générés aléatoirement et de manière indépendante selon une loi uniforme, sur 2, 10, 50 et 100 dimensions. L'asymétrie s'accentue lors de l'augmentation de la dimensionnalité.

Nous pouvons désormais définir la *hubness* d'un élément y comme :

$$H_k(y) = \sum_{i=1}^{n} V_{i,k}(y)$$

Les figures 2.7 et 2.8 indiquent la distribution de la *hubness* avec un voisinage $k = 10$ et les distances de Minkowski d'ordre 0.5 (distance fractionnelle) et 2 (distance Euclidienne). Les jeux de données utilisés dans ces figures sont composés de 1000 éléments générés aléatoirement et de manière indépendante selon une distribution uniforme d'abord (figure 2.7), puis normale (figure 2.8). Le nombre de dimensions d varie entre 2 et 100. Notons tout d'abord que la *hubness* est peu dépendante de la mesure de distance choisie. Nous pouvons observer que la distribution de la *hubness* est symétrique pour des données en dimensions 2, mais qu'elle devient fortement asymétrique quand la dimensionnalité augmente. Le phénomène de *hubness* crée, comme illustré, une faible proportion d'éléments figurant dans le voisinage de la plupart des éléments du jeu de données : ce sont ces individus en queue de distribution qui sont considérés comme des *hubs*. Certains de ces éléments figurent dans le voisinage de presque un tiers des éléments dans le dernier cas présenté. De plus, la distance employée n'affecte pas la création de ces *hubs*.

La figure 2.9 étudie la *hubness* avec les mêmes propriétés de voisinage et de distance, mais sur quatre jeux de données issus du package *datamicroarray*. Les quatres jeux de données ont été sélectionnés de manière à illustrer une dimensionnalité de plusieurs centaines jusqu'à plusieurs milliers de dimensions, et sont tous basés sur l'observation

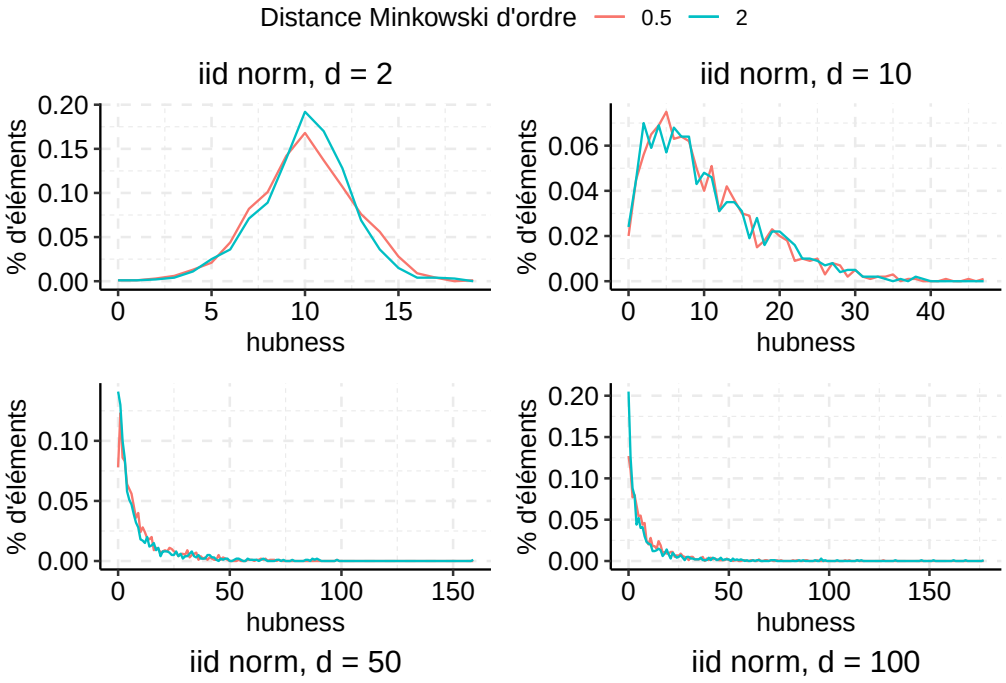


FIGURE 2.8 – Distribution de la hubness sur 1000 éléments générés aléatoirement et de manière indépendante selon une loi normale, sur 2, 10, 50, et 100 dimensions. L'asymétrie apparaît très marquée dès cinquantes dimensions.

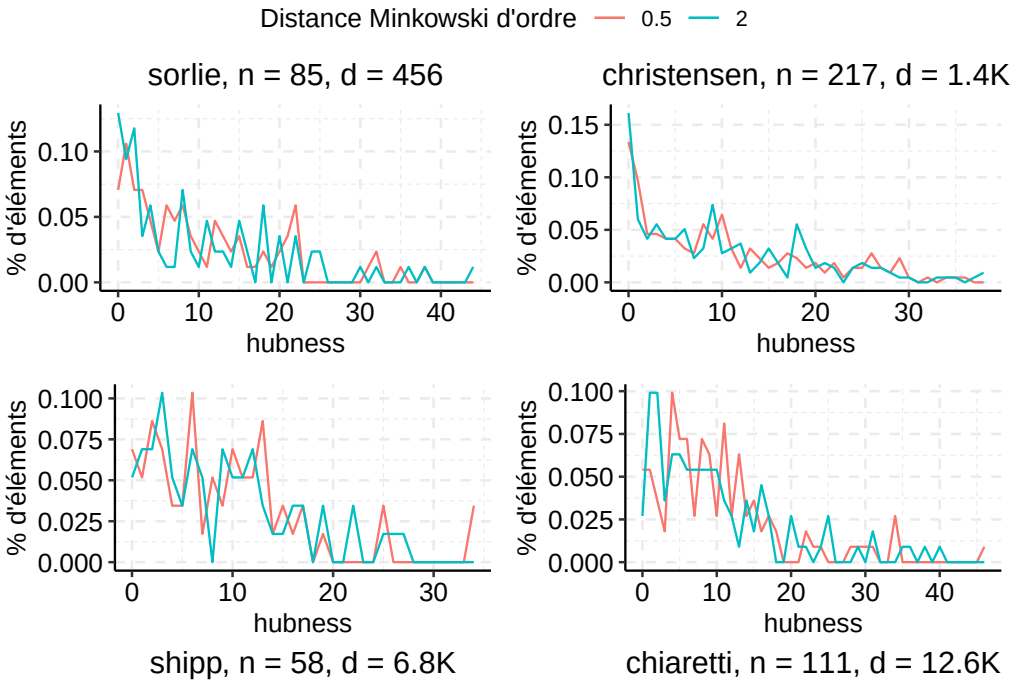


FIGURE 2.9 – Distribution de la hubness sur quatres jeux de données réelles issus du package datamicroarray et étudiant quatre pathologies différentes. Le nombre de dimensions varie considérablement mais dans chacun des jeux un petit nombre d'éléments agissent comme des hubs, voisins très populaires.

de pathologies différentes. L'asymétrie est moins forte ici, mais l'expression de la *hubness* reste marquée dans chacun des cas, avec les *hubs* les plus influents figurant dans un cinquième à la moitié des voisinages.

Inspirée de la *network science*, l'étude de la *hubness* est récente dans le domaine de l'analyse de données (Radovanovic, Nanopoulos, & Ivanovic, 2010). Toutefois certaines propriétés sont déjà attribuées aux *hubs* : pour des approches de *clustering* utilisées en haute dimension, les *hubs* sont une approximation fiable des centres de clusters proposés par les k-moyennes, et sont très proches des médoïdes issus du k-médoïdes *clustering* (Tomasev, Radovanovic, Mladenic, & Ivanovic, 2014). Ainsi, les approches basées sur les *hubs* fournissent, en haute dimension, de meilleurs résultats que les deux algorithmes cités précédemment (Buza, Nanopoulos, & Schmidt-Thieme, 2011). Soulignons aussi que les méthodes habituelles de réduction de dimensionnalité évoquées dans la sous-section suivante n'affectent pas la *hubness*, et qu'à l'inverse réduire la *hubness* ne résoud pas les problèmes de dimensionnalité (Feldbauer & Flexer, 2018).

2.2.4 Réduction de la dimensionnalité

Pour parer à ces problèmes qui émergent en grande dimension, la solution la plus courante est de réduire la dimensionnalité pour ramener les données dans un espace de description aux dimensions acceptables. Il existe plusieurs approches pour parvenir à ce résultat, mais les deux présentées ci-dessous sont les plus répandues.

Feature engineering

Il est possible d'agir directement sur les dimensions pour en réduire le nombre (Blum & Langley, 1997). Une première possibilité est de retirer les variables non pertinentes (Hira & Gillies, 2015), par exemple en enlevant les informations redondantes portées par plusieurs variables (*e.g.* une valeur exprimée dans plusieurs unités). Ces informations, en plus d'augmenter inutilement la dimension de l'espace considéré, peuvent induirent des comportements de surapprentissage. Une deuxième étape possible est l'agrégation de variables pour générer de nouvelles variables appropriées à la tâche d'apprentissage que l'on souhaite mener. Il s'agit de créer de nouvelles *features* dérivées des *features* initiales et pouvant porter la même information. Cela peut être fait en fusionnant certaines *features* ou en créant des *features* capables d'en subsumer d'autres (*e.g.* pour certaines tâches, la taille et le poids peuvent être subsumés par l'*IMC*).

Le point commun de toutes ces méthodes est qu'elles requièrent une connaissance *a priori* de la tâche, ou font l'hypothèse d'une distribution normale pour de nombreux modèles. Il est nécessaire, autant que possible, de savoir quelles informations sont présentes dans le *dataset*, quelles méthodes d'analyse et d'extraction de données sont à privilégier, et quels sont les résultats recherchés ou le type d'information à extraire. Cela implique non seulement un utilisateur expert du domaine étudié, mais surtout un investissement de temps pour la préparation des données. Dans le contexte médical qui est le notre, mais aussi dans de nombreux autres cadres applicatifs pour la recommandation et l'aide à la décision, ce second facteur n'est tout simplement pas disponible. Dans ces cas, la réduction de dimensionnalité peut toujours être envisagée grâce aux techniques de projection non-supervisées.

Techniques de projections

Le but de ces algorithmes est d'obtenir une représentation en faible dimensionnalité (souvent en deux dimensions) à partir de données décrites par un grand nombre de *features*. Ce domaine de l'intelligence artificielle offre de nombreuses méthodes et solutions, qu'il convient de grouper en deux catégories.

Les méthodes de projection linéaires tout d'abord, ont pour but d'obtenir une représentation des données qui conserve les tendances et les motifs représentés dans l'espace en grande dimension. On compte dans cette catégorie plusieurs approches comme l'Analyse en Composantes Principales (*PCA*) (Pearson, 1901), mais aussi l'analyse factorielle exploratoire (*EFA*) (Spearman, 1904), l'analyse en composantes indépendantes (*ICA*) (Hyvarinen, 1999), ou encore le positionnement multidimensionnel (*MDS*) (Torgerson, 1958) pour ne citer que quelques unes des plus utilisées.

La deuxième catégorie est celle des méthodes non linéaires ayant pour but de restituer fidèlement la cohérence des données à un niveau local. On note là aussi un panel très large de méthodes : les *t-SNE* (Maaten & Hinton, 2008) ou les *isomaps* (Tenenbaum, Silva, & Langford, 2000), ou encore le *LTSA* (Zhang & Zha, 2004) ou *KPCA* (Scholkopf, Smola, & Muller, 2006), la variante à noyau de *PCA*. Ces méthodes ont pour objectif de préserver les voisinages locaux des éléments du jeu de données ; mais la proximité entre les points de la projection résultante ne se traduit pas systématiquement un concept de similarité.

Ce large éventail d'algorithmes nécessite tout d'abord une connaissance des données, que ce soit leur distribution ou les phénomènes à leur origine, mais aussi une bonne maîtrise des méthodes disponibles afin de choisir la plus adéquate. Toutes ces approches ne préservent en effet pas les mêmes propriétés du jeu de données, certaines conservant les distances, d'autres le voisinage ou encore les dimensions. Des connaisances *a priori* sont donc là aussi nécessaires, ce qui diverge de notre hypothèse de travail.

Impact sur l'interprétabilité

Durant l'analyse de données et l'extraction de connaissances, la réduction de dimensionnalité par *feature engineering* ou projection va induire un autre problème. En plus de la nécessité d'avoir un utilisateur expert, et d'imposer l'utilisation de connaissances *a priori* dont l'absence est souvent la motivation pour une analyse exploratoire, l'application de ces méthodes se heurte au problème d'interprétabilité. Le retrait de *features*, la fusion et création de nouvelles *features*, l'ajout d'étapes de prétraitement parfois complexes et peu transparentes, des résultats plus complexes à analyser : autant d'obstacles pour une méthode claire et interprétable pour son utilisateur. L'interprétabilité, ses avantages et les conséquences de son absence sont détaillées dans la section suivante.

2.3 Interprétabilité

L'interprétabilité désigne un sous-ensemble vaste et souvent flou de méthodes, concepts et applications dans le domaine de l'intelligence artificielle. Mise en avant depuis la démocratisation des algorithmes *black box* comme les réseaux de neurones profonds, l'interprétabilité est toutefois mal délimitée et regroupe nombre de concepts dont les définitions formelles varient. Contrôle de l'utilisateur, transparence, « *right to explain* », ou explicabilité sont autant de mots, parfois synonymes, parfois distincts, que l'on regroupe souvent sous le terme « interprétabilité ».

Étudiée depuis de nombreuses années, c'est depuis la récente omniprésence de l'intelligence artificielle dans le monde moderne que l'interprétabilité s'est propulsée au coeur des préoccupations de nombreuses communautés scientifiques, entités industrielles ainsi que groupes politiques et citoyens [2]. Car en plus de l'évolution des techniques employées, le *machine learning* est maintenant indissociable de domaines applicatifs comme le médical, le militaire, la sécurité, la finance ou même la justice et l'écologie. [3] À une époque où le contenu multimédia auquel chacun est confronté se retrouve déterminé non plus par l'individu mais par des entités tierces qui se basent sur l'apprentissage machine, il devient naturel d'étudier cette situation sous le spectre de l'interprétabilité. Avant d'en détailler les facettes, il convient de définir des conventions et d'établir un vocabulaire nous permettant d'analyser l'interprétabilité d'une méthode d'analyse de données. Les termes retenus pour lever les ambiguités de l'interprétabilité sont empruntés aux travaux de Zachary Lipton (2018) qui pose un cadre sémantique et propose une unification des définitions habituellement utilisées. La classification proposée dans cette section s'en inspire majoritairement, tout en étant replacée dans le contexte médical qui est le nôtre.

Le contenu de cette section est résumé visuellement par la Figure 2.10, qui propose une taxonomie du domaine de l'interprétabilité. La première des deux sous-sections suivantes présente les motivations amenant à considérer l'interprétabilité comme facteur important d'un algorithme, en y évoquant les bénéfices techniques et sociétaux. La seconde détaille deux facettes de l'interprétabilité, ainsi que plusieurs implémentations possibles pour chacune d'elles.

2.3.1 Objectifs

Il est important de commencer par rappeler que les objectifs applicatifs sont à différencier des objectifs de l'algorithme de *machine learning*, car les deux sont souvent totalement différents et décorrélés. Pour un algorithme d'apprentissage automatique, les objectifs sont clairement définis : minimiser l'erreur, maximiser la précision, le rappel, la séparabilité ou garantir la *scabalibity* pour n'en citer que quelques uns. Toutefois, le but final des utilisateurs de ces méthodes est différent : il s'agit de détecter des corrélations interindividus, comprendre les mécanismes ayant généré les données, extraire des motifs ou encore explorer les données mises à leur disposition.

2. *Explainable IA: why we need to open the black box*, Forbes, 2019
3. *Rapport de synthèse France IA*, 2017

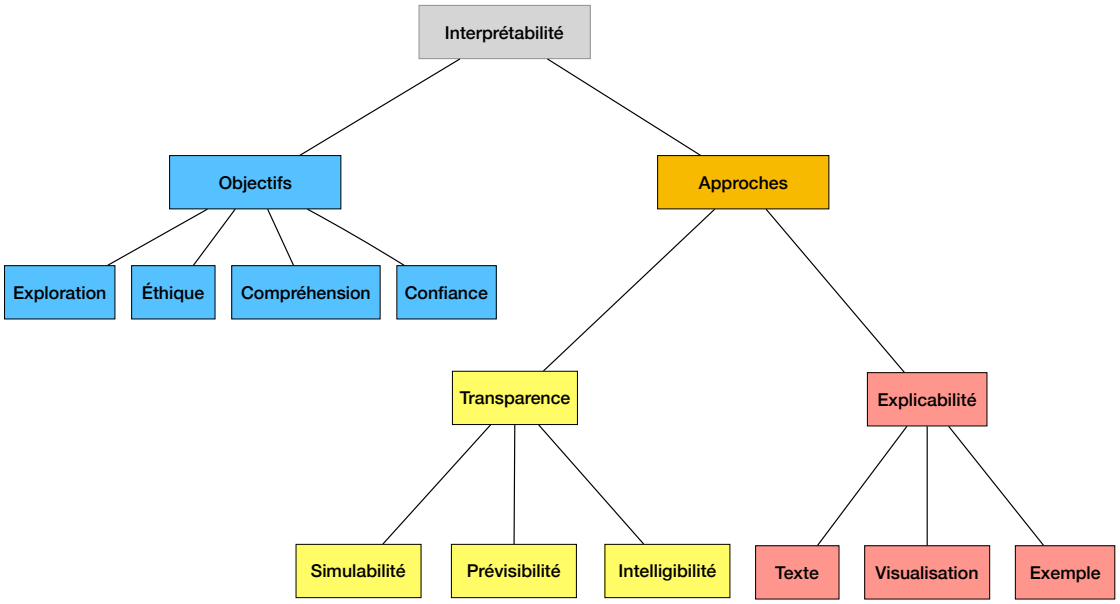


FIGURE 2.10 – Taxonomie de l'interprétabilité détaillant les différentes approches lui étant liées. On distingue l'interprétabilité comme objectif et comme méthode permettant d'atteindre ces objectifs. Ainsi, un objectif donné peut être atteint de plusieurs manières, via la transparence algorithmique ou via l'explicabilité des résultats.

Si nous prenons l'exemple d'une voiture connectée, l'objectif de l'utilisateur est que la voiture circule de manière autonomne, ce qui n'est absolument pas l'objectif des algorithmes gouvernant le véhicule qui eux tentent de maximiser la reconnaissance des formes, etc . En intelligence artificielle, les objectifs de l'interprétabilité servent l'intention de l'utilisateur final et n'ont pas de lien avec l'objectif que l'algorithme tente d'optimiser. Cette dichotomie entre l'algorithme et l'utilisateur a pour source la difficulté à retranscrire les buts concrets que servent ces algorithmes en fonctions mathématiques optimisables par la machine. Le constat est donc une interprétabilité aux finalités multiples pour ceux qui font usage de ces outils.

Confiance

Le premier but servi par l'interprétabilité est la création de confiance envers le système. De nombreuses approches partent du postulat qu'en comprenant quels faits motivent les décisions d'un algorithme, les utilisateurs du système seront plus enclins à le solliciter. Un modèle transparent va par exemple permettre de s'assurer que, même si l'algorithme a un taux de succès important, ce succès est dû non pas à un facteur chance mais à une analyse efficace des données en entrée, selon un processus rationnel. Un utilisateur suffisamment familier avec le système sera aussi plus enclin à lui laisser le contrôle d'une situation s'il en connaît le fonctionnement et est convaincu qu'il est efficace. C'est un défi à relever dans de nombreux systèmes comme la santé ou le transport, où une partie du grand public n'est pas prêt à confier des domaines aussi importants à des entités au fonctionnement opaque (Tran et al., 2019).

Éthique

De nombreuses considérations éthiques entrent en jeu lors de l'automatisation de tâches qui étaient jusque là accomplies par un être humain. La société au sens large commence à se saisir de ces problématiques [4] comme le reflète l'expression « *right to explain* » récemment adoptée dans le domaine juridique. Il est ainsi important de prendre en compte la manière dont sont appliqués ces algorithmes, l'absence ou non d'une supervision humaine, la présence de biais dans les données d'entraînement ainsi que la création de biais nouveaux à l'issue de la phase d'apprentissage.

Par exemple, si les données d'entrée ont une importance particulière en terme d'éthique c'est car elles ne font pas que risquer de pénaliser le taux de réussite de l'algorithme lors de la phase d'évaluation. Ces données peuvent aussi être responsables de la perpétuation de modèles sociaux et culturels parfois inconscients, en les inscrivant définitivement dans le processus décisionnel de sorte qu'ils se reproduisent ensuite involontairement (par un biais comme celui de confirmation). Dans ces circonstances et faute d'indicateur ou de métrique universellement accepté pour détecter le biais et la discrimination, un modèle interprétable offre la possibilité de porter un oeil critique sur le système décisionnel pour détecter ces problèmes d'équité, quantifier leur impact et pouvoir le corriger.

Compréhension

Avoir des résultats interprétables peut permettre de les utiliser de manière optimale en prenant en compte des informations dépendantes du contexte et qui échapperaient au cadre formel analysé par l'algorithme. Comme évoqué au début de cette section, les objectifs d'un algorithme de *machine learning* sont différents de ceux de son utilisateur. Ils sont une tentative de formaliser un problème du monde réel pour pouvoir lui appliquer des approches algorithmiques, statistiques ou mathématiques. En comprenant le fonctionnement d'une approche, il est ainsi possible d'apprécier la différence entre le contexte algorithmique et le contexte réel, augmentant de même l'apport du système pour l'utilisateur en offrant un moyen d'inscrire les résultats dans le contexte concret.

Exploration

L'interprétabilité peut pallier un autre problème induit par les données étudiées. En argumentant les résultats obtenus et en les expliquant, l'utilisateur peut utiliser ces outils dans le cadre d'une étude exploratoire. S'il est possible de détailler les associations découvertes par un algorithme, il devient plus aisé de les utiliser pour comprendre les phénomènes ayant généré les données étudiées (Pearl, 2009). Il peut ainsi être possible de découvrir des *patterns* dans les données, des corrélations entre variables, des erreurs de saisie ou des anomalies dans les distributions. Toutes ces informations ne sont pas observables directement grâce aux résultats, mais nécessitent de connaître les phénomènes ayant mené aux résultats.

4. *Donner un sens à l'intelligence artificielle*, Cédric Villani, 2018

2.3.2 Approches

Les approches facilitant l'interprétabilité sont regroupées sous deux notions clés que sont la transparence et l'explicabilité. La transparence fait référence à la capacité à comprendre le fonctionnement du modèle utilisé, les étapes de l'algorithme, les opérations qui sont effectuées. L'explicabilité est la capacité à comprendre, même avec un modèle de type *black box* les éléments qui ont engendré le résultat obtenu, souvent en s'appuyant sur les *features* des données d'entrée. Il s'agit ici d'établir des liens de causalité entre l'entrée du modèle et les résultats obtenus.

Transparence

La transparence, permettant d'éviter l'effet *black box*, intervient à tous les niveaux du modèle. Il s'agit dans tous les cas de la facilité de compréhension des mécanismes produisant les résultats. On peut ainsi détailler plusieurs manières d'aborder la transparence algorithmique :

— **Simulabilité** : La simulabilité implique la possibilité de comprendre le fonctionnement global de l'algorithme et pourrait être résumée comme la capacité à expliquer de manière concise le modèle utilisé. Un modèle ayant une bonne simulabilité pourrait idéalement être reproduit étape par étape par l'utilisateur si ce dernier dispose des données d'entrée et des paramètres. Cette définition n'est donc que très peu applicable aux données en haute dimension, pour lesquelles il est souvent impossible de reproduire des calculs simples à l'échelle nécessaire.
— **Intelligibilité** : L'intelligibilité peut être assimilée à l'intuitivité. Si la simulabilité est à l'échelle macroscopique, l'intelligibilité se place à échelle microscopique : chaque élément, pris séparément, remplit une fonction compréhensible pour l'utilisateur. C'est la capacité à décomposer l'algorithme en briques individuellement intuitives. Ainsi un réseau de *deep learning*, bien que peu simulable, est hautement décomposable et peut être considéré comme intelligible, chaque neurone effectuant une opération simple à partir de ses entrées.
— **Prévisibilité** : La prédictibilité du modèle représente la possibilité d'anticiper les résultats fournis par l'algorithme lorsqu'il est appliqué à des données non présentes dans le corpus d'entraînement. Un modèle prédictible permet par exemple d'identifier les cas qui seront problématiques lors de l'usage ou de discerner à l'avance les contextes sources d'erreur. Cette qualité peut être atteinte en prouvant mathématiquement certaines propriétés comme le fait que l'algortihme convergera pour toute entrée, ou encore qu'une heuristique gardera un taux d'erreur inférieur à un seuil prédéfini.

Explicabilité

L'explicabilité est la capacité à justifier, *a posteriori*, les résultats obtenus (Gosiewska & Biecek, 2019). L'explicabilité est entièrement décorrélée de la compréhension du modèle et une approche *black box* peut avoir un fonctionnement échappant à l'utilisateur mais être explicable en lui fournissant assez d'informations sur les facteurs influençant les résultats. Une difficulté majeure de l'explicabilité est que les explications de certains modèles sont additives (l'ordre dans lequel les *features* sont considérées n'a pas d'importance) tandis

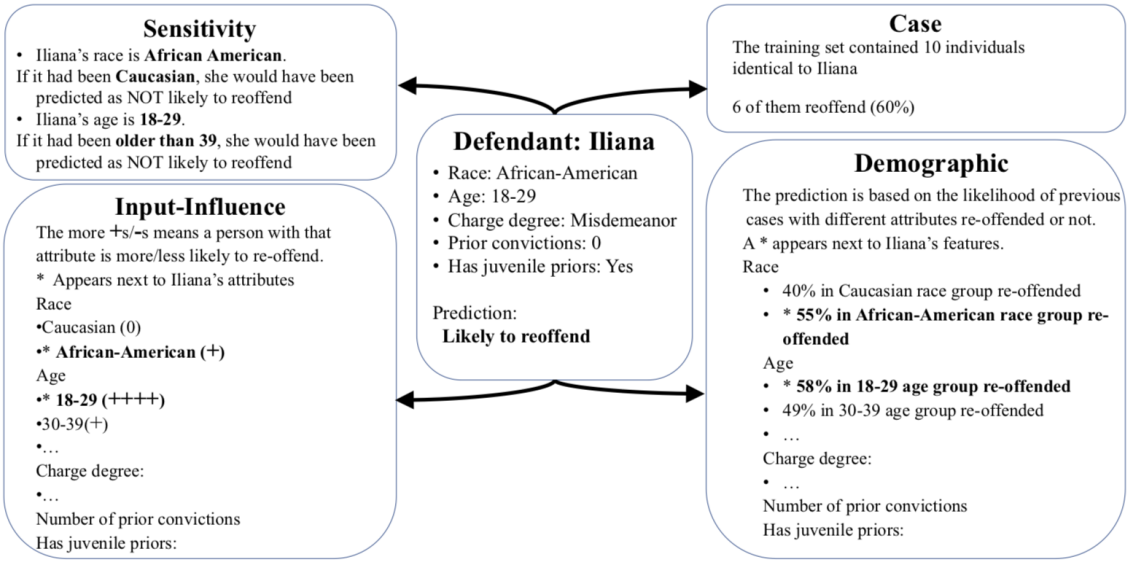


FIGURE 2.11 – Figure issue du travail de Dodge et al. (2019). Exemple d'explications textuelles du résultat d'un algorithme de classification. Ici l'objectif principal est de s'assurer de l'égalité et de l'éthique de la décision.

que pour d'autres elles sont sensibles à l'ordre dans lequel l'information est traitée. Les explications doivent donc être adaptées à la méthode. De plus, un algorithme efficace étant souvent complexe et une bonne explication souvent simple, il découle de cette dichotomie plusieurs manières d'expliciter les décisions prises par un système :

— **Texte** : Le langage naturel est l'une des options les plus couramment choisies. Ici, un bloc de texte, de préférence succinct, décrit à l'utilisateur les *features* ayant eu un impact important sur la prédiction formulée, tout en qualifiant leur impact (Schaffer, O'Donovan, Michaelis, Raglin, & Hollerer, 2019). Cette approche, illustrée par la Figure 2.11 issue du travail de Dodge, Liao, & Bellamy (2019), émule le fonctionnement d'un expert justifiant les choix qu'il a effectués et les éléments clés pour arriver à la conclusion de son analyse.

— **Visualisation** : D'autres approches, comme *LIME* (Ribeiro, Singh, & Guestrin, 2016), peuvent illustrer visuellement la contribution des *features* par rapport au résultat final (Ross, Hughes, & Doshi-Velez, 2017). La Figure 2.12, issue du travail de Bach et al. (2015), présente un exemple d'explication par visualisation. Il est aussi possible de fournir des explications visuelles en proposant à l'utilisateur une projection adaptée du résultat permettant de mettre en avant les dimensions discriminantes et les *patterns* ayant conduit à ce résultat. Les méthodes de projection de données en grande dimension citées dans la section précédente, comme les *t-SNE*, peuvent alors être utilisées pour la visualisation.

— **Exemple** : Cette méthode d'explication, sur laquelle s'appuie ce travail de thèse, est basée sur l'analogie. L'objectif est de présenter à l'utilisateur les cas qui ont été jugés similaires par l'algorithme, de sorte que l'utilisateur puisse en déduire le

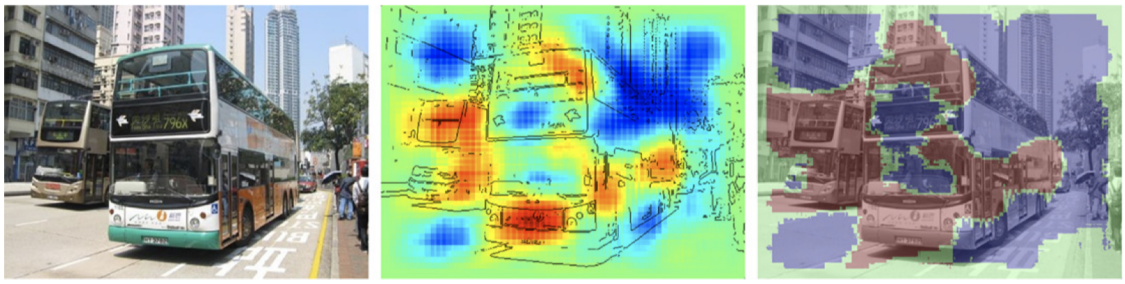


FIGURE 2.12 – Figure issue du travail de Bach et al. (2015). Dans cette explication par visualisation, nous retrouvons l'image d'origine à gauche (par l'utilisateur "tpsdave" sur Pixabay) et au centre la heatmap indiquant l'utilité des zones de l'image lors de la classifications. A droite enfin, la heatmap binarisée et superposée à l'image d'origine : les zones vertes ne sont pas utiles à la classification, les rouges renforcent l'appartenance à la classe, et les bleues la réduisent.

fonctionnement de la méthode. Ce type d'explication se base sur le raisonnement et l'apprentissage humain, dont le processus décisionnel repose en partie sur des similarités apprises par l'expérience. On peut distinguer les explications normatives et comparatives, par exemple pour expliciter à l'utilisateur les causes d'une erreur de classification sur un élément comme sur la Figure 2.13 issue du travail de Cai, Jongejan, & Holbrook (2019). L'explication normative consiste à présenter à l'utilisateur des éléments de la classe cible, pour souligner la différence entre ces éléments et l'élément mal étiquetté. Une explication comparative mettra en avant la similarité entre l'élément étudié et les données d'entraînement les plus similaires d'après le modèle. Le raisonnement par analogie est un domaine à part entière de la cognition, auquel la section suivante est consacrée.

2.4 Processus décisionnel

Des domaines scientifiques comme la psychologie, la cognition ou les neurosciences étudient depuis longtemps le processus mis en oeuvre par l'être humain lors de la prise de décision (Roy & Bouyssou, 1993). De nombreuses approches de modélisation des processus décisionnels coexistent et plusieurs d'entre elles ont fait consensus. Dans cette thèse nous nous concentrons sur le raisonnement par analogie, et cette section est donc une revue des approches les plus prévalentes dans l'étude du mécanisme de catégorisation et, par extension, de prise de décisions.

La catégorisation de stimuli est une capacité essentielle qui est utilisée dès le plus jeune âge. Elle joue un rôle important en permettant de rattacher les objets et évènements à des concepts. L'analyse du processus de catégorisation humain s'est donc imposé depuis longtemps comme domaine d'étude (Guilford, 1954). C'est par ce processus qu'il est possible de reconnaître et catégoriser des entités auxquelles nous n'avons jamais été exposées. L'idée au centre de ce domaine est celle de généralisation en concepts : en étant exposé à répétition à des entités rattachées au même concept, l'être humain en dérive une généralisation lui permettant à l'avenir d'assigner une catégorie à de

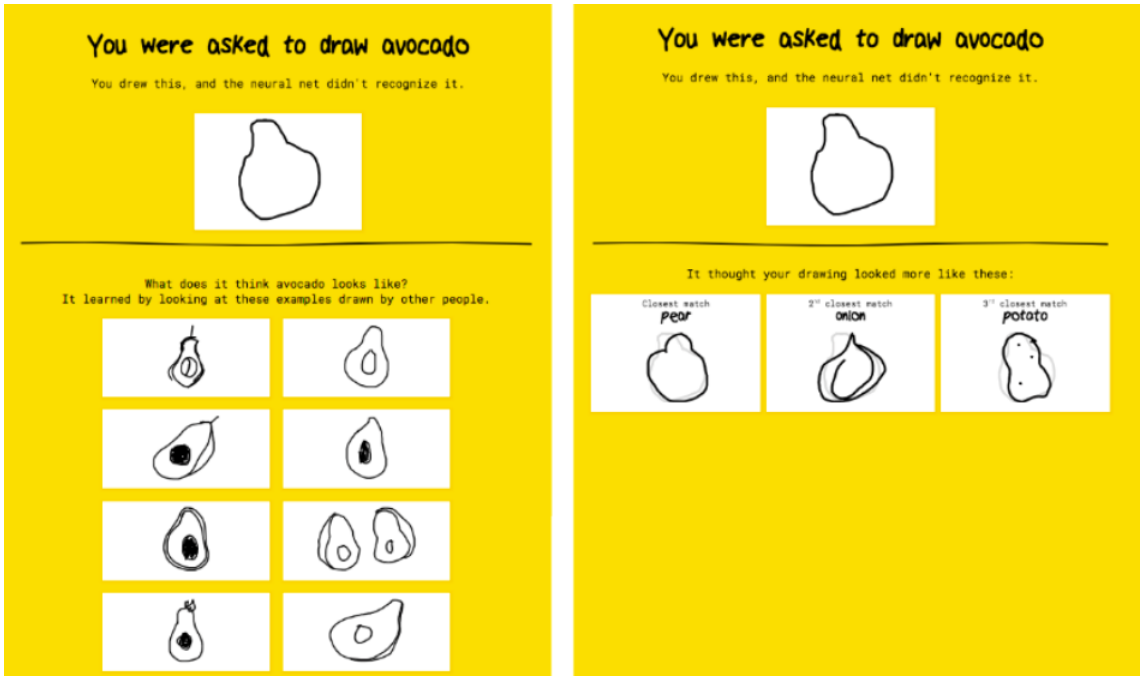


FIGURE 2.13 – Figure issue du travail de Cai et al. (2019). Cette explication par l'exemple informe l'utilisateur, si l'objet qu'il a dessiné a été mal reconnu. À gauche l'explication normative lui montre des exemples de dessins du même objet, qui ont été correctement reconnus. À droite, l'explication comparative lui montre les objets les plus similaires à son dessin, d'après l'algorithme.

nouvelles entités inconnues. C'est grâce à cet apprentissage qu'un enfant sera capable, après avoir observé plusieurs chiens de différentes races, de reconnaître tout nouveau canidé qu'il rencontrerait même si celui-ci présente certaines caractéristiques nouvelles et inhabituelles (*e.g.* un chien albinos, un loup). C'est de ce processus de catégorisation des stimuli que découle le processus de prise de décision : c'est en catégorisant un problème qu'il est possible de choisir une solution adaptée. Et c'est le contexte de la prise de décision et le type de stimuli à catégoriser qui déterminent le mécanisme de catégorisation qui sera sollicité.

Cette section présente plusieurs mécanismes utilisés par l'être humain pour mémoriser une représentation d'un concept et la solliciter. Nous évoquons d'abord la catégorisation *rule-based*, avant de décrire l'*exemplar theory* et de la situer par rapport à l'approche de la *prototype theory* à laquelle on l'oppose souvent. Enfin, nous décrirons l'équivalent informatique de cette *exemplar theory*, à savoir les algorithmes *instance-based* reposant sur les individus constituant un jeu de données.

2.4.1 Rule-based

Cette approche de la catégorisation propose une construction des concepts selon un ensemble de règles (Bower & Trabasso, 1963). Dans le cas le plus simple d'un concept dit « *well-defined* », les exemples de ce concept possèdent tous un ensemble de caractéristiques qui leur sont communes et qui font que ces éléments sont plus similaires

entre eux qu'avec des éléments d'autres catégories ne possédant pas des caractéristiques identiques (Katz & Postal, 1964). Il est ainsi possible de raisonner à partir de règles car pour un nouvel objet pour lequel plusieurs catégories sont envisageables, les règles proposent des filtres successifs à appliquer pour déterminer quelle catégorie est la plus appropriée. C'est par exemple la méthodologie au coeur des approches basées sur les arbres de décision.

Cette méthode de raisonnement est particulièrement fréquente dans plusieurs cas. Tout d'abord lorsque les stimuli à catégoriser sont suffisamment similaires pour qu'il soit difficile de percevoir des différences. Dans ce cas, et si les stimuli sont simples, il est possible d'y appliquer des règles pour en dériver une catégorie (Rouder & Ratcliff, 2006). C'est le cas par exemple pour différencier des carrés et des rectangles : s'il ne sont pas différenciables visuellement, il reste possible d'en mesurer les côtés et d'appliquer une règle (« les côtés sont-ils tous de longueur égale ? » par exemple) pour déterminer la catégorie aux caractéristiques correspondantes. Le deuxième cas favorable à la décision *rule-based* est celui d'un sujet confronté à une prise de décision dans un domaine dont il n'est pas expert. Dans ce cas, et faute d'expérience sur laquelle s'appuyer pour raisonner par analogie, les règles sont souvent utilisées, tout comme lorsqu'un sujet souhaite vérifier une décision prise intuitivement par analogie (Kahneman & Klein, 2009). En informatique, l'utilisation de règles se retrouve entre autres dans des approches classiques que sont les arbres de décision (Breiman, Friedman, Olshen, & Stone, 1984) ou l'extraction de règles d'association (Agrawal, Imielinski, & Swami, 1993).

2.4.2 Exemplar-based

L'*exemplar theory*, ou la décision *exemplar-based*, consiste à catégoriser des objets ou évènements non plus selon un ensemble de règles appliquées aux caractéristiques d'un élément, mais grâce à l'analogie. Formulée pour la première fois par Nosofsky (1986) qui s'inspirait de la *context theory* proposée par Medin & Schaffer (1978), c'est le raisonnement le plus utilisé par les experts confrontés à un problème dans leur champ d'expertise (Klein, Calderwood, & Clinton-Cirocco, 2010) ; pour cette raison nos travaux s'appuient sur cette définition du processus cognitif et de la ressemblance (Wittgenstein, 2001).

Lors de la catégorisation d'un nouvel élément, l'approche *exemplar-based* va le comparer aux exemples précédemment mémorisés, et l'affecter à la catégorie qui compte le plus d'exemples similaires à l'élément. Dans ce contexte, les exemples sont des stimuli rencontrés dans le passé, peu importe l'approche utilisée pour les catégoriser. Prenons l'exemple de la catégorisation d'un spécimen d'autruche : si l'animal n'a jamais été rencontré auparavant, ses caractéristiques (plumes, ailes, bec dépourvu de dents ou inapte au vol entre autres) seront comparées aux exemples connus d'animaux, afin de retrouver les plus similaires. Ainsi par exemple, dans les domaines de la psychologie et de la cognition, l'autruche sera catégorisée comme oiseau si la majorité des exemples similaires appartiennent à cette catégorie. Il est à noter que cette catégorie n'est pas réellement une catégorie « oiseau » mais plutôt une catégorie émergente grâce aux similarités de ses membres, qui s'avèrent être des oiseaux.

Cette méthode de catégorisation propose plusieurs avantages par rapport à l'approche *rule-based*. Tout d'abord, elle est plus souple que cette dernière car en définissant un concept grâce à un ensemble d'exemples il est possible de mémoriser des exceptions pour certaines caractéristiques. Avec une approche utilisant les règles, l'autruche n'aurait pas répondu à la règle « Apte à voler » du concept « Oiseau », et aurait nécessité l'ajout d'une règle supplémentaire pour capturer cette exception. Cet ajout de règle, augmentant la complexité du raisonnement *rule-based*, pointe aussi vers d'autres avantages de l'*exemplar theory* faisant d'elle la méthode de catégorisation utilisée par les experts de nombreux domaines, dont ceux de la santé.

Le premier de ces avantages est la rapidité avec laquelle il est possible d'attribuer une catégorie aux stimuli, le raisonnement par analogie étant instinctif. Il est ainsi utilisé dans des situations où le facteur temps est critique, que ce soit car la décision doit être prise rapidement (un pompier combattant un incendie) ou car le processus décisionnel doit être répété de très nombreuses fois à des intervalles faibles (médecin recevant plusieurs dizaines de patients dans la journée). La rapidité de ce processus décisionnel lui confère un autre avantage, à savoir le peu d'effort mental requis. La catégorisation par analogie étant inconsciente et instinctive, elle ne demande que très peu d'efforts cognitifs. Ce facteur est, là aussi, important dans un cadre où des décisions doivent être prises régulièrement, comme c'est le cas par exemple pour tout professionnel d'un domaine. Ce type de raisonnement étant basé sur l'exemple, il s'acquiert par l'expérience, en étant confronté à des objets et évènements de natures différentes.

Dans le cadre de la prise de décision médicale, c'est ce raisonnement *exemplar-based* qui est inconsciemment privilégié par les médecins et professionnels de santé confrontés à de nouveaux patients : un expert clinique va entamer son diagnostic en générant intuitivement un ensemble d'hypothèses probables, qu'il pourra vérifier si besoin (Kahneman & Klein, 2009). Cette étape de vérification est essentielle, car le taux de confiance d'un expert en son jugement n'est pas corrélé à la validité de celui-ci (Einhorn & Hograth, 1978). La vérification de ces hypothèses repose sur un mécanisme *rule-based* permettant de déterminer la catégorie finale, si l'approche *exemplar-based* ne permettait pas de discriminer les hypothèses disponibles.

L'*exemplar theory* repose donc sur la sélection, parmi les membres d'une catégorie, d'un sous-ensemble d'exemples jugés comme étant les éléments plus représentatifs de la catégorie. Plusieurs facteurs déterminent ce que nous appellerons la **représentativité** des membres d'une catégorie (Nosofsky, 1991). Le premier facteur influençant la représentativité est la typicalité d'un membre. Pour qu'il soit typique, le stimulus doit partager le plus de caractéristiques possibles avec les autres membres de la catégorie mais aussi maximiser sa différence avec les autres catégories (Lesot, Rifqi, & Bouchon-Meunier, 2008 ; Rosch & Mervis, 1975). Cela fait de l'autruche prise en exemple précédemment un membre peu typique de la catégorie « Oiseau », et donc peu à même d'être un exemple. La fréquence est une deuxième cause de représentativité. Plus un membre d'une catégorie sera rencontré fréquemment, plus le raisonnement par analogie l'associera à la catégorie qui le subsume. Ainsi, la voiture est plus représentative de la catégorie « Véhicule » que l'avion ou le bateau.

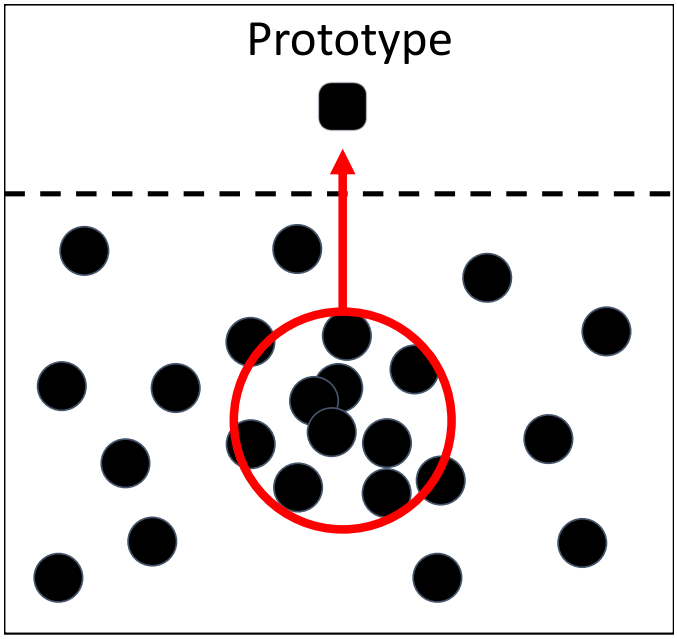


FIGURE 2.14 – Illustration du processus de construction d'un prototype pour un concept. Les stimuli les plus typiques du concept, dans le cercle rouge, sont agglomérés en un seul prototype qui devient représentant de ce concept.

Le dernier facteur impliqué dans la notion de représentativité est celui de la récence. Il est plus probable qu'un stimulus soit utilisé comme exemple s'il a été rencontré récemment. Là aussi l'exemple de la voiture s'applique, car en plus d'apparaître plus fréquemment dans nos vies, elle sera plus souvent que l'avion le dernier véhicule rencontré. Ainsi, les exemples associés à chaque catégorie sont amenés à évoluer au cours du temps, modelés par notre expérience et influencés par les évènements récents.

Prototype theory

Afin de mieux décrire *l'exemplar theory*, il convient de la situer par rapport à une autre approche : la *prototype theory*. Celle-ci a été popularisée par Eleanor Rosch (1973) et, même si les bases de l'*exemplar theory* furent posées à la même période, certaines différences fondamentales les séparent dans leur conception comme dans leurs applications.

La *prototype theory* est aussi basée sur le raisonnement par analogie, à partir des caractéristiques de l'objet, mais au lieu d'utiliser ces caractéristiques pour le rapprocher d'un ensemble d'exemples issus de plusieurs catégories, l'objet est rapproché d'un unique prototype de catégorie (Reed, 1972). Ce prototype diverge de l'exemple car, là où les exemples sont des stimuli passés, le prototype d'une catégorie est construit à partir d'un ensemble de membres représentatifs sans pour autant être l'un d'entre eux (Minda & Smith, 2002). Les exemples sont issus d'un échantillonnage, mais le prototype lui est créé de sorte à refléter les caractéristiques des éléments les plus représentatifs, comme illustré sur la figure 2.14. Le prototype, bien qu'étant une généralisation, n'est toutefois pas qu'une simple moyenne des membres représentatifs, il est plus complexe et subsume la catégorie qu'il représente grâce à un unique élément abstrait. La figure 2.15 propose

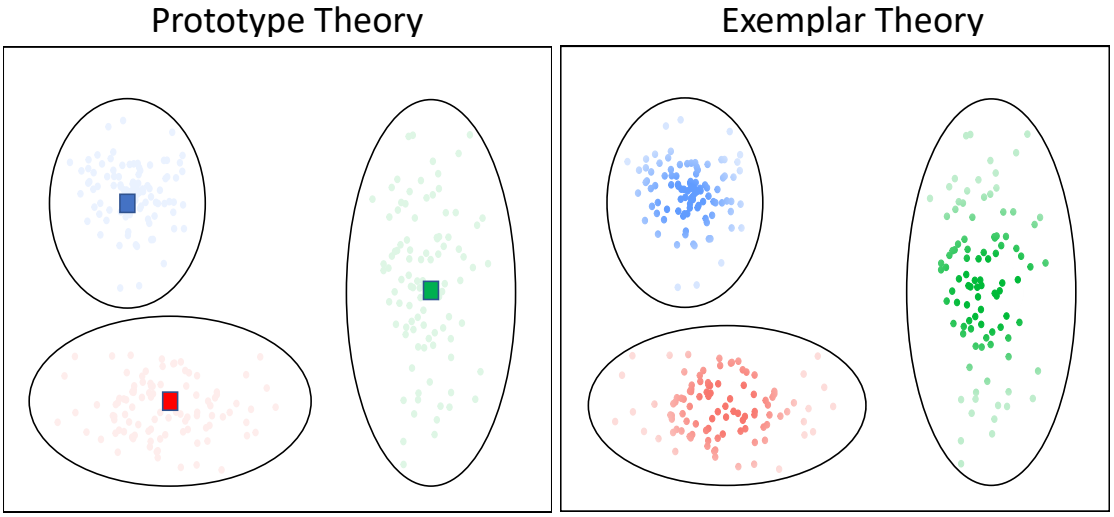


FIGURE 2.15 – Ce schéma illustre, avec trois concepts (en vert, rouge et bleu), la différence entre la prototype et l'exemplar theory. Plus un élément est opaque plus il est représentatif et typique du concept auquel il appartient et plus il est utilisé pour la catégorisation de stimuli.

une illustration des différences entre un prototype et des exemples. En *prototype theory*, un individu artificiel est créé dans chaque concept pour la représenter, les autres n'étant pas utilisés par le processus (voir la définition de Reed (1972)). En exemplar theory, la quasi totalité des éléments servent d'exemples pour identifier le concept, et les plus représentatifs sont le plus souvent utilisés. Ces deux approches sont toutes deux utilisées instinctivement par l'être humain, mais leurs différences les rendent appropriées à des contextes différents et spécifiques (Mack, Preston, & Love, 2013).

Notre choix d'appuyer nos travaux sur l'*exemplar theory* plutôt que la *prototype theory* découle du contexte dans lequel s'intègre le projet initial : afin d'être adapté à un usage clinique, il était nécessaire de se reposer sur des cas concrets plutôt que des cas abstraits et artificiels issus de moyennes statistiques. Ainsi, en utilisant le raisonnement par analogie basé sur l'exemple, il est possible de guider un praticien en le référant à des patients déjà rencontrés, dont il peut analyser le dossier médical par exemple (Thomson, Lebiere, Anderson, & Staszewski, 2015). Un « méta individu » moyen généré algorithmiquement peut faire un excellent prototype, mais ne permettra pas à un spécialiste de se reposer sur son expérience et peut même, dans certains contextes, ne correspondre à aucun cas réel.

Instance-based

Pour implémenter cette approche *exemplar-based* dans la méthode décrite dans cette thèse, nous avons fait le choix de privilégier des algorithmes « basés instance », ou *instance-based*. Ces algorithmes parviennent à éviter la généralisation observée dans la *prototype theory* en basant leurs hypothèses sur des individus issus du jeu de données étudié. Ainsi, ces individus offrent un certain niveau de généralisation et de généricité tout en restant des objets réels. Ce choix permet de favoriser le raisonnement par

analogie puisque, dans le cadre clinique, tous les résultats seront des patients réels et que le praticien a potentiellement aussi suivi, il peut donc valoriser son expérience médicale. Il existe de nombreux algorithmes *instance-based* (Bicocchi, Mamei, & Zambonelli, 2010 ; Frey & Dueck, 2007), et parmi les plus utilisés figurent ceux basés sur la méthode des k plus proches voisins. On note ausi, parmi les méthodes *instance-based* courantes, l'algorithme de clustering des k-médoïdes qui incarne une variante *exemplar-based* du populaire algorithme des k-moyennes basé, lui, sur les prototypes. Le chapitre suivant détaille la méthode proposée, qui se base sur de tels algorithmes *instanced-based*.

Chapitre 3

Algorithme de structuration

Sommaire

3.1 Représentativité 43
3.1.1 Objectifs et travaux préliminaires 44
3.1.2 Calcul de représentativité 45
3.2 Structuration 51
3.2.1 Construction du graphe 52
3.2.2 Choix du facteur k 55
3.3 Propriétés 58
3.3.1 Transparence 58
3.3.2 Explicabilité 60
3.4 Détection d'anomalies 61

Dans ce chapitre nous présentons une méthode instance-based de structuration de données multidimensionnelles, dans le but de faciliter l'analyse exploratoire des données. Cette structuration se fait selon un score de représentativité, permettant de dériver un graphe où chaque élément est relié à son représentant, ce dernier lui est similaire tout en étant plus générique et archétypal. Nous proposons ensuite une méthode de détection d'anomalies adaptée à la haute dimensionnalité. Nous présentons d'abord la représentativité et les travaux récents sur lesquels se base notre méthode de structuration, en listant les avantages que nous tentons de préserver et les faiblesses que nous avons pour objectif de corriger. Nous décrivons un algorithme de calcul de la représentativité des éléments qui traite les données dimension par dimension afin de déterminer les éléments les plus représentatifs sur chaque *feature*. En traitant les dimensions séparément avant d'agréger les résultats nous pouvons éviter les phénomènes engendrés par la haute dimensionnalité. L'approche proposée nécessite de pouvoir établir un classement des éléments, un indice de dissimilarité est donc nécessaire sur chaque dimension : nous proposons plusieurs fonctions modélisant différentes relations entre les éléments. La deuxième section détaille l'utilisation du score, calculé précédemment, afin de structurer les individus du jeu de données et obtenir un graphe de similarité sur lequel les individus sont liés deux à deux, chacun à leur représentant. Cette même section présente aussi le facteur de granularité intervenant dans l'algorithme, et son impact sur le résultat final. La troisième section énumère les propriétés de la méthode proposée par rapport aux critères d'interprétabilité exposés précédemment dans l'état de l'art. La quatrième et dernière section décrit l'algorithme de détection d'anomalie que nous proposons, ainsi que son application dans le contexte médical de nos travaux.

3.1 Représentativité

Le concept de représentativité permet de faire émerger des éléments à même d'en subsumer d'autres qui leur sont similaires : un outil puissant pour l'analyse de données massives. Afin de pouvoir déterminer quels éléments d'un ensemble sont à même d'en représenter d'autres, il est nécessaire de pouvoir déterminer une similarité inter-éléments : c'est pourquoi la proposition d'une nouvelle mesure incluant représentativité et similarité est au coeur des travaux présentés dans cette thèse.

Afin d'extraire des informations ou de faciliter la manipulation des données, la sélection d'éléments parmi un jeu de données est une approche habituelle du *data mining* et de l'extraction de connaissances. Cette étape d'échantillonnage (ou *sampling*) vise à trouver des éléments d'intérêt au sein des données, dans le but de découvrir de l'information ou de fournir une synthèse. La méthode la plus souvent retenue est celle du *clustering* (Jain et al., 1999), qui mène à un partitionnement des données : il est ensuite possible d'extraire des membres représentatifs pour chaque classe, tels les centroïdes (Lloyd, 1982) ou les médoïdes (Kaufman & Rousseeuw, 1987). Mais ce choix vient avec plusieurs contraintes et limites. Il est par exemple nécessaire de disposer de connaissances *a priori* sur les données et leur distribution ainsi que sur les informations recherchées afin de sélectionner la méthode appropriée. Parfois ce sont plusieurs populations qui sont regroupées en un seul *cluster* et seront donc mal représentées par les éléments extraits, sans compter que ces représentants sont des individus artificiels lorsque la méthode n'est pas *instance-based*.

Cette section présente d'abord une approche antérieure à ce travail qui pose toutefois Certaines bases de nos contributions à la notion de représentativité. Nous détaillons ensuite la méthode que nous proposons ainsi que la manière dont elle est implémentée. Dans cette description, nous utilisons indifféremment « élément » ou « individu » pour désigner un objet appartenant à un ensemble d'objets décrits sur plusieurs dimensions.

3.1.1 Objectifs et travaux préliminaires

Présentée par Blanchard, Ait-Younes, & Herbin (2015), l'idée du *DoR* (*Degree of Representativeness*) a été proposée comme suite logique à des travaux empiriques qui visaient d'abord à quantifier la représentativité des éléments, puis à extraire les profils les plus intéressants parmi une base de données afin d'en proposer une synthèse (Ait-Younes, Blanchard, Delemer, & Herbin, 2014 ; Blanchard, Vautrot, Akdag, & Herbin, 2010). Cet indice de représentativité a été créé pour être un outil adapté à l'analyse exploratoire menée par les professionnels de la santé. Son but est d'aider à extraire des patients typiques d'un jeu de données, afin d'assister la manipulation et l'exploration de données ainsi que le diagnostic médical ou encore l'épidémiologie. C'est dans cet objectif que nous l'avons repensé et intégré à une interface de recommandation et d'aide à la décision destinée aux professionnels de santé.

Le résultat fourni par cette méthode est un ensemble d'individus (les dossiers des patients dans le cadre médical) appelés « représentants » et sélectionnés parmi le jeu de données pour leur capacité à subsumer une partie notable de la population étudiée. Structurer les éléments du *dataset* autour de ces individus représentatifs en liant chaque élément à un représentant permet d'obtenir un graphe orienté dont les composantes connexes sont des arborescences et où chaque noeud représente un patient et chaque arrête le choix d'un représentant. Il est ainsi possible de l'utiliser pour naviguer dans le jeu de données et de l'explorer, mais aussi d'extraire un ensemble d'individus d'intérêt. Cette méthode propose plusieurs avantages par rapport à l'utilisation de l'approche classique par *clustering* :

— L'algorithme, qui s'exécute en une passe, est déterministe et sélectionne donc systématiquement les mêmes exemples pour un *dataset*. Grâce à ce comportement reproductible il est possible d'expliquer *a posteriori* des résultats en les simulant à nouveau pour les analyser.
— La méthode est robuste : le choix des exemples est très stable et l'ajout de nouveaux éléments ne remet pas en cause les exemples précédemment extraits. De plus, l'effet du bruit est faible et celui des outliers est minime de par l'utilisation du rang pour déterminer les voisinages.
— Aucune connaissance *a priori* n'est nécessaire vis-à-vis des données ou de leur distribution. Il n'est pas nécessaire non plus de connaître le nombre d'exemples souhaité, celui-ci variant selon la taille du jeu de données. Il est ainsi possible d'obtenir des résultats pertinents même dans le cas d'une analyse exploratoire préliminaire sur des données inconnues.

Malgré ces avantages, plusieurs inconvénients se présentent lors de l'utilisation de cet algorithme, réduisant l'éventail de cas où son usage est approprié :

— Le calcul repose sur une mesure de distance primaire, la distance Euclidienne. Celle-ci est utilisée dans l'espace de description entier des variables ainsi la « malédiction de la dimensionnalité » en affecte les résultats, limitant leur interprétabilité en grande dimension. C'est un obstacle majeur pour de cette méthode car le sens porté par les associations diminue avec l'augmentation du nombre de variables, nécessitant des outils d'aide à l'interprétabilité *a posteriori*.
— La navigabilité proposée par le graphe issu des résultats n'est pas optimale : certains éléments n'ont pas de représentant, voire ne sont reliés à aucun autre élément de l'ensemble.
— L'algorithme établi n'a été testé que sur un nombre réduit de jeux de données. Il n'a de plus pas été appliqué à des *datasets* en grande dimension et issus du monde réel, de sorte à évaluer ses résultats sur des données complexes. Sa robustesse sur des données d'entrée en haute dimensionnalité n'a donc pas pu être étudiée.
— Les propriétés de la structuration sous forme de graphe n'ont pas été pensées pour l'exploration. Elles n'ont pas non plus été comparées à celles de graphes générés par d'autres approches comme la structuration autour de médoïdes.

Construit sur ces fondations, l'algorithme de structuration que nous proposons dans ce manuscrit a pour but d'apporter des éléments de réponse aux points mentionnés précédemment, tout en capitalisant sur les avantages cités. En plus de ces points d'amélioration, un des aspects importants de cette contribution est de replacer cette méthode dans le contexte de la cognition et de l'interprétabilité afin de situer l'approche et son apport du point de vue de l'utilisateur. Le résultat sera donc étudié sous le spectre de l'interprétabilité, de l'explicabilité et des possibilités de raisonnement par analogie qu'il propose en tant que système d'exploration et d'aide à la décision.

3.1.2 Calcul de représentativité

L'attribution du *DoR* a pour finalité de lier chaque individu à son voisin le plus typique et digne d'intérêt parmi l'ensemble des individus. Dans un premier temps, le *DoR* agit donc comme une mesure de représentativité et de capacité à subsumer un groupe d'éléments similaires. Un parallèle peut être établit avec la *hubness* qui permet de retrouver des éléments centraux et proches des médoïdes proposés par un algorithme de *clustering* (Tomasev et al., 2014), les représentants détemrinés par le *DoR* sont des voisins « populaires » et centraux. Pour éviter les difficultés engendrées par le nombre de *features*,[1] les données sont traitées dimension par dimension avant d'agréger les résultats, contrairement à l'approche initiale décrite précédemment qui agrège les dimensions avec une distance unique. En plus de diminuer l'impact de la « malédiction de la dimensionnalité », analyser les dimensions séparément permet de quantifier l'impact de chacune sur le résultat final après l'agrégation. Ce choix favorise aussi le passage à l'échelle de cette méthode sur des volumes de données importants.

1. Pour rappel, nous utilisons indifféremment les mots variables, attributs, *features* ou dimensions pour parler des descripteurs d'un objet multidimensionnel

Exemple - Données

Tout au long de ce chapitre, des encarts vont illustrer chaque étape avec un exemple nous servant de fil conducteur. Le jeu de données sera composé de 5 éléments définis par 3 variables. Il ne s'agit pas d'un jeu de données complet mais d'un sous-ensemble utilisé à but illustratif, ainsi nous sortons du contexte de la haute dimensionnalité pour lequel cet algorithme a été créé, et les analyses n'ont qu'un but didactique.

Nous avons choisi d'utiliser des données issues du site *IMDb* [2]et extraites le 11 août 2019. L'exemple est constitué des informations relatives à quatre épisodes de la série télévisée [3]*Firefly* et au film [4]issu de la même franchise. Pour chacun de ces cinq éléments, nous avons relevé trois variables : la durée en minutes de l'épisode ou du film, la note et le nombre de critiques obtenues sur *IMDb*. Pour éviter toute ambiguité car l'épisode pilote de la série porte le même nom que le film, nous référons au premier par "Pilote" et au second par "Serenity".

	Note	Critiques	Durée
Pilote	8.9	4088	86
Train Job	8.5	3891	42
Out of Gas	9.3	4642	44
War Stories	8.8	3462	43
Serenity	7.9	271652	119

Relations entre éléments

Afin de pouvoir établir le score de représentativité de chaque individu il est nécessaire de pouvoir établir une relation entre tous les éléments de l'ensemble, deux à deux et dimension par dimension, pour pouvoir les comparer et *in fine* les classer. Nous allons dans cette partie poser le formalisme nécessaire à l'exposé de notre méthode.

On considère un ensemble de N objets $\mathscr{O} = \{o_1, o_2, \ldots, o_N\}$ sur lesquels on observe D attributs $\{a_1, a_2, \ldots, a_D\}$. Chaque attribut a_i ($i \in \{1, 2, \ldots D\}$) est une fonction de $\mathscr{O}$ dans E_i où E_i dépend du type de l'attribut. Un chapitre de Lerman (2016) y est dédié, dressant une typologie complète. En pratique, et dans de nombreux travaux, ces données sont souvent représentées par un tableau X dont :

— chaque ligne x_i, avec $i \in \{1, 2, \ldots N\}$, est l'observation des D attributs sur l'objet o_i,
— chaque colonne (ou variable) X_j, avec $j \in \{1, 2, \ldots D\}$, est l'observation de l'attribut a_j sur les objets de $\mathscr{O}$.

1. *www.imdb.com*, Internet Movie Database
2. *Firefly (TV Serie 2002-2003)*, Internet Movie Database
3. *Serenity (Movie 2005)*, Internet Movie Database

On notera $x_{i,j}$ ($i \in \{1,2,\dots N\}$, $j \in \{1,2,\dots D\}$) l'élément de la i^{me} ligne, j^{me} colonne, observation de l'attribut a_j sur l'objet o_i. On a donc :

$$\forall i \in \{1,2,\dots,N\}, \quad x_i = (a_1(o_i), a_2(o_i), \dots, a_D(o_i)) = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$$

$$\forall j \in \{1,2,\dots,D\}, \quad X_j = \begin{pmatrix} a_j(o_1) \\ a_j(o_2) \\ \vdots \\ a_j(o_N) \end{pmatrix} = \begin{pmatrix} x_{1,j} \\ x_{2,j} \\ \vdots \\ x_{N,j} \end{pmatrix}$$

On suppose maintenant que l'on dispose, pour chaque attribut, d'un indice de dissimilarité entre éléments permettant de les comparer. Notons $\{\delta_1, \delta_2, \dots, \delta_D\}$ cette famille d'indices. Les propriétés nécessaires sur ces indices sont faibles. En effet, nous allons utiliser ces indices pour ordonner les objets. Ainsi la propriété minimale est une trivialité :

$$\forall j \in \{1,2,\dots,D\}, \forall x,y,z \in \{1,2,\dots,N\},$$

$\delta_j(o_x, o_y) \leq \delta_j(o_x, o_z) \Leftrightarrow o_z$ est plus dissimilaire de o_x que o_y n'est dissimilaire de o_x

Exemple - Relations

Dans notre exemple les épisodes sont tous décrits par trois variables quantitatives, et il n'y a pas de valeurs manquantes. Nous pouvons donc choisir comme indice de dissimilarité pour chaque attribut la distance Euclidienne entre les éléments, dont la formule est rappelée dans la Table 2.2. Nous allons illustrer le calcul de dissimilarité sur la première dimension (*Note*), mais la dissimilarité est calculée sur chacune des trois dimensions.

Note		Pilote	Train job	Out of Gas	War Stories	Serenity
8.9	Pilote	0	0.4	0.4	0.1	1
8.5	Train Job	0.4	0	0.8	0.3	0.6
9.3	Out of Gas	0.4	0.8	0	0.5	0.4
8.8	War Stories	0.1	0.3	0.5	0	0.9
7.9	Serenity	1	0.6	0.4	0.9	0

Cet indice de dissimilarité peut être différent sur chaque dimension. Il est ainsi possible de l'adapter au type d'attributs, selon qu'ils soient quantitatifs ou qualitatifs par exemple. Cela permet, au fur et à mesure de l'analyse exploratoire et des connaissances acquises, d'adapter les relations pour chaque *feature* et selon les informations recherchées et la structure souhaitée.

C'est aussi par ce biais que l'expert a la possibilité de valoriser, s'il en dispose, ses connaissances préalables sur les données. Par defaut une distance conviendra tout à fait, cependant les propriétés caractéristiques des métriques ne sont pas toutes requises : seule la notion d'ordre est utilisée.

Classement

Après avoir défini ces relations de dissimilarité entre tous les éléments étudiés, nous les utilisons pour classer les éléments. Nous souhaitons obtenir, sur chacune des D dimensions, N classements relatifs. Pour chaque attribut a_j ($j \in \{1,2,\ldots,D\}$), et pour chaque objet o_i ($i \in \{1,2,\ldots,N\}$), on calcule les $\delta_j(o_i,o_1), \delta_j(o_i,o_2), \ldots, \delta_j(o_i,o_N)$.

Le tri croissant de ces valeurs permet alors de déterminer les rangs de chaque objet, en fonction de leur dissemblance avec o_x selon l'attribut a_j. On notera $Rank^j_{o_x}(o_y)$ ($x,y \in \{1,2,\ldots,N-1\}$) le rang de o_y en fonction de sa dissimilarité avec o_x selon l'attribut a_j. Par commodité et pour refléter les conventions le plus souvent adoptées, nous ne considérons pas un élément comme membre de son propre voisinage : il est donc à noter qu'un élément ne s'attribue pas de classement lui-même.

En appliquant cela à tous les objets et toutes les dimensions, nous obtenons $D \times N$ classements. Ce choix du rang plutôt que de la distance comme base pour l'attribution du score diminue l'impact des *outliers* tout en évitant de les écarter lors de cette étape, permettant de les inclure dans les possibilités d'exploration proposées par l'algorithme. Ces techniques de transformations en rangs sont une approche classique (Conover & Iman, 2012) notamment en statistiques robustes (Hampel, Ronchetti, Rousseeuw, & Stahel, 2011).

Exemple - Classement

Chaque matrice de dissimilarité est transformée en un classement, où chaque élément ordonne les autres par dissimilarité croissante. Dans cet exemple, les égalités sont réglées en utilisant le rang moyen des éléments. Pour la dimension *Note*, nous obtenons les classements suivants pour les épisodes. Les lignes correspondent aux scores attribués par cet épisode et les colonnes aux scores reçus par cet épisode, ainsi la symétrie de la matrice de dissimilarité disparaît.

	Pilote	Train job	Out of Gas	War Stories	Serenity
Pilote	-	2.5	2.5	1	4
Train Job	2	-	4	1	3
Out of Gas	1.5	4	-	3	1.5
War Stories	1	2	3	-	4
Serenity	4	2	1	3	-

Vote

Il reste maintenant à transformer ces classements en scores que l'on agrègera pour obtenir D scores par élément, correspondant à leur représentativité sur chaque dimension. Pour cela nous nous plaçons dans la théorie du choix social, où nous pouvons par exemple utiliser la méthode de Borda (1781). Nous considérons que les éléments vont voter entre eux pour s'attribuer un score selon les classements (Blanchard, Runz, Akdag, & Herbin, 2011) : avec N objets, chacun d'entre eux transforme les classements qu'il a attribués en scores relatifs. De cette manière, pour un objet o_x, les scores relatifs à cet objet selon l'attribut a_j seront exprimés comme :

$$\forall j \in D, \forall o_x, o_y \in \mathcal{O}, \quad Score^j_{o_x}(o_y) = N - Rank^j_{o_x}(o_y)$$

Dans les travaux de ce manuscrit nous choisissons, par simplicité, d'utiliser une variante de la méthode de Borda où un score n'est attribué qu'aux plus proches voisins de l'élément sur la dimension considérée. Pour deux objets o_x et o_y et $Tnn^j(o_x)$ l'ensemble des T plus proches voisins de o_x sur la dimension j, alors :

$$Score^j_{o_x}(o_y) = \begin{cases} T - Rank^j_{o_x}(o_y) + 1, & \text{si } o_y \in Tnn^j(o_x) \\ 0, & \text{sinon} \end{cases}$$

Les T plus proches voisins de o_x obtiennent donc un score $1 \leq Score^j_{o_x}(o_y) \leq T$. Si la valeur choisie pour T impacte les scores obtenus avant agrégation (et donc le score obtenu après leur agrégation), cela n'a que peu d'influence sur la structure finale créée par l'algorithme.

Pour cette raison, nous privilégions la lisibilité des résultats en fixant $T = 10$ pour conserver des scores bas et lisibles. Comme la figure 3.1 l'illustre, si l'utilisateur possède des connaissances particulières sur la distribution des données sur certaines dimensions, il est alors possible d'utiliser une fonction de scoring adaptée à cette distribution et aux relations entre les données.

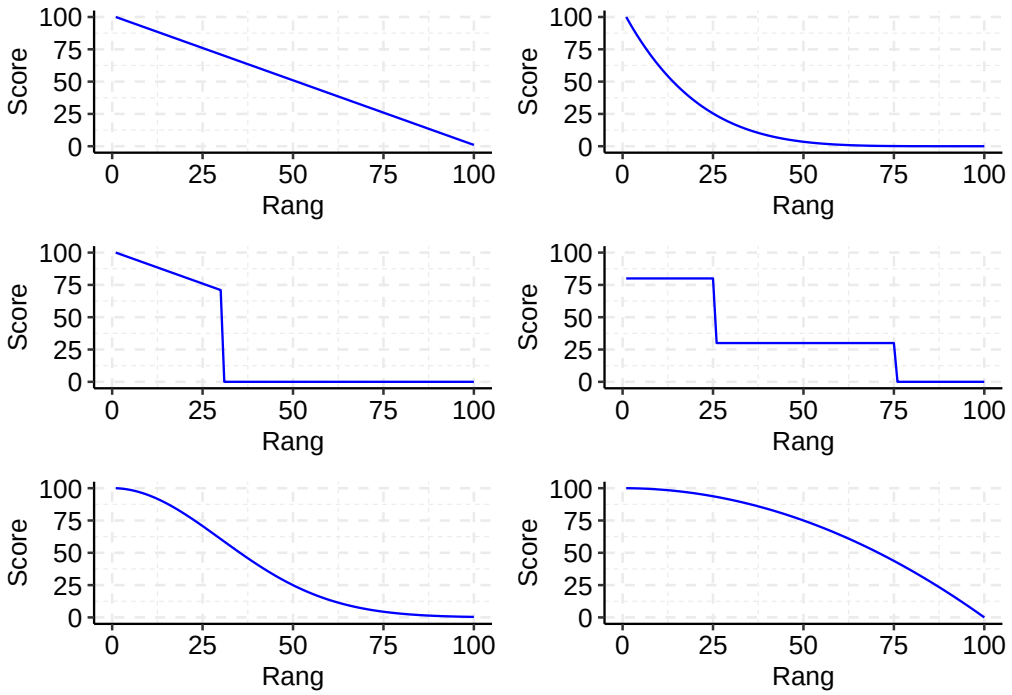


FIGURE 3.1 – Plusieurs exemples de fonctions envisageables pour transformer les rangs en score, qui ont pour seul prérequis d'être décroissantes. Le but de cette flexibilité est de pouvoir facilement intégrer les connaissances de l'expert ou celles sur les données et leur distribution, au fil de l'exploration et de la manipulation du dataset.

Exemple - Score

Nous transformons, à cette étape, les classements obtenus en scores de similarité. Nous sommes libres d'utiliser la formule de transformation adaptée à nos données et même à chaque dimension, et pour cela nous allons adapter la formule proposée précédemment en fixant $T = 3$, obtenant ainsi des scores allant de 3 pour les éléments classés premiers, à 0 pour le quatrième et dernier élément de chaque classement.

	Pilote	Train job	Out of Gas	War Stories	Serenity
Pilote	-	1.5	1.5	3	0
Train Job	2	-	0	3	1
Out of Gas	2.5	0	-	1	2.5
War Stories	3	2	1	-	0
Serenity	0	2	3	1	-

La dernière étape est celle du calcul du score de représentativité sur chaque dimension, par agrégation des N scores reçus sur la dimension. par défaut, nous utilisons la somme comme opérateur d'agrégation. Soit un invididu o_y, sa représentativité sur la dimension j sera calculée par :

$$Score^j(o_y) = \sum_{x \in \mathscr{O}} Score^j_{o_x}(o_y)$$

À cette étape, nous disposons donc d'un score, appelé $Score^j$, qui quantifie la représentativité de chaque élément du jeu de données sur une dimension. Il ne reste désormais qu'à calculer le *DoR* des éléments afin de pouvoir les associer pour obtenir une structuration, le cacul du *DoR* se basant sur le $Score^j$ défini précédemment.

Exemple - Score agrégé

Il ne reste plus qu'à faire la somme des scores obtenus par un élément, dimension par dimension, pour obtenir son $Score^j$ sur chaque dimension. Ainsi, chaque épisode possède désormais un score pour *Note*, un pour *Critiques* et un pour *Durée*, indiquant leur représentativité sur chaque variable.
Le score des épisodes pour la dimension *Note* est le suivant :

Pilote	Train job	Out of Gas	War Stories	Serenity
7.5	5.5	5.5	8	3.5

Si les données étaient assez représentatives pour le permettre, nous pourrions déduire du score de la dimension *Note* que le film *Serenity* possède la note la moins représentative de la série (7.9), alors que l'épisode le plus représentatif est *War Stories* avec 8.8. Ces observations, bien qu'à but didactique, sont cohérentes avec la note globale de 9 reçue par *Firefly* sur *IMDb* au moment où est rédigé ce manuscrit.

3.2 Structuration

Afin d'enrichir les données en les structurant, nous utilisons le score calculé précédemment pour chaque individu et dimension afin de lier tout individu du jeu de données à son représentant. « Représentant » est le terme que nous employons pour identifier l'élément choisi par chaque individu pour le subsumer. Il s'agit d'un individu, typique et générique, désigné par « *exemplar* » en *exemplar theory*. En français nous lui préférerons « représentant » afin d'éviter toute ambiguïté avec les traductions d'*exemplar*. Cette structuration autour des représentants a pour objectif de guider l'exploration et de proposer à l'utilisateur des recommandations de voisins pertinents.

3.2.1 Construction du graphe

Pour pouvoir associer les individus deux à deux selon leur similarité, il faut agréger les scores obtenus par les éléments sur chaque dimension. Considérons deux objets o_x et o_y appartenant à l'ensemble $\mathscr{O}$ composé de N objets, et le facteur k tel que $1 \leq k < N$. Nous proposons le calcul $DoR_{o_x}(o_y)$ du DoR de o_y relativement à o_x comme étant la somme des $Score^j(o_y)$ calculés précédemment, sur chaque dimension sur laquelle o_y figure parmi les k plus proches voisins de o_x :

$$\forall o_x, o_y \in \mathscr{O}, \quad \Delta = \{d \in \{1,2,\ldots,D\} / o_y \in Knn^d(o_x)\},$$
$$DoR_{o_x}(o_y) = \sum_{j \in \Delta} Score^j(o_y)$$

où Δ est l'ensemble des dimensions où o_y figure dans les k plus proches voisins de o_x. Ainsi le $DoR_{o_x}(o_y)$ exprime à quel point l'élément o_x considère l'élément o_y comme à même de le représenter. Le représentant choisi pour un individu o_x est l'individu avec le DoR_{o_x} le plus élevé, soit le meilleur représentant. Chaque individu obtient donc un et un seul représentant :

$$\forall o_x \in \mathscr{O}, \quad Rep(o_x) = \underset{o_y \in \mathscr{O}}{\arg\max}\, DoR_{o_x}(o_y)$$

Rappelons d'abord qu'un graphe est une représentation d'une relation binaire sur un ensemble d'objets. Un graphe est constitué de sommets qui représentent les objets, et d'arcs représentant la relation entre les couples d'objets. Nous utilisons dans notre cas un graphe orienté : les relations entre sommets ne sont pas nécessairement symétriques. Un graphe orienté est dit connexe si, dans le graphe non-orienté lui correspondant, il existe une chaîne entre tout couple de sommets. Enfin, une composante connexe est un sous-graphe connexe maximal, c'est-à-dire un sous-graphe connexe qui n'est inclus dans aucun autre sous-graphe connexe. Afin d'obtenir une structuration en graphe, chaque élément sera ainsi lié à son représentant par un arc. Ainsi nous créons un graphe orienté (V,E) où V correspond à l'ensemble des sommets associés aux objets, soit $V = \mathscr{O}$, et E l'ensemble des arcs $(o_x, Rep(o_x))$ pour tout élément de $\mathscr{O}$. Il est possible pour un unique individu de subsumer plusieurs autres individus l'ayant choisi comme représentant. Chaque composante connexe du graphe obtenu peut être résumée par la liste de ses représentants, ou même par un unique représentant subsumant le plus d'individus.

Nous avons donc pu structurer les éléments sous la forme d'un graphe. L'Algorithme 1 résume la méthode telle que détaillée depuis le début du chapitre, proposant une structuration basée sur le choix de représentants d'après le calcul de leur DoR, obtenu en agrégeant les scores des éléments. Avec un temps d'exécution $\mathscr{O}(n^2)$ dépendant du nombre d'éléments à structurer, l'algorithme ne peut s'exécuter en temps interactif que sur de petits volumes de données. Pour des jeux de données plus grands, cette complexité quadratique peut être en partie gérée par une implémentation adaptée de l'algorithme. En effet, chaque dimension étant traitée séparément, il est envisageable d'effectuer les traitements en parallèle sur plusieurs coeurs du processeur ou sur plusieurs noeuds de calcul afin d'avoir un nombre raisonnable de données sur chaque noeud.

Data : N elements defined on D dimensions, neighborhood size K
Result : list of N exemplars
foreach *dimension* j *in* D **do**
  Compute dissimilarity matrix of elements;
  Transform similarities into ranks;
  Convert ranks into scores with arbitrary scoring function;
  foreach *element* y *in* N **do**
    $Score^j(y) \longleftarrow \sum_{x \in \{1,2,\ldots,N\}} Score_x^j(y)$ given by x to y;
  end
end
foreach *element* y *in* N **do**
  foreach *element* x *in* N **do**
    $DoR_y(x) \longleftarrow 0$;
    foreach *dimension* j *in* D **do**
      $Knn^j(y) \longleftarrow$ *K-nearest neighbors of y, on dimension* j;
      if $x \in Knn^j(y)$ **then**
        $DoR_y(x) \longleftarrow DoR_y(x) + Score^j(x)$;
      end
    end
  end
  $exemplar(y) \longleftarrow \underset{x \in \{1,2,\ldots,N\}}{\arg\max} (DoR_y(x))$;
end
Result $\longleftarrow$ list of exemplars determined above;

Algorithme 1 : Structuration des éléments

Exemple - Calcul du DoR

Structurons maintenant les cinq épisodes présentés plus tôt. Nous avons détaillé le calcul de leur score sur la dimension *Note* comprenant les notes données par les critiques. Si nous appliquons cette méthode aux deux autres dimensions, nous obtenons les scores suivants, pour chaque individu et chaque dimension :

	Note	Critiques	Durée
Pilote	7.5	10	6
Train Job	5.5	9	5.5
Out of Gas	5.5	7	9.5
War Stories	8	4	9
Serenity	3.5	0	0

Afin de structurer les individus par représentativité, il est nécessaire de calculer le *DoR* qu'ils s'attribuent entre eux car chaque élément sera relié à l'élément auquel il accorde le *DoR* le plus élevé. Décidons de $k = 2$, c'est-à-dire que le score d'un épisode sur une dimension ne sera pris en compte dans l'agrégation que s'il est dans les 2 plus proches voisins, sur cette dimension, de l'épisode attribuant le *DoR* Par exemple lorsque l'épisode *Serenity* calcule le *DoR* qu'il attribue à l'épisode *Train Job*, il ne va prendre en compte que son score pour la dimension *Note* car sur les deux autres dimensions *Train Job* n'est pas dans les trois plus proches voisins de *Serenity*. L'illustration visuelle de la structuration est disponible dans la sous-section suivante, mais le résultat du calcul du *DoR* est le suivant, tel que sur une ligne figurent les *DoR* attribués par l'élément et sur une colonnes ceux qu'il a reçu :

	Pilote	Train job	Out of Gas	War Stories	Serenity
Pilote	-	9	16.5	**17**	0
Train Job	17.5	-	9.5	**21**	0
Out of Gas	**17.5**	14.5	-	9	3.5
War Stories	17.5	**20**	9.5	-	0
Serenity	16	5.5	**22**	0	-

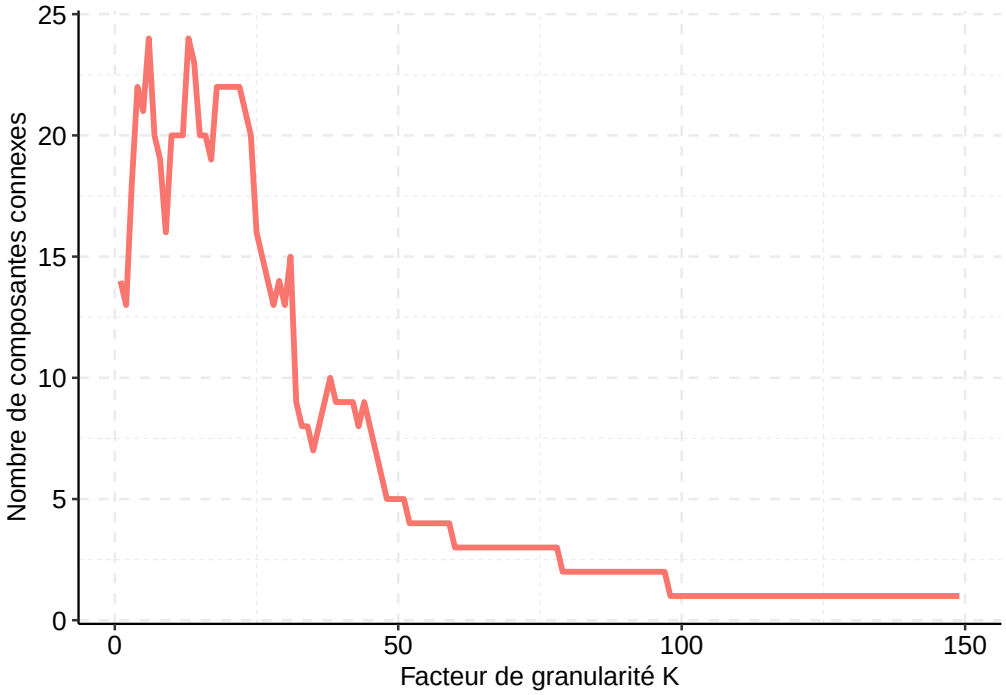


FIGURE 3.2 – Évolution du nombre de composantes connexes selon la valeur choisie pour le facteur de granularité k, lors de la structuration du jeu de données Iris.

3.2.2 Choix du facteur k

L'algorithme créant les connexions entre les éléments, deux à deux, est paramétré par k qui définit la taille du voisinage. Le graphe obtenu à la fin de l'étape précédente est donc dépendant de ce paramètre. Celui-ci agit comme un facteur de granularité affectant le *DoR*, et donc le choix du représentant de chaque élément. Ce facteur k influence le nombre total de représentants qui résument le jeu de données. Selon les informations qui sont recherchées il existe plusieurs possibilités lors de l'analyse exploratoire, selon la valeur utilisée pour k. Une faible valeur pour la taille de voisinage k permet d'obtenir une forte granularité avec de nombreux représentants, correspondant fortement aux éléments qu'ils subsument. Cette approche engendre un nombre élevé de composantes connexes, l'exploration des données en suivant les arcs du graphe est donc limitée, mais seuls les éléments très similaires sont groupés. À l'opposé, une valeur élevée pour le paramètre k fournit un petit nombre de réprésentants qui résument chacun un sous-ensemble significatif de la population étudiée. Ces représentants fournissent donc une vision typique mais condensée des données étudiées, et l'exploration est facilitée par le faible nombre de composantes connexes, permettant la découverte de nombreux éléments similaires pour chaque individu. La figure 3.2 illustre le nombre de composantes connexes obtenues selon la valeur choisie pour le paramètre k, sur le jeu de données *Iris*, fournissant un aperçu des granularités offertes par l'approche proposée.

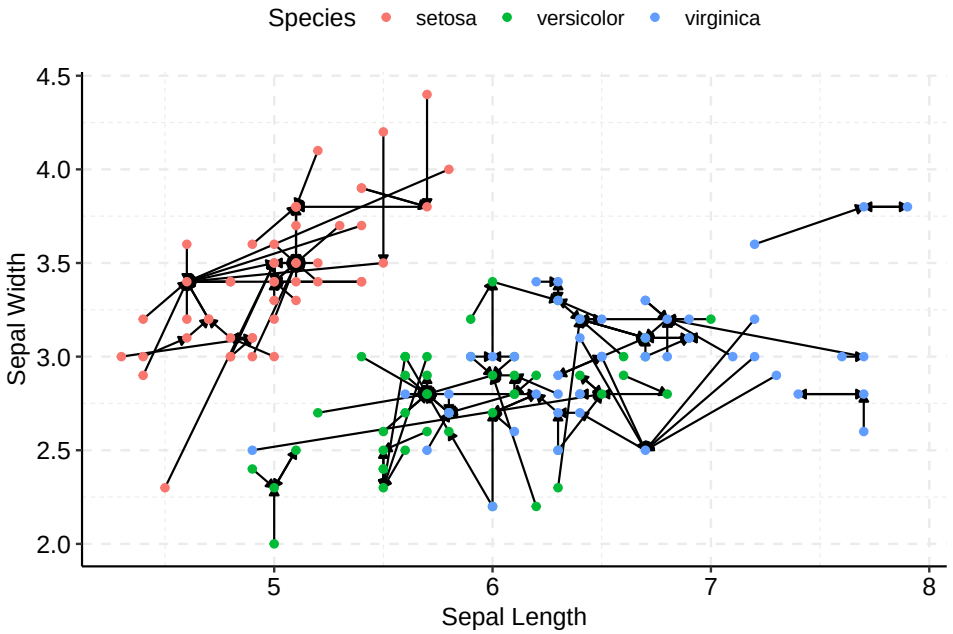


FIGURE 3.3 – Structure résultant de l'application au jeu de données Iris avec un voisinage k égal à 30. Chaque arc représente l'association d'un individu à son représentant. De nombreux représentants émergent, tous très similaires aux éléments qu'ils représentent.

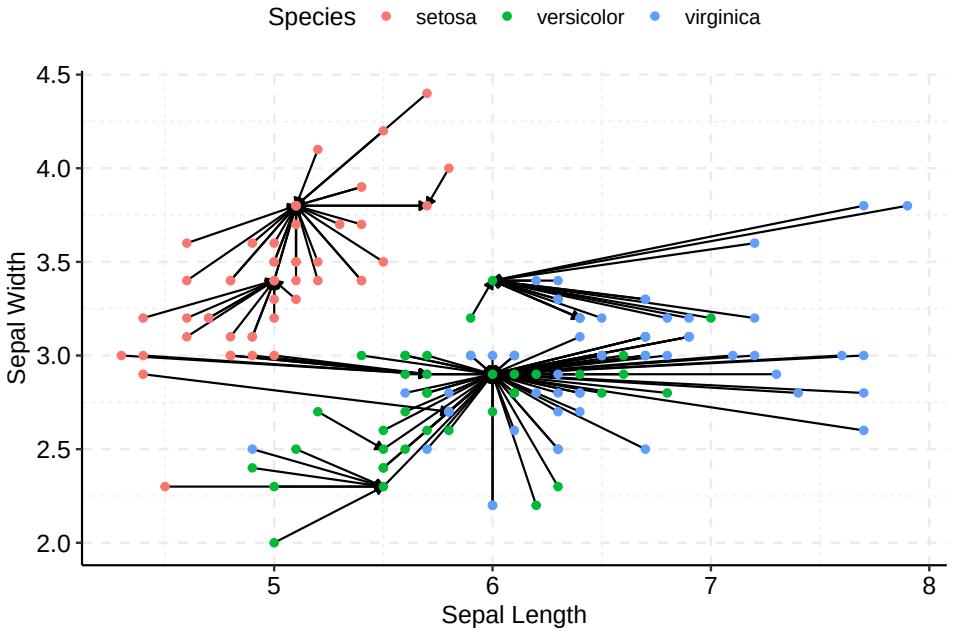


FIGURE 3.4 – Structure résultant de l'application au jeu de données Iris avec un voisinage k égal à 75. Les arcs représentent les associations entre éléments similaires, d'un élément vers son représentant. Les données sont résumées par quelques représentants génériques et typiques. Malgré la faible lisibilité des flèches, il est aisé d'identifier visuellement les éléments les plus représentatifs.

Les figures 3.3 et 3.4 illustrent, sur le jeu de données *Iris*, ces différences de granularité selon le paramètre k. Les deux valeurs de k utilisées pour ces figures ont été choisies empiriquement pour refléter différentes structurations possibles. La première figure utilise un voisinage de taille 30 et se compose de 55 représentants, avec un nombre moyen de 3 éléments subsumés par chaque représentant. Sur la seconde figure le paramètre retenu pour le voisinage est $k = 75$, ce qui donne 11 représentants, regroupant chacun en moyenne 14 individus similaires. Chacun des deux graphes peut être utilisé comme outil de recommandation, avec un représentant similaire mais plus typique et générique désigné pour chaque individu. C'est aussi un outil de visualisation pouvant guider l'utilisateur lors de l'exploration du jeu de données. Chaque arc du graphe correspond à un ensemble de *features* ayant entraîné sa création, il est ainsi possible d'analyser les causes d'une association entre deux éléments et la nature de leur similarité. La figure 3.5 reprend sous forme de graphes les associations présentées dans la figure 3.4, avec un paramètre de granularité $k = 75$. Les 11 représentants figurent en rouge et permettent une description des données avec un nombre restreint d'exemples issus du *dataset*. Une autre approche possible consiste à ne pas fixer la taille utilisée pour le voisinage, mais à étudier l'évolution du graphe lorsque k varie. En conservant un ensemble de N éléments, cela implique de structurer les éléments selon chaque valeur $1 \leq k < N$. De cette manière, nous pouvons établir d'autres mesures comme le nombre moyen d'éléments subsumés par chaque représentant, ou le nombre total d'itérations lors desquelles un élément a été choisi comme représentant (illustré sur la Figure 3.6) afin d'obtenir différents critères d'évaluation de l'importance d'un élément.

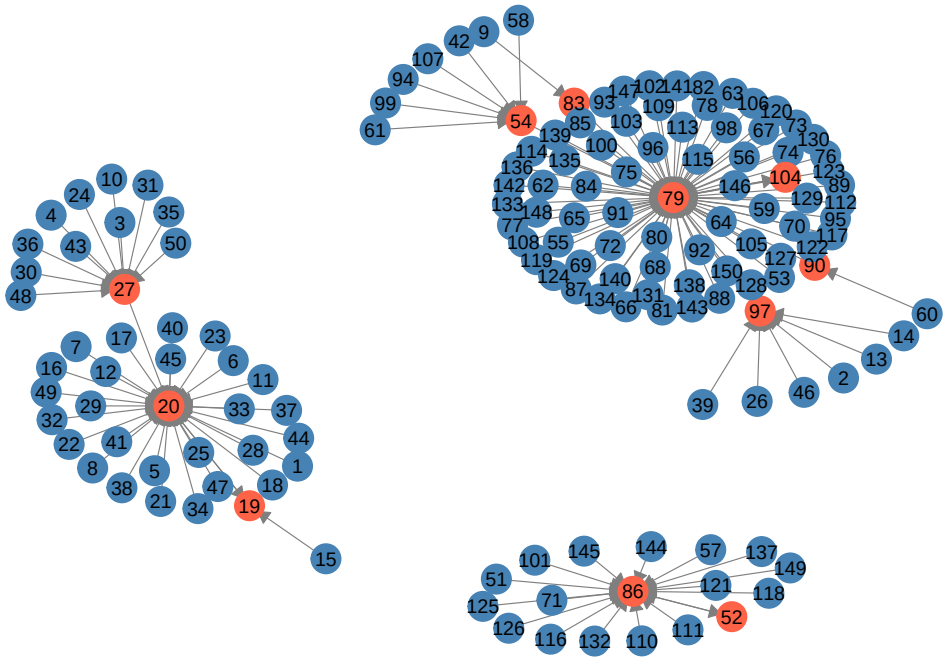
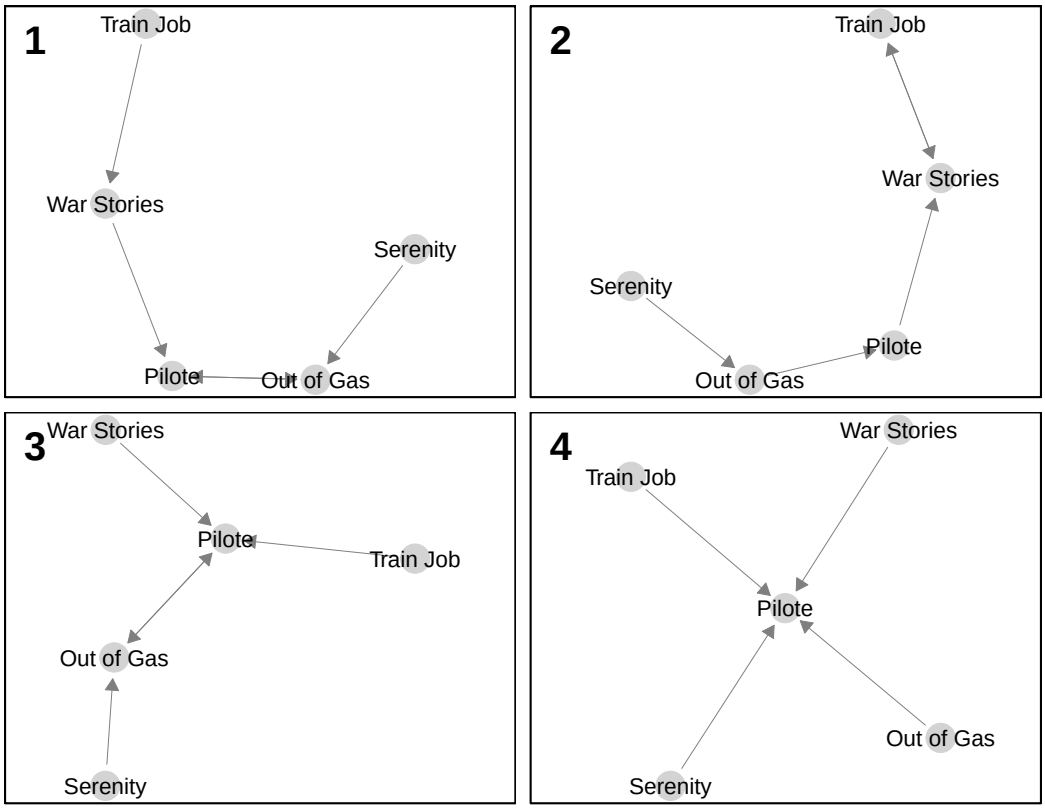


FIGURE 3.5 – Graphe illustrant les associations par similarité entre les éléments du jeu de données Iris, avec un voisinage k = 75. Les 11 représentants sont visibles en rouge et résument le jeu de données. Il est possible d'explorer les données en naviguant entre éléments similaires d'une même composante connexe.

Exemple - Choix de K

Pour le calcul du *DoR* dans l'exemple précédent nous avons choisi $k = 2$, toutefois nous aurions pu choisir toute valeur comprise entre 1 et 4. La figure ci-dessous illustre les structurations obtenues pour toute les valeurs possibles de k : nous pouvons par exemple y voir que le pilote de la série semble accomplir son rôle en étant, dans la majorité des cas, l'élément le plus représentatif des cinq épisodes choisis.



3.3 Propriétés

Nous avons défini, dans la section 2.3.2, plusieurs critères d'évaluation de l'interprétabilité d'un algorithme, ces critères étant répartis sur deux axes : la transparence et l'explicabilité. Après avoir décrit la méthode proposée et illustré le type de résultats obtenus, procédons à une analyse selon les critères d'interprétabilité évoqués précédemment afin de constater leur adéquation avec les objectifs établis pour ce travail.

3.3.1 Transparence

La notion de transparence de l'algorithme se décompose en trois facteurs, chacun décrivant un niveau d'abstraction différent de l'algorithme. Ces critères sont la simulabilité, l'intelligibilité et la prédictibilité, chacune des trois sous-sections suivantes correspondant à un de ces éléments.

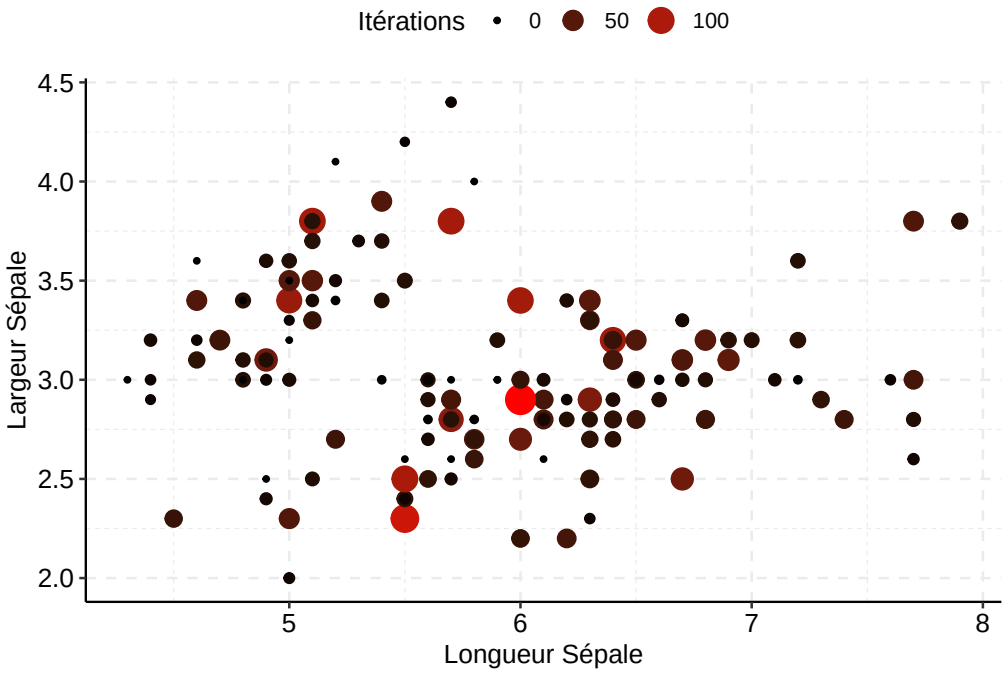


FIGURE 3.6 – Estimation de la capacité des individus du jeu de données iris à subsumer les données, mesurée comme le nombre d'itérations où chaque élément est choisi comme représentant par au moins un autre élément.

Simulabilité

Au niveau macroscopique, la simulabilité correspond à la capacité de comprendre le fonctionnement global de l'algorithme. L'approche que nous proposons est composée de peu d'étapes, et est facilement explicable dans sa globalité car reposant sur la similarité pour créer des associations. Même si ces éléments permettent de tendre vers une bonne simulabilité, la nature même des données traitées vient limiter cette aspect de la transparence : avec un algorithme conçu pour des données en haute dimensionnalité, la simulation étape par étape n'est pas concevable sur des données réelles et non plus didactiques. Ainsi, si le modèle proposé peut être considéré comme simulable de par sa simplicité et sa facilité de mise en oeuvre, il ne l'est que partiellement du fait que son exécution n'est plus aisée en grande dimension.

Intelligibilité

L'intelligibilité se place à l'inverse au niveau microscopique : c'est la capacité à décomposer en opérations élémentaires et à comprendre chacune d'elles. Les entrées tout d'abord sont interprétables car les *features* initiales ne sont pas modifiées et les éléments sont donc décrits par les mêmes variables familières à l'utilisateur. Un seul paramètre est à définir et celui-ci prend une valeur numérique entière, positive et inférieure au nom d'individus composant le jeu de données. De plus, l'impact du facteur de granularité sur le résultat est identifiable facilement visuellement. Enfin, les opérations élémentaires exécutées à chaque étape de l'algorithme sont individuellement très simples (classement, transformation des rangs en scores, agrégation des scores selon les dimensions) et leur

signification peut se traduire de manière concrète pour l'utilisateur : par exemple, la somme des $DoR^d(y)$ uniquement sur les dimensions où y est dans le voisinage de x pourrait se traduire par « l'importance de l'individu y selon cette variable ne m'intéresse que si y est similaire à x sur cette même variable ». La méthode étant *instance-based*, les résultats sont toujours des éléments du jeu de données initial et ne demandent donc pas d'effort d'instanciation de la part de l'utilisateur pour les interpréter.

Prédictibilité

La prédictibilité est la capacité à anticiper les résultats fournis. Pour la méthode proposée, le choix du facteur de granularité n'est pas forcément anticipable pour des données qui n'ont pas été rencontrées précédemment. Toutefois, lors de l'expérimentation avec les données, il est possible de déterminer empiriquement les valeurs appropriées pour ce paramètre, et l'évolution de la structure finale est, elle, tout à fait prévisible. De plus, l'approche est déterministe et peut donc être utilisée de manière répétée sur des mêmes données en sachant que la structure restera identique ce qui la rend facilement reproductible. C'est un facteur important pour des analyses médicales pour lesquelles patients et médecins peuvent souhaiter vérifier ou expliquer un résultat, dans quels cas il est nécessaire de pouvoir le reproduire. De par l'utilisation des rangs, les associations entre éléments sont aussi très peu sensibles aux *outliers* et valeurs aberrantes.

3.3.2 Explicabilité

L'explicabilité est présente à deux points de l'analyse. Tout d'abord les associations entre éléments peuvent être détaillées à l'utilisateur pour lui décrire (de manière textuelle et/ou graphique) l'impact de chacune des variables sur le choix du représentant pour chaque individu. La figure 3.7 est un exemple illustrant l'importance de chaque dimension sur la relation associant l'élément 5 à son représentant 20 sur la figure 3.5, le représentant ayant été choisi d'abord pour sa typicalité sur la dimension *Sepal Width*. Dans un second temps, il est possible d'étayer la structure choisie en laissant l'utilisateur la parcourir de proche en proche afin d'étudier les *patterns* qui en émergent dans chaque composante connexe. C'est cette fois-ci une explication par l'exemple qui est mise en œuvre : il est possible de mieux appréhender le processus de structuration en comparant les cas qui ont été jugés similaires. Le raisonnement par analogie est ici mis en avant, permettant à l'utilisateur de comprendre par l'exemple.

En résumé, nous avons présenté le *DoR* comme moyen de quantifier la représentativité des éléments d'un ensemble. Ce calcul du *DoR* est adapté aux hautes dimensions, pour lesquelles il est possible de sélectionner des mesures de dissimilarité appropriées à chaque dimesion. En se basant sur cette mesure, il est possible d'obtenir une structuration des éléments qui se prête à l'exploration, où chaque élément est rattaché à un représentant similaire mais plus générique que lui. La granularité de cette approche peut être contrôlée par un unique paramètre, et les résultats fournis sont interprétables.

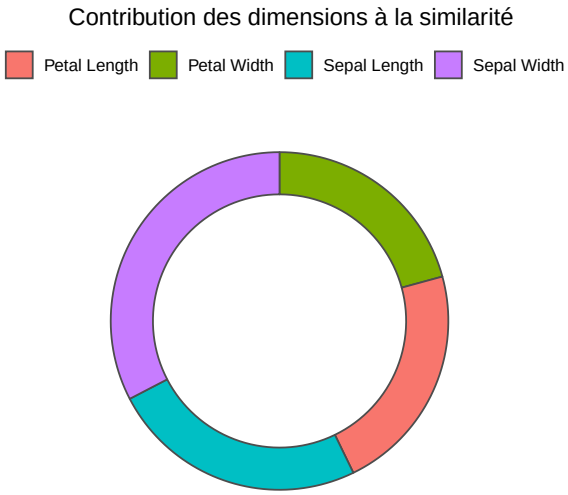


FIGURE 3.7 – Exemple d'explication visuelle de la contribution de chacune des dimensions à la similarité entre l'élément 5 et son représentant (élément 20), dans le jeu de données Iris. Un voisinage de taille k = 75 a été utilisé.

3.4 Détection d'anomalies

Peu après avoir défini pour la première fois le concept de *Degree of Representativeness* tel qu'utilisé dans les travaux présentés dans ce manuscrit, nous avons tenté d'adapter cette approche à la détection d'anomalies en haute dimensionnalité dans le but de l'intégrer au prototype développé. Dans le cadre médical, ce que l'on nomme « détection d'anomalies » correspond ici à la détection de cas rares et inhabituels. En se basant sur des travaux récents (Herbin, Gillard, & Hussenet, 2017), nous avons défini un concept de rareté dénommé *Rareness* et permettant de quantifier la propension d'un individu à pouvoir être considéré comme une anomalie, ou *outlier*. La méthode créée a été comparée aux approches faisant figure d'état de l'art dans le domaine de la recherche d'anomalie en haute dimensionnalité, mais dans le cas général n'apporte pas d'avantage substantiel vis-à-vis de l'existant. Nous la présentons toutefois pour ses propriétés d'explicabilité et afin d'ouvrir de nouvelles perspectives reposant sur ce paradigme de *Rareness*.

Constatant l'efficacité d'une approche composée d'opérations unidimensionnelles qui sont ensuite agrégées pour obtenir un résultat, nous avons souhaité étendre cette approche à d'autres problématiques. Le *DoR* propose en effet d'excellents résultats, et il nous a semblé opportun de mettre à profit les leçons que nous en avons retirées en les appliquant à la détection d'anomalies. La détection d'anomalies est une opération courante lors du pré-traitement automatique des données mais aussi lors de l'analyse d'un flux, comme en cybersécurité, pour y déceler des éléments anormaux et lever des alertes. Elle joue un rôle similaire en médecine où elle peut permettre d'alerter le personnel soignant d'un état anormal du patient par exemple.

Nous avons donc proposé le concept de *Rareness*, une mesure d'atypicité (en faisant d'une certaine manière le complètement du *DoR*). Pour détailler le calcul du *DoR*,

prenons un jeu de données Ω composé de n éléments définis sur D dimensions. Soit $Knn^d(y)$ l'ensemble des K plus proches voisins de l'élément y sur la dimensions $d \in D$, et soit $Rank_x^d(y)$ le rang de l'élément y par rapport à l'élément x, sur la dimension d. Pour calculer $Rank_x^d(y)$, l'élément x ordonne tous les autres éléments de Ω par proximité décroissante selon la dimension d. Pour un élément x, l'élément le plus proche (celui avec le rang le plus bas) obtient donc un rang $Rank_x^d(y) = 1$. Pour rappel, cette notion de proximité s'appuie sur un indice de dissimilarité, qui peut être différent pour chacune des dimensions étudiées, tant qu'il permet de comparer les individus deux à deux selon une dimension. Dans la suite, cet indice est généralement la distance, c'est-à-dire la valeur absolue de la différence des deux valeurs à comparer.

La *Rareness* locale d'un élément y est défini par rapport à un autre élément x, pour un voisinage de taille K et sur une dimension d, de la manière suivante :

$$Rareness_x^d(y) = \frac{1}{K} \cdot min(Rank_x^d(y), K)$$

La $Rareness_x^d(y)$ correspond au fait que si y est l'un des K plus proches voisins de x dans cette dimension d, $Rank_x^d(y)$ est alors inférieur à K et la *Rareness* de y dans cette dimension est inférieur à 1. Au contraire, si y n'est pas voisin de x, alors $Rank_x^d(y)$ est supérieur à K et la *Rareness* de y par rapport à x sera égale à 1. S'il est possible de calculer la *Rareness* par rapport à un élément, il est aussi possible de le faire par rapport à un ensemble d'éléments. Prenons les K plus proches voisins de y sur la dimension e : il est possible de calculer la *Rareness* de y par rapport à cet ensemble, sur la dimension d, comme étant la moyenne de la $Rareness_x^d(y)$ pour tout $x \in Knn^e(y)$, soit :

$$Rareness_{Knn^e(y)}^d(y) = \frac{1}{K} \cdot \sum_{x \in Knn^e(y)} Rareness_x^d(y)$$

Ainsi, la $Rareness_{Knn^e(y)}^d(y)$ est égale à 1 si, sur la dimension d, y ne figure plus dans le voisinage d'aucun des voisins qu'il avait sur la dimension e. Maintenant que nous avons défini la *Rareness* sur une première dimension par rapport à un ensemble de voisins sur une seconde dimension, nous pouvons définir la *Rareness* global d'un élément y : le maximum de sa *Rareness* locale pour chacun des couples de dimensions possibles :

$$Rareness(y) = \max_{\substack{d \in D \\ e \in D}} \{Rareness_{Knn^e(y)}^d(y)\}$$

Chaque élément se voit affecter une *Rareness* inférieure ou égale à 1 : nous considérons comme anomalie tout élément dont la *Rareness* globale est égale à 1, et tout élément avec une *Rareness* élevée comme étant un élément atypique. En combinant les dimensions deux à deux, nous avons la possibilité de détecter des anomalies qui appartiennent pourtant à la distribution des données sur chaque dimension prise séparément. Par exemple, sur la Figure 3.8 qui montre la répartition d'un millier de points générés selon des distributions uniformes sur 6 dimensions, les points 999 et 1000 sont au coeur de la distribution sur la majorité des dimensions, et sont donc indétectables

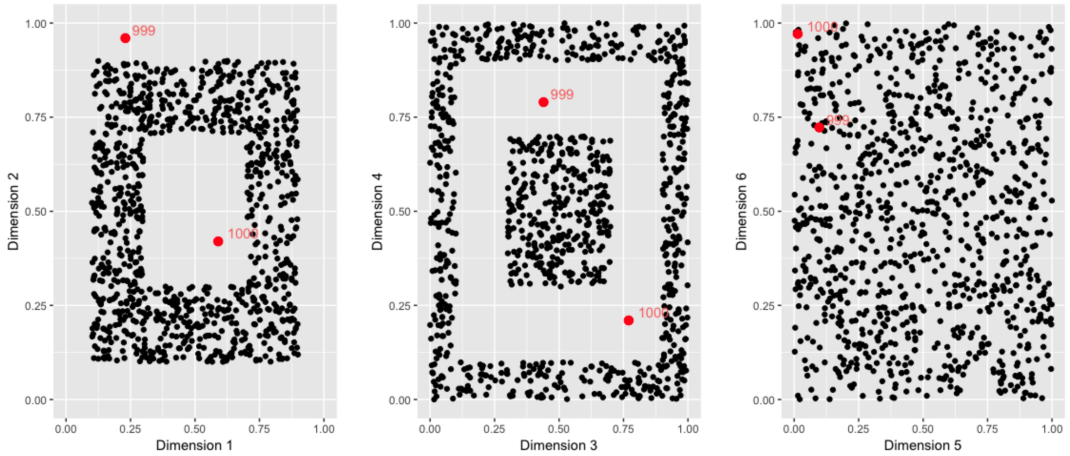


FIGURE 3.8 – Ensemble de 1000 points générés par des distributions uniformes, sur 6 dimensions. Les deux anomalies, en rouge, sont comprises au sein des distributions sur la majorité (point 999) ou la totalité (point 1000) des dimensions. Il faut donc considérer plusieurs dimensions simultanément pour identifier ces anomalies.

si l'on travaille dimension par dimension. Malgré cela, lorsque la bonne projection est observée, ces points apparaîssent immédiatement comme *outliers*, c'est pour cela que le couple de dimension d et e revêt une importance particulière. En plus de permettre d'identifier des anomalies non triviales qui sont indiscernables sur une seule dimension, ils font figure de facteurs explicatifs en permettant de pointer le couple de dimensions sur lesquelles l'individu est un anomalie. Dans un cadre médical, il est intéressant de détecter ces individus pour assister le personnel soignant : si les valeurs aberrantes sur une dimension sont facilement détectables, ce sont les combinaisons aberrantes de valeurs pourtant communes qui sont moins souvent repérées lors des consultations.

Les Algorithmes 2 et 3 présentent respectivement la recherche des anomalies pour le premier, et le calcul spécifique de la *Rareness* pour le deuxième. On peut voir que ces algorithmes sont succincts et interprétables : les variables d'entrée restent intactes et peuvent être utilisées pour expliquer le résultat et guider l'utilisateur. Nous pouvons résumer la recherche d'anomalie proposée dans cette section en rappelant qu'en plus d'être interprétable et et adaptée à la haute dimensionnalité, la *Rareness* est aussi capable de détecter des *outliers* qui ne sont pas visibles sur des projections en deux dimensions. L'inconvénient principal de cette approche est un temps de calcul élevé car toutes les combinaisons possibles de dimensions sont explorées, pour chaque individu. C'est une des raisons qui nous ont poussées à trouver, pour le prototype d'aide à l'exploration que nous avons réalisé, une autre approche qui soit exécutable rapidement sur des données contenant plusieurs milliers d'éléments et dimensions.

Dans ce chapitre nous avons décrit le *DoR* permettant de structurer des éléments par représentativité, ainsi que la *Rareness* qui détecte les anomalies au sein de données multidimensionnelles. Ces deux méthodes ont été construites dans le but d'être favorables au raisonnement par analogie et sont donc *instance_based* en plus d'être adaptées à la haute dimensionnalité. Nous allons désormais les évaluer comparativement sur différentes données, réelles et synthétiques, dans le chapitre suivant.

Data : N elements defined on D dimensions, neighborhood size K
Result : Rareness of each element
$Knn^d(n) \longleftarrow$ *K-nearest-neighbors of n, on dimension d;*
foreach *element n in N* **do**
  foreach *dimension d in D* **do**
    $scores^d(n) \longleftarrow \max_{d' \in D}(\textbf{Rareness}(n, d, Knn^{d'}(n), K))$;
  end
end
$Result \longleftarrow scores^d(n)$;

Algorithme 2 : Algorithme de détection d'anomalie

Data : element n, dimension d, neighborhood $Knn^{d'}(n)$ *of size K*
Result : rareness of element n for a given dimension d and neighborhood $Knn^{d'}(n)$
$Rank_e^d(n) \longleftarrow$ ranking of n according to its distance from e on dimension d;
foreach *neighbor e in* $Knn^{d'}(n)$ **do**
  $scores_e^d(n) \longleftarrow \frac{1}{K} \times \min(Rank_e^d(n), K)$;
end
$Result \longleftarrow \frac{1}{K-1} \times \sum_{e \in Knn^{d'}(n)} score_e^d(n)$;

Algorithme 3 : Détail du calcul de la Rareness

Chapitre 4

Expérimentations

Sommaire

4.1 Évaluation 67
4.1.1 Critères d'évaluation 67
4.1.2 Description des données 70
4.1.3 Méthodes de structuration 72
4.1.4 Méthodes d'outlier detection 73
4.2 Résultats 75
4.2.1 Structure 75
4.2.2 Pertinence 78
4.2.3 Détection d'anomalies 80

Afin d'évaluer la pertinence de notre approche dans le contexte de la structuration de données en haute dimension, nous analysons dans ce chapitre les résultats fournis par l'algorithme proposé, et les comparons aux structurations obtenues avec d'autres méthodes de structuration d'éléments par similarité. Nous comparons aussi notre algorithme de détection d'anomalies à un autre algorithme dédié au même but. Cette comparaison est réalisée sur plusieurs jeux de données réelles ou simulées. La première section précise les méthodes d'évaluation proposées. C'est aussi dans cette section que nous décrivons les jeux de données utilisés et leur création le cas échéant, et que nous détaillons les méthodes choisies pour structurer les données et détecter les anomalies. La seconde section présente les résultats obtenus et en propose une analyse selon les critères définis précédemment. Les propriétés des résultats fournis par les méthodes sont comparées, et un exemple de structuration est étudié plus en détail : nous extrayons une composante connexe issue de la structuration de données réelles en haute dimensionnalité puis nous analysons les variables ayant eu un impact sur cette structuration. Nous étudions aussi empiriquement l'impact du facteur de granularité et nous terminons en proposant une brève interprétation des résultats concernant la détection d'anomalies et justifiant notre choix d'une autre méthode pour le prototype d'exploration de données médicales décrit dans le chapitre suivant.

4.1 Évaluation

4.1.1 Critères d'évaluation

Afin d'évaluer les structures obtenues grâce à l'algorithme du *DoR*, nous les comparons à d'autres structurations des mêmes données obtenues par différentes approches. Afin de mener cette comparaison, nous retenons deux critères différents : la « navigabilité » au sein du graphe créé par les représentants, et la pertinence des relations entre chaque individu et son représentant. Ce second critère sera aussi utilisé pour évaluer les résultats des algorithmes de détection d'anomalies.

Structure

Afin d'évaluer le graphe, précisons d'abord la définition du diamètre d'une composante connexe. Le diamètre d'une composante connexe correspond à la plus grande distance qui existe entre deux sommets de cette composante, la distance entre deux sommets étant calculée comme le plus court chemin (en nombre d'arêtes empruntées) entre les sommets. Le reste du vocabulaire en rapport avec les graphes est rappelé à la fin de la sous-section 3.2.1.

Dans ce manuscrit, nous désignons par « navigabilité » le fait de pouvoir naviguer entre sommets en suivant les arcs du graphe, donc de représentant en représentant. Une bonne navigabilité assure tout d'abord que les éléments sont regroupés en un petit nombre de larges composantes connexes. La navigabilité juge aussi la possibilité de hiérarchiser par représentativité les individus appartenant à une même composante connexe afin de pouvoir identifier visuellement les individus les plus représentatifs. Il s'agit donc de la facilité à explorer le graphe selon ses arcs, et de l'information apportée par la structure.

Pour évaluer la navigabilité nous avons sélectionné deux critères. Le premier d'entre eux est la taille moyenne des composantes connexes. Cette propriété nous informe sur la possibilité de naviguer dans le graphe entre noeuds pour explorer des éléments similaires et, ce faisant, extraire des connaissances. Des composantes connexes de taille réduite limitent l'exploration aux quelques cas les plus similaires voire, dans le pire cas, simplement au représentant de l'élément choisi comme point de départ. Le second critère est le diamètre : un faible diamètre comme illustré par la figure 4.1 (A) nous indique que ces éléments ont choisi le même représentant mais ne donne aucune information sur la nature des similarités entre les éléments de la composante. À l'inverse, un diamètre plus élevée comme sur la figure 4.1 (B) nous permet d'explorer les éléments du plus spécifique au plus typique.

Pertinence

Même avec un graphe dont la structure est adaptée à l'analyse exploratoire, il est nécessaire de vérifier que les associations entre éléments résultent bien d'une similarité entre ceux-ci. Afin d'en juger, nous nous reposons sur les mesures de micro-précision, micro-rappel et micro-Fmesure telles que définies dans Sokolova & Lapalme (2009). Ces mesures sont utilisées pour apprécier la qualité des résultats fournis par les algorithmes de classification, et sont une extension des mesures de précision, rappel et Fmesure classiques. Même si la méthode que nous proposons a un objectif divergent de la classification, nous pouvons tout de même évaluer la pertinence des associations créées en nous assurant que les éléments regroupés sont globalement similaires.

Soit un ensemble de N éléments appartenant chacun à une et une seule classe parmi un total de c classes. Un algorithme de classification prédit ensuite, à partir des N éléments, la classe à laquelle chacun d'eux appartient probablement. Les prédictions obtenues sont ensuite comparées aux prédictions qui étaient attendues : un élément appartenant à la classe C_i est considéré comme vrai positif si la classe C_i lui a été prédite, et faux négatif s'il a été affecté à une autre classe que C_i. Un élément n'appartenant pas à la classe C_i est un faux positif si C_i lui a été affectée malgré tout, et c'est un vrai négatif s'il a été reconnu comme n'étant pas membre de C_i. Toujours pour une classe C_i, les vrais positifs sont notés VP_i, les faux positifs FP_i, les vrais négatifs VN_i et les faux négatifs FN_i. La micro-précision et le micro-rappel sont calculés comme suit :

$$precision_\mu = \frac{\sum_{i=1}^{c} VP_i}{\sum_{i=1}^{c} VP_i + FP_i}$$

$$rappel_\mu = \frac{\sum_{i=1}^{c} VP_i}{\sum_{i=1}^{c} VP_i + FN_i}$$

Pour expliciter le sens porté par la précision et le rappel, il convient de définir brièvement l'erreur de type 1 et celle de type 2. Pour une classe C_i, l'erreur de type 1 consiste à détecter à tort un élément comme appartenant à C_i. Son pendant, l'erreur de type 2, se présente lorsqu'on ne détecte pas un élément qui, lui, appartient à C_i. La précision quantifie l'erreur de type 1, le rappel quantifie l'erreur de type 2, ces deux mesures étant maximales en l'absence d'erreurs.

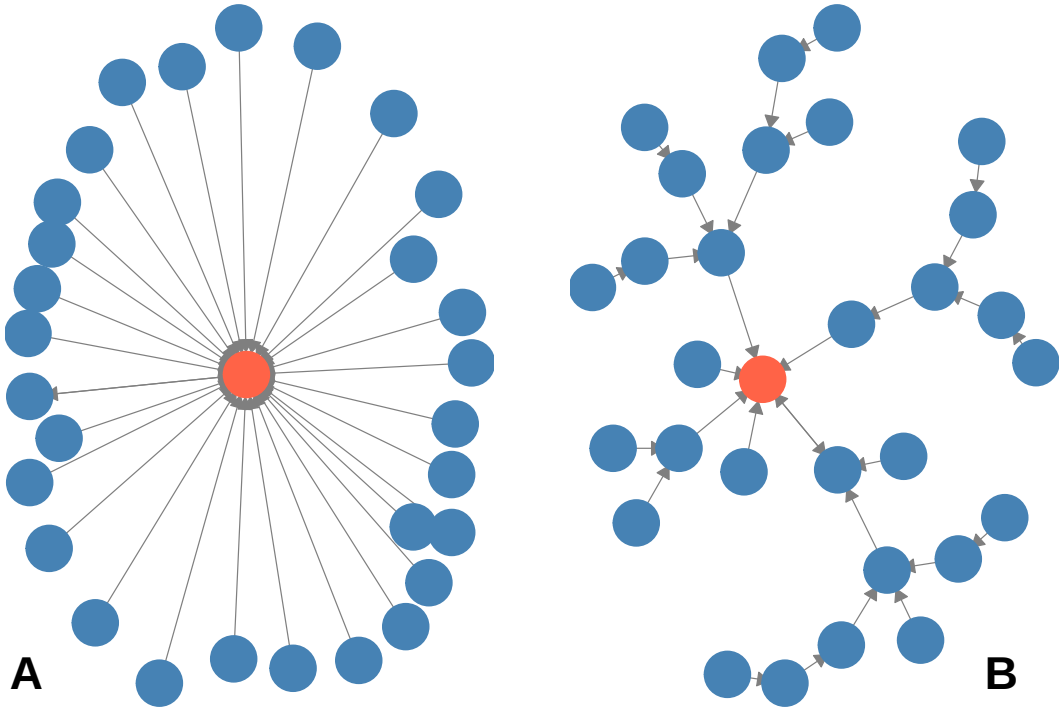


FIGURE 4.1 – Deux exemples de composantes connexes exprimant chacune la similarité de leurs éléments avec le représentant en leur centre. La structure proposée par A n'apporte pas d'information supplémentaire sur les similarités des données en les hiérarchisant, tandis que la structuration B permet d'explorer pas à pas les données par représentativité croissante en partant d'un élément.

On parle de micro (μ) précision et micro rappel quand le jeu de données étudié contient plus de deux classes. Il est possible de résumer ces valeurs grâce à la Fmesure (ou Fmesure_μ dans le cas de la précision_μ et du rappel_μ) : la Fmesure est obtenue avec la moyenne harmonique de la précision et du rappel. La Fmesure sera ainsi utilisée pour déterminer la pertinence des résultats fournis par les algorithmes de détection d'anomalies, en considérant que ces algorithmes classent les éléments en deux classes : anomalie, ou non.

Il est à noter qu'avec des données contenant plus de deux classes il est possible d'opter pour la macro-précision et le macro-rappel, c'est-à-dire la moyenne arithmétique de la précision et du rappel de chaque classe. La macro-Fmesure se calcule aussi par la moyenne harmonique des macro-précision et macro-rappel. Dans le cas des données étudiées pour la structuration, les éléments ne sont pas répartis de manière égale entre les classes, ce qui rend la micro-Fmesure plus appropriée que la macro-Fmesure. À noter aussi qu'il est possible de pondérer le calcul de la Fmesure pour accorder plus d'importance à la précision ou au rappel. Toutefois, la Fmesure que nous utilisons est un cas particulier (la F_1 mesure) qui accorde un poids égal aux deux critères.

4.1.2 Description des données

Deux ensembles de données ont été utilisés pour évaluer les méthodes proposées. L'évaluation de la structuration des élément selon le *DoR* est menée sur un ensemble composé de onze jeux de données simulés ou issus de diverses sources réelles. La détection d'anomalie est comparée à une autre approche sur un ensemble de six jeux de données réelles, ces données sont issues d'une même source et sont dédiées à l'évaluation de méthodes d'*outlier detection*.

Structuration des éléments

La table 4.1 résume les informations sur tous les jeux de données utilisés pour l'évaluation du *DoR*. La colonne « *Données* » correspond à l'abbréviation choisie pour chacun des *datasets* afin d'y référer aisément dans ce travail. Les colonnes « *éléments* » et « *dimensions* » réfèrent respectivement au nombre d'individus (N) et de variables (D) les décrivant. Une colonne « *classes* » indique, pour les données de *clustering*, le nombre de classes présentes. Enfin, les colonnes « *référence* » et « *sujet* » fournissent un contexte pour ces recueils de données en précisant le travail scientifique dont il sont issus et le domaine d'étude auxquels ils sont dédiés.

uniforme et *normal* sont des jeux de données synthétiques, composés chacun de 300 éléments décrits sur 1000 dimensions. Les variables du jeu *uniforme* sont générées aléatoirement entre 0 et 1 selon une loi uniforme, tandis que *normal* comprend des variables générées selon une loi normale centrée réduite. *normal + bruit* est un mélange des deux *datasets* précédents, dans lequel 80% des dimensions sont échantillonnées à partir d'une distribution normale centrée réduite, et les 20% de dimensions restantes sont issues d'une distribution uniforme entre 0 et 1 afin de modéliser du bruit dans les données. Ces trois jeux de données servent à illustrer la « malédiction de la dimensionnalité » présentée dans la section 2.2 : avec leur millier de dimensions, l'espace est « creux » et les distances entre éléments sont difficilement discernables.

Residential Buildings, abbrégé *residentiel* (Rafiei & Adeli, 2016) et *Communities and Crime*, abbrégé *communautes* (Redmond, 2009) sont tous les deux des jeux de données mis à disposition par le *UCI Machine Learning Repository* (Dua & Graff, 2019). Ils sont décrits chacun par plus d'une centaine de dimensions, certaines d'entre elles étant des variables à prédire permettant d'évaluer les algorithmes de prédiction. *residentiel* fournit des informations financières, économiques et physiques sur 372 projets de construction d'appartements à Téhéran, la plus grande ville et la capitale de l'Iran. *communautes* combine les données récoltées par les forces de l'ordre sur la criminalité et le contexte socio-économique de 2215 villes aux États-Unis, sur une période allant de 1990 à 1995.

L'ensemble de données *iris*, décrit précédemment lors de sa première utilisation pour illustrer l'analyse exploratoire (sous-section 2.1), est pour rappel un jeu de données comprenant 150 variables et 4 dimensions, dont chaque élément appartient à une classe parmi les trois types d'iris possibles.

Les jeux de données abrégés *alon*, *christensen*, *golub*, *sorlie* et *su* sont des données biomédicales issues de plusieurs domaines d'étude et agrégées dans le package R *datamicroarray*. [1] Ils contiennent entre 62 et 217 éléments, qui sont tous décrits en haute dimensionnalité par 456 à 7129 variables. Ces données sont uniquement quantitatives, et souvent utilisées pour du *clustering* : les éléments sont tous étiquetés avec la classe à laquelle ils appartiennent.

TABLE 4.1 – Description des jeux de données pour la structuration.

Données	Référence	Éléments	Dimensions	Sujet	Classe
uniforme	-	300	1000	-	-
normal	-	300	1000	-	-
normal + bruit	-	300	1000	-	-
residentiel	Rafiei & Adeli (2016)	372	103	immobilier	-
communautes	Redmond (2009)	2215	101	criminalité	-
iris	Anderson (1936)	150	4	botanique	3
alon	Alon et al. (1999)	62	2000	cancer colon	2
christensen	Christensen et al. (2009)	217	1413	épigénétique	3
golub	Golub et al. (1999)	72	7129	leucémie	3
sorlie	Sorlie et al. (2001)	85	456	cancer sein	5
su	Su et al. (2002)	102	5565	génome	4

Avant toute analyse, nous retirons les variables servant d'objectifs dans les jeux de données : les labels des classes pour les données de clustering, et les attributs à prédire pour les jeux utilisés en prédiction. Nous enlevons aussi les dimensions non pertinentes dans le cadre de la comparaison des performances de structuration (comme les identifiants des individus ou les dates de prise des mesures) ainsi que celles contenant des valeurs manquantes. Cela représente un total de 2 et 46 variables sur *residentiel* et *communautes* respectivement. Le nombre de dimensions indiquées sur la figure 4.1 correspond aux dimensions restantes.

Détection d'anomalies

Le comparatif des méthodes d'*outlier detection* a été réalisé avec un ensemble de données issues du *UCI Machine Learning Repository* (Dua & Graff, 2019) et filtrées par le catalogue *ODDS* (Rayana, 2016) . Composés de données numériques et multidimensionnelles, ces jeux de données ont été agrégés dans le but d'évaluer les

1. `https://github.com/ramhiser/datamicroarray`

algorithmes de détection d'anomalies. Ces jeux varient d'une centaine à plusieurs milliers d'éléments, d'une dizaine à plusieurs centaines de dimensions, et figurent parmi les plus utilisés dans l'évaluation de ce type d'algorithmes. Le taux d'anomalies dans les données utilisées pour ce comparatif varie de moins de 5% jusqu'à plus d'un tiers des éléments du jeu de données.

TABLE 4.2 – Description des jeux de données pour la détection d'anomalies.

Données	Référence	Éléments	Dimensions	Sujet	Classe
arrhytmia	Guvenir et al. (1997)	452	274	Cardiologie	66 (15%)
cardio	Ayres-de-campos et al. (2009)	1831	21	Cardiologie	171 (10%)
glass	Evett & Spiehler (1989)	214	9	Criminologie	9 (4%)
ionosphere	Sigillito et al. (1989)	351	33	Atmosphère	126 (36%)
lympho	Zwitter & Soklic (1988)	148	18	Oncologie	6 (4%)
wine	Dua & Graff (2019)	129	13	Œnologie	10 (8%)

4.1.3 Méthodes de structuration

Afin de comparer la structuration des jeux de données présentés précédemment, nous avons sélectionné plusieurs méthodes que nous décrivons ici. L'objectif, quelle que soit la méthode employée pour l'atteindre, est d'obtenir une structuration où chaque élément peut être raproché d'au moins un élément similaire et générique. Cette structuration doit être obtenue sans connaissance ou hypothèse *a priori* sur les données ou leur distribution.

Méthode proposée : DoR

Pour obtenir le graphe de similarité à partir de la méthode que nous proposons, nous devons sélectionner le paramètre k utilisé pour la taille du voisinage. Nous choisirons une valeur de k égale à 30% du nombre d'éléments composant le jeu de données que nous souhaitons structurer. Rappelons que ce paramètre pourrait être ajusté au fur et à mesure de l'exploration, selon les connaissances *a priori* de l'utilisateur ou encore selon l'objectif souhaité (un compromis entre structure de graphe appropriée et fortes similarités). Il est aussi possible de déterminer automatiquement une valeur appropriée pour ce paramètre de granularité, mais un paramètre arbitraire sera utilisé afin de conserver une équité avec les autres méthodes employées.

Plus proche voisin

Nous incluons dans ce comparatif la méthode de similarité la plus intuitive, à savoir la distance Euclidienne. Les éléments seront structurés en associant chaque élément à son plus proche voisin (ou *NN*, *nearest neighbor*) afin de former un graphe de similarité. En plus des associations selon la distance Euclidienne, nous créons aussi une structuration alternative en utilisant la distance de Minkowski d'ordre 0.5. Cette seconde distance est plus appropriée à la haute dimensionnalité que la distance Euclidienne, et devrait fournir des relations plus pertinentes entre les éléments.

Partitioning Around Medoids

Cet algorithme classique de partitionnement (Kaufman & Rousseeuw, 1987), aussi appelé *PAM*, est basé sur l'approche des k-moyennes avec un prototype de classe remplacé par un médoïde : chaque élément est associé au représentant de sa classe. Un médoïde n'est pas un élément synthétique mais un élément du jeu de données qui occupe une position centrale dans le *cluster*. Le but de cette méthode est de minimiser la distance totale entre les éléments et le représentant de leur classe respective. Cette approche nécessite de fixer *a priori* le nombre de partitions recherchées et même si cette connaissance est rarement disponible lors de l'analyse exploratoire, il est possible d'utiliser des heuristiques permettant de l'estimer, par exemple celle basée sur l'indice de silhouette (Rousseeuw, 1987). Nous allons toutefois utiliser notre connaissance préalable du nombre de partitions en le fournissant comme paramètre d'entrée pour les données dont les éléments sont étiquetés, et ne pas appliquer *PAM* aux autres.

Affinity Propagation

Surnommée *affinity propagation* (Frey & Dueck, 2007), il s'agit d'une approche structurant les données grâce à un échange itératif de messages entre les éléments. Les individus du jeu de données s'échangent ici, par itération, des informations de typicalité et de représentativité afin d'arriver à un état où les représentants choisis par les individus sont stables. Cette méthode offre plusieurs avantages similaires à ceux de la méthode que nous proposons, à savoir qu'il n'est pas nécessaire d'avoir une connaissance préalable de la manière dont les données sont structurées mais qu'il est tout de même possible, si elle existe, d'injecter cette connaissance dans le processus de structuration pour obtenir des résultats plus justes. Il n'est pas nécessaire non plus de fournir le nombre de représentants désirés.

4.1.4 Méthodes d'outlier detection

Bien que la détection d'anomalies soit un problème étudié depuis plusieurs décennies et possédant des algorithmes efficaces (Liu et al., 2017), la « malédiction de la dimensionnalité » provoque là-aussi des difficultés spécifiques (C C Aggarwal & Yu, 2001) dûes aux problèmes d'espaces « creux » et de concentration de distance (évoqués dans la Section 2.2). Si la *Rareness* propose une approche permettant de détecter des anomalies non triviales car impossible à observer sur une dimension, elle permet aussi l'analyse de données multidimensionnelles d'une manière interprétable et explicable.

Malgré ces atouts majeurs, elle n'avait pas encore été testée en grande dimension, et comparée à d'autres approches. Il existe de nombreuses méthodes non-supervisées de détection d'anomalies qui fournissent des résultats pertinents en haute dimensionnalité comme les *Isolation Forests*, les méthodes de projection et le *Local Outlier Factor* (Breunig, Kriegel, Ng, & Sander, 2000; Liu, Ting, & Zhou, 2008; Liu et al., 2017). Toutefois, selon les critères d'explicabilité détaillés dans la section 2.3, les résultats fournis par ces méthodes ne sont pas optimaux. Ainsi, ces algorithmes efficaces mais aux résultats moins compréhensibles et interprétables n'aident pas le spécialiste à comprendre sur quels critères un patient peut être considéré comme *outlier*. Pour ces raisons, nous avons retenu pour cette comparaison une méthode particulièrement adaptée à notre contexte applicatif : le *Subspace Outlier Degree* est à la fois *instance-based* et approprié aux grandes dimensions.

Méthode proposée : Rareness

Dans cette méthode basée sur la *Rareness* des individus étudiés, il est nécessaire au préalable de fournir un paramètre k agissant comme facteur de granularité définissant la taille des voisinages étudiés. Comme pour le *DoR* nous avons fait le choix d'une valeur de k égale à 30% du nombre d'éléments figurants dans le jeu de données. Il est là aussi possible d'ajuster cette valeur empiriquement pour l'adapter aux données étudiées, mais nous conserverons une valeur générique afin de ne pas optimiser la méthodes par rapport aux données étudiées.

Subspace Outlier Degree

La méthode appelée *Subspace Outlier Degree* (Kriegel et al., 2009), ou *SOD*, propose une détection d'anomalies pensée pour être efficiente en haute dimension. Comme nous l'avons vu dans la section 2.2.2, en haute dimensionnalité les distances se concentrent et les individus deviennent difficiles à différencier les uns des autres. *SOD* se base justement sur la distance des individus par rapport au centre de la distribution, un critère qui peut sembler inadapté à de nombreuses dimensions au vu des effets de la dimensionnalité. Pour y remédier, l'algorithme n'utilise pas la distance dans l'espace entier de projection : *SOD* va effectuer des projections dans des sous-espaces choisis de manière stochastique, réduisant ainsi la concentration des distances. Si les données sont initialements définies en D dimensions, *SOD* va sélectionner de manière aléatoire e dimensions parmi les D initiales, avec $e << D$. Au sein de ces projections, tout élément s'éloignant du centre de la distribution de plus de 80% de la variance sera considéré comme anomalie.

L'approche *SOD* se base elle aussi sur les plus proches voisins d'un point afin de déterminer s'il peut être considéré comme une anomalie. Nous avons là-aussi utilisé un voisinage d'une taille de 30% du nombre d'éléments composant leu jeu de données. Ainsi pour N éléments, chaque élément se basera sur ses $0.3 * N$ plus proches voisins au sens des *shared nearest neighbors* pour déterminer s'il s'agit d'un élément inhabituel.

Maintenant que nous avons décris les données utilisées ainsi que les méthodes comparées et nos critères d'évaluation, nous pouvons étudier les résultats obtenus. D'abord ceux de la méthode de structuration proposée sous la forme du *DoR*, puis les résultats comparant la *Rareness* au *SOD* en haute dimensionnalité.

4.2 Résultats

Concernant la structuration des éléments, les résultats des critères de navigabilité figurent dans la première sous-section au sein des Tables 4.3 et 4.4. Les résultats concernant la pertinence des associations sont dans la Table 4.5 figurant dans la deuxième sous-section. Les méthodes sont abrégées par *DoR* pour la méthode proposée, *NNe* et *NNm* pour le plus proche voisin selon la distance euclidienne et minkowski d'ordre 0.5 respectivement, *PAM* pour les k-médoïdes et *AP* pour l'*affinity propagation*.

La comparaison des approches de détection d'anomalies figure dans la troisième sous-section. La Table 4.6 propose la comparaison des résultats offerts par la *Rareness* et par le *Subspace Outlier Degree*, en se basant sur la Fmesure calculée à partir de ces résultats.

4.2.1 Structure

La Table 4.3 présente les moyennes de la taille des composantes connexes. Nous pouvons voir que la taille moyenne des structures de notre algorithme est régulièrement plus basse que celle proposée par *affinity propagation*, et souvent similaire aux tailles des composantes crées par les méthodes de plus proche voisins. *Partitioning Around Medoids*, à l'inverse, propose de très grandes composantes connexes du fait que le nombre de partitions est fixé en paramètre d'entrée afin de correspondre aux *clusters* présents dans les données. Il serait toutefois possible d'augmenter le facteur k choisi pour notre méthode de structuration afin d'obtenir des résultats similaires, le diamètre moyen serait réduit en faveur de composantes connexes plus grandes.

Concernant les diamètres (figurant sur la Table 4.4), les structures créées par *PAM* ont systématiquement un diamètre égal à 1, car la hiérarchie engendrée par cet algortihme est horizontale : tous les éléments d'une partition ne sont subsumés que par le médoïde qui les représente. *AP*, dont le diamètre est parfois égal à 1, propose toutefois souvent des composantes au diamètre moyen inférieur à 1. Ce résultat s'explique par le fait que certains éléments sont jugés comme *outliers* et ne sont rattachés à aucun autre. En terme d'exploration des données, un résultat comme celui-ci empêche toute exploration de proche en proche à partir d'un élément isolé. Les plus grands diamètres sont, de manière consistante, ceux des structures créées par l'approche présentée dans ce manuscrit.

TABLE 4.3 – Tailles moyennes des composantes connexes générées par les différentes méthodes de structuration choisies, sur des données réelles et synthétiques.

Données	DoR	NNe	NNm	PAM	AP
uniforme	8.57	7.32	5.00	-	12.00
normal	20.00	12.50	8.11	-	17.65
normal + bruit	11.54	15.00	8.33	-	21.43
residentiel	5.24	3.54	3.48	-	14.88

communautes	7.77	5.54	5.03	-	18.31
iris	18.75	4.17	3.95	50	13.64
alon	4.13	6.20	4.13	31	4.77
christensen	8.68	9.43	8.35	72.33	15.50
golub	6.55	6.00	6.00	36	4.80
sorlie	7.08	12.14	9.44	17	9.44
su	5.37	5.37	5.67	25.5	8.50

TABLE 4.4 – Diamètres moyens des composantes connexes générées par les différentes méthodes de structuration choisies, sur des données réelles et synthétiques.

Données	DoR	NNe	NNm	PAM	AP
uniforme	3.69	2.46	2.38	-	1.00
normal	5.40	2.71	2.49	-	1.00
normal + bruit	4.19	3.05	2.50	-	1.00
residentiel	2.52	2.06	2.06	-	1.00
communautes	3.05	2.52	2.43	-	0.89
iris	3.75	2.31	2.11	1	1.00
alon	2.07	2.80	2.27	1	0.69
christensen	2.68	2.78	2.81	1	0.71
golub	2.73	2.17	2.17	1	0.47
sorlie	3.17	3.14	2.78	1	0.78
su	2.68	2.26	2.67	1	1.00

Illustrons cette contrainte de navigabilité avec les Figures 4.2 et 4.3 qui proposent deux structurations du jeu de données *Iris* : en liant chaque élément à son plus proche voisin selon la distance Minkowski d'ordre 0.5 (4.2) et en utilisant la méthode que nous proposons (4.3). La Figure 4.2 contient de nombreuses composantes connexes et n'est pas favorable à l'exploration selon les arcs du graphe, les plus grandes composantes ayant moins d'une dizaine d'éléments. Sur la Figure 4.3 la contrainte de pouvoir naviguer aisément entre les éléments par similarité est comparativement mieux satisfaite, particulièrement car aucun élément ne peut être isolé (comme c'est parfois le cas avec *affinity propagation* par exemple). Notons qu'il est aussi possible de parcourir le graphe de proche en proche par généricité croissante, les composantes ayant un diamètre beaucoup plus élevé qu'en utilisant le plus proche voisin selon la distance de Minkowski.

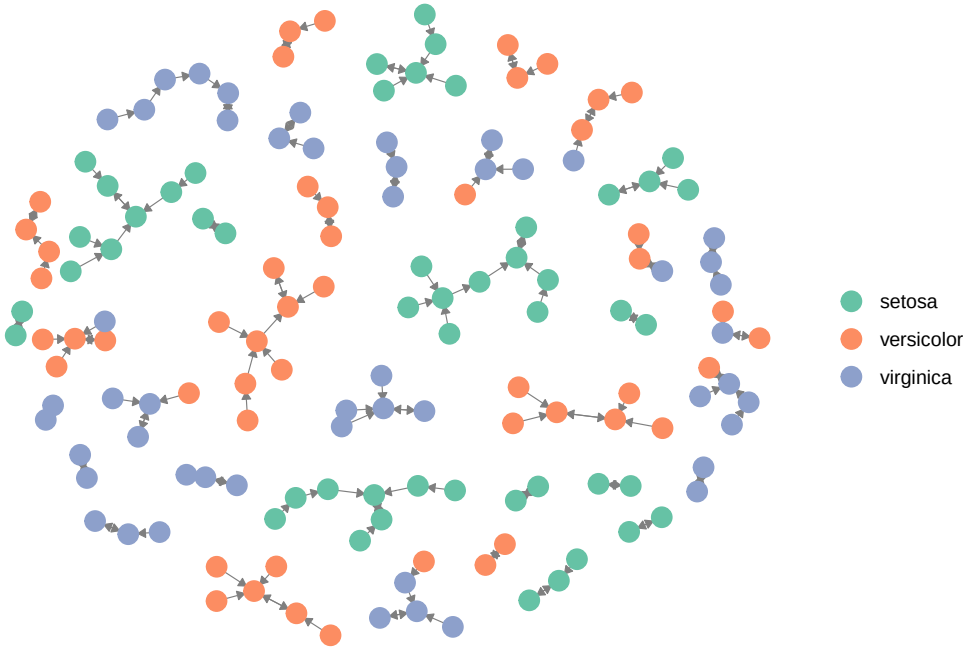


FIGURE 4.2 – Structuration du jeu de données Iris en associant chaque élément à son voisin le plus proche selon la distance de Minkowski d'ordre 0.5. Le grand nombre de composantes contenant peu d'éléments n'est pas favorable à l'exploration des données selon leur similarité.

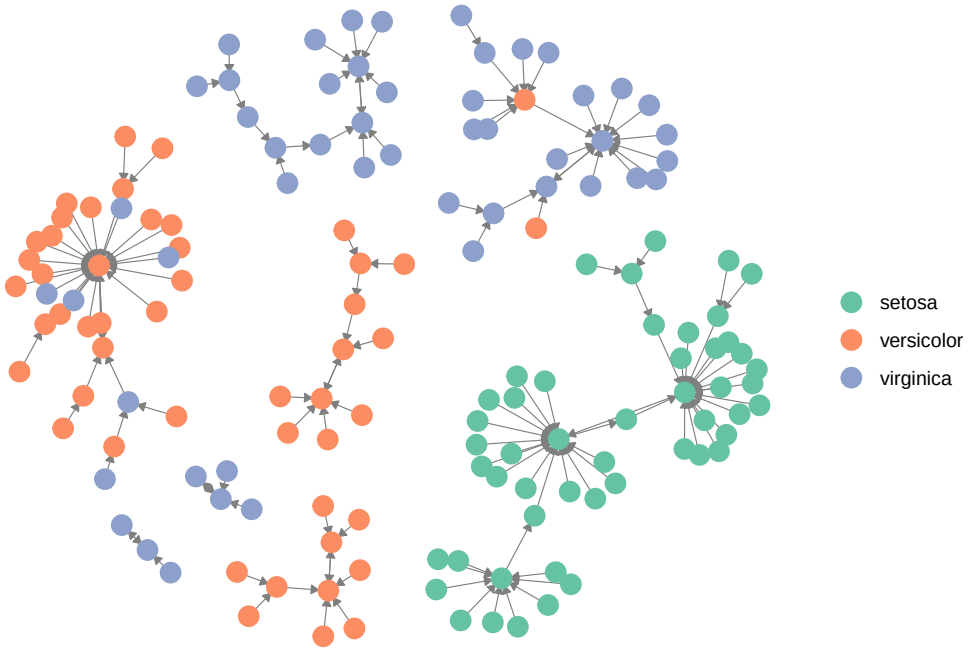


FIGURE 4.3 – Structuration de Iris selon la méthode proposée (k = 45, soit 30% du nombre d'éléments). Les composantes connexes obtenues sont de diamètre élevé, et regroupent de nombreux éléments semblables mais hiérarchisés selon leur similarités.

4.2.2 Pertinence

Si la structure retournée est appropriée à une démarche d'exploration des données, le deuxième critère à étudier est la pertinence des associations qui forment cette structure. Pour cela, la Table 4.5 contient les micro-Fmesures obtenues avec différentes méthodes, sur les données appropriées à cette évaluation : celles dont les éléments appartiennent chacun à une classe. Cette mesure est obtenue, pour rappel, avec la moyenne harmonique des formules de micro-précision et micro-rappel de la section 4.1.1.

Bien que cette mesure ait été élaborée pour évaluer des problèmes de classification, elle peut être utilisée comme indicateur global de la similarité entre éléments associés. Nous pouvons voir que, dés que les données sont décrites en au moins une centaine de dimensions, les structures que nous proposons s'avèrent généralement plus pertinentes que les autres approches proposées. Cela s'explique par le fait que les méthodes traditionnelles sont, depuis longtemps, optimisées pour les données à faible dimensionnalité. Il est intéressant de noter que les k-médoïdes, méthode incontournable de par sa simplicité, fournit des résultats très inégaux en haute dimensionnalité bien que nous connaissions *a priori* le nombre de partitions recherchées, qui est une information rarement disponible lors de l'analyse exploratoire de données. Malgré les problèmes évoqués précédemment concernant le faible diamètre des composantes créées et les éléments isolés, *affinity propagation* propose par contre des associations pertinentes même en grande dimension et sans nécessiter de connaissances préalables, ce qui en fait l'alternative principale à l'approche que nous proposons.

Enfin, s'il est plus aisé de naviguer par similarité grâce aux relations créées par l'association au plus proche voisin selon la distance Euclidienne, les relations proposées par la distance fractionnelle de Minkowski s'avèrent plus pertinentes en grande dimension. Ce résultat illustre la préférence pour la distance fractionnelle lorsque le nombre de dimensions dépasse la dizaine, les éléments étant ainsi plus séparables qu'en distance Euclidienne.

TABLE 4.5 – Évaluation de la pertinence des relations grâce aux micro-Fmesures obtenues par les méthodes retenues et sur les jeux de données composés d'éléments étiquetés par une classe.

Données	DoR	NNe	NNm	PAM	AP
iris	0.89	0.95	0.92	0.89	0.96
alon	0.73	0.74	0.71	0.65	0.85
christensen	1.00	0.99	1.00	1.00	1.00
golub	0.94	0.88	0.90	0.65	0.90
sorlie	0.78	0.71	0.75	0.62	0.69
su	0.99	0.96	0.98	0.94	0.98

Nous pouvons, dans un second temps, analyser la nature des relations entre éléments au sein d'une composante connexe, afin d'en savoir plus sur leur pertinence et le sens qu'elles portent. Pour cela, la Figure 4.4 présente une composante connexe issue de la structuration, avec l'algorithme proposé, du jeu de données *residentiel*, en utilisant un paramètre de granularité $k = 111$, soit 30% de la taille du jeu de données. En étudiant les relations entre ces éléments et en rappelant que *residentiel* est composé de 103 dimensions, nous pouvons noter que :

— Les éléments **188** et **203** ont tous les deux choisi l'élément **140** comme représentant car cet élément figure parmi leurs k plus proches voisins sur au moins 82 dimensions, ils sont donc similaires à **140** sur la majorité de leurs dimensions. 81 des 82 dimensions associant **188** à **140** sont aussi celles associant **203** à **140**.
— Si **188** et **203** sont similaires l'un à l'autre selon 87 variables, ils ne sont toutefois par représentants l'un de l'autre car ils attribuent chacun à **140** un *DoR* de plus de 4388 tandis qu'ils s'attribuent l'un l'autre un $DoR_{188}(203) = 3979$ et un $DoR_{203}(188) = 3878$. **140** est donc un peu moins similaire mais sensiblement plus typique, d'où le choix de cet élément comme représentant afin d'être subsumé par un individu plus caractéristique.
— Enfin, **140** est similaire à **127** sur 90 dimensions pour un *DoR* total de 4908. Ce degré de représentativité élevé confirme la généricité et la typicalité de **127**. Précisons aussi que si **188** et **203** étaient similaires à **140** sur 82 dimensions, 80 d'entre elles sont aussi parmi les 90 dimensions associant **140** à **127**.

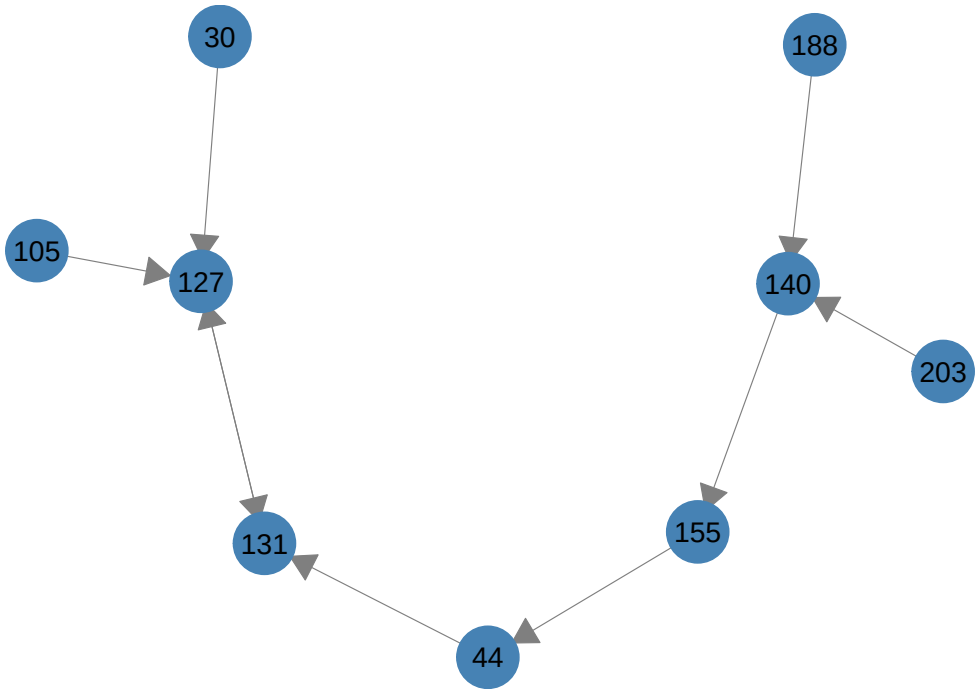


FIGURE 4.4 – Composante connexe extraite de la structuration du jeu de données Residentiel en utilisant la méthode proposée dans ce manuscrit, avec un facteur de granularité k = 111 (30% du nombre d'éléments).

4.2.3 Détection d'anomalies

Pour la détection d'anomalies, l'approche proposée et basée sur la *Rareness* s'avère très interprétable, mais le *Subspace Outlier Degree* propose la même qualité : il est possible de regarder sur quelles dimensions du sous-espace l'*outlier* sort de la distribution afin de pouvoir isoler ses spécificités et les communiquer clairement à l'utilisateur. La différence se fait toutefois sur la performance de *SOD* par rapport à l'approche que nous proposons, du point de vue de la Fmesure. Comme l'illustre la Table 4.6, l'approche par projections dans des sous-espaces s'avère dans tous les cas offrir de meilleurs résultats. Il n'y a qu'avec certaines distributions de données spécifiques que l'approche proposée fournit des résultats similaires.

C'est pour ces raisons que, lors de la recherche d'une méthode de détections d'anomalies pour enrichir le prototype, nous avons décidé en faveur de *SOD* qui offre à la fois efficacité et interprétabilité. La *Rareness*, efficace et explicable avec un nombre plus réduit de dimension, propose toutefois un temps de calcul prohibitif en haute dimensionnalité. De plus, elle ne répond à l'objectif de détection d'anomalies non triviales que dans des cas plus spécifiques, ses résultats étant inconstant suivant le type de données analysées.

TABLE 4.6 – Comparaison de la Fmesure obtenue sur la détection d'outliers, sur plusieurs jeux de données réelles. SOD fournit de meilleures performances sur la totalité des jeux de données, et l'approche proposée ne s'avère efficace que dans des conditions extrêmements spécifiques.

Données	Rareness	SOD
arrhythmia	0.06	0.89
ionosphere	0.56	0.80
lympho	0.07	0.99
wine	0.92	0.92
cardio	0.06	0.95
glass	0.28	0.31

Concernant l'algorithme de structuration proposé, nous avons pu vérifier empiriquement dans ce chapitre que les résultats fournis par le *DoR* s'avèrent à la fois pertinents mais aussi appropriés à la tâche d'exploration. Ils permettent en effet une structuration qui peut être naviguée aisément entre éléments similaires, et dont on peut extraire les éléments les plus représentatifs de chaque groupe de données. Ces éléments peuvent être repérés aussi bien visuellement que par analyse de leurs *DoR*. Pour la détection d'anomalies, nous avons constaté que la *Rareness* ne propose des résultats satisfaisants que pour certains cas spécifiques. Pour cela nous privilégions le *Subspace Outlier Degree* pour la détection d'anomalies : cet algorithme offre de meilleures performances autant en terme de temps d'exécution qu'en terme de qualité des résultats, et il fournit lui aussi des résultats interprétables et explicables directement à partir des données d'entrée. En résumé, employer le *DoR* et le *SOD* sur des données médicales, comme dans le chapitre suivant, est ainsi possible et adapté.

Chapitre 5

Application aux données médicales

Sommaire

5.1 Données médicales 85
5.1.1 Contexte 85
5.1.2 Données 85
5.1.3 Objectif 86
5.2 Pré-traitement 86
5.2.1 Généralisation 86
5.2.2 Interface et résultats 87
5.3 Prototype d'exploration 91

Ce travail de recherche s'inscrivant dans un contexte médical, ce chapitre est dédié à l'application de la structuration à des données médicales issues du suivi de patients diabétiques. La première section décrit le projet *Datadiab* dans lequel s'intègre ce travail, nous y rappellerons brièvement le contexte applicatif avant de décrire les données qui y sont associées, leurs particularités et les résultats espérés. Dans la section suivante, nous détaillerons la phase de prétraitement qui a été nécessaire afin de transformer des données brutes en données manipulables. La troisième et dernière section décrit un prototype d'interface web permettant aux experts médicaux de structurer ces données et de faciliter leur exploration. Nous passerons en revue les vues et fonctionnalités fournies par cet outil destiné au service Diabète-Endocrinologie-Nutrition du centre hospitalier de Reims, avec lequel nous collaborons.

5.1 Données médicales

Notre collaboration avec les médecins et le personnel hospitalier du centre hospitalier universitaire de Reims ayant donné naissance aux travaux présentés ici, nous avons souhaité appliquer cette méthode aux jeux de données dont dispose le service d'endocrinologie du CHU. Pour cela nous nous sommes reposés sur la base de données ORNICARE, alimentée et rendue disponible par le réseau de santé de Champagne-Ardenne appelé CARéDIAB.

5.1.1 Contexte

Association de professionnels de la santé, CARéDIAB fédère depuis 2004 des médecins, personnels hospitaliers, pharmaciens, infirmiers et travailleurs sociaux auteur d'un projet d'accompagnement des patients atteints de diabète. Ce réseau a pour but de faciliter l'échange entre les spécialistes impliqués et ainsi de promouvoir l'émergence de soins adaptés. Inspiré par ADDICA, un réseau de support pour le suivi de l'addiction, CARéDIAB a désormais fusionné avec ADDICA afin de mettre en commun leurs savoirs et leur expertise.

Afin de faciliter la collaboration, une plateforme de partage de données à été mise en place. Nommée ORNICARE, il s'agit d'une plateforme à laquelle ont accès les professionnels membres de CARéDIAB, et les patients suivis. Les dossiers médicaux des patients diabétiques y sont hébergés de manière à être accessibles par tous les acteurs du parcours de soin, facilitant la transmission d'informations et la prise de décision rapide. En plus de cela, un système de partage d'annotations offre une manière directe pour les professionnels d'échanger entre eux et de s'assister mutuellement, tout en incluant le patient dans ce processus, peu importe le lieu où seront réalisés les soins.

5.1.2 Données

Nous travaillons sur un sous-ensemble de données anonymisées issues de la base ORNICARE et extraites par les professionnels du service Diabète-Endocrinologie-Nutrition du CHU de Reims. Agrégées en 2014, ces données patients contiennent 51456 enregistrements, chacun doté de 644 variables et décrivant un

total de 6957 patients atteint de diabète de Type 1. Nous référerons à ce jeu de données comme *patients14*. Il contient indifféremment des informations qualitatives et quantitatives, incluant entre autres des données physiologiques, les complications liées au diabète, l'état général du patient, ses traitements en cours et passés ou les examens effectués, souvent sous forme de texte libre. Chaque enregistrement correspond à une visite ou un acte médical, lors duquel le professionnel a rempli une partie des 644 champs à sa disposition, selon la nature de l'examen. Ainsi, chaque élément de la base *patients14* correspond à une mise à jour de certains champs, plusieurs éléments pouvant être rattachés à un seul et même patient : autant que d'examens médicaux subis par ce patient.

Un second jeu de données a été extrait de la base ORNICARE en 2017, toujours en se concentrant sur les patients atteints de diabète de Type 1. Cette fois, les données ont été enrichies manuellement par les médecins responsables de la création du jeu de données, pour la réalisation d'une étude clinique. 28073 actes médicaux ont été extraits, définis par 92 variables, et concernant 192 patients. Nous réfèrerons à ce jeu de données plus récent et plus complet sous le nom de *patients17*.

5.1.3 Objectif

L'objectif final, sur les deux jeux de données extraits, est de pouvoir structurer les individus par similarité afin de pouvoir explorer la cohorte de patients et faire de l'analyse épidémiologique ou prédire la survenue prochaine de complications liées au diabète. Pour cela, une étape de prétraitement a été nécessaire afin de rendre les données exploitables pour la méthode que nous proposons. La seconde étape est celle de la création d'un outil permettant de structurer les patients puis naviguer ce graphe de similarité, de sorte que les professionnels puissent être autonomes dans leur usage de cette solution.

5.2 Pré-traitement

Afin d'exploiter ces données pour établir une similarité entre les patients, il est tout d'abord nécessaire de nettoyer le jeu de données en retirant les valeurs manquantes ou aberrantes, les erreurs de saisie et de mesure, puis de fusionner les données redondantes pour obtenir un historique et des informations exploitables pour chaque patient.

5.2.1 Généralisation

Ce travail a été effectué une première fois par Élodie Lavoisier en 2014 avec la création de scripts *R* (R Core Team, 2019) spécifiques au jeu de données *patients14* en notre possession à l'époque. Toutefois ce travail n'était adapté qu'au fichier de patients dont nous disposions, et lorsqu'une version plus récente de ce fichier nous a été communiquée avec *patients17*, il a été nécessaire d'effectuer à nouveau cette étape de pré-traitement. Nous avons alors envisagé la création d'un outil de traitement plus générique, qui puisse être adapté à toutes les données ultérieures issues de la base ORNICARE.

Cet objectif s'est concrétisé avec le travail de Danaé Proix, dont le stage que j'ai encadré a permis la création d'une suite d'outils de pré-traitement de ces données. Ces opérations sur les données étaient d'abord réalisées sous la forme d'une série de scripts *R* à l'exécution automatisée et entièrement paramétrable, avant que nous choisissions d'opter pour une interface web grâce à *Shiny* (Chang, Cheng, Allaire, Xie, & McPherson, 2019). Cette seconde étape nous a permis cette fois de simplifier et rationaliser les opérations préalables pour mettre en forme les données, réduisant ainsi drastiquement le temps nécessaire pour cette opération amenée à être répétée régulièrement.

5.2.2 Interface et résultats

L'outil créé durant le stage permet ainsi, sous la forme d'une interface web pouvant tourner en local ou sur un serveur distant, de charger un fichier de données brutes et d'y appliquer divers traitements automatisés listés sur la Figure 5.1. Cette approche propose, en plus d'être plus rapide à utiliser qu'un script à modifier manuellement, la généricité nécessaire pour être utilisé pour tout autre type de jeu de données médicales ou non. Parmi les traitements proposés on peut énumérer, de manière non exhaustive :

— le retrait des colonnes vides ou superflues et le renommage de groupe de celles restantes, selon une convention de nommage (Annexe A.1).
— la conversion du format de colonnes (Figure 5.2) et la correction automatique de fautes d'orthographe et erreurs de saisie.
— la définition de règles d'imputation type « si âge > 50 ans ET bio-clairance > 60mL/min ALORS insuffisance rénale » (Figure 5.4).
— la fusion de toutes les opérations concernant chaque patient, de sorte à avoir un unique enregistrement par patient. Sur la Figure 5.3 l'utilisateur peut, selon les colonnes, retenir uniquement la valeur la plus récente (pour l'âge par exemple) ou l'historique entier des valeurs pour cette variable (pour l'évolution du poids par exemple).

Cet outil nous permet d'obtenir des données traitées plus simples à manipuler : *patients14* passe de 644 variables et 96% de valeurs manquantes à 510 variables et 81% de valeurs manquantes. Pour *patients17* nous évoluons de 92 colonnes et 59% de données manquantes à 28 colonnes et 23%. Ce second jeu de données, enrichi et vérifié manuellement par les médecins, a donc beaucoup moins de colonnes et de valeurs manquantes.

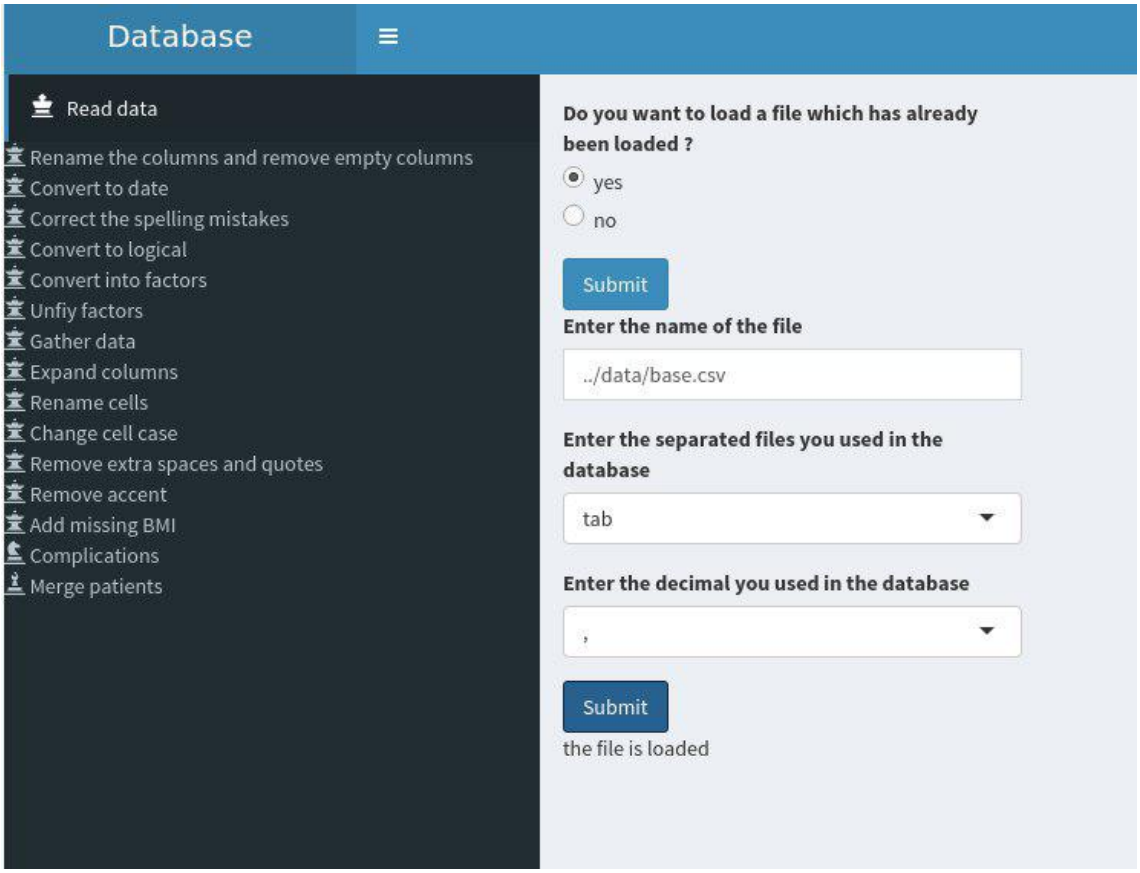


FIGURE 5.1 – Écran d'accueil une fois un fichier de données patients chargé dans le logiciel. En plus des informations relatives aux délimiteurs du fichier, l'utilisateur dispose dans le panneau gauche d'un large éventail de fonctions permettant de traiter, nettoyer, fusionner et enfin effectuer un contrôle du jeu de données.



FIGURE 5.2 – Panneau de conversion de colonnes en dates. Le format employé est flexible et peut être spécificé indépendamment pour chaque groupe de colonnes. Plusieurs formats cohabitants au sein d'une même colonne peuvent être généralisés en un format de date unique.

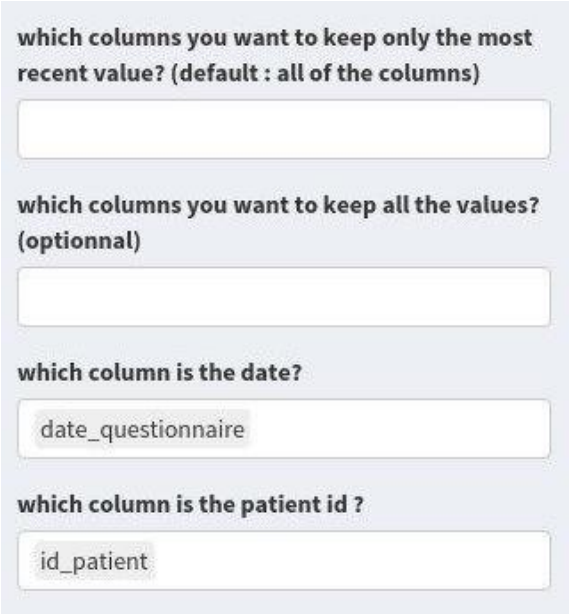


FIGURE 5.3 – Agrégation en un seul élément de tous les enregistrements concernant un patient. Il est possible de spécifier si toutes les valeurs doivent être gardées ou simplement la plus récente.

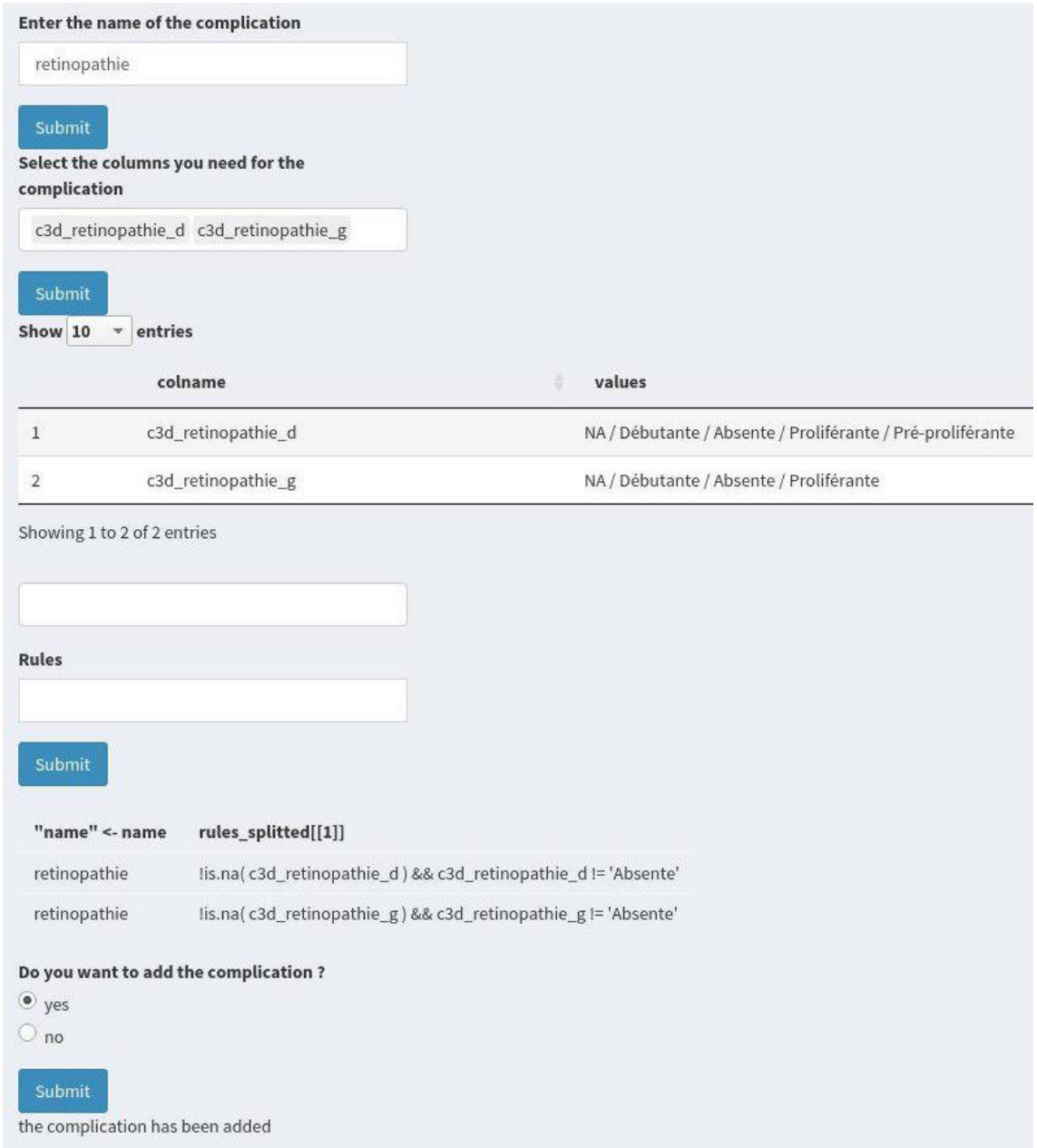


FIGURE 5.4 – Panneau de définition de règles d'imputation pour les complications, à partir des colonnes existantes. Dans cet exemple, les variables de rétinopathie pour l'oeil droit et l'oeil gauche, pouvant chacun avoir différentes modalités, sont utilisées pour définir une nouvelle colonne binaire indiquant simplement la présence ou l'absence d'une rétinopathie chez le patient.

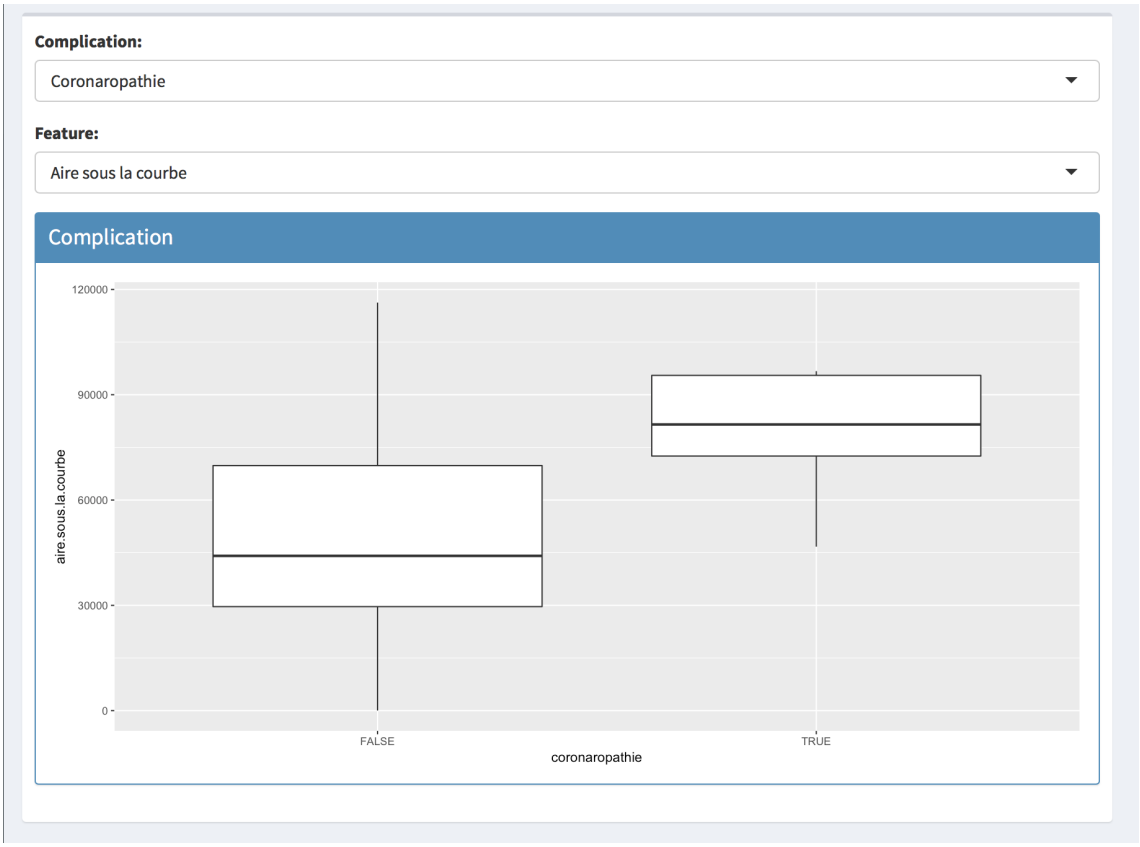


FIGURE 5.5 – Analyse des corrélations entre les variables et les complications. Il est possible de sélectionner des couples variable-complication pour vérifier manuellement la présence de corrélations potentielles entre la variable sélectionnée et la survenue d'une complication.

5.3 Prototype d'exploration

Une fois les données pré-traitées et afin de mettre notre méthode de structuration et visualisation à disposition des utilisateurs, nous avons réalisé un prototype d'interface web permettant d'explorer, par similarité, les cohortes précédemment traitées. Réalisé en *R* et *Shiny*, nous avons pu le déployer et le maintenir aisément grâce à un container Docker [1] de sorte à pouvoir aussi facilement l'exécuter sur un serveur que sur toute autre machine personnelle en ayant simplement installé Docker au préalable. L'application est divisée en deux zones, visibles sur la Figure 5.7 : un panneau latéral de sélection de la tâche et de paramétrage, et un large panneau central dédié à la tâche actuellement sélectionnée. Le cas d'usage habituel de l'application comporte les étapes suivantes :

— **Filtrage**. L'utilisateur commence par sélectionner les patients qu'il souhaite étudier (Annexe A.2). Il a la possibilité de choisir les patients atteints ou non de certaines complications du diabète, de retenir ceux dont la charge glyquée est supérieure à un certain seuil, ou dont le diabète est déclaré depuis un certain nombre d'années. Ces filtres permettent de sélectionner précisément la population qui sera étudiée.

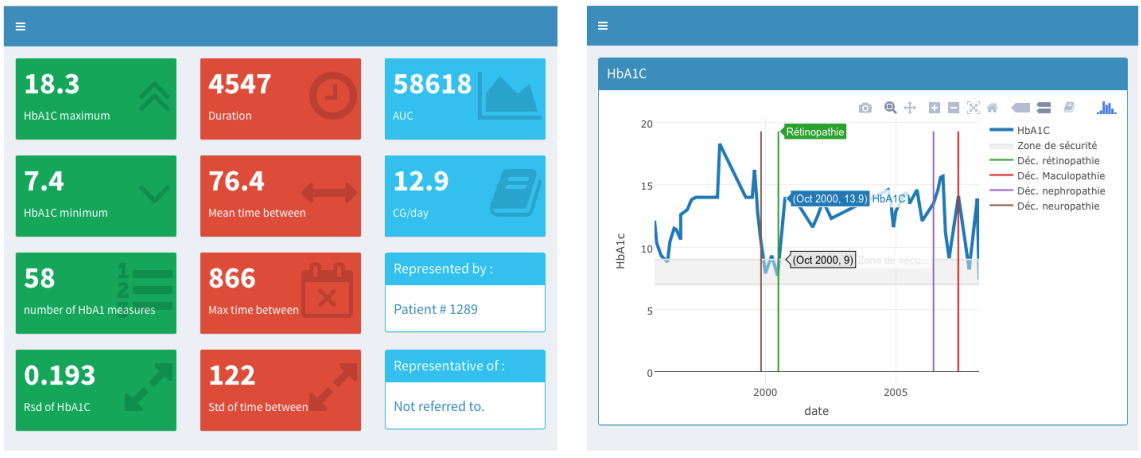
1. `https://www.docker.com`

— **Vérification**. Via une présentation sous forme de tableau dynamique, il est possible de vérifier le résultat du filtrage précédent pour éventuellement itérer jusqu'à sélection de l'échantillon des patients désirés (Annexe A.3). L'utilisateur peut, à cette étape, exporter ses données au format tableur et ainsi uniquement se servir du prototype comme interface simplifiée pour émettre des requêtes sur la base de patients.

— **Analyse exploratoire**. Il est possible de commencer l'exploration du jeu de données en étudiant la survenue des complications du diabète. Pour cela, le prototype permet de visualiser la survenue d'une complication choisie en fonction d'une variable du jeu de données (Figure 5.5). Cette étape manuelle fournit à l'utilisateur un moyen de formuler des hypothèses initiales qu'il peut continuer d'étudier grâce aux autres outils disponibles.

— **Exploration par similarité**. En choisissant ou non un patient de départ grâce à la fonction de recherche, l'expert médical peut ensuite utiliser le graphe de similarité visible sur la Figure 5.7 pour naviguer au sein de la cohorte par similarité entre patients, au sein de composantes connexes composées de patients similaires.

— **Étude d'un patient**. En sélectionnant un patient dans le graphe précédent ou via la fonction de recherche, l'utilisateur peut consulter sa fiche médicale ici anonymisée (Figure 5.6). Celle-ci comporte les informations relatives à la maladie diabétique tels que l'historique d'hémoglobine glyquée, les complications rencontrées, et le détail de son suivi. La fiche patient contient aussi, pour chaque individu, le patient choisi comme représentant par l'individu actuel, et la liste des patients ayant choisi l'individu actuel comme représentant. Cela permet de naviguer de proche en proche et de pouvoir vérifier les corrélations entre différents motifs relatifs à la charge glyquée (HbA1c) et la survenue de complications.



(A) (B)

FIGURE 5.6 – La fiche médicale d'un patient. On y retrouve des indicateurs relatifs au suivi du patient sur la durée et à son hémoblogine glyquée au court du temps (A), ainsi qu'un graphique représentant cette HbA1c et les complications liées au diabète (B).

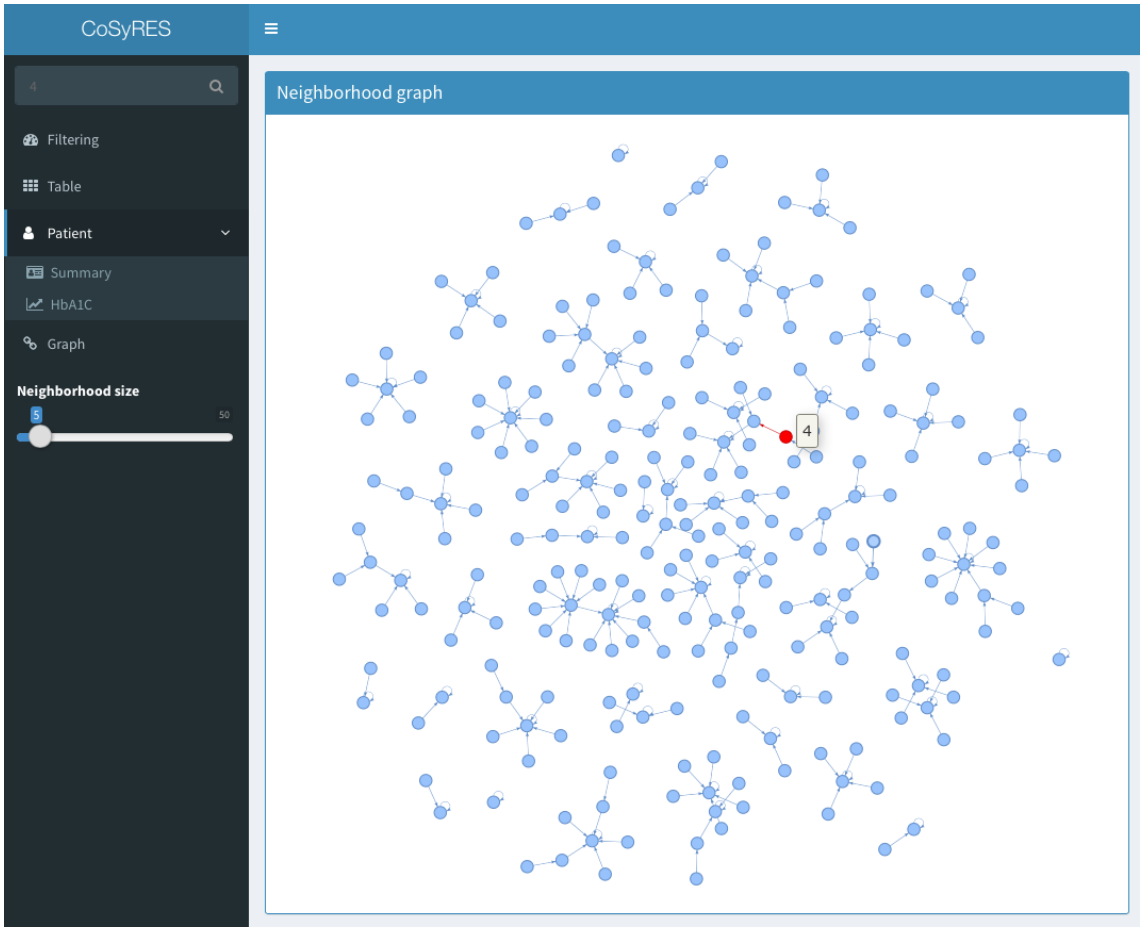


FIGURE 5.7 – Graphe obtenu au sein du prototype d'application. Les patients de la cohorte étudiée sont reliés par similarité afin de proposer à l'utilisateur une exploration guidée de ses données. Le patient actuellement étudié figure en rouge.

Chapitre 6

Conclusion

Sommaire

6.1 Structuration par représentativité 97
6.1.1 Contribution 97
6.1.2 Discussion 98
6.2 Prototype d'aide au diagnostic médical 99
6.2.1 Contribution 99
6.2.2 Discussion 99
6.3 Perspectives 100

Dans le dernier chapitre de ce manuscrit, nous présentons un résumé des contributions au domaine de l'extraction d'information et de l'aide à la décision introduites par ce travail de thèse. Nous mettons aussi en perspective et discutons les avantages et faiblesses des méthodes exposées. Pour cela, la première section se concentre sur l'algorithme de structuration par représentativité et l'algorithme de détection d'anomalies et la deuxième est dédiée au prototype d'exploration de données médicales. Dans chacune de ces deux sections nous résumons tout d'abord la contribution apportée avant d'en discuter les points positifs ainsi que les critiques qui peuvent lui être adressées. Enfin, la dernière section dresse un bilan des perspectives ouvertes par les travaux présentés et les poursuites que nous pouvons anticiper à ce travail.

6.1 Structuration par représentativité

6.1.1 Contribution

Nous proposons dans cette thèse une méthode de structuration des données par représentativité et similarité. Pour enrichir les données grâce à cette structure, nous proposons tout d'abord un calcul du degré de représentativité (*DoR*) d'un élément décrit sur plusieurs dimensions. Ce *DoR* quantifie la similarité et la capacité d'un élément à agir comme représentant d'autres éléments en les subsumant. Le *DoR* est calculé pour un élément à partir d'un autre élément, différent du premier. Ainsi un élément donné se verra attribuer un *DoR* élevé par un élément similaire, mais un *DoR* très faible par un élément différent qu'il n'est pas à même de représenter. Il est ensuite possible d'obtenir une structure de graphe en liant chaque élément d'un ensemble à l'élément auquel il attribue le plus haut *DoR*, et d'isoler ainsi les attributs ayant contribué à chaque association.

La structure obtenue, un graphe composé d'arcs entre chaque élément et son représentant, peut être utilisé afin d'explorer les données de proche en proche, par similarité. Il est aussi possible de l'utiliser pour extraire les éléments centraux de chaque composante connexe afin d'obtenir une vue résumée des données. Comparativement aux graphes obtenus par d'autres méthodes de recherche d'éléments similaires, le graphe résultant de la méthode proposée présente des avantages pour l'analyse exploratoire des données : peu de composantes connexes, et un diamètre généralement plus élevé de 10% à 100%. De plus, dans un jeu de données composé de plusieurs classes, les relations établies entre les éléments sont pertinentes du point de vue de la micro-fmesure et en comparaison avec d'autres approches de structuration.

Nous avons tenté d'ajouter au prototype un algorithme de détection d'anomalies qui soit adapté aux problèmatiques de haute dimensionnalité et d'interprétabilité. Pour cela, nous avons implémenté une méthode de calcul de *Rareness* en se basant sur une approche unidimensionnelle, comme pour le *DoR*. Ainsi le score de *Rareness* d'un individu indique la probabilité qu'il puisse être considéré comme *outlier* au sein des données. La performance de la *Rareness* a été comparée à une autre méthode courante pour les données en haute dimension : le *Subspace Outlier Degree*.

6.1.2 Discussion

La méthode dévelopée dans ce manuscrit propose des résultats qui, d'après les critères de transparence et d'explicabilité, peuvent être jugés comme interprétables. Il est en effet possible d'isoler les dimensions qui sont sources de similarité entre deux éléments et les résultats sont basés sur des éléments réels, issus des données étudiées. Cette utilisation de données concrètes dans le résultat final permet de favoriser le raisonnement par analogie, majoritairement utilisé par les spécialistes tels que le personnel soignant qui était le public visé par le prototype créé. De cette manière, l'expérience de l'utilisateur peut être mise à profit puisque l'algorithme, basé instance, capitalise dessus.

Le second point clé de cette méthode est son adaptabilité : il est possible de valoriser les connaissances *a priori* de l'utilisateur en appliquant des mesures de similarité spécifiques et adaptées, pour chaque dimension. L'algorithme travaille de manière unidimensionnelle, avant d'agréger les résultats. En plus d'être plus facile à appréhender dans un but d'explicabilité, cela permet de travailler sur des données en haute-dimension avec un impact réduit de la « malédiction de la dimensionnalité ». Ainsi, des problèmes comme ceux de la concentration des distances ou des espaces « creux » ne se présentent pas dans ce cas d'utilisation.

Enfin, dans le cadre de l'analyse exploratoire de données, il est possible d'obtenir une structure plus simple à visualiser qu'une projection des points dans un espace à deux ou trois dimensions. Les éléments étudiés sont en effet présentés par similarité, mais aussi hiérarchisés, ce qui permet à l'utilisateur d'être guidé dans son exploration et d'isoler les éléments les plus représentatifs du *dataset* par exemple, ou encore de naviguer de proche en proche afin d'explorer les données par similarité. De plus, la granularité de la structure peut être facilement modifiée par un unique paramètre afin de varier l'importance accordée à la similarité.

Ces points positifs sont à pondérer entre autres par la complexité en temps de l'algorithme, relativement élevée. Les dimensions étant traitées séparément, chaque nouvelle dimension augmente d'autant la durée du calcul. Ainsi, sur des données à plus d'une centaine de dimensions ou avec plusieurs milliers d'éléments, il est impossible d'utiliser l'algorithme en temps interactif. Cela n'est toutefois pas un obstacle majeur, puisque la structure des données peut être recalculée périodiquement, en tâche de fond, lorsque l'utilisateur ne visualise pas le graphe par exemple. L'un des avantages proposés peut aussi s'avérer être un incovénient dans certains cas d'utilisation : la définition d'un indice de dissimilarité sur chaque dimension impacte directement la qualité du résultat obtenu. Bien que les indices de dissimilarité les plus classiques soient efficaces, il est nécessaire d'être attentif lors de la redéfinition de ces indices.

Notons aussi le fait que le *DoR*, bien que quantifiant la similarité, n'est pas une métrique de similarité. Ainsi il est impossible de comparer le *DoR* d'un premier couple d'éléments avec un second couple d'éléments. Il n'est pas possible non plus de donner une échelle absolue : dans un jeu de données structuré sous la forme d'un graphe à deux composantes connexes, les éléments les plus représentatifs de la première et de la seconde composante peuvent avoir des *DoR* variant très largement, ce qui rend la comparaison inappropriée.

La détection d'anomalies permet de déceler les anomalies plus complexes, qui ne peuvent être observées par la distribution sur une seule dimension. Elle présente aussi l'avantage d'une explicabilité accrue en pouvant isoler sur quelle dimension et par rapport à quels éléments l'individu étudié est une anomalie. Cependent les tests de cette approche sur des données en grande dimension révèlent une méthode qui n'est efficace que dans une minorité de cas. C'est pourquoi nous lui préférons, dans notre contexte applicatif, d'autres approches d'*outlier detection* multidimensionnelles telles que le *Subspace Outlier Degree*. Précisons aussi que le calcul de la *Rareness* exigeant toutes les combinaisons deux à deux possibles pour les dimensions, la durée des calculs rend cette méthode prohibitive à mettre en oeuvre sur des données conséquentes aux centaines d'éléments et de descripteurs.

6.2 Prototype d'aide au diagnostic médical

6.2.1 Contribution

Nous avons développé une plateforme web afin de pouvoir implémenter la méthode proposée dans un contexte applicatif concret. L'origine de cette plateforme vient du souhait exprimé par les profesionnels de santé avec qui nous avons travaillé de pouvoir mieux comprendre et manipuler leurs données afin d'en tirer un intérêt concret dans l'exercice de leur métier. Cet outil que nous proposons se base sur des technologies comme *R*, *Shiny* et *Docker* afin d'offrir un déploiement facilité et une interface réactive et adaptée à tout type de support, y compris tablettes et mobiles. Après avoir prétraité les données à l'aide d'une autre interface web que nous avons développée, nous permettons au personnel soignant avec lequel nous collaborons de visualiser une base de données de suivi de patients diabétiques.

Dans cet outil d'aide au diagnostic, les individus sont structurés par similarité dans une visualisation interactive sous forme de graphe orienté où chaque point correspond à un patient, relié par un arc à un autre patient qui peut être vu comme son représentant. Il est par exemple possible d'analyser chaque composante connexe de ce graphe comme une typologie différente de patients, et d'extraire les individus les plus représentatifs de cette typologie comme exemples. Les individus centraux peuvent être identifiés comme étant des individus très archétypaux et à même de résumer un large ensemble de cas.

6.2.2 Discussion

Le prototype d'exploration et de recommandation de patients similaires propose des avantages tels que la simplicité d'accès pour l'utilisateur : seul un navigateur web est nécessaire et aucune installation n'est requise. C'est d'ailleurs pour cette raison que le prototype est déjà disponible auprès du service Diabète-Endocrinologie-Nutrition avec lequel nous collaborons. Concernant la visualisation proposée, elle offre un résumé concis de la structure des éléments et de la manière dont se découpe le jeu de données ainsi que des typologies de patients qui le composent. Il est facile de modifier l'unique paramètre de granularité afin de voir l'évolution de la structuration, et de manière à obtenir différents niveaux d'information.

Toutefois le prototype n'a pas été évalué formellement en terme d'utilisabilité et d'expérience utilisateur. De plus, un déploiement dans un cadre applicatif plus large nécessiterait plus de temps de développement pour ajouter des fonctionnalités adaptées : par exemple il n'est pas possible dans la version actuelle de charger des données dans un autre format que celui utilisé pour les données de suivi des patients diabétiques. L'absence de ces fonctionnalités est à relativiser du fait de la construction même du prototype. En effet, grâce à une programmation en micro-services reposant sur des briques logicielles distinctes, il est possible d'ajouter facilement de nouvelles fonctionnalités, de modifier celles déjà existantes, et éventuellement de pouvoir créer plusieurs versions adaptées à différents contextes applicatifs. La plateforme a en effet été conçue pour être une base robuste et générique sur laquelle plusieurs méthodes d'analyse exploratoire peuvent cohabiter, plutôt qu'un outil de visualisation uniquement dédié à la structuration par *DoR*.

6.3 Perspectives

Nous pouvons, à partir de là, envisager plusieurs poursuites à ce travail. Tout d'abord technologiquement et concernant le prototype, il est envisageable d'y intégrer de nouvelles méthodes de visualisation et d'analyse de données afin d'offrir un plus large éventail de solutions. On peut imaginer intégrer des solutions alternatives de *clustering* et de structuration, plusieurs algorithmes pour la détection d'anomalies ou encore des algorithmes de classification. Il serait intéressant d'ajouter de nouvelles méthodes de prétraitement des données, permettant de charger des jeux de données d'autres domaines médicaux. Optimiser l'implémentation de l'algorithme de structuration serait un gain direct, afin de réduire les temps d'exécution avec une exécution en parallèle ou grâce à des améliorations plus proches de la machine comme les *SSE*. Nous pouvons citer la mise en place d'une étude d'utilisabilité sur l'interface comme une perspective complémentaire, où il serait possible de comparer ce type de structuration et de visualisation à ceux offerts par d'autres outils professionnels existants. Toutes ces perspectives en lien avec le prototype médical ont pour objectif commun d'augmenter l'aide apportée aux médecins dans la valorisation des données. Nous pouvons ainsi espérer extraitre plus d'informations des données existantes et, par un système fonctionnant sur un principe de recommandation, aider l'utilisateur à naviguer et explorer ces données qui sont habituellement sans structure inhérente.

Plusieurs poursuites sont envisageables pour l'algorithme de structuration. Tout d'abord, élargir le *benchmarking* à d'autres types de *datasets* que les données bio-informatique majoritairement utilisées dans nos expérimentations. Il serait aussi possible d'étendre ce travail à d'autres domaines que le médical, en vérifiant son apport à des rôles reposant sur la prise de décision : la finance, la cybersécurité, l'assurance pour n'en citer que quelques uns. Ensuite, si nous revenons à la structuration sous forme de graphe, de nombreuses perspectives existent : la première serait d'appliquer des méthodes d'analyse de graphe à cette structure. Par exemple s'en servir comme base pour de la détection de communauté afin d'isoler certains sous-groupes d'individus. Il est aussi possible de transposer de nombreux indicateurs aux individus du graphe, comme celui de centralité par exemple, et d'étudier leur pertinence et leur sens dans ce contexte.

Il serait intéressant d'étudier toutes ces propriétés au fur et à mesure de l'évolution du paramètre de granularité, plutôt que d'en fixer un seul. Nous n'utilisons à l'heure actuelle qu'une seule structure finale, mais il serait toutefois possible de réaliser une structuration en graphe pour chaque dimension des données. Ainsi, plutôt qu'étudier quel élément est le plus représentatif des données, il devient possible d'analyser la représentativité des éléments sur chaque variable pour étudier un niveau de granularité différent et éventuellement faire émerger des hypothèses différentes. Toujours concernant l'algorithme de structuration, il serait envisageable d'intégrer un mécanisme d'apprentissage par renforcement : en donnant un moyen à l'utilisateur d'indiquer si une association entre éléments lui a été utile, il serait possible de pondérer l'importance accordée à chaque variable lors du processus de structuration. Cette pondération pourrait ainsi évoluer au gré du retour des utilisateurs, et chaque utilisateur se verrait affecter un profil contenant les pondérations obtenues grâce au *feedback* qu'il a fourni, chacun obtenant des structurations personnalisées pour ses besoins. Des profils pourraient aussi être établis suivant les spécialités cliniques par exemple.

Les perspectives évoquées pour ces travaux laissent envisager un horizon où l'intelligence artificielle, loin de supplanter le médecin, pourrait l'assister pour en augmenter les capacités. Dans ce but, l'intelligence fournie par l'outil prendrait la forme de suggestions d'exploration, d'aide à la manipulation et à la compréhension des données pour permettre au professionnel de valoriser le coeur de son métier : mettre son expérience, ses connaissances et son savoir au profit du patient. En infusant les briques logicielles des systèmes de santé d'une intelligence basée sur le raisonnement des professionnels qui s'en servent, l'utilisation de ces logiciels devient plus transparente, naturelle et efficace : des atouts majeurs pour du personnel de santé souvent très sollicité par des patients en nombre toujours plus grand. Mais au-delà du domaine médical, le modèle proposé est en décrochement avec l'ubiquité des boîtes noires qui influent sur le quotidien de chacun. Ce manuscrit n'est donc qu'une contribution à une vision plus transparente et éthique de l'intelligence artificielle, dans laquelle l'utilisateur garde plus de contrôle sur les outils qu'il choisit et où la compréhensibilité lui permet de garder un regard critique sur les résultats obtenus.

Annexe A

Annexes

A.1 Traitement des colonnes

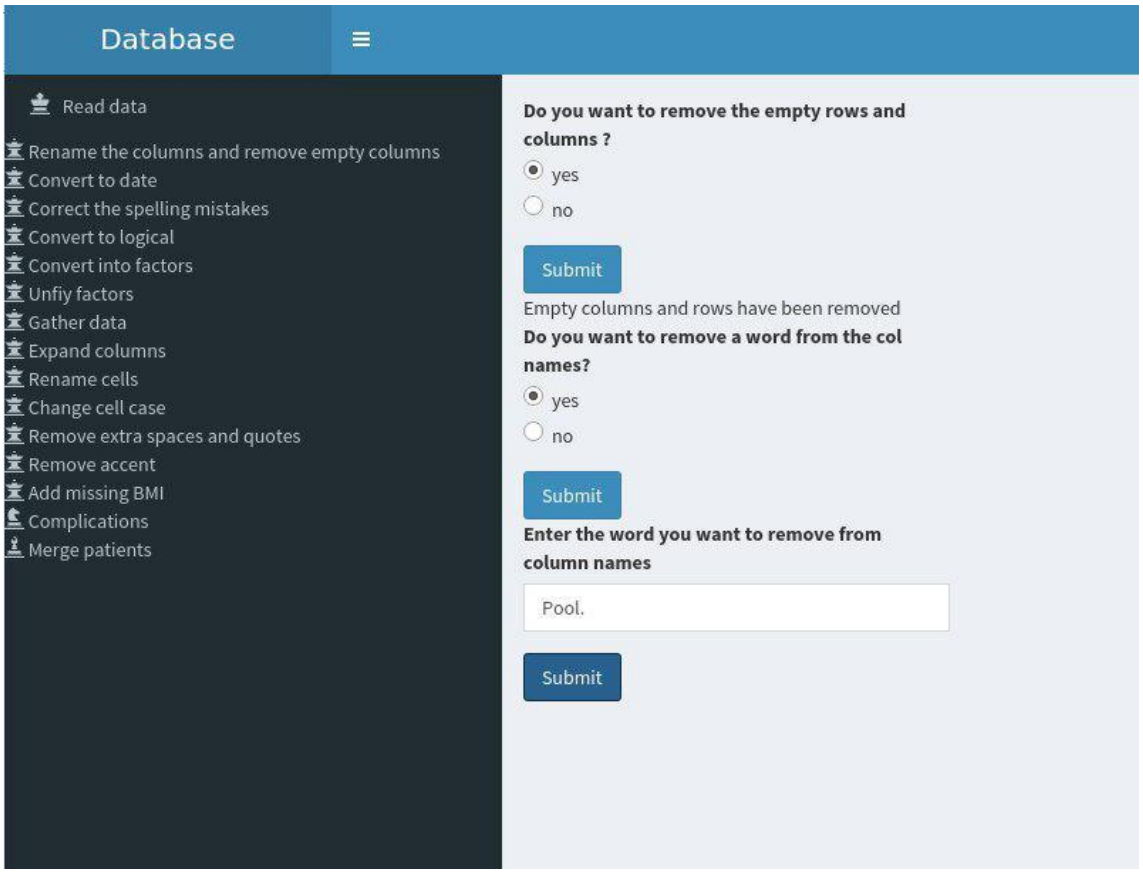


FIGURE A.1 – Interface de gestion des colonnes d'un jeu de données, avec la possibilité de retirer les colonnes vides (ainsi que les lignes vides) et de renommer par lot l'ensemble de colonnes en retirant par exemple un préfixe ou suffixe superflu.

A.2 Filtrage des patients

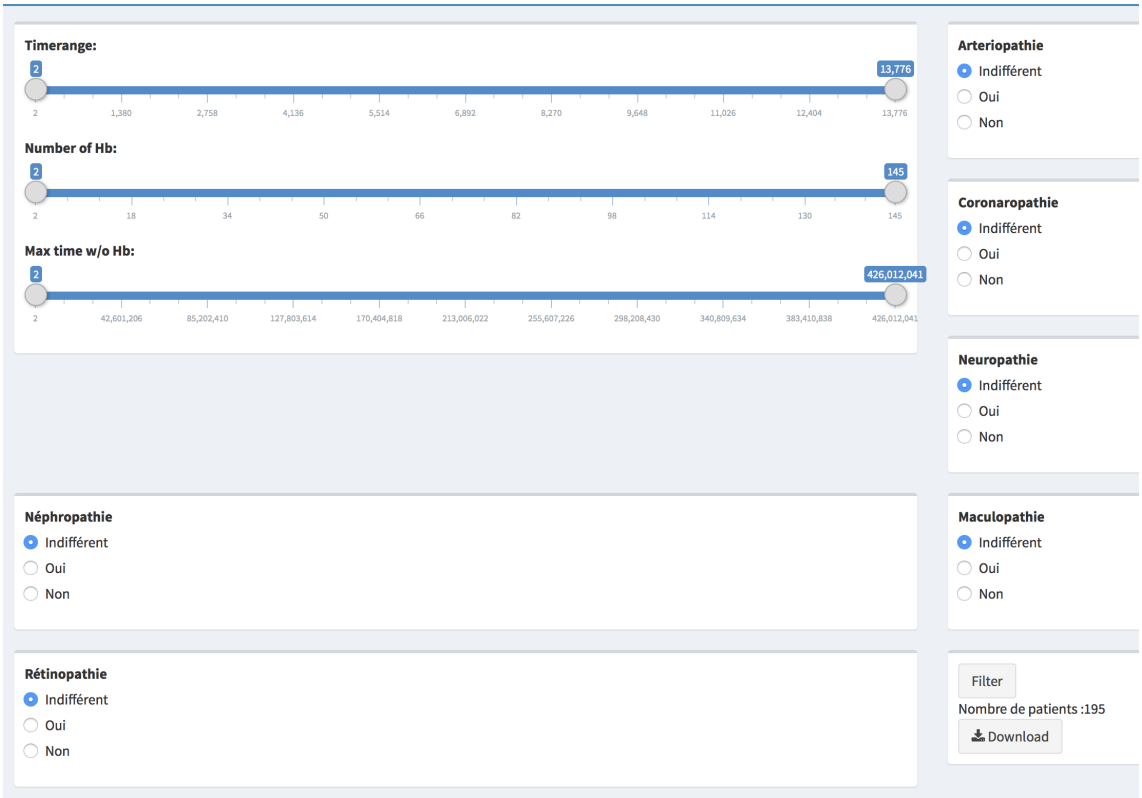


FIGURE A.2 – Interface de filtrage des patients étudiés grâce au prototype. Il est possible de sélectionner uniquement les patients suivis depuis un certains temps, ayant fait un certains nombre de prélèvement, ou affectés par certaines complications spécifiques du diabète. Il est aussi possible de télécharger un fichier CSV comprenant les patients sélectionnés.

A.3 Vérification du filtrage

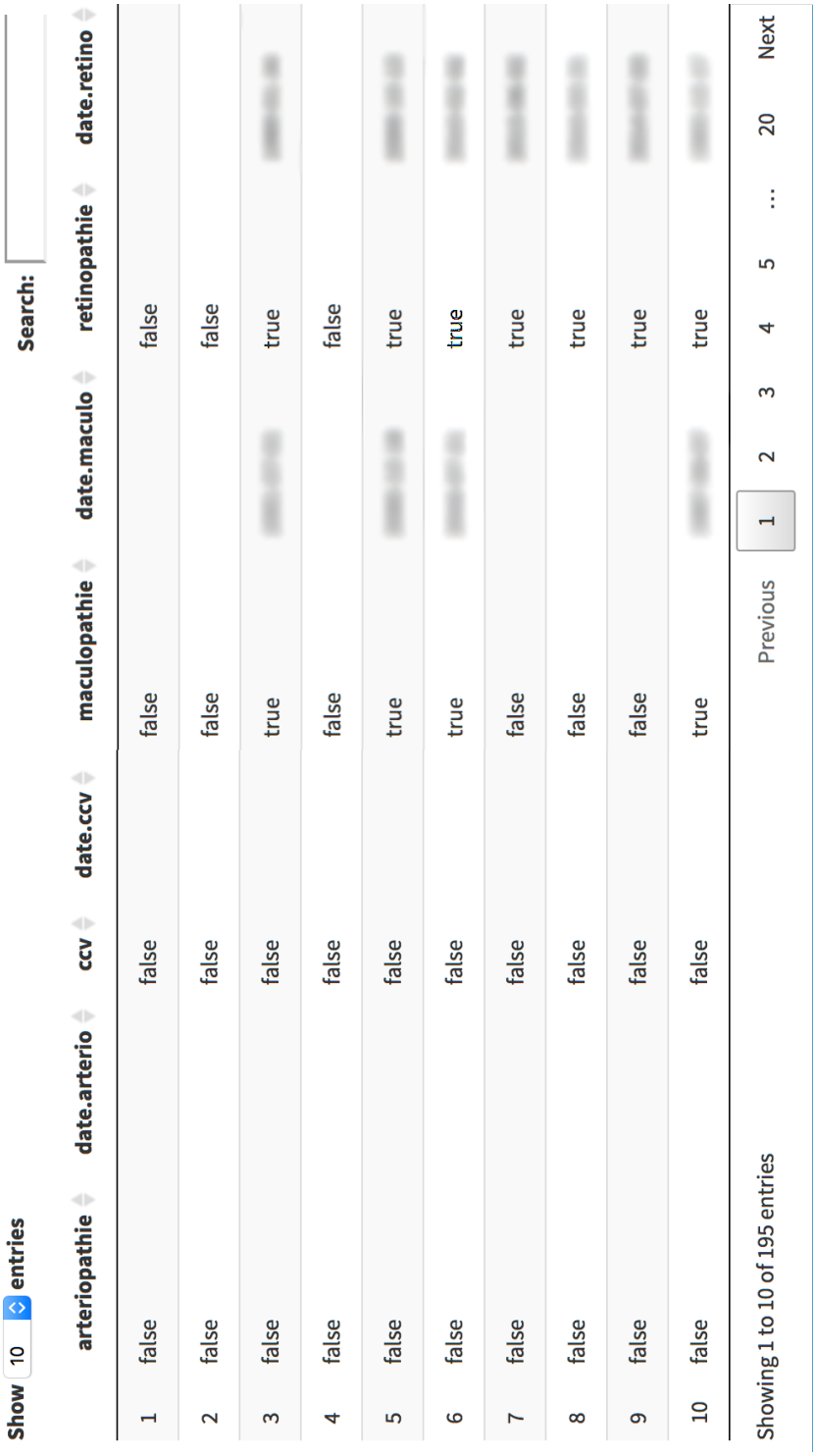


FIGURE A.3 – Écran récapitulatif de la sélection des patients choisis. Une fonction de recherche est disponible, et il est possible de trier chaque colonne. Cette interface permet d'itérer sur les options de sélectionner jusqu'à pouvoir identifier la population cible désirée.

Colophon

Les interlignes sont de 1 et la police utilisée pour le corps du texte est de 12pt. Le manuscrit est écrit en *RMarkdown*, le code *R* executé par *knitr* et intégré dans un fichier *Markdown* puis transformé en *LaTeX* grâce à *pandoc* pour être finalement compilé en *pdf* par *pdflatex*. Les packages utilisés pour la mise en page sont *bookdown*, *thesisdown* et *huskydown*.

Le *template* utilisé à inspiré de celui fournit par l'université de Reims Champagne-Ardenne et mis à jour par Nicolas Courilleau. Les consignes de mise en page sont celles imposées par l'école doctorale Sciences du Numérique et de l'Ingénieur de l'URCA.

Les fichiers source et les données utilisés pour générer les illustrations, les tables et tous les calculs des *benchmarks* sont disponibles à l'adresse suivante : `https://github.com/elesday/thesis`. Toute question peut être adressée à l'auteur par email à *joris@datacrunch.sh*.

Ce manuscrit a été généré le 2019-11-27 11 :21 :49 et utilise les fichiers du *commit* suivant :

```
Commit:  62b7a9d40a3e1cfe6d92e66ba4ffa430292d826b
Author:  Datacrunch.sh <joris.falip@univ-reims.fr>
When:    2019-11-26 15:16:07

     Added acknowledgements

3 files changed, 75 insertions, 63 deletions
_book/thesis.pdf | - 2 + 2  in  1 hunk
_book/thesis.tex | -60 +62  in 12 hunks
index.Rmd        | - 1 +11  in  1 hunk
```

L'environnement de développement utilisé pour générer cette version est le suivant :

```
setting  value
version  R version 3.6.1 (2019-07-05)
os       Windows 10 x64
system   x86_64, mingw32
ui       RTerm
language (EN)
collate  French_France.1252
ctype    French_France.1252
tz       Europe/Paris
date     2019-11-27
```

package	loadedversion	date	source
apcluster	1.4.8	2019-08-21	CRAN (R 3.6.1)
bookdown	0.13	2019-08-21	CRAN (R 3.6.1)
cluster	2.1.0	2019-06-19	CRAN (R 3.6.1)
datamicroarray	0.2.3	2019-09-13	Github (ramhiser/datamicroarray@a95f6f6)
devtools	2.2.0	2019-09-07	CRAN (R 3.6.1)
doParallel	1.0.15	2019-08-02	CRAN (R 3.6.1)
dotCall64	1.0-0	2018-07-30	CRAN (R 3.6.1)
dplyr	0.8.3	2019-07-04	CRAN (R 3.6.1)
fields	9.8-6	2019-08-19	CRAN (R 3.6.1)
FNN	1.1.3	2019-02-15	CRAN (R 3.6.1)
foreach	1.4.7	2019-07-27	CRAN (R 3.6.1)
ggnet	0.1.0	2019-09-13	Github (briatte/ggnet@da9a7cf)
ggplot2	3.2.1	2019-08-10	CRAN (R 3.6.1)
ggpubr	0.2.3	2019-09-03	CRAN (R 3.6.1)
git2r	0.26.1	2019-06-29	CRAN (R 3.6.1)
gridExtra	2.3	2017-09-09	CRAN (R 3.6.1)
HighDimOut	1.0.0	2015-04-02	CRAN (R 3.6.1)
igraph	1.2.4.1	2019-04-22	CRAN (R 3.6.1)
iterators	1.0.12	2019-07-26	CRAN (R 3.6.1)

knitr	1.24	2019-08-08	CRAN (R 3.6.1)
magrittr	1.5	2014-11-22	CRAN (R 3.6.1)
maps	3.3.0	2018-04-03	CRAN (R 3.6.1)
network	1.15	2019-04-02	CRAN (R 3.6.1)
purrr	0.3.2	2019-03-15	CRAN (R 3.6.1)
readr	1.3.1	2018-12-21	CRAN (R 3.6.1)
readxl	1.3.1	2019-03-13	CRAN (R 3.6.1)
rmatio	0.14.0	2019-03-18	CRAN (R 3.6.1)
scales	1.0.0	2018-08-09	CRAN (R 3.6.1)
sna	2.4	2016-08-08	CRAN (R 3.6.1)
spam	2.2-2	2019-03-08	CRAN (R 3.6.1)
statnet.common	4.3.0	2019-06-02	CRAN (R 3.6.1)
thesisdown	0.0.2	2019-09-13	Github (ismayc/thesisdown@077f725)
tibble	2.1.3	2019-06-06	CRAN (R 3.6.1)
usethis	1.5.1	2019-07-04	CRAN (R 3.6.1)
yardstick	0.0.4	2019-08-26	CRAN (R 3.6.1)

Bibliographie

Aggarwal, C. C. (2012). Outlier ensembles - position paper. *SIGKDD Explorations*, *14*, 49.

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the Surprising Behavior of Distance Metrics in High Dimensional Space. In *Database theory - icdt 2001* (Vol. 1973, pp. 420–434).

Aggarwal, C. C., & Yu, P. S. (2001). Outlier detection for high dimensional data. *ACM Sigmod Record*, *30*, 37–46.

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, *22*, 207–216.

Ait-Younes, A., Blanchard, F., Delemer, B., & Herbin, M. (2014). Singular profile of diabetics. *I4CS*, 104–107.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, *96*, 6745–6750.

Anderson, E. (1936). The species problem in Iris. *Annals of the Missouri Botanical Garden*, *23*, 457.

Ayres-de-campos, D., Bernardes, J., Garrido, A., Marques-de-sa, J., & Pereira-leite, L. (2009). SisPorto 2.0 : A Program for Automated Analysis of Cardiotocograms. *Journal of Maternal-Fetal and Neonatal Medicine*, *9*, 311–318.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Muller, K.-R., & Samek, W. (2015). On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, *10*, 0130140.

Behrens, J. T. (1997). Principles and procedures of exploratory data analysis., *2*, 131–160.

Bellman, R. E. (1961). *Adaptive Control Processes : A Guided Tour*. Princeton University Press.

Beyer, K. S., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When Is ”Nearest Neighbor” Meaningful ? *ICDT*, *1540*, 217–235.

Bicocchi, N., Mamei, M., & Zambonelli, F. (2010). Detecting activities from body-worn

accelerometers via instance-based algorithms. *Pervasive and Mobile Computing*, *6*, 482–495.

Blanchard, F., Ait-Younes, A., & Herbin, M. (2015). Linking Data According to Their Degree of Representativeness (DoR). *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, *2*.

Blanchard, F., Runz, C. de, Akdag, H., & Herbin, M. (2011). Representativite et graphe de representants : une approche inspiree de la theorie du choix social pour la fouille de donnees relationnelles. In *Extraction et gestion des connaissances workshops*.

Blanchard, F., Vautrot, P., Akdag, H., & Herbin, M. (2010). Data Representativeness Based on Fuzzy Set Theory. *Journal of Uncertain Systems*, *4*, 216–228.

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, *97*, 245–271.

Borda, J. C. de. (1781). *Memoire sur les Elections au Scrutin*.

Bower, G., & Trabasso, T. (1963). Concept identification.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, J. S. (1984). *Classification and regression trees*.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF - Identifying Density-Based Local Outliers. *SIGMOD Conference*, 93–104.

Buza, K., Nanopoulos, A., & Schmidt-Thieme, L. (2011). INSIGHT : Efficient and Effective Instance Selection for Time-Series Classification. In *Advances in knowledge discovery and data mining* (Vol. 6634, pp. 149–160).

Cai, C. J., Jongejan, J., & Holbrook, J. (2019). The effects of example-based explanations in a machine learning interface. In *Intelligent user interface* (pp. 258–262).

Chang, W., Cheng, J., Allaire, J. J., Xie, Y., & McPherson, J. (2019). *shiny : Web Application Framework for R*.

Christensen, B. C., Houseman, E. A., Marsit, C. J., Zheng, S., Wrensch, M. R., Wiemels, J. L., … Kelsey, K. T. (2009). Aging and Environmental Exposures Alter Tissue-Specific DNA Methylation Dependent upon CpG Island Context. *PLOS Genetics*, *5*, e1000602.

Conover, W. J., & Iman, R. L. (2012). Rank Transformations as a Bridge between Parametric and Nonparametric Statistics. *The American Statistician*, *35*, 124–129.

Dodge, J., Liao, Q. V., & Bellamy, R. K. E. (2019). Explaining Models : An Empirical Study of How Explanations Impact Fairness Judgment. In *Intelligent user interface* (pp. 275–285).

Donoho, D. L. (2000). High-Dimensional Data Analysis : The Curses and Blessings of Dimensionality. *AMS Math Challenges Lecture*, *1*, 32.

Dua, D., & Graff, C. (2019). UCI Machine Learning Repository.

Einhorn, H. J., & Hograth, R. M. (1978). Confidence in judgment : Persistence of the

illusion of validity. *Psychological Review*, *85*, 395–416.

Erevelles, S., Fukawa, N., & Swayne, L. (2016). Big Data consumer analytics and the transformation of marketing. *Journal of Business Research*, *69*, 897–904.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *KDD*.

Evett, I. W., & Spiehler, E. J. (1989). *Rule induction in forensic science*. Halsted Press.

Feldbauer, R., & Flexer, A. (2018). A comprehensive empirical comparison of hubness reduction in high-dimensional spaces. *Knowledge and Information Systems*, *59*, 1–30.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*, 179–188.

Fix, E., & Hodges, J. L. (1951). *Discriminatory analysis, non-parametric discrimination, USAF School of Aviation Medicine, Randolph Field, TX*.

Frey, B. J., & Dueck, D. (2007). Clustering by Passing Messages Between Data Points. *Science*, *315*, 972–976.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., … Lander, E. S. (1999). Molecular classification of cancer : class discovery and class prediction by gene expression monitoring. *Science*, *286*, 531–537.

Gosiewska, A., & Biecek, P. (2019). iBreakDown : Uncertainty of Model Explanations for Non-additive Predictive Models. *arXiv.org*, *cs.LG*.

Guilford, J. P. (1954). Psychometric methods.

Guvenir, H. A., Acar, B., Demiroz, G., & Cekin, A. (1997). A supervised machine learning algorithm for arrhythmia analysis. In *Computers in cardiology* (pp. 433–436).

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., & Stahel, W. A. (2011). *Robust Statistics*. John Wiley & Sons.

Herbin, M., Gillard, D., & Hussenet, L. (2017). Concept of Observer to Detect Special Cases in a Multidimensional Dataset. *I4CS*, *717*, 34–44.

Hinneburg, A., Aggarwal, C. C., & Keim, D. A. (2000). What Is the Nearest Neighbor in High Dimensional Spaces ? *VLDB*.

Hira, Z. M., & Gillies, D. F. (2015). A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data. *Advances in Bioinformatics*, *2015*, 1–13.

Houle, M. E., Kriegel, H.-P., Kroger, P., Schubert, E., & Zimek, A. (2010). Can Shared-Neighbor Distances Defeat the Curse of Dimensionality ? *SSDBM*, *6187*, 482–500.

Hyvarinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, *10*, 626–634.

Indyk, P. (2004). Nearest Neighbors in High-Dimensional Spaces. *Handbook of Discrete*

and Computational Geometry, 2nd Ed.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering : a review. *ACM Computing Surveys (CSUR)*, *31*, 264–323.

Jarvis, R. A., & Patrick, E. A. (1973). Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers*, *C-22*, 1025–1034.

Jayaram, B., & Klawonn, F. (2012). Can unbounded distance measures mitigate the curse of dimensionality ? *International Journal of Data Mining, Modelling and Management*, *4*, 361.

Jebb, A. T., Parrigon, S., & Woo, S. E. (2017). Exploratory data analysis as a foundation of inductive research. *Human Resource Management Review*, *27*, 265–276.

Josephson, J. R., & Josephson, S. G. (1996). *Abductive Inference*. Cambridge University Press.

Kahneman, D., & Klein, G. (2009). Conditions for intuitive expertise : A failure to disagree. *American Psychologist*, *64*, 515–526.

Katz, J. J., & Postal, P. M. (1964). *An Integrated Theory of Linguistic Descriptions*.

Kaufman, L., & Rousseeuw, P. (1987). Clustering by Means of Medoids, 14.

Klein, G., Calderwood, R., & Clinton-Cirocco, A. (2010). Rapid Decision Making on the Fire Ground : The Original Study Plus a Postscript. *Journal of Cognitive Engineering and Decision Making*, *4*, 186–209.

Komorowski, M., Marshall, D. C., Salciccioli, J. D., & Crutain, Y. (2016). Exploratory Data Analysis. In *Secondary analysis of electronic health records* (pp. 185–203).

Kotsiantis, S. B. (2007). Supervised Machine Learning - A Review of Classification Techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*.

Kriegel, H.-P., Kroger, P., Schubert, E., & Zimek, A. (2009). Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data. *PAKDD*, *5476*, 831–838.

Lerman, I.-C. (2016). Foundations and Methods in Combinatorial and Statistical Data Analysis and Clustering. *Advanced Information and Knowledge Processing*, 647.

Lesot, M.-J., Rifqi, M., & Bouchon-Meunier, B. (2008). Fuzzy Prototypes : From a Cognitive View to a Machine Learning Principle. In *Fuzzy sets and their extensions : Representation, aggregation and models* (Vol. 220, pp. 431–452).

Liao, S.-H., Chu, P.-H., & Hsiao, P.-Y. (2012). Data mining techniques and applications - A decade review from 2000 to 2011. *Expert Systems with Applications*, *39*, 11303–11311.

Lipton, Z. C. (2018). The Mythos of Model Interpretability. *ACM Queue*.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. *ICDM*, 413–422.

Liu, H., Li, X., Li, J., & Zhang, S. (2017). Efficient Outlier Detection for

High-Dimensional Data. *IEEE Transactions on Systems, Man, and Cybernetics : Systems*, *48*, 1–11.

Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Trans. Information Theory*, *28*, 129–137.

Maaten, L. van der, & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

Mack, M. L., Preston, A. R., & Love, B. C. (2013). Decoding the Brain's Algorithm for Categorization from Its Neural Implementation. *Current Biology*, *23*, 2023–2027.

Mayer-Schönberger, V., & Cukier, K. (2013). *Big Data : A Revolution that Will Transform how We Live, Work, and Think*.

McAfee, A., & Brynjolfsson, E. (2012). Big data : the management revolution. *Tarjomefa.com*.

Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review*, *85*, 207–238.

Minda, J. P., & Smith, J. D. (2002). Comparing prototype-based and exemplar-based accounts of category learning and attentional allocation. *Journal of Experimental Psychology : Learning, Memory, and Cognition*, *28*, 275.

Mirza, N. A., Danesh, N. A., Noesgaard, C., Martin, L., & Staples, E. (2014). A concept analysis of abductive reasoning. *Journal of Advanced Nursing*, *70*, 1980–1994.

Nosofsky, R. M. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology : General*, *115*, 39–57.

Nosofsky, R. M. (1991). Typicality in logically defined categories : Exemplar-similarity versus rule instantiation. *Memory & Cognition*, *19*, 131–150.

Pearl, J. (2009). Causal inference in statistics : An overview. *Statistics Surveys*, *3*, 96–146.

Pearson, K. (1901). *On Lines and Planes of Closest Fit to Systems of Points in Space*.

Radovanovic, M., Nanopoulos, A., & Ivanovic, M. (2010). Hubs in Space : Popular Nearest Neighbors in High-Dimensional Data. *Journal of Machine Learning Research*, *11*, 2487–2531.

Rafiei, M. H., & Adeli, H. (2016). A Novel Machine Learning Model for Estimation of Sale Prices of Real Estate Units. *Journal of Construction Engineering and Management*, *142*, 04015066.

Rayana, S. (2016). ODDS Library.

R Core Team. (2019). *R : A Language and Environment for Statistical Computing*.

Redmond, M. (2009). Communities and crime data set.

Reed, S. K. (1972). Pattern recognition and categorization. *Cognitive Psychology*, *3*, 382–407.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *Why Should I Trust You ? : Explaining the Predictions of Any Classifier*.

Rosch, E. H. (1973). Natural categories. *Cognitive Psychology*, *4*, 328–350.

Rosch, E., & Mervis, C. B. (1975). Family resemblances : Studies in the internal structure of categories. *Cognitive Psychology*, *7*, 573–605.

Ross, A. S., Hughes, M. C., & Doshi-Velez, F. (2017). Right for the Right Reasons : Training Differentiable Models by Constraining their Explanations. *IJCAI*.

Rouder, J. N., & Ratcliff, R. (2006). Comparing Exemplar- and Rule-Based Theories of Categorization. *Current Directions in Psychological Science*, *15*, 9–13.

Rousseeuw, P. J. (1987). Silhouettes : A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.

Roy, B., & Bouyssou, D. (1993). *Aide multicritere a la decision : methodes et cas*.

Schaffer, J., O'Donovan, J., Michaelis, J., Raglin, A., & Hollerer, T. (2019). I can do better than your AI - expertise and explanations. *IUI*, 240–251.

Scholkopf, B., Smola, A., & Muller, K.-R. (2006). Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, *10*, 1299–1319.

Sigillito, V. G., Wing, S. P., Hutton, L. V., & Baker, K. B. (1989). Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, *vol. 10*, 262–266.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*, 427–437.

Sorlie, T., Perou, C. M., Tibshirani, R., Aas, T., Geisler, S., Johnsen, H., … Borresen-Dale, A.-L. (2001). Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences*, *98*, 10869–10874.

Spearman, C. (1904). "General Intelligence" Objectively Determined and Measured, *15*, 201.

Su, A. I., Cooke, M. P., Ching, K. A., Hakak, Y., Walker, J. R., Wiltshire, T., … Hogenesch, J. B. (2002). Large-scale analysis of the human and mouse transcriptomes. *Proceedings of the National Academy of Sciences*, *99*, 4465–4470.

Tenenbaum, J. B., Silva, V. de, & Langford, J. C. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, *290*, 2319–2323.

Thomson, R., Lebiere, C., Anderson, J. R., & Staszewski, J. (2015). A general instance-based learning framework for studying intuitive decision-making in a cognitive architecture. *Journal of Applied Research in Memory and Cognition*, *4*, 180–190.

Tiercelin, C. (2013). *C. S. Peirce et le pragmatisme*. College de France.

Tomasev, N., & Mladenic, D. (2011). Nearest Neighbor Voting in High-Dimensional Data : Learning from Past Occurrences. In *International conference on data mining workshops* (pp. 1215–1218).

Tomasev, N., Radovanovic, M., Mladenic, D., & Ivanovic, M. (2014). The Role of Hubness in Clustering High-Dimensional Data. *IEEE Transactions on Knowledge and Data Engineering*, *26*, 739–751.

Torgerson, W. S. (1958). *Theory and methods of scaling*. John Wiley & Sons.

Tran, V.-T., Riveros, C., & Ravaud, P. (2019). Patients' views of wearable devices and AI in healthcare : findings from the ComPaRe e-cohort. *Npj Digital Medicine*, *2*, 53.

Tufte, E. R. (1983). *The Visual Display of Quantitative Information*.

Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley.

Turner, V., Gantz, J. F., Reinseil, D., & Minton, S. (2014). The digital universe : Rich data and the increasing value of the internet of things. *Australian Journal of Telecommunications and the Digital Economy*, *16*.

Weber, R., Schek, H.-J., & Blott, S. (1998). A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. *VLDB*.

Wittgenstein, L. (2001). *Philosophical Investigations*.

Zhang, Z., & Zha, H. (2004). Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment. *SIAM Journal on Scientific Computing*, *26*, 313–338.

Zwitter, M., & Soklic, M. (1988). Lymphography domain.