HAL
open science

Gestion et exploitation de données capteurs : une approche basée sur la réduction de données

Khedidja Boulanouar

To cite this version:

Khedidja Boulanouar. Gestion et exploitation de données capteurs : une approche basée sur la réduction de données. Autre [cs.OH]. ISAE-ENSMA Ecole Nationale Supérieure de Mécanique et d'Aérotechique - Poitiers; Université Saad Dahlab de Blida (Algérie), 2021. Français. NNT : 2021ESMA0011 . tel-03483342

HAL Id: tel-03483342

https://theses.hal.science/tel-03483342

Submitted on 16 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

pour l'obtention du Grade de

Docteur de l'École Nationale Supérieure de Mécanique et d'Aérotechnique et **de l'Institut d'Aéronautique et des Études Spatiales, Université Saad Dahlab Blida 1**
(Diplôme National - Arrêté du 25 mai 2016)

Secteur de Recherche : Informatique et Applications

Présentée par :

Khedidja BOULANOUAR

****

Gestion et Exploitation de Données Capteurs : une Approche Basée sur la Réduction de données

****

Directeurs de thèse : **Allel HADJALI** et **Mohand LAGHA**

****

Soutenue le 17 novembre 2021
devant la Commission d'Examen

****

JURY

Président :

Salah BOUKRAA	Professeur, Université Saad Dahleb Blida 1, Blida, Algérie

Rapporteurs :

Karima AKLI-ASTOUATI	Professeur, USTHB, Alger, Algérie
Sofian MAABOUT	MCF-HDR, Université de Bordeaux, Bordeaux, France

Membres du jury :

Daniela GRIGORI	Professeur, Université de Paris Dauphine, Paris, France
Allel HADJALI	Professeur, ISAE - ENSMA, Poitiers, France
Mohand LAGHA	Professeur, Université Saad Dahleb Blida 1, Blida, Algérie

Intelligence is the ability of a system to adjust appropriately to a changing world.

— Christopher Evans

Remerciements

Soyons reconnaissants aux personnes qui nous donnent du bonheur ; elles sont les charmants jardiniers par qui nos âmes sont fleuries.

—Marcel Proust

À la phase de finalisation de cette thèse, je suis convaincue que ce travail ne pourra jamais vu le jour sans le soutien des nombreuses personnes que je voudrais les remercier du fond du cœur :
Pour commencer, je tiens à remercier mes directeurs de thèse : Prof. Allel HADJALI et Prof. Mohand LAGHA ; qui m'ont soutenu tout au long de mon parcours, je tiens à leur exprimer ma profonde gratitude et reconnaissance pour leurs conseils avisés et leurs encouragements dans les moments les plus difficiles.
Je remercie très sincèrement les membres du jury d'avoir accepté d'évaluer ce travail, et de m'avoir orienté vers des nouveaux champs de recherche à travers ses questions et commentaires. Je remercie donc prof. Salah BOUKRAA, président du jury, MCF. Sofiane MAABOUT, prof. Karima AKLI-ASTOUATI et prof. Daniala GRIGORI.

Les travaux menés dans cette thèse sont les fruits d'une collaboration entre le Laboratoire des Sciences Aéronautiques (LSA), Institut d'Aéronautique et des Études Spatiales (IAES), université de Saad Dahlab, Blida et le laboratoire d'informatique et d'automatique pour les systèmes (LIAS), École nationale Supérieure de mécanique et d'Aérotechnique (ENSMA). À ce titre, j'adresse mes sincères remerciements aux membres des deux laboratoires, plus particulièrement je remercie Prof. Ladjel BELLATRECHE, responsable de l'équipe Ingénierie des Données et des Modèles de laboratoire LIAS, pour ses encouragements et ses précieux conseils durant la réalisation de ce projet. Je tiens à remercier Mickael BARON pour sa disponibilité, son aide et son soutien tout au long de mon séjours au LIAS.
Mes remerciements vont au directeur adjoint du laboratoire LIAS prof. Emmanuel GROLLEAU pour ses qualités humaines, son aide et sa sympathie.
Je voudrais également remercier Rachid ALLOUCHE pour toutes nos discussions et ses conseils qui m'ont accompagné tout au long de mon cursus.

Un immense merci, tout à fait particulier, à mes parents je suis consciente que je ne serais jamais ce que je suis aujourd'hui sans leur soutien leur patience, leur sacrifice et surtout leur amour qu'ils m'ont apporté toute ma vie. J'espère que je serais toujours une source de bonheur et de fierté pour eux.

Je souhaite adresser également mes chaleureux remerciements à mes frères : Abdrahmen ; Habib ; Dahou et Senouci, à mes sœurs : Mbarka ; Malika ; Fatiha, à mes beau frères et belles sœurs pour leur soutien et leur amour.

Un grand merci à mes collèges et mes amis plus particulièrement : Rayane MERDAOUI, Abir FAROUZI, Ishaq ZAOUAGHI et sans oublier Houssameddine MOHAMMEDI, cette thèse n'aurait jamais été possible sans leur soutien et leur encouragement.

Enfin, je tiens à remercier tous ceux qui ont contribué à la réalisation de cette thèse de près ou de loin. Merci d'être toujours à mes côtés, merci de me soutenir dans les moments les plus difficiles. Tous les mots ne peuvent pas exprimer ma gratitude et ma reconnaissance.

Table des matières

Table des matières vii

Liste des figures xi

Liste des tableaux xiii

Introduction générale 1

I Préliminaires et état de l'art 5

1 Notions de base 7
1.1 Introduction . . . 9
1.2 Théorie des ensembles flous . . . 9
1.2.1 Concepts fondamentaux . . . 9
1.2.2 Variable floue et valeur linguistique floue . . . 10
1.2.3 Opérateurs ensemblistes . . . 11
1.3 Aperçu sur les bases de données . . . 12
1.3.1 Système de gestion de base de données (SGBD) . . . 12
1.3.2 Avantage d'un SGBD . . . 12
1.3.3 Types de SGBD . . . 13
1.3.4 Langage des bases des données . . . 13
1.4 Flux de données . . . 14
1.4.1 Principe d'un flux de données . . . 14
1.4.2 Gestion de fenêtrage . . . 15
1.4.3 Modèles de flux de données . . . 15
1.4.4 Types de données . . . 16
1.4.5 Systèmes de gestion de flux de données . . . 17
1.5 Conclusion . . . 19

2 Travaux connexes : Techniques de réduction des données 21
2.1 Introduction . . . 23
2.2 Définition d'un résumé de données . . . 23
2.3 Applications des résumés . . . 23
2.4 Aéronautique et résumés de données . . . 24
2.5 Techniques de résumé de données . . . 25
2.5.1 Clustering . . . 26
2.5.2 Échantillonnage . . . 29
2.5.3 Histogramme . . . 32
2.5.4 Compression par Ondelettes . . . 32
2.6 Conclusion . . . 33

II Contributions **35**

3 Construction de résumés de données : Approches fondées sur l'intelligence computationnelle **37**
3.1 Introduction . . . 39
3.2 Résumés basées sur les quantificateurs linguistiques . . . 39
3.2.1 Protoformes du résumé linguistique . . . 40
3.2.2 Structure de base d'un résumé linguistique standard . . . 41
3.2.3 Résumé linguistique avec restriction . . . 44
3.2.4 Mesures de qualité . . . 45
3.3 Résumés de données fondées sur la typicité . . . 47
3.3.1 Notion de la valeur typique . . . 48
3.3.2 Approches de calcul de la valeur typique . . . 48
3.4 Étude Expérimentale . . . 52
3.4.1 Ensemble de données . . . 52
3.4.2 Cas des Résumés linguistiques . . . 53
3.4.3 Cas des valeurs typiques . . . 56
3.4.4 Étude Comparative . . . 57
3.5 Conclusion . . . 58

4 Résumés linguistiques dans le contexte des séries temporelles **59**
4.1 Introduction . . . 61
4.2 Séries temporelles . . . 61
4.3 Résumé linguistique des séries temporelles . . . 62
4.3.1 Protoformes des résumés linguistiques . . . 62
4.3.2 Segmentation linéaire par morceaux . . . 63
4.3.3 Processus de résumé . . . 67
4.4 Étude expérimentale . . . 72
4.5 Conclusion . . . 76

5 Algorithmes génétiques multiobjectif au service des résumés linguistiques **77**
5.1 Introduction . . . 79
5.2 Principe des algorithmes génétiques . . . 79
5.2.1 Codage d'individu . . . 80
5.2.2 Fonction objectif . . . 81
5.2.3 Taille de population . . . 81
5.2.4 Opérateurs génétiques . . . 81
5.2.5 Paramètres de dimensionnement . . . 83
5.2.6 Discussion . . . 84
5.3 Optimisation multi-objectif . . . 84
5.3.1 Problème multi-objectif . . . 84
5.3.2 Panorama des systèmes d'optimisation multi critères . . . 85
5.3.3 Méta-heuristiques pour l'optimisation multi-objectif . . . 85
5.3.4 Notion de dominance . . . 86
5.3.5 Frontière de Pareto . . . 86
5.4 Algorithme génétique multi objectif NSGA II . . . 87
5.5 Étude expérimentale . . . 89
5.5.1 Modèle de l'algorithme génétique . . . 89
5.5.2 Sélection multi-critère basée sur NSGA II . . . 91
5.6 Conclusion . . . 94

Conclusion générale **95**

Bibliographie **99**

A Acronymes **109**

Liste des figures

1.1 Exemple de fonction d'appartenance trapézoïdale 10
1.2 Fonction d'appartenance décrivant le terme jeune 11
1.3 Exemple de flux de données 16
1.4 Comparaison entre l'architecture des SGBDs (DBMSs) et celle des SGFDs (DSMSs) 18

2.1 Taxonomie des techniques de résumé de données 25
2.2 Exemple d'un histogramme équi-largeur 33

3.1 Exemple des quantificateurs relatifs 43
3.2 Valeur typique "environ 11" 50
3.3 Quantificateurs relatifs 54
3.4 Ensembles flous représentant l'altitude 54
3.5 Ensembles flous représentant la vitesse sol 55
3.6 Variation de température du campus intelligent 55
3.7 Ensembles flous représentant la température du campus 56
3.8 Comparaison des méthodes de résumé en fonction du temps d'exécution : (a) base de données ADSB ; (b) Campus intelligent NeOCampus. 57

4.1 Série temporelle originale (a) et son approximation linéaire par morceau (b) . . . 64
4.2 Comparaison de trois algorithmes de segmentation : (a) représente les trois algorithmes avec régression ; (b) représente les trois algorithmes avec interpolation. . 68
4.3 Processus de résumé des séries temporelles 68
4.4 Représentation des angles des tendances 70
4.5 Caractéristiques dynamiques des tendances (a) Création des tendances (b) Histogramme de la dynamique du changement (c) Histogramme de la durée de changement (d) Histogramme de la variabilité 73
4.6 Ensembles flous caractérisant les caractère dynamique des tendances (a) la dynamique du changement (b) la durée (c) la variabilité 74

5.1 Principe de l'algorithme génétique 80
5.2 Sélection par roue de loterie 82
5.3 Croisement en un point 83
5.4 Exemple de mutation 84
5.5 Exemple de dominance 86
5.6 Frontière de Pareto 87
5.7 Principe de fonctionnement de NSGA II 88
5.8 Exemple d'un individu 89
5.9 Évolution de l'algorithme génétique (a) Temps d'exécution en fonction du nombre d'itérations (b) Temps d'exécution en fonction du taux de mutation (c) Taux de réussite par rapport au nombre d'itérations 91

5.10 Évolution de l'algorithme génétique multi-objectif (a) Temps d'exécution en fonction du nombre d'itérations (b) Temps d'exécution en fonction du taux de mutation (c) Front de Pareto final 92

Liste des tableaux

1.1 Gestion du fenêtrage 15
1.2 Systèmes de gestion des flux de données [BBD+02, GÖ03] 18
1.3 Comparaison SGFD v.s. SGBD 19

2.1 Caractéristiques des clusters hiérarchiques ; N désigne le nombre d'objet dans un cluster [HTG+15] 27
2.2 Caractéristiques des clusters de partitionnement ; N designe le nombre des objets dans le cluster et k le nombre des clusters [HTG+15] 30

3.1 Calcul de degré de satisfaction 44
3.2 Résumé linguistique avec restriction 45
3.3 Exemple illustratif 51

4.1 Exemple de caractéristiques dynamiques des tendances 72
4.2 Résumés linguistiques de la série temporelle avec le protoforme classique Q y sont S 74
4.3 Mesures de qualité pour le protoforme Q R y sont S 75

5.1 Résumés linguistiques obtenus en utilisant l'algorithme génétique 90
5.2 Résumés linguistique en utilisant NSGA II 93

Introduction générale

Contexte et problématique

Le monde assiste à une évolution numérique sans précédent touchant les domaines scientifiques et économiques (transport, astronomie, énergie, environnement, sécurité, santé, etc.). Cette évolution s'est accompagnée d'une énorme explosion de volume de données produites, par exemple, par des réseaux de capteurs ou des plates-formes IoT (Internet of Things). Dans certains contextes et applications, le traitement de l'ensemble entier de ces données n'est pas souvent requis pour une prise de décision. Une représentation plus concise de ces données, permettant de retourner des réponses approximatives ou d'exhiber des tendances des données, suffit pour répondre aux besoins des décideurs. L'avantage de ce type d'approche est qu'elle conduise à des procédures moins coûteuses en terme de temps de calcul et en terme d'énergie. Ce qui pourrait être hautement désirable dans certaines applications du monde réel, comme par exemple, les applications de type temps réel.

Récemment, l'approche utilisant le principe de réduction de données a suscité un réel engouement. Le principe de cette approche est de réécrire les données originales sous une forme compacte et concise dans le but de réduire le volume de données en entrée du processus de traitement. Dans ce cadre, même si les réponses aux requêtes sont de nature approximative elles apporteront suffisamment d'informations pour être acceptables. Il existe de nombreuses techniques de réduction de volumes de données, dont les structures de résumé font partie.

De nombreuses méthodes de résumé de données sont proposées dans la littérature [Ahm19a]. Parmi ces méthodes, on peut citer : l'échantillonnage, les clusters et les histogrammes. Malheureusement, chaque méthode souffre de certaines insuffisances qui limiteront grandement son utilisation pratique telles que : (i) le manque de la représentativité des données ; (ii) les résumés construits sont difficilement compréhensibles par des utilisateurs non experts car leurs expressions ne sont pas exprimées en langage naturel. Ces formes de résumé sont ainsi loin de pouvoir refléter une vraie perception humaine.

De ce fait, les chercheurs scientifiques se sont intéressés aux résumés linguistiques, en raison de leur grande intelligibilité et de leur forte représentativité des données cibles. Un des premiers travaux dans ce contexte a été réalisé en [Yag82] où l'auteur propose d'utiliser une proposition linguistique quantifiée au sens de Zadeh [Zad65]. Ce concept a considérablement été développé dans [Yag88, KYZ00, KYZ02]. L'idée majeure est de représenter les données sous forme d'expressions du langage naturel, et chaque expression représente une instance des protoformes générales *"Q y sont S"* ou *"Q R y sont S "* où Q est un quantificateur linguistique, S et R sont des étiquettes floues qui représentent les attributs cibles.

En pratique, nous ne faisons pas seulement face aux grands volumes des données qui dépassent les capacités de traitement des systèmes actuels, mais nous devons aussi considérer les données avec la dimension temporelle qui joue un rôle critique dans de nombreuses applications du monde réel. C'est pour cette raison que les chercheurs ont accordé une grande importance aux données ayant cette spécificité temporelle, connues dans la littérature sous le nom de *séries temporelles*. Différentes descriptions compactes et concises des séries temporelles ont été étudiées lors des deux dernières décennies. [COMS09] propose de résumer les propriétés des séries temporelles

sur des intervalles de temps hiérarchiques. Dans [CBMB99, CBMB00], les auteurs ont suggéré des protoformes comme *"pendant les 30 dernières minutes, la température était élevée"*, qui permettent d'étudier l'occurrence ou la durée d'un phénomène dans une série temporelle. Selon [KWZ06a, KWZ10], le résumé de série temporelle fait, généralement, référence aux caractéristiques dynamiques des tendances associées à la série. Ces tendances sont identifiées avec des segments d'approximation linéaire par morceaux de la série temporelle en utilisant l'algorithme de Sklansky et Gonzalez présenté dans [SG80]. Cependant, des études, permettant d'extraire des tendances de série temporelle, ont été menées telles que : la méthode de fenêtre glissante et les méthodes de Bottom-Up et Top down [KCHP01, KCHP04, NP15].
Le processus d'élaboration des résumés peut être considéré comme un problème d'optimisation permettant de sélectionner les meilleurs résumés parmi un large éventail de candidats. Dans la littérature, plusieurs méta-heuristiques sont proposées pour améliorer la solution du problème d'optimisation ; tandis que dans le cadre des résumés linguistiques, les chercheurs se sont concentrés beaucoup plus sur l'exploitation d'une classe d'algorithmes génétiques. Un des premiers travaux proposés dans ce contexte, nous citons l'étude menée dans [KWZ06b] où chaque résumé extrait est considéré comme un chromosome et son degré de vérité est exprimée par une fonction d'évaluation utilisée pour identifier les résumés.
En fait, un résumé linguistique peut être évalué selon plusieurs critères : degré de vérité, degré de pertinence, degré de couverture et degré d'imprécision. Ces critères, nommés aussi mesures de qualité, sont généralement contradictoires, c'est-à -dire, l'amélioration d'un critère provoque la détérioration de l'autre, c'est pourquoi les techniques d'optimisation multi-objectif ont pris tout leur intérêt. Parmi ces techniques, nous proposons d'utiliser Non dominated Sorting Genetic Algorithm II (NSGA II) [DPAM02] qui est une méthode élitiste basée sur la notion de dominance de Pareto [Par97] et qui possède aussi de bonne performance en terme de temps d'exécution. En plus, elle permet d'assurer la diversité dans l'espace de recherche et de garantir la convergence vers la solution optimale.

Principales contributions

Ce travail s'inscrit dans le cadre de la réduction de données issues de différents capteurs, il a pour objectif de faire face aux volumes gigantesques de données, et de garantir un processus d'exploitation moins coûteux (en termes de calcul et de temps) en vue d'une prise de décision fiable et pertinente. Il consiste, notamment, à étudier une famille de méthodes de réduction dont les fondations théoriques sont issues des domaines de l'intelligence computationnelle et du soft computing.
Dans un premier temps, un état de l'art a été effectué sur les différentes structures de résumés de données. Pour chaque modèle de résumé, les éléments en faveur ou en défaveur de ce modèle sont abordés d'une manière explicite. En particulier, une étude est menée sur deux méthodes de résumés de données (issues de l'Intelligence Computationnelle), à savoir, le modèle fondé sur les quantificateurs linguistiques et celui basé sur le concept de la typicité. Un ensemble de propriétés de ces modèles a été établi. L'aspect algorithmique et l'implémentation des deux modèles ont été également largement discutés. Une comparaison entre les deux approches proposées a été conduite afin de montrer celle qui répond au mieux dans le cas des données massives. La validation de nos propositions a été réalisée par une série d'expérimentations sur des données réelles issues de deux projets : le projet ADSB [ads16] en aéronautique et le projet Neocampus [iri17] dans le domaine des campus intelligents.
Nous avons ensuite appliqué l'approche de résumé linguistique pour l'extraction des connaissances à partir des séries temporelles et identifier les caractéristiques dynamiques de ces séries. Nous avons exploité la méthode de segmentation linéaire par morceaux "Bottom up" pour segmenter la série temporelle en un ensembles des tendances caractérisées par sa vitesse du changement, la durée du changement et la variabilité des données. Nous utilisons les deux protoformes

classiques de résumé linguistique "Q y sont S" et "Q R y sont S" pour fournir des résumés décrivant les caractéristiques des tendances.
Cependant, cette approche génère un grand nombre de résumés et la sélection des meilleurs résumés (par rapport aux critères cités précédemment) n'est pas une tâche facile. Une solution est de formuler la question de la sélection sous forme d'un problème d'optimisation multi-objectifs.
Pour atteindre cet objectif ; nous avons subdivisé notre solution en deux étapes. La première vise à extraire les résumés linguistiques des caractéristiques dynamiques des tendances qui caractérisent les séries temporelles. Cela peut être fait en utilisant l'algorithme génétique traditionnel où la fonction d'évaluation représente le degré de vérité de la proposition linguistique quantifiée. Dans un second temps, nous avons abordé le problème de l'optimisation multi-critères où nous utilisons différentes mesures de qualité comme des cibles d'amélioration. Nous proposons d'utiliser Non dominated Sorting Genetic Algorithm II (NSGA II) comme un algorithme d'optimisation multi objectif.
Enfin, une série d'expérimentations a été réalisée afin de valider nos contributions et montrer la fiabilité de l'approche de résumé proposée.

Structure de la thèse

Ce manuscrit est structuré en deux parties principales. La première partie dédiée à l'état de l'art, elle est constituée de deux chapitre. Dans le premier chapitre, nous présentons les notions préliminaires de la théorie des ensembles flous sur lesquelles nous nous appuyons pour développer nos contributions. Puis, nous introduisons quelques notions liées à la gestion des bases de données tout en mettant l'accent sur les avantages et les inconvénients des SGBDs traditionnels. Nous rappelons ensuite quelques concepts de représentation et de traitement des flux de données.
Le chapitre 2 passe en revue les différentes approches qui ont abordé la problématique de réduction des masses de données. Il décrit notamment les structures de résumés de données, comme par exemple, l'échantillonnage, les histogrammes et les clusters. Ces techniques sont capables de réduire, d'une manière efficace, des données massives. À partir de ces résumés, nous pouvons extraire une information utile et concise à des fins de prise de décision.

La seconde partie de cette thèse est consacrée à nos contributions et détaille aussi les différentes et riches expérimentations afin de valider nos propositions. Elle est structurée en trois chapitres.
Après avoir étudié les différentes structures de résumés de données, nous présentons dans le chapitre 3 deux techniques de résumés basées sur l'intelligence computationnelle et qui reflètent la perception humaine de la sémantique des données. Ces deux structures : le résumé basé sur les quantificateurs linguistiques et celui utilisant le concept de typicité, se distinguent par deux particularités : (i) l'intelligibilité des résumés construits et ; (ii) la génération des résumés qui décrivent les données à des niveaux d'abstraction différents. À la suite de cette étude, nous détaillons une série d'expérimentations visant à montrer la fiabilité de ces techniques. Nous proposons aussi une étude comparative entre les deux méthodes dans le cas des données provenant de multiples capteurs.
Le chapitre 4 est dédié à l'étude de réduction de données dans le contexte des séries temporelles. En premier lieu, nous définissons les séries temporelles tout en présentant les différentes méthodes de segmentations de ces séries et les caractéristiques dynamiques. Enfin, nous appliquons l'algorithme de résumé linguistique présenté dans le chapitre 3 pour extraire l'information à partir des tendances d'une série temporelle.
Dans le chapitre 5, nous proposons une approche améliorée de résumé linguistique de séries temporelles en exploitant l'algorithme génétique multi objective NSGA II qui nous permettra de sélectionner les résumés les plus fiables à partir d'un ensemble de candidats. Nous utilisons aussi cet algorithme génétique pour choisir les meilleurs résumés dans le cas multi-critère.

Nous clôturons ce manuscrit par une conclusion générale où nous résumons nos contributions et discutons les différents résultats obtenus. À la fin, nous dressons un ensemble de perspectives que nous jugeons intéressantes et prometteuses pour de futures travaux.

Liste des publications

Journal International

1. **Khedidja BOULANOUAR**, Allel HADJALI, Mohand LAGHA. Data Summarization for Sensor Data Management : Towards Computational-Intelligence-Based Approaches. International Journal of Computing and Digital Systems (IJCDS).
`http://dx.doi.org/10.12785/ijcds/090505`

Communications Internationales

1. **Khedidja BOULANOUAR**, Allel HADJALI, Mohand LAGHA. A Hybrid Approach for Linguistic Summarization of Time Series. 2020 International Conference on Data Analytics for Business and Industry : Way Towards a Sustainable Economy (ICDABI), Sakheer, Bahrain 26-27 October, 2020.
`http://dx.doi.org/10.1109/ICDABI51230.2020.9325701`
2. **Khedidja BOULANOUAR**, Allel HADJALI, Mohand LAGHA. Flight Data Summarization : A Fuzzy-Set-Theory-Based-Approach, 5th International Workshop Unmanned and Swarming Conference Research Challenges for Future Unmanned Systems and Autonomous Swarming, Bordeaux, France, 10-11 October, 2018.

Journées et séminaires

1. **Khedidja BOULANOUAR**, Allel HADJALI, Mohand LAGHA. Data Summarization Based on Computational Intelligence Methods, 1er Symposium du GDR Madics, Rennes, France, 26-28 juin, 2019.
2. **Khedidja BOULANOUAR**, Allel HADJALI, Mohand LAGHA. Gestion du Big data à l'aide des techniques de réduction de données, Journées nationales Doctoriales, Université de Abderahmane Mira, Béjaia, Algérie, 1-6 août, 2017.

Première partie

Préliminaires et état de l'art

Chapitre 1

Notions de base

Une période d'échec est un moment rêvé pour semer les graines du succès.

—Emmeline Raymond

Sommaire

1.1 Introduction **9**
1.2 Théorie des ensembles flous **9**
1.2.1 Concepts fondamentaux 9
1.2.2 Variable floue et valeur linguistique floue 10
1.2.3 Opérateurs ensemblistes 11
1.3 Aperçu sur les bases de données **12**
1.3.1 Système de gestion de base de données (SGBD) 12
1.3.2 Avantage d'un SGBD 12
1.3.3 Types de SGBD 13
1.3.4 Langage des bases des données 13
1.4 Flux de données **14**
1.4.1 Principe d'un flux de données 14
1.4.2 Gestion de fenêtrage 15
1.4.3 Modèles de flux de données 15
1.4.4 Types de données 16
1.4.5 Systèmes de gestion de flux de données 17
1.5 Conclusion **19**

1.1 Introduction

Afin de comprendre un sujet/thème, il est important d'avoir une connaissance préalable de tous les domaines liés au sujet en question et qui permet de se familiariser avec celui-ci. C'est pourquoi ce chapitre se limite à un rappel des notions de base utilisées dans la suite de ce manuscrit. Il est structuré en trois parties. Dans la première, nous présentons une introduction à la théorie des ensembles flous qui nous permet de représenter, manipuler et de gérer la gradualité et l'imprécision inhérentes aux relations et aux termes du langage naturel. Dans la Section 1.3, nous présentons un ensemble de notions sur les systèmes de gestion de bases de données tout en précisant certaines de leurs limitations. Enfin, nous introduisons le paradigme de flux de données en décrivant ses modèles, plus précisément, le modèle de série temporelle sur lequel se fonde une partie des travaux décrits dans cette thèse.

1.2 Théorie des ensembles flous

La logique floue [1] représente un outil très puissant pour traiter certains problèmes complexes du monde réel. Par exemple, récemment, la gestion et l'exploitation des données imparfaites a vu un regain d'intérêt sans précédent. Les ingénieurs et les scientifiques sont généralement confrontés à des problèmes réels impossibles à modéliser et à résoudre à l'aide des règles mathématiques traditionnelles (logique classique, ensembles classiques, etc.). À cet effet, ils ont introduit une nouvelle théorie, dite "la théorie des ensembles flous", en s'inspirant de la perception humaine du monde et de son mécanisme de raisonnement. Ainsi, nous pouvons caractériser et modéliser un système dont le modèle n'est pas connu ou mal défini [TC87]. La théorie de la logique floue a donc la capacité de capturer l'imprécision des termes linguistiques présents dans les déclarations exprimées en langage naturel. Cela a permis de fournir des bases et mécanismes rationnels reproduire le raisonnement humain.
La théorie des ensembles flous [Zad65] est donc une généralisation de la théorie des ensembles classiques. Elle a pour objectif de représenter les classes d'objets dont les limites sont imprécises ou mal définies. Elle permet de décrire la transition entre l'appartenance totale (représentée par le degré 1) et la non-appartenance (représentée par le degré 0) en introduisant la notion de l'appartenance graduelle (ou nuancée). Elle représente également une extension de la logique binaire en proposant une zone floue entre ce qui est vrai et ce qui est faux.
Souvent, nous raisonnons en termes de classes/termes flous. Prenez le cas de l'âge des personnes, dans la logique classique nous pouvons définir le prédicat "jeune" comme les personnes ayant l'âge entre 18 et 35 ans, donc une personne qui a 36 ans n'est pas du tout considérée comme jeune (alors qu'elle est très proche des personnes dites jeunes) . Pour pallier ce problème de seuil brutal, la théorie des ensembles flous autorise la notion d'appartenance dite graduelle (entre les valeurs 0 et 1). Dans ce cas, une personne, dont l'âge est 36 ans, appartient à l'ensemble jeune avec un certain degré appartenant à l'intervalle $]0, 1[$. Ce degré dépend généralement de la définition de la fonction (caractéristique ou d'appartenance) associée au terme flou. Cela signifie que la sémantique d'un ensemble flou dépend du contexte de l'application considérée.

1.2.1 Concepts fondamentaux

La logique classique nous permet d'écrire l'information par deux états : soit totalement vrai ou totalement faux.

Définition 1 *Soit un ensemble X (dit aussi référentiel ou univers de discours). Un sous ensemble flou F de X est caractérisé par sa fonction d'appartenance μ_F correspondant à la fonction carac-*

1. Dans tout le document, la logique floue et la théorise des ensembles flous sont utilisées d'une manière interchangeable

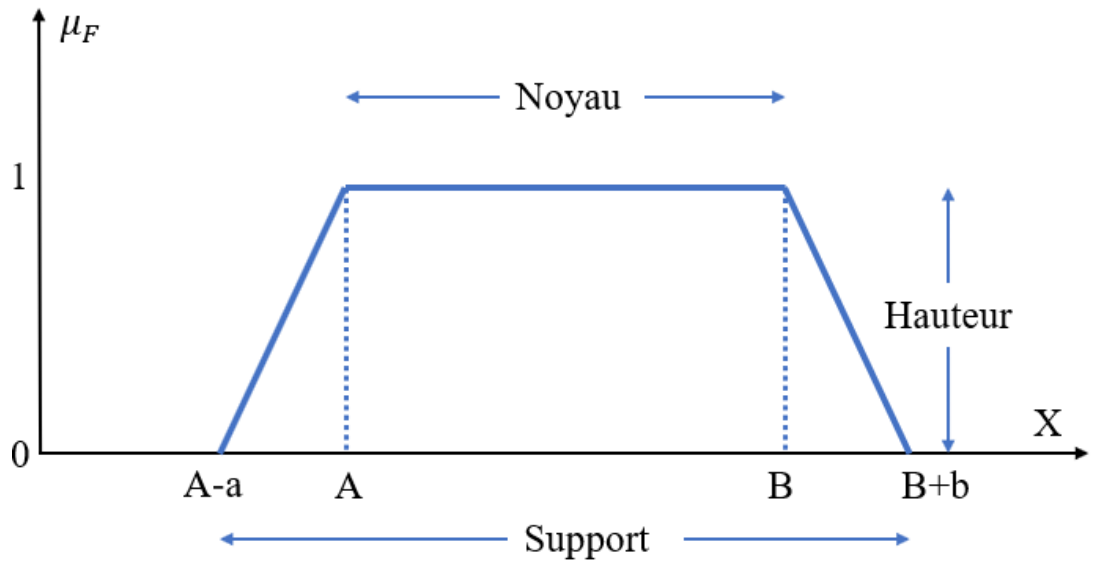


FIGURE 1.1 – Exemple de fonction d'appartenance trapézoïdale

téristique en logique classique. Cette fonction attribue à chaque élément de X une valeur dans l'intervalle [0,1] $\mu_F : X \to [0,1]$.

La définition d'un ensemble flou nécessite la détermination de certaines propriétés liées à cet ensemble. Ci-dessous nous décrivons les plus importantes [God99, BMM03] :

— Le type : la fonction d'appartenance [2] peut avoir diverses formes selon la définition : triangulaire, trapézoïdale, Gaussienne, Sigmoïdes. Le choix de la fonction d'appartenance dépend de l'application et faite par les experts du domaine. Généralement, et pour des aspects calculatoires, l'ensemble flou F est associé à une fonction d'appartenance trapézoïdale (t.m.f) exprimée par le quadruplet (A,B,a,b) (où le noyau $C(F) = [A, B]$ et le support $S(F) = [A - a, B + b]$, voir ci-dessous)

— La hauteur : elle représente le plus fort degré avec lequel un élément de X appartient à F $H(F) = Sup_{x \in X}(\mu_F(x))$. Un sous ensemble flou est normalisé si sa hauteur est égale 1.

— Le noyau : il représente l'ensemble de tous les éléments qui appartiennent totalement à F, $C(F) = \{x \in X, \mu_F(x) = 1\}$.

— Le support : il désigne l'ensemble de tous les éléments qui appartiennent un tant soit peu au sous ensemble. Les éléments du support doivent vérifier $S(F) = \{x \in X, 0 < \mu_F(x) < 1\}$.

— La cardinalité : elle désigne le nombre d'éléments appartenant à F pondéré par leur degré d'appartenance $card(F) = \sum_{x \in X} \mu_F(x)$.

Définition 2 *$\alpha - cut$ d'un sous ensemble flou F est le sous ensemble des éléments dont le degré d'appartenance est supérieur ou égal à α : $\alpha - cut(F) = \{x \in X, \mu_F(x) \geq \alpha\}$*

Définition 3 *Un sous ensemble flou F est dit normalisé si et seulement si sa hauteur $H = 1$. En pratique, il est extrêmement rare de travailler sur des ensembles flous non normalisés.*

1.2.2 Variable floue et valeur linguistique floue

Une variable floue est un terme du langage naturel caractérisé par un triplet (V, X_V, T_V)

2. Dans la littérature, il existe plusieurs méthodes pour définir la sémantique de ces fonctions, voir par exemple [Hud16].

- V : nom de la variable(température, vitesse, poids)
- X_V : univers de discours représentant les valeurs prises par V
- T_V : ensemble des valeurs linguistiques que peut prendre V.

Exemple Nous pouvons définir la variable linguistique V = âge dans l'univers de discours X = [0, 90]. Les valeurs linguistiques liées à cette variable peuvent être T = {enfant, jeune, adulte, âgé}. Par exemple, le terme jeune est défini par la fonction d'appartenance (1.1).

$$S(x) = \begin{cases} 0 & x < 15 \\ \frac{1}{3}x - 5 & 15 \leq x < 18 \\ 1 & 18 \leq x \leq 35 \\ \frac{-1}{10}x + 4.5 & 35 < x \leq 45 \\ 0 & x > 45 \end{cases} \tag{1.1}$$

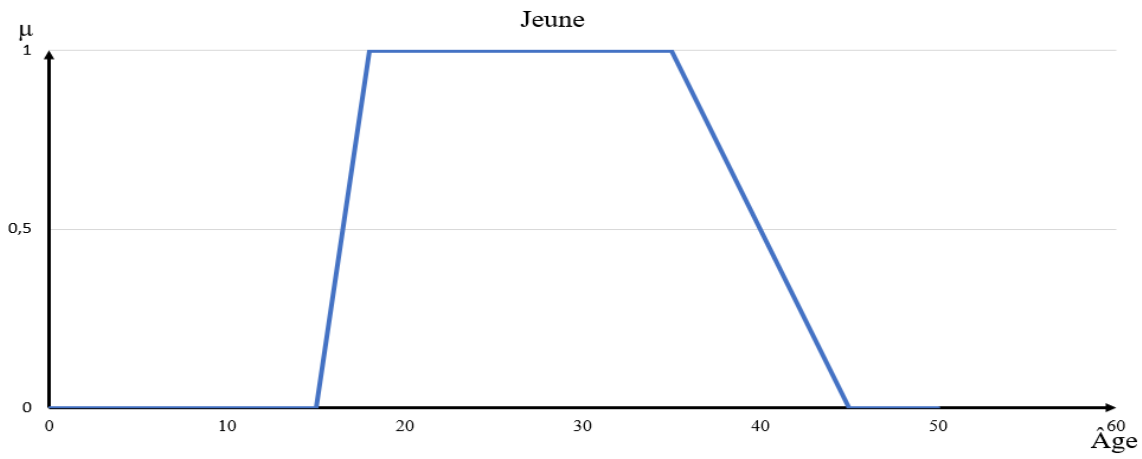


FIGURE 1.2 – Fonction d'appartenance décrivant le terme jeune

1.2.3 Opérateurs ensemblistes

Soit F et G deux ensembles flous sur l'univers de discours X, les opérations d'union et d'intersection de ces deux ensembles peuvent être définies en utilisant les opérations de normes triangulaires (t-normes) et de co-normes triangulaires (co-normes). Une application t-norme $\mathbb{T}$ ou co-norme $\perp$ doit satisfaire les axiomes suivants :

— Commutativité : $\mathbb{T}(a, b) = \mathbb{T}(b, a)$ et $\perp (a, b) = \perp (b, a)$

— Associativité : $\mathbb{T}(a, (b, c)) = \mathbb{T}(\mathbb{T}(a, b), c)$ et $\perp (a, \perp (b, c)) = \perp (\perp (a, b), c)$

— monotonie : $\mathbb{T}(a, b) \geq \mathbb{T}(c, d)$ si $a \geq b$ et $c \geq d$ et $\perp (a, b) \geq \perp (c, d)$ si $a \geq b$ et $c \geq d$

— Élément neutre : $\mathbb{T}(a, 1) = a$ et $\perp (a, 0) = a$

— Idempotence pour 0 (respectivement l'élément 1) : $\mathbb{T}(0, 0) = 0$ et $\perp (1, 1) = 1$

Union. L'union de deux ensemble flous F et G peut être définie comme l'ensemble de tous les éléments de X appartiennent à F ou G associés avec le plus grand degré d'appartenance. En utilisant l'opérateur$\perp$, l'union est donnée par : $\forall x \in X$, $\mu_{F \cup G}(x) = \perp (\mu_F(x), \mu_G(x))$. L'opérateur max est un exemple de co-normes.

Intersection. L'intersection comprend les éléments appartiennent à F et à G associés avec le plus petit de degrés. En utilisant l'opérateur $\mathbb{T}$, l'intersection est donnée par : $\forall x \in X$, $\mu_{F \cap G}(x) = \mathbb{T}(\mu_F(x), \mu_G(x))$. L'opérateur min est un exemple de t-normes.

Complément. Le Complément de F, noté $\overline{F}$, peut être défini par la fonction d'appartenance suivante : $\forall x \in X$, $\mu_{\overline{F}}(x) = 1 - \mu_F(x)$.

1.3 Aperçu sur les bases de données

Les bases de données font aujourd'hui une partie intégrante de l'environnement informatique associé à chaque organisation. Cette dernière doit disposer de données (de niveaux de qualité variés) pour une prise de décision efficace et fiable. À cette fin, ces données sont organisées sous forme d'une base de données. D'une façon globale, une base de données peut être définie comme une collection des données connexes stockées et offrant la possibilité d'accès et de récupération des informations de la base selon les besoins utilisateurs.

La puissance des bases de données est due aux connaissances et technologies qui se sont développées au cours des dernières décennies et qui sont incorporées dans un système spécialisé appelé système de gestion de base de données, ou SGBD, ou plus familièrement un "système de base de données". Un SGBD est un outil puissant pour créer et gérer efficacement des grandes quantités de données et leur permet de persister sur de longues périodes de temps, en toute sécurité [Feu10].
L'objectif de cette section est de donner un aperçu sur les bases de données, d'expliquer leurs composants et de présenter l'architecture d'un système de gestion de base de données.

1.3.1 Système de gestion de base de données (SGBD)

La technologie des bases de données peut être décrite comme l'un des domaines de l'informatique et des sciences de l'information qui a connu un intérêt considérable et une croissance extrêmement rapide. Elle est apparue à la fin des années 60 à la suite de la combinaison de diverses circonstances [Feu10]. Cette technologie, appelée généralement "technologie de gestion de base de données", a émergé pour traiter des données de divers types et le logiciel qui en résulte est connu sous le nom de " Système de Gestion de Bases de Données" (SGBD), un système permettant de stocker et de manipuler une grande masse de données partagée par plusieurs utilisateurs d'une manière efficace.
Un SGBD est composé généralement de trois éléments essentiels :

1. Un système de gestion de fichiers : permet la gestion de stockage sur un support physique ;
2. SGBD interne : gère l'ordonnancement des données ;
3. SGBD externe : représente l'interface avec l'utilisateur.

1.3.2 Avantage d'un SGBD

L'utilisation d'un SGBD présente de nombreux avantages, on peut citer :

— Indépendance des données : les programmes d'application doivent être indépendants autant que possible des détails de la représentation et du stockage des données. Le SGBD peut fournir une vue abstraite des données pour isoler le code d'application de ces détails.

— Accès efficace aux données : un SGBD utilise une variété de techniques sophistiquées pour stocker et récupérer efficacement les données cibles. Cette fonction est particulièrement importante si les données sont stockées sur des périphériques de stockage externes.

— Intégrité et sécurité des données : si les données sont toujours accessibles via le SGBD, le SGBD peut appliquer des contraintes d'intégrité sur les données. Par exemple, avant l'insertion des informations de salaire pour un employé, le SGBD vérifie que le budget du service n'est pas dépassé. En outre, le SGBD applique des contrôles d'accès qui régissent les données visibles par différentes classes d'utilisateurs.

— Administration des données : lorsque plusieurs utilisateurs partagent les données, la centralisation de l'administration des données peut offrir des améliorations significatives. Des professionnels expérimentés qui comprennent la nature des données gérées et la manière dont différents groupes d'utilisateurs les utilisent, peuvent être responsables de l'organisation de la représentation des données pour minimiser la redondance et d'affiner le stockage des données pour rendre la récupération efficace.

— Accès simultané et récupération après incident : un SGBD planifie des accès simultanés aux données de manière à ce que les utilisateurs puissent considérer les données comme étant accessibles par un seul utilisateur à la fois. De plus, le SGBD protège les utilisateurs des effets des pannes du système.

1.3.3 Types de SGBD

Plusieurs critères sont utilisés pour classer les SGBDs [Feu10]. Le premier critère est le modèle de données sur lequel repose le SGBD. Le modèle de données principal, utilisé dans de nombreux SGBDs commerciaux actuels, est le modèle de données relationnel. Le modèle de données d'objets a été mis en œuvre dans certains systèmes commerciaux mais n'a pas été largement utilisé. De nombreuses applications héritées fonctionnent toujours sur des systèmes de base de données basés sur les modèles de données hiérarchiques et réseau. Les SGBD relationnels évoluent en permanence et, en particulier, incorporent de nombreux concepts développés dans les bases de données d'objets. Cela a conduit à une nouvelle classe de SGBD appelée SGBD relationnel objet. Nous pouvons donc catégoriser les SGBD en fonction du modèle de données : relationnel, objet, objet-relationnel, hiérarchique, réseau, etc.

Le deuxième critère utilisé pour classer les SGBDs est le nombre d'utilisateurs pris en charge par le système. Les systèmes mono-utilisateur ne prennent en charge qu'un seul utilisateur à la fois et sont principalement utilisés avec des ordinateurs personnels. Les systèmes multi-utilisateurs, qui incluent la majorité des SGBDs, prennent en charge plusieurs utilisateurs simultanément.

Un troisième critère est le nombre de sites sur lesquels la base de données est distribuée. Un SGBD est centralisé si les données sont stockées sur un seul site informatique. Il peut prendre en charge plusieurs utilisateurs, mais le SGBD et la base de données eux-mêmes résident totalement sur un seul site informatique. Un SGBD distribué peut avoir la base de données réelle et le logiciel SGBD distribués sur de nombreux sites, connectés par un réseau informatique. Les homogènes utilisent le même logiciel de SGBD sur plusieurs sites. Une tendance récente est de développer des logiciels pour accéder à plusieurs bases de données autonomes préexistantes stockées sous SGBD hétérogènes. Cela conduit à un SGBD fédéré (ou système multi-bases de données), dans lequel les SGBDs participants sont faiblement couplés et ont une certaine autonomie locale. De nombreux SGBDs utilisent une architecture client-serveur [Feu10].

1.3.4 Langage des bases des données

Pour manipuler, interroger ou plus précisément interagir avec la base de données, plusieurs langages peuvent être utilisés [RG99] :

- Langage de définition des données DDL : il est utilisé pour définir la structure ou le modèle de la base de données. En plus, il permet de créer des schémas, des tables, des index, des

contraintes, etc. dans la base de données. Le stockage des informations des métadonnées comme le nombre de tables et de schémas, leurs noms, index, colonnes dans chaque table, contraintes, etc. est possible à l'aide des instructions DDL.

- langage de manipulation des données DML : il est utilisé pour accéder et manipuler des données dans une base de données. Il permet de réaliser les tâches suivantes :
  - La récupération des informations stockées dans la base de données
  - L'insertion des nouvelles informations dans la base de données
  - La suppression des informations de la base de données
  - La modification des informations stockées dans la base de données
- langage de contrôle de données DCL : ce langage est utilisé pour contrôler l'accès des utilisateurs à la base de données. Il est lié aux problèmes de sécurité.

En pratique, les langages de définition et de manipulation et de contrôle de données ne sont pas des langages distincts ; ils font simplement partie d'un seul langage de base de données, tel que le langage Structured Query Language (SQL) qui est le langage des requêtes le plus courant et le plus utilisé dans les SGBDs actuels. Il a été initialement développé chez IBM, il a été conçu principalement pour stocker et manipuler les données. La première version commercialement de SQL a été présenté en 1979 par Oracle Corporation. Puis elle a été adoptée comme norme internationale en 1987 [RG99].

1.4 Flux de données

Avec l'explosion de la quantité de données produite, d'une manière continue, dans plusieurs domaines et la vitesse avec laquelle ces données arrivent, les systèmes de gestion de données traditionnels et leur langages de requêtes s'avèrent incapables de gérer et de traiter ces données et leurs caractéristiques. Ce qui impose le développement d'une nouvelle architecture adaptée à la notion de flux de données.

1.4.1 Principe d'un flux de données

D'après [Cse08], un flux de données (en anglais, Data Stream) est une séquence de données structurées, que l'on peut considérer comme infinie. Cette séquence étant constituée d'éléments générés de façon continue et avec un rythme important. La capacité de traitement et de stockage n'est donc pas en mesure de répondre à la vitesse d'arrivée de nouveaux éléments. Cette même vitesse d'arrivée peut s'avérer constante ou variable. À partir de cette définition ; nous pouvons résumer les caractéristiques principales d'un flux de données comme suit [Chi09] :

- Rapidité : représente la vitesse avec laquelle les données arrivent.
- Volumétrie : décrit la quantité des données qui est potentiellement illimitée et qui dépasse la capacité des systèmes de gestion des bases de données traditionnels.
- Variété : les données proviennent d'une grande variété de sources et sont généralement de l'un des trois types suivants : données structurées, semi-structurées et non structurées. La diversité des types de données nécessite souvent des capacités de traitement distinctes et des algorithmes spécialisés.
- Véracité : fait référence à la qualité des données analysées. Les données à haute véracité contiennent de nombreux enregistrements précieux à analyser et qui contribuent de manière significative aux résultats globaux. Les données à faible véracité, en revanche, contiennent un pourcentage élevé de données dénuées de sens.

Chaque élément d'un flux F possède une étiquette temporelle correspondant à l'instant de sa création :

$$F = X_1, X_2, ..., X_i \tag{1.2}$$

où X_i est un élément du flux qui représente un vecteur de dimension d :

$$X_i = x_i^1; x_i^2, ..., x_i^d \tag{1.3}$$

où x_i est de nature numérique ou modale.

1.4.2 Gestion de fenêtrage

Le flux de donnée est caractérisé par sa nature infinie, pratiquement, il est impossible de le traiter dans son intégralité. À cet effet, il est important de définir une portion du flux, nommée fenêtre, sur laquelle s'effectuer les opérations de traitement.

— **Fenêtre logique et physique :** (voir Tableau 1.1) nous distinguons deux types de fenêtres selon la modélisation du temps utilisée. Dans le cas où on utilise l'unité temporelle on parle de fenêtre physique ou temporelle (par exemple, du 15/01/2020 au 15/02/2020). Quant à la fenêtre logique, nous considérons le nombre des éléments (par exemple : du l'élément 880 à l'élément 980).
Nous pouvons aussi distinguer trois types des fenêtres selon le début et la fin.

- Fenêtre fixe : les bornes de cette fenêtre sont bien définies, par exemple du 01/juin au 30/juin.
- Fenêtre point de repère : l'une des bornes de cette fenêtre est bien précise et l'autre est relative, par exemple du 20 mars à l'instant t actuelle.
- Fenêtre glissante : elle se caractérise par ses bornes non fixées, par exemple les données des derniers trois jours.

Fenêtre	**Bornes**	**Exemple**
Fenêtre logique fixe	Fixes	du 200ème élément au 520ème
Fenêtre physique fixe	Fixes	du 24/01/2019 au 05/08/2020
Fenêtre logique point de repère	Une fixe, une relative	du 200ème élément au dernier élément
Fenêtre physique point de repère	Une fixe, une relative	du 24/01 à l'insatnt courant t
Fenêtre logique glissante	Relatives	les dix derniers éléments
Fenêtre physique glissante	Relatives	les dix dernières minutes

Tableau 1.1 – Gestion du fenêtrage

1.4.3 Modèles de flux de données

Selon [MR19], il existe trois grands modèles de flux de données, qui différencient dans la manière dont les éléments du flux sont liés et s'influencent les uns les autres : le modèle de tourniquet, le modèle de caisse enregistreuse et le modèle de séries temporelles (en anglais, time series). Le modèle le plus général est le modèle de tourniquet. Dans ce modèle, le flux est modélisé comme un vecteur d'éléments, et chaque élément est une mise à jour (incrémentation ou décrémentation) d'un élément du signal sous-jacent. La taille du vecteur dans ce modèle est le domaine des éléments de flux. Ce modèle est également le modèle généralement utilisé dans les systèmes de base de données traditionnels, où il y a des insertions, des suppressions et des mises à jour

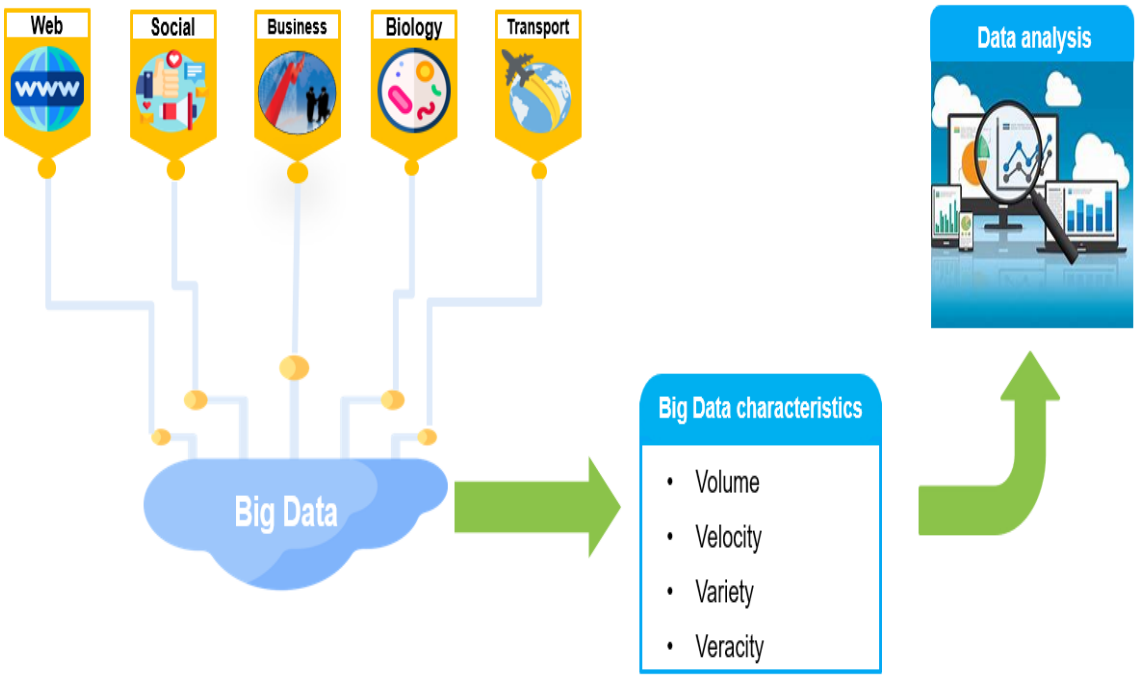


FIGURE 1.3 – Exemple de flux de données

de la base de données. Dans le modèle de caisse enregistreuse, les éléments du flux ne sont que des ajouts au vecteur sous-jacent, mais les éléments ne peuvent jamais quitter le vecteur. Ceci est similaire aux bases de données enregistrant l'historique des relations. Ces deux modèles représentent un modèle généraliste qui décrit la variation du flux.

Enfin, le modèle de série chronologique traite chaque élément du flux comme une nouvelle entrée vectorielle indépendante. En conséquence, le modèle sous-jacent est un vecteur en constante augmentation et généralement illimité. Étant donné que chaque élément peut être traité individuellement dans ce modèle, il est fréquemment utilisé dans les moteurs de traitement de flux actuels. Nous détaillons ce modèle dans le chapitre 4.

1.4.4 Types de données

Due à la grande variété de sources, les données arrivent sous différents formats. Pour cette raison une étape de pré-traitement est cruciale pour passer des données brutes à des formats adaptés aux algorithmes. Nous pouvons distinguer deux principaux formats :

— **Données quantitatives :** regroupent les données sous format numérique, elles proviennent de mesures ou de calculs mathématiques. Le domaine de définition est soit continu soit discrèt. Un exemple de ce type est les mesures de température, de pression, de vitesse, etc.

— **Données qualitatives :** prennent un nombre fini de valeurs qu'on appelle des modalités. Les données qualitatives sont classées en deux catégories : ordinale si les modalités respectent un certain ordre comme petit, moyen et grand, sinon les données sont dites nominales comme les couleurs.
Les données binaires sont considérées comme un cas particulier des données qualitatives où elles prennent les valeurs vraie ou fausse.

En plus de son objectif principal, l'étape de pré-traitement a pour but aussi de nettoyer les données et les normaliser. Le premier sert à traiter, par exempe, les données manquantes ou aberrantes. La deuxième opération a pour objectif de comparer les valeurs de données après les normaliser à un même domaine de définition.

1.4.5 Systèmes de gestion de flux de données

Les SGBDs traditionnels actuels ne répondent pas parfaitement aux besoins liés au flux de données. L'architecture des systèmes de gestion de ce type de données doit s'adapter aux caractéristiques des flux de données telles que la volumétrie et la rapidité. Ce qui a incité au développement d'une nouvelle technologie permettant le traitement et l'analyse des flux en temps réel et qui remplace le paradigme des SGBDs traditionnels.

Les systèmes de gestion de flux de données SGFD traitent les données comme une séquence d'enregistrements définis par rapport au temps d'arrivée d'une manière continue et rapide. Un SGFD doit assurer, en plus des fonctionnalités traditionnelles des SGBDs, des nouvelles fonctionnalités en prenant en considération le caractère continu, rapide et volumineux et temps réel. La pluaprt de ces fonctions sont décrites dans [SÇZ05], nous pouvons citer :

— Traitement des requêtes continues : le langage utilisé dans le traitement de requêtes dans le SGFD est très similaire à SQL, le traitement de requêtes s'effectue à la volé en se basant sur le temps. À cet effet, la gestion du fenêtrage temporelle doit être prise en considération pour s'adapter aux spécificités du flux.

— Traitement des données continues et statiques : le système doit avoir la capacité de combiner des données statiques et les bases des données dynamiques.

— Disponibilité des données : SGFD doit assurer et garantir la disponibilité des données à tout instant. De plus, il doit faire face aux pannes système. Ainsi une opération de sauvegarde du contenu de la mémoire du SGFD doit être prévue [Cse08].

Dans ce qui suit, nous décrivons à titre d'exemple deux systèmes de gestion de flux de données existant dans le marché commercial (voir Tableau 1.2 pour plus de détails) :

— **Aurora :** un système de gestion des flux de données généraliste pour les applications de supervisions. Il dispose d'une interface graphique pour la spécification du plan de requête. La version commerciale de ce système a été proposée en 2003 sous le nom de StreamBase [ACC+03, BBC+04]. Le langage des requêtes utilisé est une extension du langage SQL (StreamSQL) incluant la définition de la structure de flux, la définition de fenêtres et la manipulation de données statiques.

— **TelegraphCQ :** un SGFD généraliste développé dans le cadre d'un projet mené par l'Université de Berkeley. C'est une extension du système de gestion de base de données PostgreSQL permettant de poser des requêtes continues sur des flux de données. TelegraphCQ a été conçu afin de faire face aux flux de haut débit [KCC+03, CCD+03].

Le tableau 1.2 résume les principales caractéristiques de certains SGFDs (y compris des systèmes non décrits ci-dessus), alors que la figure 1.4 et le tableau 1.3 montrent le mécanisme de fonctionnement des SGFDs et des SGBDs et une comparaison entre ces deux architectures.

Système	Applications	Modèle	Fenêtre	Bornes
Aurora	capteurs	procédural	logique et physique	fixe, point de repère et glissante
Aquery	réseau et stock	relationnel	logique et physique	fixe, point de repère et glissante
Telegraph CQ	capteurs	relationnel	logique et physique	fixe, point de repère et glissante
Medusa	réseau	distribué	logique et physique	fixe, point de repère et glissante
Stream	Général	relationnel	logique et physique	glissante
Cougar	capteurs	orienté objet	logique et physique	fixe, point de repère et glissante
Hanckock	réseau	procedural	logique	fixe et point de repère

TABLEAU 1.2 – Systèmes de gestion des flux de données [BBD+02, GÖ03]

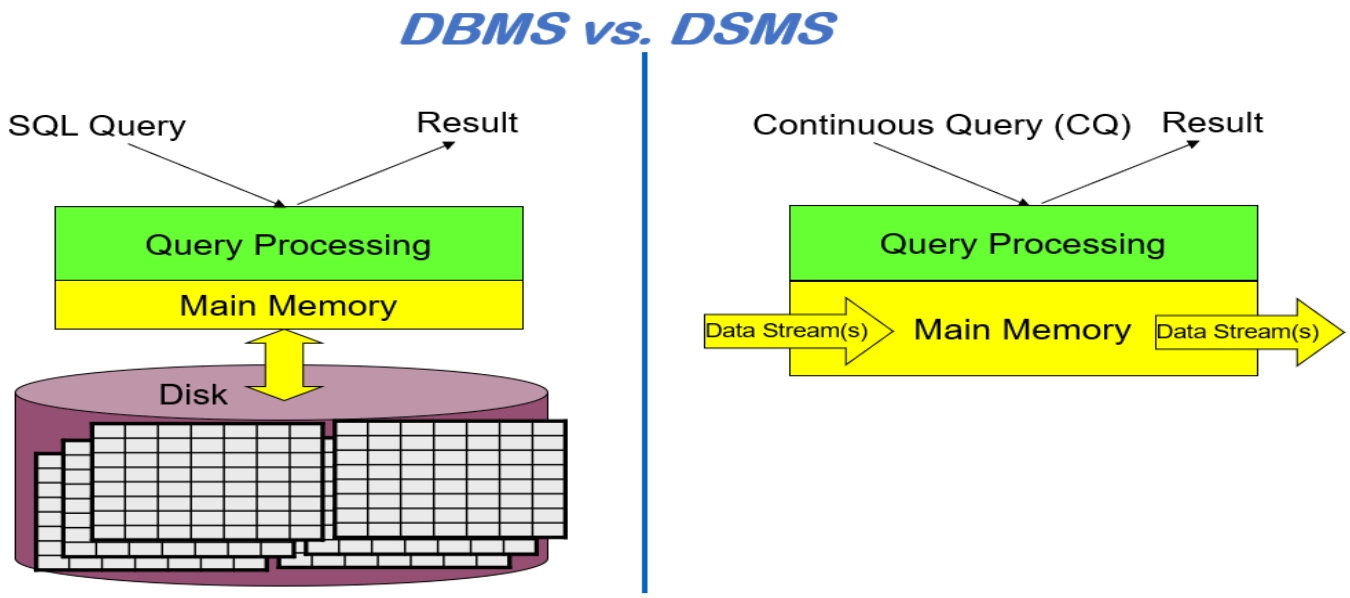


FIGURE 1.4 – Comparaison entre l'architecture des SGBDs (DBMSs) et celle des SGFDs (DSMSs)

SGBD	SGFD
• Relations persistantes (relativement statiques, stockées) • Requêtes ponctuelles • Accès aléatoire • Stockage de disques "illimité" • Seul l'état actuel compte • Pas de services en temps réel • Taux de mise à jour relativement faible • Des données à toute granularité • Des données précises • Plan d'accès déterminé par le processeur de requêtes, conception physique de la base de données	• Flux transitoires (analyse en ligne) • Requêtes continues (CQ) • Accès séquentiel • Mémoire principale limitée • Les données historiques sont importantes • Exigences en temps réel • Taux d'arrivée éventuellement multi-GB • Données avec une granularité fine • Données imprécises • Arrivée et caractéristiques des données imprévisibles / variables

Tableau 1.3 – Comparaison SGFD v.s. SGBD

1.5 Conclusion

Dans ce Chapitre, nous avons introduit les notions de base utilisées dans la suite du manuscrit. Nous avons donné une vision globale sur la théorie des ensembles flous, cadre théorique sur lequel se fonde nos travaux de recherche. Après avoir défini les SGBDs traditionnels et mentionner ses limitations, nous avons présenté les SGFDs qui sont le fruit de développement technologique pour faire face à l'explosion des données produites, sous forme continue, dans divers domaines. Cette explosion a donné naissance à un nouveau type de données nommé "flux de données" qui est caractérisé par sa nature volumineuse et sa rapidité. Ces particularités rendent irréalisable le traitement traditionnel des données et le stockage de ces données dont sa totalité, pratiquement impossible. Cependant, il est important pour certaines applications réelles de garder une trace des données ou d'avoir une forme synthétique/compacte de ces données ; soit dans le but de prise de décision en temps réel soit pour effectuer une prévision sur les données.
Pour répondre à ces besoins, des structures de résumé de données sont donc utilisées. Ces structures permettent de fournir une version compacte mais informative de données. Par ailleurs, les structures de résumé répondent aussi parfaitement au besoin d'obtenir rapidement des réponses approximatives dans des contextes où l'obtention d'une réponse exacte est un processus long, extrêmement coûteux et surtout consommateur d'énergie. Une réponse approximative est souvent désirable et apporte suffisamment d'informations pour être acceptable.

Chapitre 2

Travaux connexes : Techniques de réduction des données

L'intelligence n'est pas la capacité de stocker des informations, mais de savoir où les trouver.

— Albert Einshtein,

Sommaire

2.1 Introduction **23**
2.2 Définition d'un résumé de données **23**
2.3 Applications des résumés **23**
2.4 Aéronautique et résumés de données **24**
2.5 Techniques de résumé de données **25**
2.5.1 Clustering 26
2.5.2 Échantillonnage 29
2.5.3 Histogramme 32
2.5.4 Compression par Ondelettes 32
2.6 Conclusion **33**

2.1 Introduction

Dans la plupart des applications modernes, les données représentent la matière principale, le volume de ces données est énorme, ce qui rend leur exploitation difficile. Cette matière première dépasse la capacité des applications pour la gérer, surtout, avec des ressources limitées en termes de mémoire, d'espace de stockage et de temps d'exécution. Pour faire face aux contraintes de la volumétrie et la vitesse avec laquelle ces données arrivent, des techniques de résumé ont été proposées afin de réduire la taille de données à traiter, d'extraire l'information à partir d'une grande masse de données et d'obtenir une réponse approximative (moins coûteuse) dans le cas où une réponse exacte n'est pas très utile. Ainsi, les résumés de données est un paradigme aidant à la prise de décision fiable en temps réel et en exploitant le moindre ressources. Selon Hudec [Hud16], le résumé des données est un processus qui est désormais nécessaire à tout système intelligent dédié aux applications du monde réel.
Le résumé de données peut être donc défini comme un processus de création d'une version concise, mais informative, des données originales [Ahm19a, Ahm19b]. Les termes concis et informatifs sont assez génériques et dépendent du domaine d'application. Cette définition de résumé est très exigence, elle requiert des contraintes impliquant l'utilisation des méthodes totalement différentes. Ainsi, plusieurs techniques de résumé sont proposées dans la littérature, à titre d'exemple nous pouvons citer : l'échantillonnage, le clustering et les histogrammes.

L'objectif de ce chapitre est de définir la notion de résumé de données, de présenter certaines applications de résumé et d'examiner, en particulier, la relation entre l'aéronautique et le résumé de données. Après nous détaillons les différentes techniques de réduction de données traditionnelles. Ensuite, nous clôturons le chapitre par une conclusion permettant de dresser un bilan global du chapitre.

2.2 Définition d'un résumé de données

L'idée majeure derrière le concept de résumé de données est de garder une trace des données cibles pour être utilisée ultérieurement. Par conséquence, le résumé de données est un processus qui consiste à créer et de conserver une représentation compacte et informative des données originales. Dans la littérature, il existe trois principales catégories de résumés [Gab11] :

— **Synopsis :** les techniques de synopsis permettent de répondre à des requêtes bien définies avant l'arrivée des données. Parmi les structures de synopsis, nous pouvons citer les sketchs et le filtre de Bloom.

— **Historiques :** ou structures archivées, ces approches utilisent des agrégats spécifiés pour répondre aux besoins de l'utilisateur dans le contexte d'un SGFD. Ce type de requêtes est défini avant l'arrivée du flux de données.

— **Résumés généralistes :** le but de ces structures est de répondre d'une façon approximative à n'importe qu'elle requête sur l'historique de flux de données. Parmi les structures généralistes, nous pouvons citer l'échantillonnage et les clusters.

2.3 Applications des résumés

Le résumé de données a été largement étudié dans de nombreux domaines tels que l'analyse de texte, la surveillance du trafic réseau, le domaine financier, le secteur de la santé et bien d'autres. Comme le résumé utilise le contenu sémantique des données, il s'est avérée être une technique d'analyse de données utile et efficace pour interpréter des ensembles de données à large échelle. Le résumé est une étape importante pouvant accélérer la découverte des connaissances et facilitant ainsi les tâches de fouillé de données (qui prennent énormément de temps), tout

en réduisant intelligemment la taille des données traitées. Ci-dessous, nous présentons certaines domaines d'applications où les structures de résumé ont été utilisées :

— **Informatique médicale :** la biologie, la génétique, la clinique sont des exemples de sciences médicales qui appliquent certaines techniques de résumés pour répondre à différents objectifs. Comme les données médicales collectées peuvent être très importantes, le traitement et l'analyse est donc un processus extrêmement coûteux. Il est donc intéressant de pouvoir présenter une version compacte des données décrivant la totalité des données cliniques, tout en extrayant des informations médicales cruciales en moindre temps [WKA11].

— **Analyse financière :** le marché commercial génère une quantité énorme de données. Le traitement de ces données, la détection des fraudes, la détection des tendances, la prédiction de séries financières, l'analyse des stocks et d'autres opérations financières nécessitent l'utilisation de l'opération de résumé de données.

— **Télécommunication :** une quantité énorme de données peut être générée d'une manière continue à travers les réseaux de communication. À titre d'exemple, les informations des clients et les détails sur les appels téléphoniques qui peuvent être utiles pour la détermination des habitudes des clients et par la suite l'exploitation à des fin de marketing.

  La gestion des matérielles et des logiciels dans les entreprises de télécommunication génèrent aussi des données importantes utilisées pour la détection des anomalies et de défaillances, ce qui permet de mieux gérer les réseaux de télécommunication. Plus de détails sur les tâches réalisées en utilisant ces données sont décrits dans [Wei05].

— **Réseaux sociaux :** ces dernières années, les réseaux sociaux tels que Facebook et Twitter prennent de plus en plus une place au seine de la population. Une grande quantité de données est ainsi générée quotidiennement. L'analyse et le traitement de ces masses de données résultent en des tâches consommatrices de temps et d'espace. Par conséquent, le résumé de données prend tout son intérêt pour résoudre ces problèmes. Le résumé de texte, la recherche exploratoire et l'échantillonnage des réseaux de diffusion, la synthèse des modèles de communication dans les réseaux sociaux à large échelle peuvent être considérées comme des exemples d'application de résumé pour le Big Data [ZRH+16].

— **Réseau de capteurs :** selon [Maz07], un réseau de capteurs peut être défini comme un réseau ad-hoc auto configurable composé de capteurs intelligents, chacun ayant une certaine capacité de calcul, de stockage et de communication sans fil. Les capteurs sont utilisés dans plusieurs domaines à titre d'exemples ; la surveillance de l'environnement, les maisons intelligentes, la médecine, etc. De nos jours, les réseaux de capteurs produisent une grande quantité de données qui doivent être analysée pour extraire des informations intéressantes à des fins de prise de décision. L'évaluation des requêtes est une tâche courante dans les réseaux de capteurs, qui doivent traiter de grand volumes de données retourner une réponse à l'utilisateur. Cependant, parfois des réponses approximatives pourraient satisfaire aux besoins de l'utilisateur, d'où l'intérêt grandissant des structures de résumé dans ce contexte [JPK+19].

2.4 Aéronautique et résumés de données

Au cours des dernières années, la collecte des données des systèmes à bord des avions modernes est de plus en plus appliquée. Par exemple, les moteurs à réaction collectent des informations à 5000 points de données par seconde ; un Boeing 787 génère en moyenne 500 Go de données système par vol et un Airbus A380 est équipé de 25000 capteurs [dat]. Une grande partie de ces données est traitée pour planifier, par exemple, la maintenance, positionner les pièces de rechange et anticiper les pannes des composants. Ces systèmes spécialement conçus représentent l'un des précurseurs de ce qui est devenu le Big Data. Ils ont permis aux opérateurs de réaliser

d'importantes économies. Ils ont également permis aux fabricants d'avions et de moteurs d'offrir des nouveaux modèles commerciaux lucratifs basés sur les données.
Les données provenant de multiples capteurs fournissent également des renseignements en vol pour faire fonctionner l'avion plus efficacement et en toute sécurité tels que la position géographique, les paramètres de vol, les données météorologiques ainsi que les données opérationnelles d'autres avions à proximité ou sur la même route. Elles permettent aussi aux opérateurs d'avoir une perspective globale de gestion des risques opérationnels.

Par ailleurs, les données peuvent être collectées pour des centaines (voir des milliers) de vols sur toute une flotte et offrent ainsi un excellent aperçu opérationnel de types d'avion ou de routes spécifiques. L'analyse de ces données volumineuses pourraient produire d'excellents scénarios de formation pour la compétence continue des pilotes ainsi que des défis réalistes sur les simulateurs de vol. Cela permettrait aux compagnies aériennes non seulement d'assurer la sécurité, mais aussi de réaliser des économies de coûts potentielles. Par exemple, une formation basée sur les données pour améliorer le contrôle de l'accélérateur, le réglage de l'aéronef et l'arrondi à l'atterrissage pourrait faire économiser aux exploitants des coûts importants en termes de consommation de carburant et de fatigue des composants.

Les résumés de données prendraient donc tout leur intérêt dans le domaine aéronautique à des fins de prise de décision fiable, efficace et en temps réel.

2.5 Techniques de résumé de données

Comme indiqué ci-dessus, le résumé des données a une large utilisation dans différents domaines du monde réel. De nombreuses techniques de construction de résumé sont proposées dans la littérature [HTG+15, Ahm19a, Chi09, Cse08]. La Figure 2.1 présente une taxonomie des approches de résumé où les techniques sont divisées en deux grandes catégories pour les données structurées et non structurées. Dans le reste de ce chapitre, nous donnons un aperçu des techniques de résumé de données les plus utilisées dans la littérature. Les méthodes présentées incluent le clustering, l'échantillonnage, les ondelettes et les histogrammes.

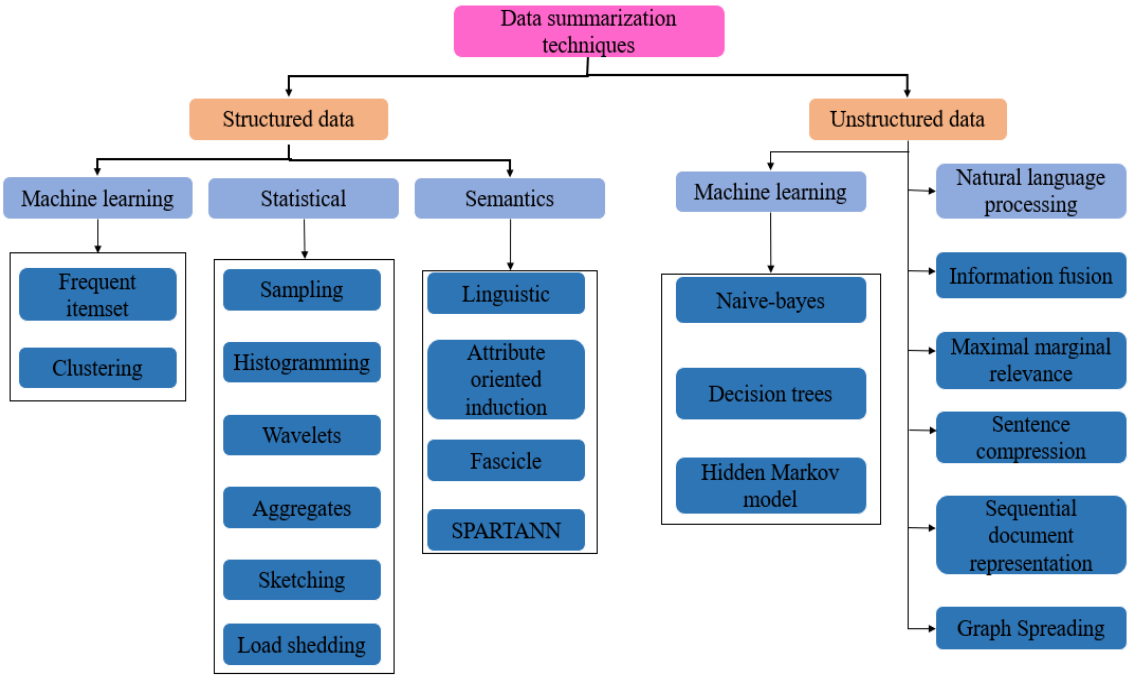


FIGURE 2.1 – Taxonomie des techniques de résumé de données

2.5.1 Clustering

Le clustering est une technique de classification non supervisée. Le cluster est utilisé généralement dans l'objectif de réaliser un résumé de données, il vise à rassembler les objets similaires dans un groupe appelé cluster. La similitude entre les objets peut être déterminée à travers différentes métriques comme la distance (qui a pour objectif de minimiser la distance entre les objets de même classe et de maximiser la distance entre les classes). Les algorithmes de clustering peuvent être classés en différents modèles tels que hiérarchique, partitionnement et les clusters basés sur la densité [KR09, SAWH14, JMF99].
Le choix des techniques de clustering ainsi que la mesure de similarité dépendent essentiellement du type de données cibles (discuté dans la section 1.4.4).

Clustering hiérarchique : également appelé clustering basé sur la connectivité, est l'une des approches classiques de résumé de données. Le principe fondamental est de créer des clusters sous la forme d'un arbre, dans lequel chaque cluster est représenté comme un nœud. En se basant sur la mesure de la proximité utilisée, les objets proches sont regroupés dans un cluster. Par conséquent, la distance entre les objets joue un rôle central dans le regroupement des objets [VM02]. Parmi les algorithmes de clustering les plus utilisés dans le contexte de résumé de données, nous trouvons BIRCH, ROCK et CURE (voir Tableau 2.1).

- **BIRCH :** (Balanced Iterative Reducing and Clustering using Hierarchies), l'une des techniques de clustering hiérarchique les plus utilisées dans le contexte des données massives. BIRCH utilise une structure arborescente, appelée CF Tree (Clustering Feature Tree). Cet arbre est équilibré en hauteur et se compose de nœuds feuilles et intermédiaires où chacun d'eux a certaines entrées. Un avantage de BIRCH est sa capacité à regrouper d'une manière incrémentale et dynamique des points de données métriques multidimensionnels entrants, dans le but de produire la meilleure qualité de clustering pour un ensemble donné de ressources (contraintes de mémoire et de temps). Dans la plupart des cas, BIRCH ne nécessite qu'une seule passe pour analyser les données [ZRL96].
- **CURE :** (Clustering Using REpresentatives), est un algorithme de regroupement adapté aux grand volumes de données. Contrairement à BIRCH, CURE est robuste contre les valeurs aberrantes et peut traiter des clusters de forme arbitraire [GRS01]. Il commence par considérer chaque point comme un cluster puis fusionner de manière récursive deux clusters existants en un seul jusqu'à ce que nous ayons que K clusters.
- **ROCK :** (RObust Clustering using linKs) [GRS00], est l'algorithme le mieux adapté pour regrouper des données catégorielles, car il peut utiliser des coefficients de similarité pour découvrir la similitude entre les deux points de données et de plus, il utilise l'idée de liens pour déterminer les voisins.

Partitionnement clustering : dans les algorithmes de partitionnement, un ensemble de données est partitionné en k partitions avec n objets dans chaque partition à l'aide d'une fonction objectif prédéfinie.
L'une des fonctions objectifs les plus utilisées est la minimisation d'erreur quadratique et qui est calculée comme suit :

$$E = \sum \sum \|p - m_i\| \tag{2.1}$$

où p est un point de données dans un cluster et m_i est le centre du cluster. Nous présentons dans ce qui suit les principaux algorithmes de partitionnement qui ont prouvé leur efficacité théorique et pratique dans le résumé de données massives.

- **K means :** [For65, M$^+$67], est probablement l'un des algorithmes de partitionnement les plus connus. Il divise les objets de données en k partitions de manière que chaque objet soit affecté au centre de cluster le plus proche. Cette opération se poursuit jusqu'à la

Algorithmes	Type de données	Forme de cluster	Complexité temporelle	Complexité spatiale
BURCH	quantitative	sphérique	O(N)	-
CURE	quantitative	arbitraire	$O(N^2_{sample} logN_{sample})$	O_{sample}
ROCK	qualitative	-	$(N^2_{sample} logN_{sample} + (N^2_{sample} + kN_{sample})$	$O(\min n^2, nm_m m_a)$

TABLEAU 2.1 – Caractéristiques des clusters hiérarchiques ; N désigne le nombre d'objet dans un cluster [HTG+15]

visite de tous les objets de données, puis le centre de gravité est recalculé pour obtenir un meilleur regroupement. Le nombre de clusters (à savoir k), l'initialisation du cluster et la distance métrique sont des paramètres spécifiés par l'utilisateur, dans lesquels la sélection de k est la tâche la plus difficile. Par conséquent, K-means est un algorithme heuristique et s'exécute plusieurs fois pour trouver les meilleures partitions avec la plus petite erreur quadratique car il vise à minimiser la somme des carrés intra-cluster. K-means est un algorithme glouton avec une complexité temporelle de O(TKN), où N est le nombre d'objets, K est le nombre de clusters et T est le nombre d'itérations. T et K peuvent être ignorés car ils sont négligeables par rapport à N. Par conséquent, l'algorithme K-means est évolutif et convient aux grands ensembles de données en raison de sa complexité linéaire.

- **CLARA :** (Clustering LARge Applications) [KR09], est un algorithme de partitionnement qui utilise PAM (Partitioning Around Medoid), qui a une complexité temporelle de $O(k(n-k)^2)$, où k est le nombre d'objets médoïdes et n est le nombre d'objets non médoïdes. Malgré la tentative de corriger les limites des algorithmes de clustering, PAM n'est pas un algorithme approprié à utiliser pour de grands ensembles de données en raison de sa complexité temporelle. Médoïde est un point de données situé à peu près au centre d'un cluster. PAM commence par trouver k médoïdes au hasard en tant que représentants de chaque cluster et forme k clusters. Ensuite, grâce à une approche par force brute, il trouve les meilleurs k médoïdes entre toutes les paires de l'ensemble de données pour regrouper parfaitement k partitions. C'est évidemment la raison de sa grande complexité. CLARA tire parti de l'algorithme PAM en l'appliquant sur un échantillon aléatoire de l'ensemble de données au lieu de l'ensemble complet. La technique prend plusieurs échantillons de l'ensemble de données, puis applique PAM sur chaque échantillon pour trouver les meilleurs k médoïdes parmi les données échantillonnées. Après cela, elle tente de découvrir les points de données les plus similaires pour chaque k médoïdes de l'ensemble de données entier pour former k clusters. Cependant, là ne garantit pas que l'algorithme peut trouver les meilleurs k médoïdes pendant le processus d'échantillonnage et n'obtient pas non plus le meilleur regroupement. Le problème avec PAM est qu'il stocke toutes les distances par paire entre les objets par conséquence, il est consommateur de l'espace de stockage et du temps, ce qui ne permet pas d'être une option applicable pour des grands ensembles de données.

- **CLARANS :** (Clustering Large Applications based upon Randomized Search) [NH02], est une version améliorée de CLARA en terme de qualité. CLARANS peut être appliqué pour des ensembles de données volumineux et de grandes dimensions, car il utilise une recherche aléatoire pour regrouper les points de données. CLARANS convient également pour rechercher des objets polygonaux. Le processus de regroupement dans CLARANS est similaire à un processus de recherche dans un graphe. Chaque nœud du graphe est un représentant d'un ensemble de k médoïdes. Deux nœuds sont voisins si leur ensemble de médoïdes diffère d'un médoïde. L'algorithme commence par un nœud aléatoire et les voisins sont vérifiés de manière aléatoire pour trouver une meilleure partition. Si les voisins fournissent une meilleure partition, ce processus reprend avec un nouveau nœud ; sinon, la recherche s'arrête en trouvant un minimum local. Cette itération continue de trouver plusieurs optimaux locaux, et le "meilleur" optimum local est considéré comme une sortie de clustering. CLARANS et CLARA sont similaires en terme d'échantillonnage. Cependant, il existe une différence entre eux lors du choix d'échantillons à partir d'un ensemble de données. Bien que CLARANS échantillonne un ensemble de voisins d'un nœud et ne considère pas tous les voisins d'un nœud, il ne limite pas la recherche à une zone localisée. Cela signifie que CLARA tire un échantillon de l'ensemble de données, puis travaille sur l'échantillon sélectionné, tandis que CALARANS dessine un échantillon de voisins et modifie dynamiquement cet échantillon et fonctionne donc sur tous les ensembles de données et pas seulement sur un échantillon particulier de l'ensemble de données. Étant donné

que CLARANS prend en compte la zone locale à chaque étape, il peut détecter les valeurs aberrantes plus précisément que CLARA et il est plus résistant à la dimensionnalité croissante.

Le tableau 2.2 résume certaines caractéristiques de K-means, CLARA et CLARANS en termes de complexité, de type de données et de forme de cluster. Ces algorithmes de clustering partitionnel arborescents mentionnés ci-dessus peuvent être appliqués pour de grands ensembles de données numériques.

Cluster basé sur la densité : [PLS11] L'outil de regroupement basé sur la densité fonctionne en détectant les zones où les points sont concentrés et celle où les point sont séparés par des zones vides ou éparses. Les points qui ne font pas partie d'un cluster sont étiquetés comme du bruit. Parmi les techniques du cluster basées sur la densité les plus étudiées dans la littérature nous trouvons :

- **DBSCAN :** (Density-Based Spatial Clustering of Applications with Noise) [EKS+96], est un algorithme de regroupement basé sur la mesure de la densité, il peut couvrir les clusters de formes et de tailles différentes à partir d'une grande quantité de données, qui contient du bruit et des valeurs aberrantes. Il utilise deux paramètres **minPts :** le nombre minimum de points (un seuil) regroupés pour qu'une région soit considérée comme dense. Et eps (ϵ) : une mesure de distance qui sera utilisée pour localiser les voisins de n'importe quel point.

  Ces paramètres peuvent être compris si nous explorons le concept appelé accessibilité de densité. L'accessibilité en terme de densité établit un point accessible depuis un autre s'il se trouve à une distance particulière (eps) de celui-ci. Cet algorithme commence par un point aléatoire ou arbitraire, et si suffisamment de voisins sont entourés dans la plage de voisinage eps d'un nœud sélectionné, un cluster est alors formé. Sinon, le point est considéré comme du bruit.
- **DBCLASD :** (Distribution Based Clustering of Large Spatial Databases) [XEKS98] est un algorithme de clustering basé sur la densité incrémentale qui utilise une distribution uniforme des points de données dans un cluster. Contrairement aux algorithmes de partitionnement tels que CLARANS, DBCLASD construit des clusters de forme arbitraire. En outre, DBCLASD ne nécessite aucun paramètre d'entrée, contrairement à l'algorithme de clustering DBSCAN nécessitant deux paramètres d'entrée qui peuvent être difficiles à fournir pour les grandes bases de données. La distance du voisin le plus proche est un paramètre clé par lequel des clusters se forment. Cet algorithme crée des clusters de manière incrémentale, ce qui signifie qu'il ne nécessite pas de charger l'ensemble des données dans la mémoire et qu'il traite chaque point de données à temps.

2.5.2 Échantillonnage

L'échantillonnage est l'une des techniques de résumé de données qui fournit une représentation concise et informative de l'ensemble des données. La définition de l'échantillonnage du dictionnaire Merriam Webster est "l'acte, le processus ou la technique de sélection d'une partie représentative d'une population dans le but de déterminer les paramètres ou les caractéristiques de l'ensemble de la population". Sur la base de cette définition, l'échantillonnage peut être considéré comme une technique de résumé qui pourrait réduire le temps et l'espace en observant simplement une partie de l'ensemble de données qui est encore informative au lieu de l'ensemble des données.

Avec l'avènement de la technologie numérique, de nombreux stockages de données supportent un énorme volume de données qui doivent être traitées et analysées pour répondre aux besoins des utilisateurs. Cependant, compte tenu cette de l'énorme quantité de données, cela demande beaucoup de ressources. Ainsi, pour résoudre ces problèmes, les techniques d'échantillonnage ont été largement appliquées dans de nombreux domaines de recherche tels que l'exploration

Algorithmes	Type de données	Forme de cluster	Complexité temporelle	Complexité spatiale
K-means	quantitative	sphérique	O(NKd)	O(N+k)
CLARA	quantitative	arbitraire	$O(k(40+k)^2 + k(N-k))$	-
CLARANS	quantitative	quadratique	-	-

TABLEAU 2.2 – Caractéristiques des clusters de partitionnement ; N designe le nombre des objets dans le cluster et k le nombre des clusters [HTG+15]

de données, la gestion des données, l'optimisation des requêtes, la réponse approximative aux requêtes, l'estimation des statistiques et le traitement des flux de données qui répondent à l'objectif de résumé. Avant d'expliquer les différentes techniques d'échantillonnage, il serait peut-être préférable de décrire quelques préliminaires utiles pour que le lecteur comprenne les techniques d'échantillonnage.

— Qu'est-ce qu'un échantillon ? Un échantillon est un représentant d'un plus grand groupe (population) qui préserve les mêmes caractéristiques de la population et l'étude est menée sur l'échantillon plutôt que sur la population.

— Qu'est-ce que la population ? La population est le grand groupe de données à partir duquel un échantillon est prélevé pour l'étudier.

Il existe deux catégories principales de techniques d'échantillonnage, à savoir celle basée sur les probabilités et celle non-probabiliste, qui fournissent un moyen efficace de résumer des grandes quantités de données.

2.5.2.1 Échantillonnage probabiliste

— **Échantillonnage aléatoire simple :** la technique de l'échantillonnage de base, l'idée principale consiste à choisir d'une manière aléatoire les individus de l'échantillon de telle sorte tous les éléments de la population ayant la même probabilité soient sélectionnés. L'avantage de cette technique réside dans sa simplicité de compréhension et d'implémentation avec un minimum d'informations à partir de l'ensemble des données. Cependant, il doit avoir une liste de toute la population [Coc07].

  Il existe une vaste étude d'échantillonnage aléatoire pour diverses applications dans la littérature. Dans le contexte d'un ensemble de données volumineux, où il n'est pas possible de stocker l'ensemble des données tel qu'un flux de données. Par conséquent, pour accélérer l'exécution de ces tâches, un échantillonnage aléatoire (qui ne nécessite pas de pré-connaissance des données) peut être utile pour traiter et analyser les données.

  Échantillonnage réservoir : dans cet algorithme, un réservoir maintient un échantillon aléatoire uniforme de taille fixe k qui est tiré pendant un passage séquentiel à travers l'ensemble des données. Cela signifie que les n premiers points de données sont ajoutés à un réservoir. Ensuite, en arrivant au n+ 1 ème point de données, l'un des points de données existants dans le réservoir est choisi au hasard pour être supprimé et remplacé par le nouveau point dans le réservoir [Vit85].

— **Échantillonnage systématique :** ou échantillonnage par intervalle, ce qui signifie qu'il existe un intervalle ou un écart entre les unités sélectionnées (il utilise le principe de fenêtre glissante). Pour construire un tel échantillon il faut suivre les étapes suivantes : préciser l'intervalle d'échantillonnage K en divisant le nombre des éléments (unités) N par la taille de l'échantillon n. L'étape suivante consiste à choisir aléatoirement un nombre entre 1 et K qui sera le point de départ de notre échantillon. Pour sélectionner les autres individus, il suffit d'ajouter le pas K au premier élément [Coc07].

— **Échantillonnage stratifié :** consiste à divisé la population en L groupes non chevauchés appelés strates, Ensuite, à partir de chaque strate on sélectionne des échantillon. Pour la sélection de l'échantillon à l'intérieur de strate, on peut utiliser n'importe quelle des méthodes d'échantillonnage mentionnées ci-dessus. Si la méthode utilisée pour le choix de l'échantillon dans le strate est l'échantillonnage aléatoire, on l'appelle la méthode échantillonnage aléatoire stratifié [Coc07].

2.5.2.2 Échantillonnage non probabiliste

Les techniques d'échantillonnage non probabilistes sont rarement utilisées. Elles consistent à choisir les échantillons d'une façon non aléatoire. De ce fait les individus n'ont pas la même probabilité d'être sélectionnés. Ce genre d'échantillonnage peut provoquer de biais de sélection lorsque l'échantillon choisi ne représente pas parfaitement la population étudiée. Un exemple de cette stratégie est l'échantillonnage par Quota qui est un peu similaire à l'échantillonnage stratifié dans lequel l'ensemble des données ou la population est divisé en groupes mutuellement exclusifs. Ensuite, par un jugement, des échantillons sont tirés de chaque groupe satisfaisant une proportion déterminée [Coc07].

2.5.3 Histogramme

L'histogramme est une méthode graphique utilisée pour représenter un grand volume de données de manière compacte. Il est généralement employé pour le résumé de données quantitatives et qualitatives. L'histogramme montre la distribution et la répartition des données selon une caractéristique. L'idée principale consiste à partitionner les données en un ensemble de groupes ou de classes disjoints. Mathématiquement parlant, un histogramme est une fonction qui représente la quantité des données se trouvant dans les classes et représente les fréquences des données dans ces classes. La fonction doit satisfaire la condition suivante :

$$y = \sum_{i=1}^{n} x_i \tag{2.2}$$

où y est le total des données, n est le nombre total des groupes et x_i représente la quantité des données dans la classe i. Cette fonction peut être représentée graphiquement (voir Figure 2.2), où chaque groupe est représenté par un rectangle et sa surface dépend proportionnellement au nombre des éléments dans le groupe. Les histogrammes sont largement étudiés dans la littérature. Dans ce qui suit nous donnons un aperçu non exhaustif de certaines approches des histogrammes.

- **Histogramme Equi-largeur :** catégorise les plages continues de valeurs d'attributs en N intervalles égaux. La largeur des intervalles est calculée en fonction des valeurs maximale (Max) et minimale (Min) de l'attribut comme suit : $W = (Max - Min)/N$ [Koo81].
- **Histogramme Equi-Fréquence :** également connu sous le nom d'histogramme à hauteurs égales, les frontières des groupes sont choisies de tel sorte que dans chaque groupe les nombres d'effectifs sont égaux [MD88].
- **Histogramme V optimal :** cette approche a pour but de réduire l'erreur d'approximation ou l'erreur de la variance entre les classes pour produire des cases homogènes en utilisant la fonction d'erreur quadratique [GSW04].
- **Histogramme Fin Biaisée :** permet de répondre à des requêtes de types *Iceberg* [FSGM$^+$], qui sont destinées à trouver les valeurs d'agrégats au-dessus d'un certain seuil. Cette approche consiste à regrouper, en premier, les fréquences les plus élevées et les plus basses dans des groupes et les autres fréquences vont se trouver dans d'autres groupes [IP95].

2.5.4 Compression par Ondelettes

Les ondelettes peuvent être considérées comme une technique de résumé principalement utilisée dans les applications de traitement d'images et de requêtes. Différentes transformations d'ondelettes telles que Haar et la transformé de fourrier sont utilisées pour transformer les données d'un domaine à un autre. L'idée majeure derrière cette technique est de décomposer les données en une série de coefficients et ne garder que les coefficients à ordre supérieur. De ce fait, la compression par ondelettes provoque une perte d'informations et produit une erreur lors de la reconstitution des données originales [SDDS96].

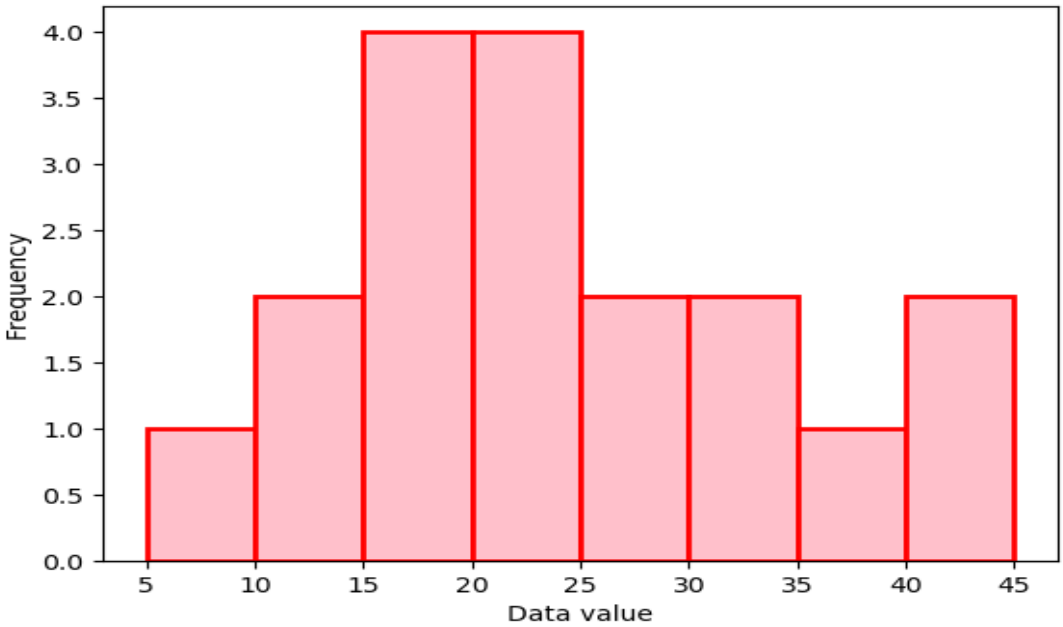


FIGURE 2.2 – Exemple d'un histogramme équi-largeur

2.6 Conclusion

De nos jours, les techniques de résumé de données sont de plus en plus utilisées pour faire face à l'explosion des quantités de données produites dans différentes domaines d'applications. Ces techniques s'avèrent pertinentes pour la prise de décision malgré la caractère approximatif des réponses retournées, mais elles sont peu coûteuses en termes de temps et d'énergie. Dans le présent chapitre nous avons discuté les principales techniques de résumé de données comme l'échantillonnage, les histogrammes et les clusters. Il faut noter que nous n'avons pas abordé ici toutes les techniques proposées dans la littérature comme les sketchs, micro-clustering, etc., pour plus de détails, nous recommandons aux relecteurs les articles [Ahm19a, HTG+15].
Bien que ces techniques soient efficaces pour la construction de résumés de données, elles souffrent cependant de certaines limitations non négligeables à titre d'exemple : le manque de représentativité et la non-utilisation du langage naturel. Ce qui rend les résumés produits difficilement compréhensibles pour les non experts. Pour surmonter ces lacunes, nous proposons d'utiliser une autre technique de résumé de données, elle est fondée sur l'utilisation des termes linguistiques (issus du langage naturel) et pouvant traiter des ensembles de données relativement volumineux. Cette proposition fera l'objet de chapitre suivant.

Deuxième partie

Contributions

Chapitre 3

Construction de résumés de données : Approches fondées sur l'intelligence computationnelle

Sans la curiosité de l'esprit, que serions-nous ?
Telle est bien la beauté et la noblesse de la science : désir sans fin de repousser les frontières du savoir, de traquer les secrets de la matière et de la vie sans idée préconçue des conséquences éventuelles.

— Marie Curie

Sommaire

3.1 Introduction **39**
3.2 Résumés basées sur les quantificateurs linguistiques **39**
3.2.1 Protoformes du résumé linguistique 40
3.2.2 Structure de base d'un résumé linguistique standard 41
3.2.3 Résumé linguistique avec restriction 44
3.2.4 Mesures de qualité 45
3.3 Résumés de données fondées sur la typicité **47**
3.3.1 Notion de la valeur typique 48
3.3.2 Approches de calcul de la valeur typique 48
3.4 Étude Expérimentale **52**
3.4.1 Ensemble de données 52
3.4.2 Cas des Résumés linguistiques 53
3.4.3 Cas des valeurs typiques 56
3.4.4 Étude Comparative 57
3.5 Conclusion **58**

3.1 Introduction

Bien que les techniques présentées dans le chapitre précédent soient faciles à mettre en œuvre pour extraire certaines connaissances et produire un résumé d'un ensemble de données, elles souffrent cependant de certaines insuffisances limitant leurs utilisations. Dans la vie réelle, les personnes sont capables de mieux comprendre les déclarations en langage naturel, pourtant elles sont moins précises que les chiffres. C'est pour relever ce défi que les scientifiques ont proposé la construction de résumés linguistiques de données. L'idée est d'utiliser des phrases quantifiées exprimées en langage naturel pour décrire les informations les plus pertinentes d'une grande quantité de données et de les présenter sous forme cohérente [Hud16]. Nous pouvons ainsi constater que le résumé linguistique est capable d'extraire les connaissances utiles et potentiellement abstraites à partir de données numériques et catégoriques. Ce qui explique l'intérêt grandissant, pour ce type de résumé, suscité par de nombreux chercheurs dans diverses domaines, y compris dans le contexte des séries chronologiques [KWZ08, KWZ10], et d'autres domaines : activités humaines, rapports financiers, énergie, analyse du trafic, réseaux sociaux, systèmes de recommandation etc. La plupart des études dans le domaine de résumés linguistiques utilisent le cadre théorique des ensembles flous pour modéliser les concepts, les termes et les prédicats du langage naturel.
Le présent chapitre est dédié à la présentation de deux techniques inspirées du raisonnement et du comportement humain. La première traite le principe des résumés linguistiques de données en décrivant les différentes composantes du résumé, les protoformes et les qualités de mesures utilisées pour évaluer la validité d'un résumé. La seconde approche, s'appuie sur la notion de la valeur typique d'un ensemble de données, pour construire un résumé de cet ensemble. Ensuite, nous présentons l'implémentation de ces deux méthodes avec quelques expérimentations menées sur différentes bases de données (données issues de vols réels et collectées dans le cadre de projet ADSB [ads16] de L'ENSMA et données réelles d'une cité intelligente collectées dans le cadre de projet NeOCampus [iri17]). Enfin, une étude comparative entre les deux approches proposées est menée.

3.2 Résumés basées sur les quantificateurs linguistiques

Pour les utilisateurs finaux, il est plus facile de comprendre un énoncé en langage naturel tel que *"la plupart des employés sont jeunes"* que de comprendre les caractéristiques statistiques d'attributs tels que la médiane, la valeur moyenne, l'écart type, etc. Une telle déclaration fournit une description concise et intelligible du contenu sémantique des données, elle est connue sous le nom de *Résumé Linguistique Flou* (RLF) dans la littérature. Le résumé linguistique de données propose une représentation textuelle et synthétique de l'information, permettant une interprétation simple en un temps raisonnable. Ce type de représentation est d'autant plus utile aujourd'hui que la quantité de données créée explose et que leur analyse exhaustive n'est plus envisageable.

L'approche classique de résumé linguistique a été proposée la première fois par Yager [Yag82] et considérablement développée par Kacprzyk et son équipe [KYZ00, KYZ02, KWZ06a, KW09] et par d'autre chercheurs [NP15, DRSV14]. Cette approche se fonde sur les principes de la logique floue et permet de produire des phrases dans le langage naturel décrivant un aspect particulier de l'ensemble des données en étude, comme "la plupart des jeunes sont grands" ou "peu de températures en hiver sont élevées". Ces phrases s'appuient sur des modèles ou des schémas globaux nommés protoformes et utilisent des modalités de variables linguistiques comme jeune ou grand et des quantificateurs comme peu ou la plupart. Chaque phrase est associée à un degré de validité indiquant la vérité de la déclaration quantifiée. Un aperçu des développements récents de ce domaine peut être trouvé dans [BAY16, BMM12, Yag21]. La section suivante aborde les

protoformes utilisés pour produire les propositions linguistiques quantifiées.

3.2.1 Protoformes du résumé linguistique

Un intérêt croissant a été exprimé plus récemment pour les résumés linguistiques flous qui fournissent une description textuelle des données numériques. Ils ont été introduits il y a des décennies et sont de plus en plus étudiés en raison de la difficulté actuelle de gérer efficacement l'immensité des données disponibles. La représentation textuelle des informations peut être plus efficace que les autres représentations décrites dans le chapitre précédent dans différents cas. Par exemple, les données peuvent être décrites dans des domaines de grandes dimensions, éventuellement, difficiles à représenter et à montrer graphiquement (cas des histogrammes), auquel cas le résumé linguistique est une alternative intéressante. De plus, il a été montré que les informations affichées, sous forme de texte, à l'utilisateur sont interprétées plus rapidement par rapport aux représentations numériques comme l'échantillonnage. Enfin, un résumé linguistique peut être lu par un système de synthèse texte-parole lorsque l'attention visuelle ne doit pas être perturbée, lors de l'exécution d'une tâche complexe par exemple [AT13], ou lorsqu'elle est déficiente.

La construction de résumés linguistiques est l'une des capacités/fonctionnalités de base dont tout système "intelligent" a désormais besoin pour une utilisation pertinente dans les applications de la vie réelle [Hud16]. Comme mentionné précédemment, et en raison de la disponibilité d'outils matériels et logiciels extrêmement développés, fonctionnelles et efficaces, nous sommes généralement confrontés à une abondance de données qui dépasse les compétences cognitives et de compréhension humaines. Dans [KZ05], les résumés linguistiques de données au sens de Yager [Yag82], illustrés par *la plupart des employés sont jeunes et bien payés* (associé avec un certain degré de vérité) à partir d'une de base de données des personnels, traduisent une compréhension intuitive cohérente humaine et un outil de découverte de connaissances (basé sur le langage naturel) des données cibles.
Des nombreuses variantes de protoformes ont été développées pour construire des résumés linguistiques dont les plus célèbres sont ceux proposées par Yager [Yag82] : "Q y sont S" et "Q R y sont S" où Q représente un quantificateur linguistique, R et S sont respectivement, le résumeur et le qualificateur linguistique. Ces notions sont détaillées dans la section suivante. Le choix du protoforme dépend à la fois du type d'informations à extraire et du type de données auxquelles la technique de résumé est appliquée. À cet effet, trois principales catégories de protoformes peuvent être distinguées [LMBM16] :

— **Les protoformes classiques :** expriment des résumés liés à des attributs particuliers sur des ensembles de données numériques et des connaissances relationnelles entre les attributs en utilisant des quantificateurs tels que la plupart, environ la moitié, quelques-uns. Ces résumés sont illustrés par des phrases quantifiées comme "la plupart des employés sont bien payés".

— **Les protoformes des séries temporelles :** ou série chronologique, ces séries expriment le comportement des attributs en fonction du temps. C'est-à-dire, des données dont la description comprend un attribut temporel. Leur spécificité induit différents types d'informations extraites et donc différents types de résumés. Une première approche consiste à extraire des attributs numériques de séries temporelles, c'est-à-dire, des approximations linéaires par morceaux de séries temporelles (les tendances), décrites par leur pente, leur durée ou leur qualité d'approximation (plus de détails seront donnés dans le chapitre 4). Ces protoformes conduisent à des résumés tels que "les tendances croissantes ont une grande variabilité". Ce genre de résumé est regroupé deux types :(i) un résumé décrivant une série temporelle [KWZ08] comme "la plupart des tendances de la consommation d'énergie étant de variabilité moyenne", (ii) un résumé considérant plusieurs séries temporelles [ALBM$^+$13] où le

protoforme "Q R y sont S $Q_t$time" est proposé, illustré par une phrase telle que "peu de patients ont la plupart du temps une valeur moyenne de la fréquence cardiaque".

— **Les protoformes temporels :** Ce sont des types distincts de résumés linguistiques pour les séries temporelles prenant en compte la spécificité de l'attribut temps : ils n'appliquent pas de quantificateurs flous mais considèrent le mode M comme une indication temporelle. Cette approche conduit par exemple à des protoformes de la forme "régulièrement y sont S", où l'adverbe "régulièrement" décrit dans quelle mesure "y sont S" s'applique compte tenu d'un ajustement temporel spécifique [MLBM13, ML16]. Un tel protoforme peut être enrichi avec des informations sur la période, comme "Adv toutes les p unités, y sont S ", où Adv est un adverbe comme "grossièrement" ou "exactement", p est une approximation de la période et "unité" est l'unité considérée. Un exemple de ce type de résumés est "au cours des 30 dernières minutes, la température était élevée".

D'autre protoformes sont existent afin de décrire les données multidimensionnelles tels que les arbres conceptuels implémentés dans le système SaintEtiQ [Ras01, RM02], et qui utilisent également des outils de logique floue, mais présentent les résultats différemment. Ils ne sont pas affichés comme des phrases en langage naturel mais comme un concept hiérarchique, où le résumé le plus général est la racine et les résumés les plus détaillés sont les feuilles.
Notons que d'autres techniques, pour trouver des représentations plus concises des données, ont été développées. Néanmoins, elles n'expriment pas de vraies phrases linguistiques. Par exemple, l'extraction de règles floues, décrite dans [CMPV99], où des relations floues entre les données peuvent être établies. Cette approche fournit une sorte de résumé non linguistique considérant que les données similaires peuvent être supprimées (puis résumées). La même remarque s'applique à la généralisation, comme dans [CH98]. Ici, les données se résument par généralisation, c'est-à-dire en trouvant des catégories qui les englobent. De cette façon, cette approche est plus proche du regroupement que du résumé linguistique.

Dans ce qui suit, nous nous focalisons sur les protoformes classiques proposés par Yager[Yag82] et développés par [KZ05]. Nous décrivons l'idée de base d'un résumé linguistique et nous discutons les mesures utilisées pour évaluer les résumés obtenus.

3.2.2 Structure de base d'un résumé linguistique standard

De manière générale, un protoforme est un modèle de phrase composé de parties constantes en langage naturel et de parties variables pouvant faire référence à des expressions linguistiques. Dans le cadre de résumé linguistique flou standard introduit par [Yag82], les protoformes " **Q y sont S**" et "**Q R y sont S**" contiennent respectivement deux et trois parties variables sous forme de sous ensemble flou : un quantificateur Q, une modalité (Résumeur) S et une modalité R dans le second cas. L'instanciation d'un protoforme est associée à une mesure de qualité liée à son adéquation aux données appelée degré de vérité comme détaillé dans ce qui suit. Nous présenterons ici brièvement l'approche de base de résumé linguistique d'ensembles de données. Cela fournira un point de départ pour notre analyse plus approfondie sur des résumés plus compliqués et réalistes. Dans l'approche de Yager [Yag82] soit :

1. V : une qualité (attribut) par exemple : V peut désigner l'âge, le salaire. ou bien n'importe quelle autre qualité.
2. Y := $\{y_1, y_2, ..., y_n\}$: un ensemble d'enregistrements satisfont la qualité V où V(y) sont les valeurs de la qualité. C'est à dire, V(y_i) la valeur de la qualité V pour l'objet y_i.
3. D : = $\{V(y_1), ..., V(y_n)\}$: l'ensemble des données que nous voulons résumer.

Un résumé linguistique d'un ensemble de données peut s'écrire donc "**Q y sont S**" avec :

1. S : désigne un "résumeur" exprimé par une valeur linguistique et représenté par un ensemble flou associé à une fonction d'appartenance qui prend des valeurs dans l'intervalle unitaire [0 1] (ex : jeune). Pour chaque attribut un ensemble des modalités ou des résumeurs lui sont associés.

2. Q : désigne un quantificateur représentant le nombre des données qui satisfont le "résumeur", cette quantité est aussi un ensemble flou qui peut prendre des valeurs absolues comme :"environ de 5", "plus ou moins 100", ou bien des valeurs relatives comme : "la plupart", "quelques".

3. T : désigne le degré de validité du résumé (degré de vérité) indiquant la vérité de la déclaration, nous pouvons le calculer en utilisant l'approche de Yager comme indique dans l'équation (3.1) :

$$T = \mu_Q[\frac{\sum_{i=1}^{n} \mu_S(y_i)}{n}] \tag{3.1}$$

   où n est la cardinalité scalaire d'un ensemble de données (nombre de tuples), $\frac{\sum_{i=1}^{n} \mu_S(y_i)}{n}$ est la proportion d'objets dans un ensemble de données qui satisfait S, et Q est la fonction d'appartenance d'un quantificateur relatif bien choisi. Les quantificateurs décrivent dans quelle mesure le résumé est valable pour l'ensemble des données considérés.

   Le résumé pourrait concerner plus d'une condition atomique jointe par le connecteur "et" [Hud16]. Si le résumé se compose de plusieurs prédicats atomiques j, $\mu_s(x_i)$ est calculé de la manière suivante :

$$\mu_s(y_i) = \mathbb{T}(\mu_{s_j}(y_i)) \tag{3.2}$$

Considérons maintenant un ensemble de données D. Nous pouvons émettre l'hypothèse de tout résumeur S approprié et de tout quantificateur Q, et la mesure de vérité supposée indiquera la vérité de la déclaration du résumé. Commençons par expliquer les deux premiers éléments, puis abordons la question de savoir comment calculer le degré de vérité.

Le résumeur Puisque pour l'être humain, le langage naturel est le premier moyen de communication nous supposons que le résumeur S est une expression linguistique modélisée sémantiquement par un ensemble flou [KZ12]. Par exemple, un résumeur comme "jeune" serait représenté comme un ensemble flou dans l'univers du discours 1, 2,. . ., 90, c'est-à-dire contenant les valeurs possibles de l'âge d'une personne compatibles avec le prédicat "jeune".

Quantificateur La quantité en accord Q également appelée quantificateur flou dans [Zad83]. C'est un terme linguistique représentant le nombre d'éléments des données qui satisfont le résumeur. Deux types de quantificateurs peuvent être employés [Lié95] :

- **absolu :** représente un ensemble flou qui prend des valeurs dans l'ensemble des nombres réels non négatifs. Par exemple "environ 5", "plus ou moins 100", "plusieurs", etc.

- **relatif :** représente un ensemble flou de l'intervalle unitaire. Par exemple "quelques-uns", "plus ou moins la moitié", "la plupart", "presque tous", etc.

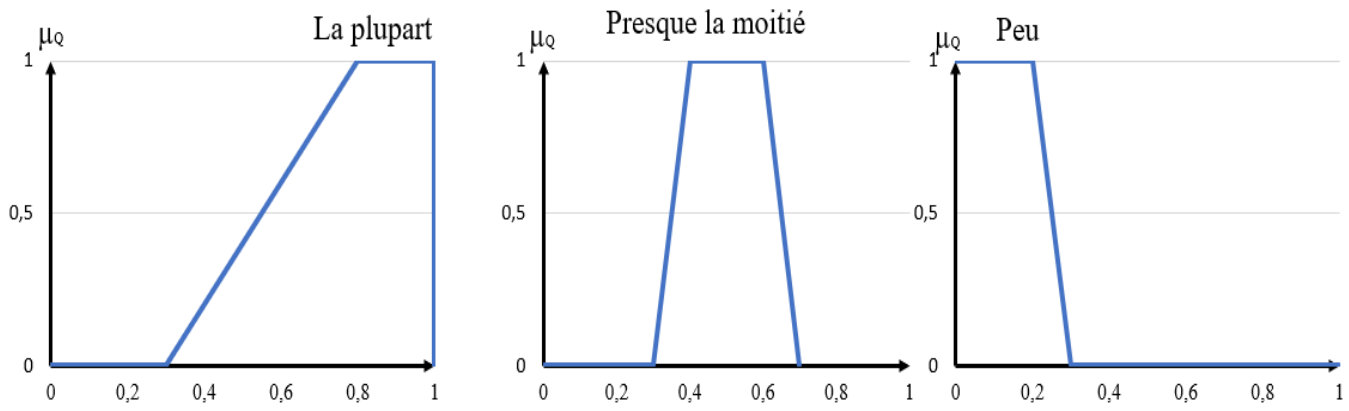


FIGURE 3.1 – Exemple des quantificateurs relatifs

Tous ces quantificateurs peuvent être formalisés par des fonctions non linéaires : Gaussien pour le quantificateur environ la moitié et fonction sigmoïde pour les quantificateurs au moins, peu nombreux, environ la moitié et la plupart. En pratique, la forme linéaire est utilisée en raison de sa simplicité et pour des aspects calculatoires. La figure 3.1 donne quelques exemples de quantificateurs relatifs les plus utilisés.
L'élicitation des fonctions d'appartenance est une tâche délicate, car les paramètres choisis de ces fonctions influencent la solution. Ainsi, les fonctions d'appartenance doivent être soigneusement construites. Les ensembles flous sont des concepts qui dépendent du contexte, "grand" dans le contexte de personnes n'a pas la même sémantique que dans le contexte des immeubles, par exemple. Les phrases linguistiques sont un moyen approprié d'expliquer les résumés, d'extraire des connaissances et de les exploiter lorsque les concepts flexibles de R et S sont correctement formalisés par des fonctions d'appartenance pour couvrir les particularités de l'attribut en étude.

Le degré de vérité Le calcul de la vérité (ou, plus généralement, de la validité ou de la qualité) d'un résumé linguistique considéré dans cette section équivaut au calcul de la valeur de vérité (dans l'intervalle unitaire) d'une déclaration quantifiée linguistiquement (comme "la plupart des employés sont jeunes"). Cela peut être réalisé par deux techniques : soit le calcul de Zadeh [Zad83] d'instructions quantifiées linguistiquement, soit l'opérateur OWA proposé par Yager [Yag88].
En utilisant le protoforme classique " Q y sont S" où Q est le quantificateur flou, y est l'ensemble de données et S est le résumeur. Le degré de vérité d'une proposition quantifiée est obtenu en utilisant la formule (3.3) :

$$T = \mu_Q[\frac{\sum_{i=1}^{n} \mu_S(y_i)}{n}] \tag{3.3}$$

Exemple illustratif : Supposons que nous ayons une collection de données décrivant l'âge des personnes :

$$\mathrm{D} = \{15, 18, 25, 19, 24, 27, 30, 43, 41, 40\}.$$

Supposons qu'un résumé proposé de cette collecte de données est "*la plupart des employés sont jeunes*" où le résumeur est "jeune" et le quantificateur est "la plupart". Ces deux concepts sont définis par l'utilisateur comme deux sous-ensembles flous. La fonction d'appartenance du résumeur " jeune" peut être définie comme indiqué dans l'équation (3.4) :

$$\mu_S(x) = \begin{cases} \frac{1}{3}x - 5 & 15 \leq x < 18 \\ 1 & 18 \leq x \leq 35 \\ \frac{-1}{10}x + 4.5 & 35 < x \leq 45 \end{cases} \tag{3.4}$$

Le degré d'appartenance de chaque tuple au sous ensemble flou "jeune" est indiqué dans le tableau 3.1.

Tuple d_i	**degré d'appartenance** $\mu_S(d_i)$	**Tuple** d_i	**degré d'appartenance** $\mu_S(d_i)$
15	0	27	1
18	1	30	1
25	1	43	0.4
19	1	41	0.8
24	1	40	0.5

TABLEAU 3.1 – Calcul de degré de satisfaction

Le quantificateur "la plupart" peut être défini comme indiqué dans l'équation(3.5) [Lié95, KYZ00]

$$\mu_Q(x) = \begin{cases} 1 & x \geq 0.8 \\ 2x - 0.6 & 0.3 \leq x \leq 0.8 \\ 0 & x < 0.3 \end{cases} \tag{3.5}$$

Soit maintenant $r = \frac{1}{n} * \sum_{i=1}^{n} \mu_S(d_i) = 0.77$, la partie de D qui satisfait S. Nous pouvons facilement vérifier que $T = Q(r) = Q(0.77) = 0.94$. Donc la validité de résumé *"la plupart des employés sont jeunes"* est égale à 0.94

3.2.3 Résumé linguistique avec restriction

Dans [KZ12], les auteurs ont introduit une autre forme de proposition quantifiée plus complexe représentée par *"Q R y sont S"* où R est une expression linguistique indiquant le qualificateur (par exemple : *La plupart des jeunes employés sont bien payés*). Dans ce cas, le degré de vérité peut être calculé par la formule (3.6).

$$T = \mu_Q[\frac{\sum_{i=1}^{n} min(\mu_S(y_i), \mu_R(y_i))}{\sum_{i=1}^{n} \mu_R(y_i)}] \tag{3.6}$$

Le qualificateur peut être atomique où composé, le rôle du qualificateur est de se concentrer sur une partie des données cibles.

Exemple illustratif Le but de cet exemple est de savoir si les livres les plus chers ont un petit nombre de pages. Les livres et leurs paramètres sont décrit dans le tableau 3.2.

Livre	Prix £	Nombre de page	$\mu_{cher}(B_i)$	$\mu_{petit}(B_i)$
B1	88	225	0.8	0.5
B2	81	210	0.1	0.9
B3	87.5	162	0.75	1
B4	52	210	0	0.9
B5	63	295	0	0
B6	82.5	220	0.25	0.6
B7	76	230	0	0.4
B8	92	188	1	1
B9	72	290	0	0

TABLEAU 3.2 – Résumé linguistique avec restriction

L'expression linguistique "prix élevé" est donnée par la fonction caractéristique (3.7), tandis que l'équation (3.8) indique la sémantique du prédicat "le petit nombre de pages".

$$\mu_{cher}(x) = \begin{cases} 0 & \text{si } x \leq 80 \\ \frac{-x}{10} - 8 & 80 \leq x \leq 90 \\ 1 & \text{ailleurs} \end{cases} \tag{3.7}$$

$$\mu_{petit}(x) = \begin{cases} 1 & \text{si } x \leq 200 \\ \frac{-x}{50} + 5 & 200 \leq x \leq 250 \\ 0 & \text{ailleurs} \end{cases} \tag{3.8}$$

La première étape consiste à calculer la portion qui satisfait le résumé :

$$\frac{\sum_1^n min(\mu_S(y_i).\mu_R(y_i))}{\sum_{i=1}^n \mu_R(y_i)} = \frac{0.5+0.1+0.75+0.25+1}{0.8+0.1+0.75+0.25+1} = 0.865$$

L'étape suivante consiste à insérer cette valeur dans le quantificateur "la plupart" étant exprimé par (3.5). De toute évidence, la validité égale 1, ce qui conduit à l'acceptation totale et entière de la déclaration testée "la plupart des livres qui ont un petit nombre de pages sont chers". Mais, il faut être prudent en tirant des conclusions, car d'autres aspects peuvent influencer la solution.

3.2.4 Mesures de qualité

Le pas le plus crucial dans le résumé linguistique flou est bien sûr l'évaluation de ce dernier, donc la sélection de méthode appropriée pour évaluer les résumés linguistiques dans le sens de différentes vues comme la qualité, la quantité, la pertinence et la simplicité devient indispensable. Selon [KZ05], le critère de validité de base, c'est-à-dire la vérité d'une déclaration linguistique est certainement le plus important. Cependant, il ne décrit pas tous les aspects d'un résumé linguistique. Certaines tentatives pour concevoir d'autres critères de qualité (validité) ont été proposées dans [KZ05, KWZ06a].

Dans son article, Yager [Yag82] a proposé la mesure de spécificité d'une modalité d'une variable linguistique. Cette mesure consiste à évaluer la faculté à référencer précisément un élément d'un ensemble de données. Elle a été définie comme la somme de l'inverse du nombre d'éléments des

$\alpha - cut$ de la fonction d'appartenance caractérisant la modalité floue. Le degré d'imprécision proposé dans [KZ05] décrit le rapport entre la taille du support de la fonction d'appartenance et le nombre des éléments de l'univers de discours. Ces deux mesures sont liées au modalité linguistique (le résumeur S). En revanche, nous pouvons distinguer des mesures de qualité dépendant des quantificateurs utilisés, comme le degré de couverture, et d'autres reposent sur le choix de protoforme comme le degré de pertinence et la longueur de résumé [Moy16]. Nous présentons dans la suite les mesures de qualité les plus pertinentes, utilisées pour évaluer un résumé linguistique, introduites dans [KZ05].

- **Degré de vérité :** la mesure la plus étudiée dans la littérature pour estimer la fiabilité d'un résumé. Elle représente la mesure du degré d'adéquation des données à une phrase linguistiquement quantifiée. La méthode la plus utilisée pour calculer le degré de vérité d'un protoforme "Q y sont s" est celle proposée par Yager [Yag82] et indiquée dans l'équation (3.1). La vérité d'un résumé est caractérisée par de nombreuses propriétés. Tout d'abord, un résume doit être résistible à la permutation de données, l'ordre d'arrivée de données ne doivent pas influencer et perturber le calcul du degré de vérité. En plus, un résume doit assurer la propriété d'inclusion et d'intersection ; prenons l'exemple deux modalités A et B, si $A \subset B$ alors $T_A \geq T_B$.
  Il faut noter que le calcul de degré de vérité doit être fonctionnelle avec tous les types de quantificateurs relatifs et absolus, universel ou existentiel. Enfin, la dernière propriété qui devrait être garantie est la cohérence de résumé, qui regroupe l'égalité d'antonymie, la négation externe et la dualité. L'égalité d'antonymie vérifie si les deux déclarations " la plupart des températures sont hautes" et "peu de températures ne sont pas hautes" ont le même degré de vérité. La négation externe assure que la vérité de la déclaration " la plupart des températures sont hautes" est au complément de " la plupart de températures ne sont pas hautes". La dualité implique que la valeur de vérité de " Beaucoup de jeunes ne sont pas grands" est au complément de " Pas beaucoup de jeunes sont grands".
- **Degré d'imprécision :** est un critère de validité important et évident. Il décrit le rapport entre la taille du support de la fonction d'appartenance et le nombre des éléments de l'univers de discours.
  Supposons un "résumeur" S présenté par un ensemble des valeurs floues $S = s_1, s_2, ..., s_m$. Pour un ensemble flou s_j, j=1,...,m, nous pouvons définir le degré d'interférence comme

$$in(s_j) = \frac{card\{x \in X_j : \mu_s(x) > 0\}}{card(X_j)} \tag{3.9}$$

  $card(X_j)$ représente le nombre des éléments dans l'ensemble des données, donc le degré d'imprécision est défini comme indiqué dans la formule (3.10). Ce degré dépend de la forme de résumé et non pas de la base de données cible.

$$T_2 = 1 - \sqrt[m]{\prod_{j=1,...,m} in(s_j)} \tag{3.10}$$

- **Degré de couverture :** indique combien d'objets dans l'ensemble des données correspondant au qualificateur R (indiqué ici par w_g) sont "couverts" par le résumé particulier, le résumeur S. La valeur de ce degré dépend clairement du contenu de la base de données (voir équation (3.11)). Son interprétation est simple ; supposons que le degré de vérité de la déclaration "la plupart des tendances croissantes ont une faible variabilité" est élevé mais elle ne contient qu'un seul élément de la base de données donc son degré de couverture est faible et la déclaration est trompeuse car un seul élément est cohérent avec le résumé. Cette définition de la couverture permet donc de favoriser les phrases décrivant une quantité importante de données et d'ignorer les phrases à faible nombre de données.

$$T_3 = \frac{\sum_{i=1}^{n} t_i}{\sum_{i=1}^{n} h_i} \tag{3.11}$$

avec
$t_i = \begin{cases} 1 & \text{si } \mu_S(y_i) > 0 \text{ et } \mu_{w_g}(V_g(y_i)) > 0 \\ 0 & \text{ailleurs} \end{cases}$
$h_i = \begin{cases} 1 & \mu_{w_g}(V_g(y_i)) > 0 \\ 0 & \text{ailleurs} \end{cases}$
où $V_g(y_i)$ sont des valeurs de qualité V_g pour l'objet y_i.

- **Degré de pertinence :** décrit la dépendance entre les attributs. Il est utilisé pour le protoforme "Q R y sont S". Si les deux attributs R et S sont dépendants la valeur de pertinence est élevée. L'idée derrière cette mesure est : une phrase dont les deux attributs R et S sont dépendants est plus intéressante qu'une autre dont les attributs ne le sont pas [KY01, KYZ02]. La formule (3.14) permet de calculer le degré de pertinence.

$$S_j(y_i) = \mu_{sj}(V_j(y_i)) \tag{3.12}$$

et

$$r_j = \frac{\sum_{i=1}^{n} h_i}{n} \tag{3.13}$$

avec $h_i = \begin{cases} 1 & S_j(y_j) \\ o & ailleurs \end{cases}$
et le degré de pertinence, T_4 est défini par

$$T_4 = abs(\prod_{j=1,..,m} r_j - T_3) \tag{3.14}$$

- **Longueur de résumé :** peut être défini comme la taille de la déclaration linguistique, elle dépend de nombre des prédicats utilisés dans le résumeur et le qualificateur. Une déclaration plus courte est valorisée par rapport à une autre plus longue. L'équation (3.15) montre le calcul de la longueur de résumé.

$$T_5 = 2(0.5^{cardS}) \tag{3.15}$$

Chacune des mesures partielles décrites précédemment fournit une information sur le résumé concernant un aspect donné. Yager [Yag88] a proposé une mesure globale de la qualité d'un résumé en combinant l'ensemble de ces mesures partielles. Cette mesure s'appelle **degré de validité**, elle est définie comme la somme pondérée de ces 5 mesures de qualité, elle est donnée par l'équation :

$$T = \sum_{i=1,2,..,5} w_i T_i \tag{3.16}$$

avec $w_1, ..., w_5$ les poids associés à chaque degré de vérité dans l'intervalle unitaire vérifiant :

$$\sum_{i=1,2,...,5} w_i = 1$$

Il est à noter que d'autres chercheurs ont tenté de concevoir d'autres critères de validité comme l'utilité, la simplicité, l'informativité, etc., pour plus de détails voir [Hud16, COMS09, COMST12].

3.3 Résumés de données fondées sur la typicité

Dans de nombreuses applications sur l'analyse intelligente de données, la question de la détermination d'un objet typique (prototypique) à partir d'une collection d'objets se pose. Cette question revêt une importance particulière dans le développement des systèmes de découverte des connaissances. L'idée d'une valeur typique joue un rôle important dans les systèmes de raisonnement où les valeurs typiques sont souvent utilisées comme des valeurs par défaut [DP93].

Il convient de noter que pour les ensembles de données issues des applications modernes, les méthodes classiques de la détermination de la valeur typique tel que la moyenne, la médiane ne peuvent pas fournir une valeur qui pourrait être l'élément le plus similaire de l'ensemble des données.
Dans ce qui suit, nous définissons le concept de la valeur typique, puis nous introduisons quelques méthodes, fondées sur le cadre théorique de la logique floue, pour calculer ce concept tout en expliquant leurs limitations dans le cas où nous avons un gros volume de données.

3.3.1 Notion de la valeur typique

Le concept de la typicité est largement étudié dans la littérature afin d'extraire de la connaissance à partir d'un ensemble de données. Certains auteurs considèrent la valeur typique comme une méthode classique permettant d'offrir une vision globale sur l'ensemble des données cibles. Ils l'ont défini comme la valeur moyenne, le mode et la médiane. En revanche, d'autres chercheurs ont exploité la logique floue pour déterminer l'intervalle optimal de la valeur typique d'un ensemble de données.
Soit une collection d'observations, nous utilisons ces observations comme une base pour trouver s'il y a une valeur typique caractérisant cette collection. De manière informelle, une valeur typique est une valeur qui est identique ou très similaire à la plupart des observations dans les données cibles que nous voulons caractériser [Yag97] [DPR98] [Zad96]. La valeur typique pourra être caractérisée par :

- Elle représente une collection d'observations.
- C'est la valeur la plus similaire à la plupart des observations que nous essayons de caractériser.
- Elle peut être un ensemble de valeurs ou une valeur linguistique représentée par un sous-ensemble flou.

Nous discutons ci-dessous deux approches pour la détermination de la valeur typique d'un ensemble de données/observations/objets.

3.3.2 Approches de calcul de la valeur typique

3.3.2.1 Approche de Yager

Dans ce qui suit, nous examinons l'approche proposée par Yager [Yag97] permettant de déterminer la valeur typique d'une collection de données. L'idée est que la typicité est une question de degré et donc la possibilité qu'une collecte de données n'ait pas du tout une valeur typique (qui est fondamentale dans cette approche). Par exemple, la collection $\{10, 20, 30, 40, 50, 60, 70\}$ n'a pas de valeur typique. Dans cette approche, la valeur typique n'est pas nécessairement une valeur précise particulière. Elle peut être une valeur linguistique représentée par un sous-ensemble flou.

Dans son approche, Yager suppose que V est une variable dont le domaine de discours X est un sous-ensemble de valeurs représenté par un intervalle $[a, b]$. Soit D est une collection de n observations de V, parmi X.
D'après Yager, une valeur typique de V basée sur les données D est un sous-ensemble flou unimodal A de X. De plus, pour que l'ensemble A soit une valeur typique de D, il doit satisfaire deux critères :

— La première condition est que la plupart des éléments de la collection D soient compatibles avec le concept A suggéré comme valeur typique. Cette condition assure que la valeur typique est bien représentative de l'ensemble de données D.

— La deuxième condition associée à une valeur typique acceptable est qu'elle doit être "étroite" (moins imprécise) en bande passante spécifique. En substance, bien que l'approche permette aux ensembles flous d'être des valeurs typiques, nous souhaitons que ces ensembles flous soient "étroites" (moins imprécis).

La nécessité de satisfaire ces deux conditions conduit à une mesure de typicité associée à tout sous-ensemble A. Nous montrons maintenant les formules permettant de mesurer le degré auquel un sous-ensemble flou proposé A satisfait l'idée d'être une valeur typique pour une collection de données D.

Index de compatibilité : se réfère au degré de satisfaction qu'un sous-ensemble flou proposé A satisfait les conditions pour être une valeur typique pour une collection de données D. Supposons que A est une valeur typique proposée associée à la collection D, nous définirons l'indice de compatibilité de A par rapport à D, où n est la cardinalité de D, comme :

$$comp(A/D) = \sum_{d_i \in D} \frac{\mu_A(d_i)}{n} \tag{3.17}$$

Index de spécificité : l'exigence de l'étroitesse (précision) de la valeur typique nécessite l'introduction du concept de spécificité. Supposons que A est un sous-ensemble flou de l'univers de discours X et d'un intervalle [a, b], l'indice de spécificité de A par rapport à X est alors défini comme :

$$Sp[A/X] = 1 - \frac{\int_a^b \mu_A(x)dx}{a-b} \tag{3.18}$$

Il est intéressant de noter que ces indices doivent appartenir à l'intervalle unitaire [0, 1].

Degré de typicité : après avoir introduire les deux mesures : compatibilité et spécificité, la mesure de la typicité d'un ensemble flou A par rapport à une collection de données D est obtenue en considérant la conjonction de ces deux exigences :

$$Typ(A/D) = Sp(A/X) \wedge comp(A/D) \tag{3.19}$$

où $\wedge$ désigne le symbole de conjonction. Notons que tout opérateur t-norme peut être utilisé pour modéliser la conjonction (comme le minimum, le produit, etc.).

Exemple illustratif :

Supposons que nous avons un ensemble de données D de l'intervalle [0, 100] :

$$D = \{12, 12, 11, 9, 13, 10, 9, 9, 11, 14\}.$$

Dans ce cas, la valeur moyenne est $\bar{d} = 11$. Considérons maintenant la valeur typique (ensemble typique flou) A = "environ 11", comme montre la figure 3.2.

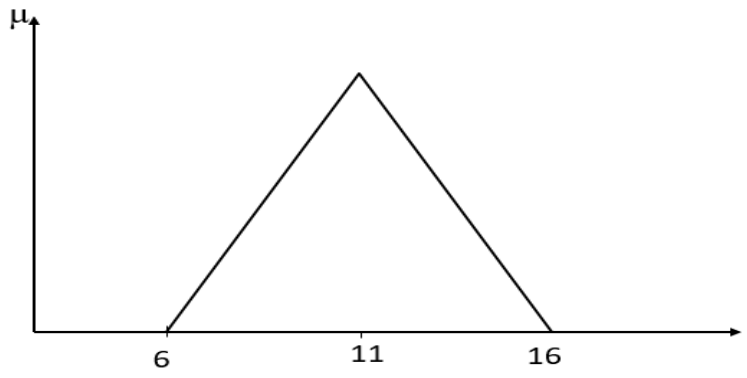


FIGURE 3.2 – Valeur typique "environ 11"

L'indice de compatibilité et de spécificité sont calculés en utilisant les équations (3.17) et (3.18) respectivement, nous trouvons :

$$\text{comp(A/D)=0.72}$$

$$Sp(A/X) = 0.95$$

La mesure $Typ(A/D)$ est calculée en utilisant le minimum entre la spécificité et la compatibilité :

$$Typ(A/D) = min(Sp(A/X), comp(A/D)) = 0.72$$

La principale critique de cette approche est le fait qu'elle pourrait conduire à un nombre important d'ensembles typiques ayant le même degré de spécificité. De plus, chaque ensemble que nous déterminons pourrait être relatif à la même collection d'observations, donc il faut connaître l'ensemble des données et voir une vue globale sur de données pour proposer une valeur typique ce qui est pratiquement irréalisable. Pour surmonter ce problème, Yager a introduit une nouvelle idée basée sur le calcul de la probabilité de chaque ensemble.

3.3.2.2 Approche de Dubois et Prade

Dans [DPR98], les auteurs ont constaté que la méthode présentée précédemment présente certaines limites non négligeables, à titre d'exemple, l'ensemble typique proposé pourrait être plus large que l'ensemble initial des données. Par ailleurs, il peut aussi exister plusieurs ensembles typiques ayant le même degré de typicité, ce qui rend le choix de l'ensemble typique une tâche délicate nécessitante une connaissance préalable du domaine, et qui peuvent être difficile à comprendre pour un utilisateur non familier avec la notion de valeur typique. À cet effet, Dubois et Prade ont suggéré une nouvelle approche permettant de définir un intervalle typique optimal au lieu d'un ensemble flou typique. Cette approche a pour but de trouver I* le meilleur intervalle optimal, elle est basée sur la densité de classification et d'extraction de ℓ* le pas optimal entre les deux bornes de l'intervalle I*.

Le principe de l'approche est décrit dans ce qui suit : supposons que nous avons un ensemble de données D, nous pouvons définir X comme le domaine de définition de D, $X = [D_{min}, D_{max}]$. Supposons que $L = D_{max} - D_{min}$ est la longueur de l'intervalle X. Nous définissons $I = [x_i, x_i + \ell] \subset X$ avec $\ell \in [1, L]$ et la fonction du coût f(x_i,ℓ) qui est proportionnelle à la probabilité de sélection des valeurs de l'intervalle I par rapport aux éléments de l'ensemble D.

$$f(x_i, \ell) = \frac{|I(x_i, \ell)|}{n} \tag{3.20}$$

Pour une valeur donnée de ℓ nous déterminons l'intervalle qui maximise la fonction de coût :

$$f(x_\ell, \ell) = \frac{|I^*(x_i, \ell)|}{n} = \max_{x_k \in D} f(x_k, \ell) \tag{3.21}$$

Les valeurs induites sont alors les intervalles optimaux pour chaque pas. Pour obtenir la valeur typique de l'ensemble de départ D, on doit calculer $f(x_i, \ell) - \frac{\ell}{L}$.
Dans le paragraphe suivant, une brève explication par un exemple de la méthode est présentée pour déterminer la valeur typique d'un ensemble de données D.
Exemple illustratif :
Supposons que nous avons une collection d'observations :

$$D = \{0, 3, 4, 5, 6, 7, 8, 9, 12, 23\}$$

Avec l'occurrence de chaque $d_i \in$ D est respectivement $f_i \in$ U :

$$U = \{1, 1, 1, 4, 7, 5, 3, 5, 2, 1\}$$

La longueur de D est L=23, et ℓ le pas que l'on pourrait prendre dans D avec le maximum de la longueur L, on peut noter ici l'intervalle I$(x_i, l) = [x_i, x_i + \ell] \subset X$ avec $\ell \in [1, L]$ on utilise la fonction de coût (3.20).

ℓ	I*(x_l,ℓ)	f(x_l,ℓ)	$\frac{\ell}{L}$	f*(x_l,L)-$\frac{\ell}{L}$
1	[6,7]	12/30	1/23	0.3565
2	[5,7]	16/30	2/23	0.4460
3	[6,9]	20/30	3/23	0.5360
4	[5,9]	24/30	4/23	0.6266
5	[4,9]	25/30	5/23	0.6159

Tableau 3.3 – Exemple illustratif

Pour obtenir la valeur typique de l'ensemble de départ D, nous devons calculer f$(x_i, \ell) - \frac{\ell}{L}$ et considérer l'intervalle I* comme le meilleur intervalle selon l'équation 3.21. Par simple interpolation, nous obtenons ℓ*=4, avec la meilleure fonction de coût $f(x_\ell, \ell) = 24/30$. Nous pouvons déduire ensuite l'intervalle [5,9] comme l'intervalle typique optimale dans D, voir Tableau 3.3. L'algorithme 1 représente la méthode à suivre pour avoir l'intervalle typique.
À cause de sa complexité de calcul, l'algorithme proposé dans [DP93] est consommateur en termes de temps d'exécution et d'espace mémoire, car il doit balayer toutes les données à chaque fois pour extraire l'intervalle optimal pour chaque pas. Il nécessite en plus, plusieurs passes pour déterminer l'intervalle optimal typique. Dans la section suivante, nous montrons la validité de cette hypothèse en proposant une série d'expérimentations sur différentes bases de données.

Algorithm 1 L'algorithme de DUBOIS et PRADE

```
for ℓ = 1 to L do
    for x_i = x_0 to n do
        f(x_i, ℓ) = Card(I(x_i, ℓ))/n
        if f(x_i, ℓ) >= max1 then
            max1 ← f(x_i, ℓ)  I ← I(x_i, ℓ)
        end
    end
    if f(x_i, ℓ) > max2 then
        max2 ← f(x,ℓ) − ℓ/L
        I* ← I
    end
end
```

3.4 Étude Expérimentale

Les procédures de résumés proposées dans les sections précédentes ont été implémentées sur deux bases de données réelles. La première est une base de données statique représentant des enregistreurs de données de vol ; ces données sont recueillies dans le cadre du projet ADSB (Automatic Dependent Surveillance-Broadcast) [ads16] de l'ENSMA. La deuxième base de données représente un flux de données collectées dans le cadre du projet NeOCampus soutenu par l'Université de Toulouse III [iri17].

Dans ce qui suit, nous présentons les deux bases de données utilisées pour validés les approches proposées. Ensuite, nous détaillons les expérimentations menées sur ces bases. Nous clôturons cette section en donnant une étude comparative entre les deux approches : Résumé linguistique quantifié et le concept de la valeur typique.

3.4.1 Ensemble de données

3.4.1.1 Base de données ADSB

L'objectif du projet ADSB est de stocker sur les serveurs de l'ENSMA toutes les informations des avions survolant l'école [ads16]. Les données émises par les transpondeurs des avions peuvent être captées avec des antennes. La base de données est séparée en 3 sous-bases de données différentes :

— **Données avion en vol :** affichent les informations relatives au vol de l'avion, par exemple, l'aéroport de départ, l'heure de départ, la destination,. . ..

— **Données météorologiques locales :** le but est de donner la météo relative à un aéronef. Cela permet de connaître la vitesse du vent et la pression. Ces données sont fournies par l'API Web Open Weather Map

— **Données avion :** informations sur les avions (modèle, constructeur, ...), les aéroports (position GPS, ICAO et IATA des villes et pays, ...) et les itinéraires (numéro de vol, arrivée, destination, ...)

Dans cette étude, nous avons choisi deux attributs : l'altitude et la Vitesse sol qui sont considérés parmi les attributs les plus critiques lors d'un vol. Ils dépendent de plusieurs attributs tels que la pression, la vitesse du vent, l'angle d'attaque. Le choix optimal de la vitesse et de l'altitude minimise le coût global du vol et assure la sécurité des aéronefs et des passagers.

3.4.1.2 Base de données NeOCampus

Ce projet de recherche est soutenu par l'Université de Toulouse III [iri17]. Son objectif est de démontrer les compétences des chercheurs de différents domaines de l'Université pour la conception du campus du futur. Trois objectifs majeurs sont identifiés : faciliter la vie de l'utilisateur du campus, réduire l'empreinte écologique, contrôler la consommation d'énergie. Le campus est vu comme une ville intelligente où plusieurs milliers de flux de données proviennent de capteurs intérieurs et extérieurs hétérogènes (CO2, vent, humidité, luminosité, présence humaine, consommation d'énergie et de fluides ...). Des techniques d'intelligence artificielle sont utilisées pour comprendre les objectifs et le comportement des citoyens à partir de sous-ensembles de données sélectionnés manuellement. Nous pouvons distinguer différents types de données :

— Données brutes : ce sont les données de consommation d'énergie (eau, électricité, gaz).

— Données spécifiques à l'activité : il s'agit de données prétraitées issues de la fusion de données brutes (activités pédagogiques, occupation des salles ...).

— Données spécifiques aux incidents : il s'agit des pannes identifiées sur le campus (échauffement des équipements informatiques, pannes réseau, ...)

— Données ambiantes : elles concernent le contexte dans lequel se déroule le scénario (température, météo, niveau de CO2 dans l'air).

3.4.2 Cas des Résumés linguistiques

Nous appliquons sur les deux base de données présentées ci-dessus deux algorithmes de résumé : le résumé linguistique basé sur l'utilisation des propositions linguistiques quantifiées dans le sens de Zadeh [Zad83], et le concept de la valeur typique décrit par l'algorithme 1. Le but de cette étude est de produire des résumés d'une large base de données et voir la méthode qui se comporte au mieux dans le cas des données massives. Cette étude a fait l'objet d'une publication scientifique [KAM20].

3.4.2.1 Données statiques

Dans notre cas d'étude, nous avons choisi quatre quantificateurs relatifs pour implémenter l'algorithme de résumé linguistique flou et tester les deux bases de données : la plupart, certains, environ la moitié et peu. Ces quantificateurs sont caractérisés par des fonctions d'appartenance permettant de calculer la portion de la base adéquate au quantificateur. Le terme "La plupart" est défini par la formule (3.5) indiqué dans la page 44, le quantificateur relatif "Certains" peut être défini comme suit :

$$\mu_Q(x) = \begin{cases} 10x - 1 & 0.1 \leq x < 0.2 \\ 1 & 0.2 < x \leq 0.3 \\ -5x + 2.5 & 0.3 < x \leq 0.5 \\ 0 & \text{ailleurs} \end{cases} \tag{3.22}$$

Le quantificateur "presque la moitié" est donné par :

$$\mu_Q(x) = \begin{cases} 5x - 1.5 & 0.3 \leq x < 0.5 \\ 1 & x = 0.5 \\ -5x + 3.5 & 0.5 < x \leq 0.7 \\ 0 & \text{ailleurs} \end{cases} \tag{3.23}$$

Et le dernier quantificateur "peu" est caractérisé par la fonction :

$$\mu_Q(x) = \begin{cases} 10x - 1 & 0.1 \leq x < 0.2 \\ 1 & 0.2 < x \leq 0.3 \\ 0 & \text{ailleurs} \end{cases} \tag{3.24}$$

La figure 3.3 présente les fonctions d'appartenance des quantificateurs relatifs (peu, certains, environ la moitié et la plupart).

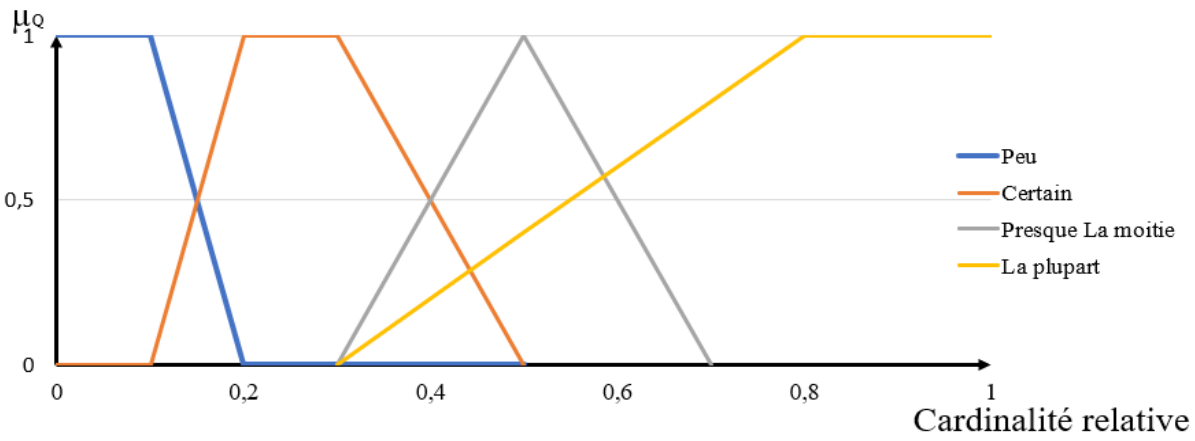


FIGURE 3.3 – Quantificateurs relatifs

Dans un premier temps, nous appliquons les tests sur les données de vol du projet ADSB de l'ENSMA stockées dans PostgreSQL. Ces données sont à caractère statique. La mise à jour de la base de données ADSB est interrompue depuis deux ans. Nous avons choisi deux attributs, l'altitude de l'avion (ALT) mesurée en ft, le second attribut représente la vitesse sol (GS) mesurée en nœuds. Pour la première variable, nous utilisons 4 valeurs linguistiques (faible, moyenne, élevée et très élevée) comme montre la figure 3.4.

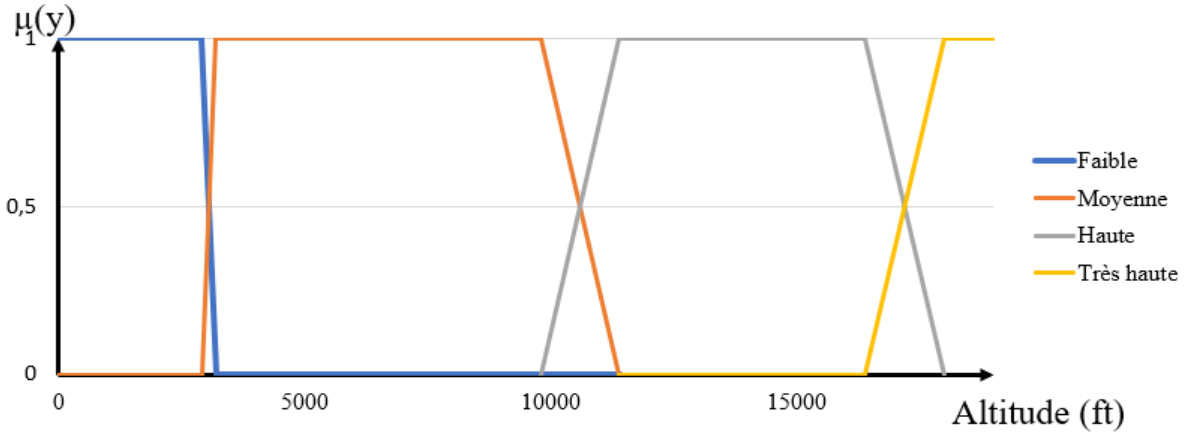


FIGURE 3.4 – Ensembles flous représentant l'altitude

La figure 3.5 décrit les valeurs linguistiques utilisées pour caractériser la vitesse sol. Les attributs choisis sont considérés parmi les attributs les plus critiques lors d'un vol, ils dépendent de plusieurs autres attributs tel que la pression, la vitesse du vent, l'angle d'attaque.

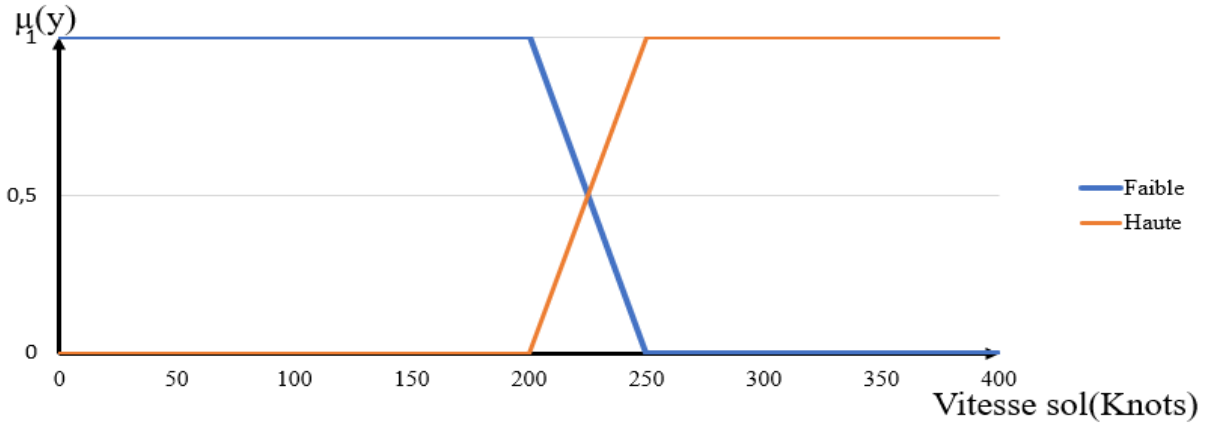


FIGURE 3.5 – Ensembles flous représentant la vitesse sol

La base de données utilisée pour mener les expériences contient un million de tuples. Parmi les résultats obtenus nous présentons ici seuls qui ont le degré de vérité le plus élevé. Nous obtenons les résultats suivants :

— T (peu d'altitude sont très élevées) = 1 qui représente le meilleur résumé pour l'attribut Altitude.

— T (peu de vitesse sol sont faibles) = 0,87 ce qui représente le meilleur résumé pour l'attribut Vitesse sol.

— Nous appliquons un test de quantificateurs sur les données de l'enregistreur de vol qui prend environ 3,328 secondes comme temps d'exécution.

3.4.2.2 Données dynamiques

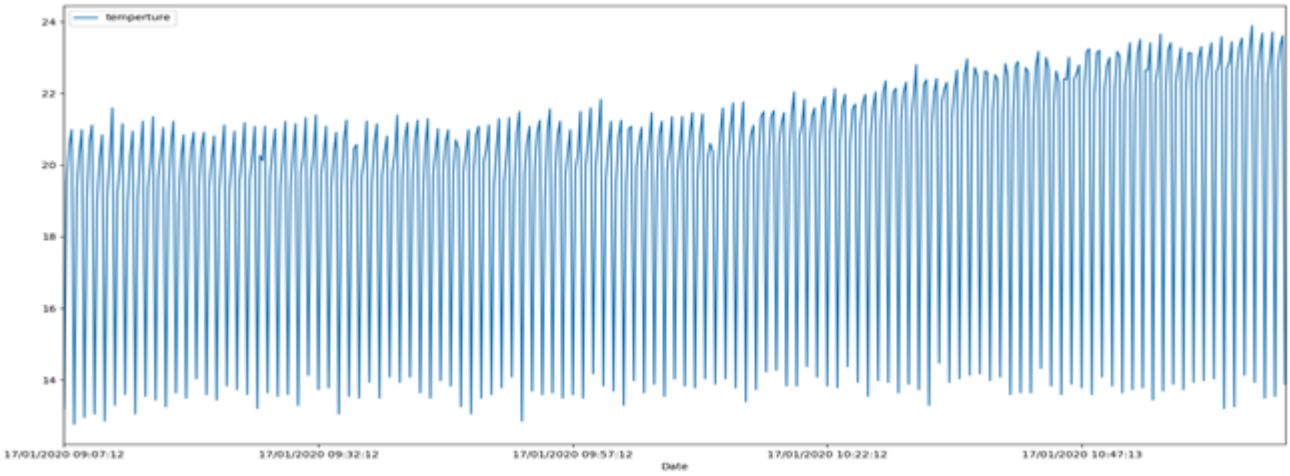


FIGURE 3.6 – Variation de température du campus intelligent

Nous appliquons l'algorithme de résumé linguistique sur les températures collectées à partir de la ville intelligente du projet NeOCampus mesurées en (°c) où la distribution des étiquettes linguistiques utilisées pour décrire la température est représentée dans la figure 3.7, la variation de ces températures pendant une demi-heure est illustrée par la figure 3.6.

— T (la plupart des températures sont moyennes) = 0,84

— T (peu de températures sont basses) = 1 représente le meilleur résumé de l'attribut Température

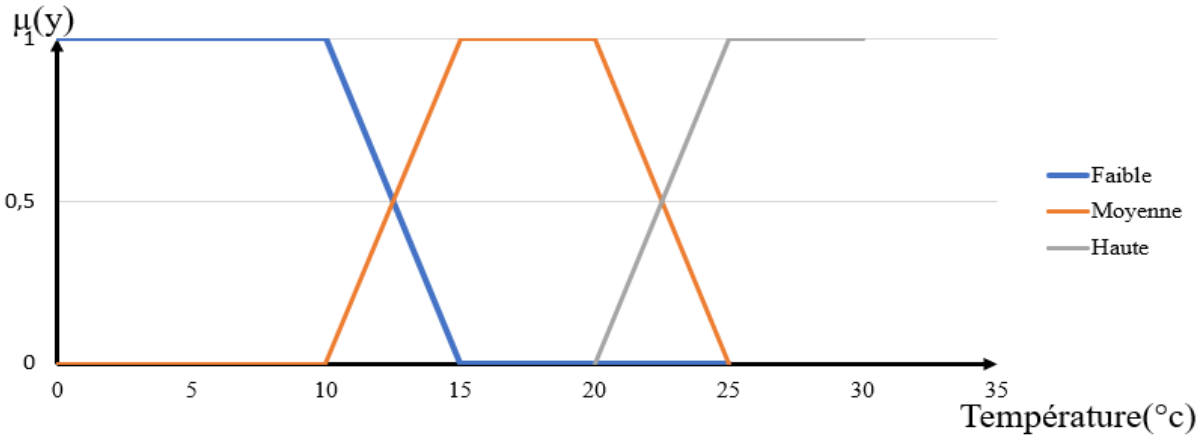


FIGURE 3.7 – Ensembles flous représentant la température du campus

L'utilisation de deux quantificateurs "la plupart" et "peu" nous permet de dessiner les propriétés suivantes :

- **Non-contradiction :** cela signifie que si un résumé a un grand degré de vérité, l'opposé de ce résumé doit avoir un faible degré de vérité, en prenant comme exemple le cas des altitudes, la première phrase "la plupart des altitudes sont élevées" a un degré égal à 1. Sa négation "peu d'altitudes sont élevées" a 0 comme degré de vérité. On peut également remarquer que les phrases ont un degré de vérité complémentaire.
- **Double négation :** un résumé qui possède la négation de deux paramètres d'un autre résumé doit nous donner le même degré de vérité que le premier par exemple ; pour les deux résumé "la plupart de l'altitude est élevée" et "peu d'altitude sont faibles" le degré de la vérité est 1.

3.4.3 Cas des valeurs typiques

Afin d'obtenir les valeurs typiques des données étudiées, nous proposons d'implémenter une version améliorée et adaptée aux spécificités des données de l'algorithme 1 décrit dans la page 52. Nous fournissons dans ce qui suit une partie des résultats après l'exécution de l'algorithme 1.

— Pour la base de données ADSB, la valeur typique de l'ensemble d'altitude (ALT) est donnée sous forme d'un intervalle [27000, 41025].

— Pour le flux de données NeOCampus accumuler dans une fenêtre temporelle de 30 minutes, le résultat obtenu confirme que la valeur typique de la température dans le campus intelligent est entre [21, 23].

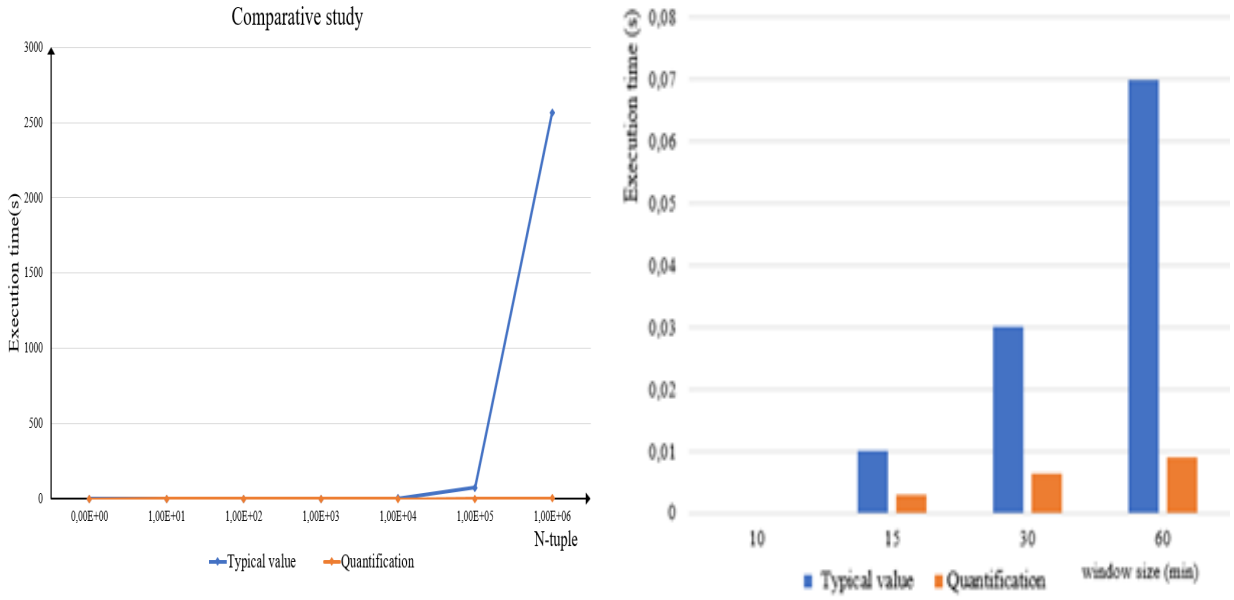
L'algorithme 1 (de Dubois et Prade) a de bonnes performances tout en fournissant des résultats intéressants pour réduire un gros volume de données, mais il présente des inconvénients non négligeables, à savoir : une consommation importante d'espace de mémoire ; un temps d'exécution important et il augmente proportionnellement avec la taille des données étudiées.

L'exécution de l'algorithme 1 nécessite plusieurs passes pour balayer l'ensemble de données ce qui résulte en un processus extrêmement long et consommateur dans le cas des données massives.

La première étape de l'algorithme consiste à ordonner les données pour avoir l'occurrence de chacune et pour déterminer la longueur de l'intervalle de départ. Toutes ces critiques rendent l'exploitation de cet algorithme pratiquement impossible dans le cas, par exemple, d'un flux de données.

3.4.4 Étude Comparative

Deux procédures de résumé de données ont été décrites, étudiées et appliquées sur plusieurs contextes de données. Les expériences ont été réalisées sur le même jeu de données. Dans cette partie, nous donnons une comparaison entre deux méthodes de résumé vues dans les sections précédentes et nous discutons les résultats en termes de temps d'exécution. Pour la base de données statique représentée par le projet ADSB, les expériences montrent que l'approche de résumé linguistique consomme moins de temps par rapport à l'algorithme Dubois et Prade. La figure 3.8a décrit les résultat du temps d'exécution en fonction de la taille de la base de données. Comme mentionné dans la section précédente, l'algorithme pour les valeurs typiques a le temps d'exécution le plus élevé en raison de l'utilisation d'une grande combinaison de calcul.
En ce qui concerne le flux de données provenant du campus intelligent NeOCampus, nous prenons comme cas d'étude l'attribut température. La comparaison entre les deux approches en terme de temps d'exécution est illustrée dans la figure 3.8b. On note que le temps d'exécution de l'algorithme de la valeur typique augmente avec l'augmentation de la taille des fenêtres temporelles utilisés, c'est-à-dire, la taille de l'ensemble d'étude. En revanche, l'algorithme de résumé linguistique possède une faible augmentation de l'ordre de (ms). À partir de ces résultats, nous pouvons constater que le résumé linguistique est mieux adapté à des fins de réduction dans le contexte des données massives et où une réponse approximative d'une requête suffit pour répondre aux besoins de l'utilisateur ou du décideur.



(a) (b)

FIGURE 3.8 – Comparaison des méthodes de résumé en fonction du temps d'exécution : (a) base de données ADSB ; (b) Campus intelligent NeOCampus.

3.5 Conclusion

Le résumé de données constitue un outil puissant d'extraction de connaissances à partir d'un grand ensemble de données. Il a été utilisé avec succès dans de nombreux domaines. Il permet de fournir des informations pertinentes et efficaces à des fins de prise de décision. Ce chapitre a examiné les méthodes de construction de résumés de données en utilisant les techniques de la logique floue afin de faire face au volume croissant de données créées et stockées. Dans un premier temps, nous avons commencé par étudier deux méthodes de résumé. Nous avons discuté l'approche de Yager basée sur le paradigme du quantificateur linguistique. Le résumé fondé sur la typicité a également été abordé dans ce chapitre. À la lumière de cette étude, les techniques proposées sont utilisées et mises en œuvre, une étude comparative entre ces deux méthodes pour obtenir un résumé efficace à partir d'un large ensemble de données d'entrée a également été décrite.

Nous avons constaté que le résumé linguistique possède un certain nombre d'avantages par rapport aux autres méthodes : il a la capacité de fournir des phrases en langage naturel compréhensible par un utilisateur non expert, il fournit de nombreux résumés à des fins spécifiques. En plus, il peut résumer des données numériques et non numériques. Il convient de noter que le résumé linguistique s'adapte mieux dans le cas des données massives à cause de sa consommation faible en termes d'énergie et de temps d'exécution.

Dans l'étude des données provenant du projet NeOCampus, nous avons considéré le protoforme temporel en prenant une fenêtré temporelle bien définie. Nous montrons dans le prochain chapitre comment on peut exploiter le caractère dynamique des données provenant de multiples capteurs en les considérant comme des séries temporelles.

Chapitre 4

Résumés linguistiques dans le contexte des séries temporelles

Il n'y a qu'une seule partie de l'univers que nous pouvons changer d'une façon certaine : soi-même.

— Aldous Huxley

Sommaire

4.1 Introduction **61**
4.2 Séries temporelles **61**
4.3 Résumé linguistique des séries temporelles **62**
4.3.1 Protoformes des résumés linguistiques 62
4.3.2 Segmentation linéaire par morceaux 63
4.3.3 Processus de résumé 67
4.4 Étude expérimentale **72**
4.5 Conclusion **76**

4.1 Introduction

Récemment, avec l'omniprésence croissante des données avec une dimension temporelle, en particulier les séries temporelles, diverses tentatives de recherche et de nouveaux développements ont émergés dans le domaine de la gestion et d'exploitation de données. Les séries temporelles ou chronologiques constituent une classe importante de données temporelles et peuvent être facilement obtenues à partir des applications scientifiques et financières (par exemple, électrocardiogramme (ECG), température quotidienne, totaux des ventes hebdomadaires...). Une série temporelle est un ensemble d'observations effectuées selon un ordre chronologique. La nature des données de ces séries est caractérisée par : une taille importante de données, une dimensionnalité élevée et une mise à jour continue. De plus, les données de série temporelle se caractérisent par leur nature numérique et continue, ce qui nécessite le développement des nouvelles techniques adaptées à ces propriétés. Le résumé linguistique pourrait être une solution adéquate permettant de fournir une synthèse de ces données tout en expliquant l'évolution de la série temporelle cible.

L'objectif du présent chapitre est d'examiner en détails l'exploitation des résumés linguistiques dans le contexte des séries temporelles. Afin d'atteindre cet objectif, nous introduisons d'abord plusieurs notions, dont la définition des séries temporelles, leurs utilisations pour résoudre certains problèmes complexes des applications réelle. Aussi, les techniques d'extraction des tendances des séries temporelle, en se basent sur la segmentation linéaire par morceaux, seront abordées. Nous proposons également une étude comparative permettant de choisir la méthode de segmentation la plus efficace. Par la suite, nous discutons l'utilisation des résumés linguistiques pour identifier les caractéristiques des tendances d'une série temporelle. Enfin, nous présentons une série d'expérimentation réalisée, en combinant la segmentation linéaire par morceaux et l'approche de résumé linguistique, menée sur les données provenant des capteurs du campus intelligent NeOCampus [iri17].

4.2 Séries temporelles

Les données des séries temporelles sont omniprésentes. Récemment, il y a eu un intérêt grandissant pour la question liée à la gestion et à l'exploitation des bases de données de séries chronologiques. Cela n'est guère surprenant étant donné que les séries chronologiques représentent une grande partie des données produites par des applications commerciales, médicales et scientifiques. Contrairement aux bases de données transactionnelles avec des éléments discrets, les données de séries chronologiques sont caractérisées par leur nature numérique et continue [CFLN02]. De plus, les bases de données de séries temporelles sont souvent extrêmement volumineuses et continuellement croissantes. Ainsi, toutes les caractéristiques intrinsèques des données des séries chronologiques rendent le traitement, l'analyse et l'exploitation des données des tâches difficilement réalisables. À partir de ces propriétés, [Moy16] a proposé de définir une série chronologique comme un ensemble de valeurs numériques associées à des étiquettes temporelles ou une séquence d'éléments de données mesurés généralement à des moments uniformément espacés, comme montre l'équation (4.1), où n est la taille de la série temporelle, t_i fait référence à une date réelle et x_i la valeur à l'instant t_i.

$$T = (t_i, x_i)_{i=1,\dots n} \tag{4.1}$$

Pour décrire la série temporelle, trois éléments doivent donc être déterminés : la valeur, l'étiquette temporelle et le fenêtrage temporel. Ces concepts sont discutés dans ce qui suit :

Valeurs : Selon les valeurs associées à la série temporelles, nous distinguons deux catégories principales : série univariée si à chaque instant, une valeur est associée. Si plusieurs valeurs sont

attribuées au même instant ; la série est multivariée. En plus de ces deux catégories, les séries temporelles pouvant être classées en série numérique ou en une série symbolique.

Étiquette temporelle : Pour une série temporelle l'ordre d'arrivée des données est crucial pour le traitement, il est indiqué par la date où l'instant d'arrivée t_i. Pour déterminer la nature des séries, un calcul d'écart δ entre deux instants successifs est établi. Si l'écart est constant pour toute la série on dit que la série est à temps globalement régulier et la fréquence d'échantillonnage est donnée par $1/\delta$. Dans le cas où δ est régulier pour certains instants la série est localement régulière. Dans certain cas la mesure ne s'effectue pas avec un écart bien défini, mais dans des instants totalement différents et ne s'expriment pas par une fréquence d'échantillonnage, la série est donc irrégulière.

Fenêtre temporel : Une fenêtre temporelle est une technique courante utilisée pour l'analyse des tendances dans les données de séries chronologiques. Une fenêtre temporelle est définie comme un ensemble d'observations dans un ordre chronologique qui décrivent une séquence d'observations continues sur une période spécifiée par cette fenêtre temporelle [KB05].

4.3 Résumé linguistique des séries temporelles

Le résumé de séries temporelles a été étudié dans [OF06], où les auteurs ont proposé des techniques de résumé en ligne basées sur la transformation de Fourier ou des ondelettes comme mentionnées dans le chapitre 2. Nous avons montré que ces techniques possèdent de nombreux désavantages non négligeables tels que le manque de représentativité et la non-utilisation de langage naturel. Pour surmonter ces contraintes, nous proposons d'appliquer le résumé linguistique pour décrire la variation des tendances liées aux séries temporelles.
Dans le chapitre précédent, nous avons considéré le cas général de résumés linguistiques de données. Dans ce chapitre, nous nous intéressons à la description d'une série temporelle par ce type de résumés. Dans un premier temps, nous présentons les différentes approches de résumé utilisées, la seconde partie est dédiée à l'étude de la segmentation de séries temporelles, à la suite de cette étude, nous détaillons l'approche proposée.

4.3.1 Protoformes des résumés linguistiques

Dans le cadre de résumés linguistiques des séries temporelles, plusieurs protoformes sont proposés selon le type de la série uni-variée ou multi-variée et selon le type de l'information cherchée et souhaitée. Dans ce qui suit nous représentons les principaux protoformes.

4.3.1.1 Séries univariées

Dans la plupart des cas, les séries temporelles sont considérées comme des séries univariées. Dans cette section, nous présentons différents exemples de génération de résumés linguistiques à partir de ces séries :

Étiquettes temporelles linguistiques. Dans l'article [COMS09], les auteurs proposent d'utiliser une variable linguistique pour décrire l'attribut en question, et une autre variable linguistique décrivant l'étiquettes temporelle, comme **"la plupart des jours où le temps est doux, le nombre des patients est moyen**. Cette approche est proposée dans le cadre d'analyse des données médicales décrivant la fréquentation d'un hôpital au cours du temps en fonction d'informations météorologiques.

Tendances temporelles linguistiques. L'idée de cette approche est de segmenter la série temporelle en un ensemble de segments (droites linéaires) ou, plus formellement, **tendances**. Une méthode de segmentation linéaire par morceaux [SG80] est utilisée. Par la suite, on associe à l'attribut tendance trois caractéristiques dynamiques : (i) vitesse du changement, (ii) durée du changement et (iii) variabilité de la tendance. Chaque caractéristique représente une variable linguistique associée à des modalités floues. Dans [KWZ08], les auteurs ont proposé d'utiliser les modalités : très décroissante, décroissante, constante, très croissante... pour modéliser l'attribut dynamique du changement. Le résume linguistique obtenu par la suite utilise les quantificateurs linguistiques et les protoformes ordinaires : "Q y sont S" comme **"la plupart des tendances sont décroissantes"**, et "Q R y sont S" comme **"la plupart des tendances à haute variabilité sont courtes"**.

Propositions temporelles floues. Le protoforme "durant T, A" a été proposé dans [CBMB99, CBMB00], il consiste à déterminer l'occurrence de la durée d'un phénomène en utilisant une variable linguistique décrivant le fenêtrage temporel, et une autre variable linguistique pour décrire l'attribut cible. Comme **"Durant les 30 dernières minutes, la température était élevée"**.

4.3.1.2 Série multivariée

Comme mentionnée ci-dessus, une série temporelle multivariée est une série où on attribue à chaque instant plusieurs valeurs, elle peut être considérée comme un ensemble des séries univariées. Deux protoformes sont utilisés pour les résumés linguistiques de séries multivariées.

Échelles hiérarchiques temporelles. Dans cette approche, une série multivariée est considérée comme plusieurs séries univariées, elle est basée sur la comparaison entre ces séries. Comme, **La plupart des jours, les deux séries varient dans la même direction**. Dans [COMS11], les auteurs proposent de hiérarchiser la variable linguistique décrivant le temps selon trois axes : annuel, chaque 5 ans et décennie, puis il utilise le protoforme classique "Q y sont S". Les résumés les plus fiables sont ceux qui couvrent l'intervalle le plus large.

Mesure d'exceptionnalité. Le protoforme a été proposé dans [vdHT09], dont l'objectif est de fournir un résumé décrivant la moyenne ou l'écart type d'un attribut particulier pour chaque intervalle de temps défini en utilisant le fenêtrage temporel, par exemple "la plupart des jours, la consommation le matin est inférieure à la consommation le soir".

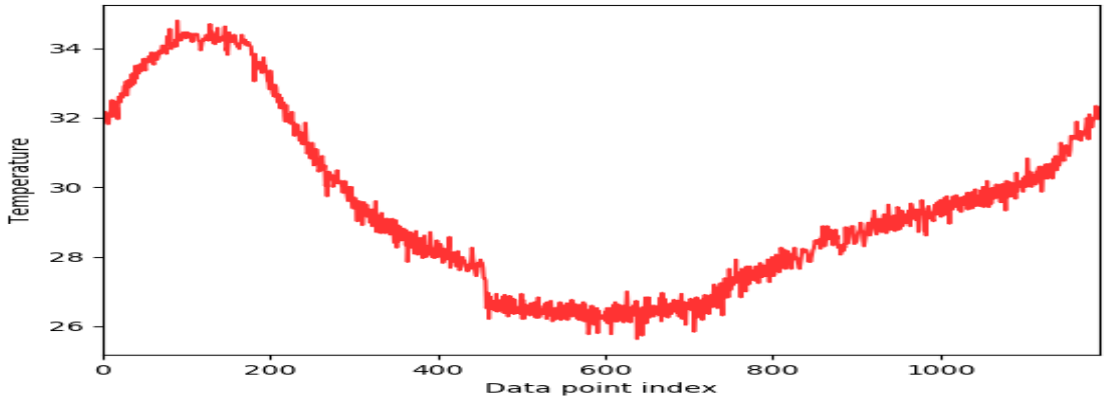
Dans notre étude, les résumés proposés se réfèrent aux résumés des tendances des série temporelles identifiées ici avec des segments de lignes droites et d'une approximation linéaire par morceaux. Nous montrons d'abord comment construire une telle approximation. Puis, nous définissons les caractéristiques dynamiques des tendances crées en utilisant ces approximations.

4.3.2 Segmentation linéaire par morceaux

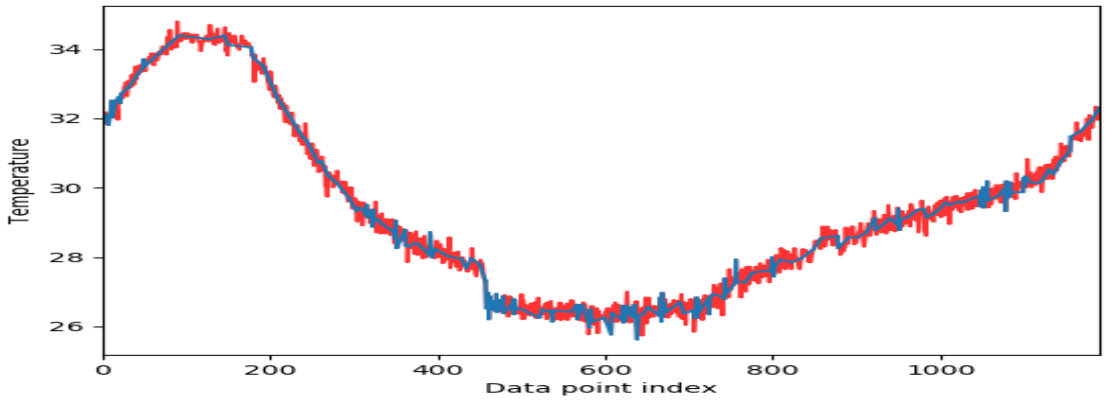
Comme pour la plupart des problèmes informatiques, la représentation des données est la clé des solutions efficaces [KCHP01]. Plusieurs représentations de haut niveau de séries temporelles ont été proposées afin d'exploiter d'une manière efficace les quantités massives des données de ces séries, y compris les transformées de Fourier, ondelettes (plus de détails dans le chapitre 2) et la segmentation linéaire par morceaux. Cette dernière est l'une des techniques les plus couramment utilisées par divers chercheurs pour réaliser un processus de regroupement, classification, indexation et extraction de règles d'association. La segmentation des séries temporelles a été discutée dans la littérature dans différents contextes, par de nombreux auteurs [CFNL04, FCNL01]. Le

problème de segmentation peut ainsi être qualifié : comme une étape de pré-traitement et une tâche principale pour diverses opérations de traitement et d'analyse.
L'approximation linéaire par morceaux est une approche permettant de fournir une représentation linéaire d'une série chronologique. Elle consiste à diviser la série en un ensemble de segments et en approximant chaque segment avec une ligne droite, voir par exemple la Figure 4.1. De nombreux algorithmes ont été proposés pour la représentation des séries temporelles sous leurs formes segmentées et la détermination d'un nombre adéquat de segments homogènes. Les trois algorithmes de segmentation les plus courants, basés sur la représentation linéaire par morceaux (PLR), sont les suivants : algorithme Top-Down, Bottom-Up et Sliding Window [KCHP04].

Dans ce qui suit nous présentons un aperçu général sur les trois algorithmes de segmentation et nous proposons une étude empirique permettant de les comparer afin de définir l'algorithme le mieux adapté à notre ensemble de données.



(a)



(b)

FIGURE 4.1 – Série temporelle originale (a) et son approximation linéaire par morceau (b)

4.3.2.1 Sliding windows

Le processus de segmentation à l'aide de l'algorithme de fenêtre glissante commence par déterminer la limite gauche (ancre) du premier segment potentiel (généralement le premier point de données d'une série chronologique), qui représente le point de départ de la fenêtre glissante, et permet ainsi d'identifier et de sélectionner des segments qui satisfont le critère de segmentation

prédéfini (seuil spécifié par l'utilisateur). En glissant dans la série, la taille de la fenêtre augmente progressivement, car tous les points de données visités sur son parcours deviennent automatiquement des éléments potentiels du segment, jusqu'à ce que l'erreur du segment potentiel ne dépasse pas le seuil spécifié par l'utilisateur.

À ce stade, la limite droite de la fenêtre mobile cesse d'être inconnue. De cette manière, la longueur de ce segment est déterminée et le point d'arrêt du segment nouvellement formé devient la nouvelle ancre, c'est-à-dire le point de départ du segment potentiel suivant. Ce processus de formation des segments, se répète jusqu'à ce que toute la série temporelle soit convertie en représentation linéaire par morceaux. Le pseudo-code de l'algorithme de fenêtre glissante est illustré par l'algorithme 2.

Algorithm 2 Sliding window algorithm

```
Input: T // time serie
 max_error
Result: Result_segment
anchor = T[0]
 end = anchor
 i = 0
while i < length(T) do
    i=i+1
     end = T[i]
     test_segment = create_segment(anchor, end)
     error = calculate_error(anchor, end)
     if error < max_error then
         result_segment = test_segment
           anchor=anchor+i
    else
         break
    end
end
```

4.3.2.2 Top down

Souvent appelé " scission binaire", l'algorithme commence par l'observation conditionnelle de séries chronologiques non segmentées comme un segment majeur. Pour choisir les deux segments initiaux, l'algorithme prend toutes les variantes possibles et les compare pour identifier le point de rupture (frontière du segment) qui divise la série temporelle en deux segments. S1 (segment gauche) et S2 (segment droite) sont choisis de telle manière que la différence entre ces deux segments soit maximale. Ces deux segments sont ensuite testés en terme du niveau de l'erreur d'approximation. Si l'erreur d'approximation du segment observé est inférieure au seuil défini par l'utilisateur, la procédure de segmentation s'arrête et le segment testé est accepté. D'un autre côté, si l'erreur d'approximation est supérieure au seuil défini par l'utilisateur, une nouvelle division du segment testé en deux nouveaux (sous) segments est effectuée. Pour chacun des segments nouvellement formés, le processus de division en deux nouveaux segments est répété de manière identique, sans effet sur l'emplacement du point de rupture déterminé lors de l'itération précédente (étape). L'algorithme répète ces étapes jusqu'à ce que le critère d'arrêt défini soit satisfait, c'est-à-dire lorsque la division ultérieure ne contribue plus à la minimisation de l'erreur de segmentation. Le pseudo code, donné dans l'algorithme 3, illustre le fonctionnement de l'algorithme de segmentation Top down.

Algorithm 3 Top down algorithm

```
best_left_error, best_left_segment = inf
best_right_error, best_right_segment = inf
for i = 0 to len(T) do
    breakpoint = i
    segment_left = create_segment(T, (0, breakpoint = i; ))
    error_left = compute_error(T, segment_left))
    segment_segment = create_segment(T, (i, len(T))
    error_right = compute_error(T, segment_right))
    if error_left, error_right = best_left_error, best_right_error then
        best_left_error, best_right_error = error_left, error_right
        best_left_segment, best_right_segment = segment_left, segment_right
    end
end
if best_left_error ≤ max_error then
    left_segment = [best_left_segment]
end
```

4.3.2.3 Bottom up

Souvent appelée "fusion itérative" en complément naturel à l'algorithme Top Down [KCHP01], l'algorithme commence par diviser la série temporelle d'origine, en un grand nombre de très petits segments de longueurs égales. Ensuite, il compare chaque paire de segments consécutifs (y compris les voisins gauche et droit). Les paires qui causent la plus petite augmentation de l'erreur sont identifiées, et par conséquent fusionnées dans un nouveau segment plus grand. L'algorithme répète ces étapes jusqu'à ce que tous les segments aient des erreurs en dessous du seuil défini comme un critère d'arrêt de processus de segmentation [LMS14]. L'algorithme 4 illustre ce type de segmentation.

Algorithm 4 Bottom Up algorithm

```
for i = 0 to len(T) do
    Segment = create_segment(T[i], T[i + 1])
end
for i = 0 to len(T) − 1 do
    merge_cost(i) = calculate_error(merge(segment(i), segment(i + 1)
end
while min(merge_cost < max_error) do
    i = index(min(merge_cost)
    segment = merge(segment(i), segment(i + 1)
    deltete(segment(i + 1))
    merge_cost(i) = calculate_error(merge(segment(i), segment(i + 1))
    merge_cost(i − 1) = calculate_error(merge(segment(i − 1), segment(i + 1))
end
```

4.3.2.4 Comparaison de techniques de segmentation

La sélection de l'algorithme de segmentation adapté aux données cibles, qui fournit la meilleure approximation linéaire et assure la couverture de toute la série temporelle, est une tâche délicate et nécessite la prise en compte un certain nombre d'éléments : le premier enjeu consiste à la détermination de la nature de la série temporelle en question, c'est-à-dire : stationnaire [1], non stationnaire, périodique [2], non périodique...etc. Puisque les algorithmes mentionnés ci-dessus se distinguent de point de vue des paramètres d'entrées. En plus, le niveau et la qualité d'approximation souhaitée jouent un rôle important dans le choix de l'algorithme. Le dernier critère pour la sélection concerne le temps d'exécution de l'algorithme de segmentation.
L'approximation linéaire d'une série temporelle peut être réalisée au moins de deux façons : l'interpolation et la régression linéaire. Ces deux méthodes sont utilisées pour prédire les valeurs d'une variable (Y) pour une valeur donnée d'une autre variable (X).

- Interpolation linéaire : Ce modèle cherche simplement à établir une relation linéaire entre une variable X représentant ici les instants t_i de la série, et les valeurs de la série temporelle.
- Régression linéaire : la régression est un processus d'ajustement d'un certain nombre de points à une courbe passant par ou près des points avec une erreur quadratique minimale.

Dans cette partie, nous fournissons une comparaison empirique des performances des trois algorithmes (Sliding windows , Bottom-Up et Top down) appliqués sur la même quantité de données. Les résultats de la recherche expérimentale et l'analyse comparative sont illustrés dans la figure 4.2. Cette recherche a été menée sur les données provenant du campus intelligent "NeOCampus" et qui se caractérisent par leur nature numérique, non stationnaire et non périodique. Il faut également prendre en compte que le seuil d'erreur pour chaque algorithme est défini par l'utilisateur, et nous le considérons comme le seul paramètre d'entrée pour les trois algorithmes. Les deux méthodes d'approximation (régression et interpolation) sont utilisées dans notre étude.

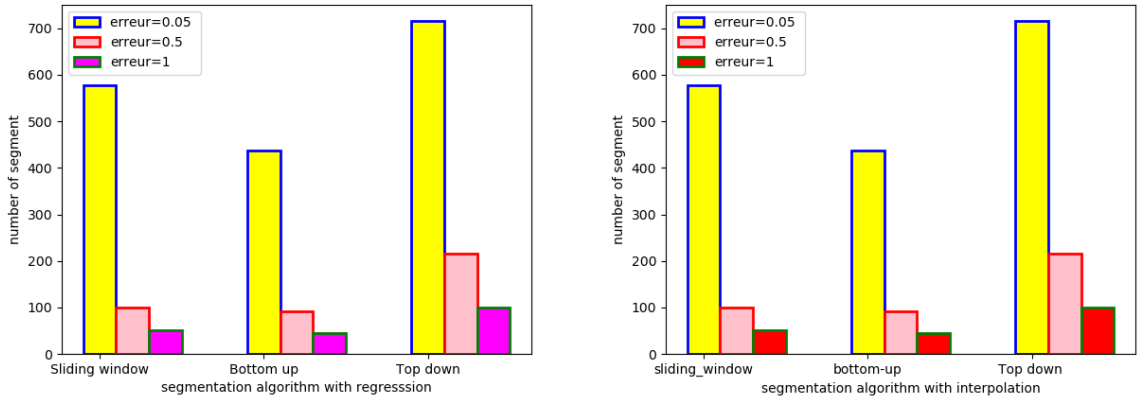
- L'algorithme Top down s'est avéré être l'algorithme avec les pires performances ; il génère un nombre de segments très important par rapport aux autres algorithmes.
- Dans plusieurs essais, l'algorithme Bottom–Up a montré des résultats légèrement meilleurs que l'algorithme Sliding windows.

À la base de cette étude, nous considérons l'algorithme Bottom-Up comme l'algorithme de segmentation le plus adapté pour la nature des données en question.

4.3.3 Processus de résumé

Les résumés proposés des séries temporelles se réfèrent ici aux résumés des caractéristiques des tendances (segments) liées à la série temporelle. Nous commençons par l'identification des tendances avec des segments de lignes droites et une approximation linéaire par morceaux en utilisant l'un des algorithmes proposés précédemment (dans notre cas nous avons choisi l'algorithme Bottom-up qui montre sa fiabilité pour le type de données en question). Puis, nous associons un ensemble de paramètres permettant de décrire les tendances. Chacun de ces paramètres (nommés caractéristiques dynamiques des tendances) est associé par la suite à un ensemble de valeurs linguistiques floues pour représenter la variation des tendances. En utilisant les protoformes proposés dans [KWZ06a] et illustrés par "Q y sont S" ou "Q R y sont S", où y représente les tendances R et S des valeurs floues modélisant les caractéristiques dynamiques des tendances et Q le quantificateur linguistique, nous obtenons les résumés linguistiques de la série temporelle. La figure 4.3 montre le processus utilisé pour extraire le résumé linguistique d'une série temporelle.

1. La série temporelle est stationnaire si les caractéristiques d'espérance et de variance sont constantes
2. La période mesure le temps séparant deux occurrences du même événement répété dans la série temporelle.



(a) (b)

FIGURE 4.2 – Comparaison de trois algorithmes de segmentation : (a) représente les trois algorithmes avec régression ; (b) représente les trois algorithmes avec interpolation.

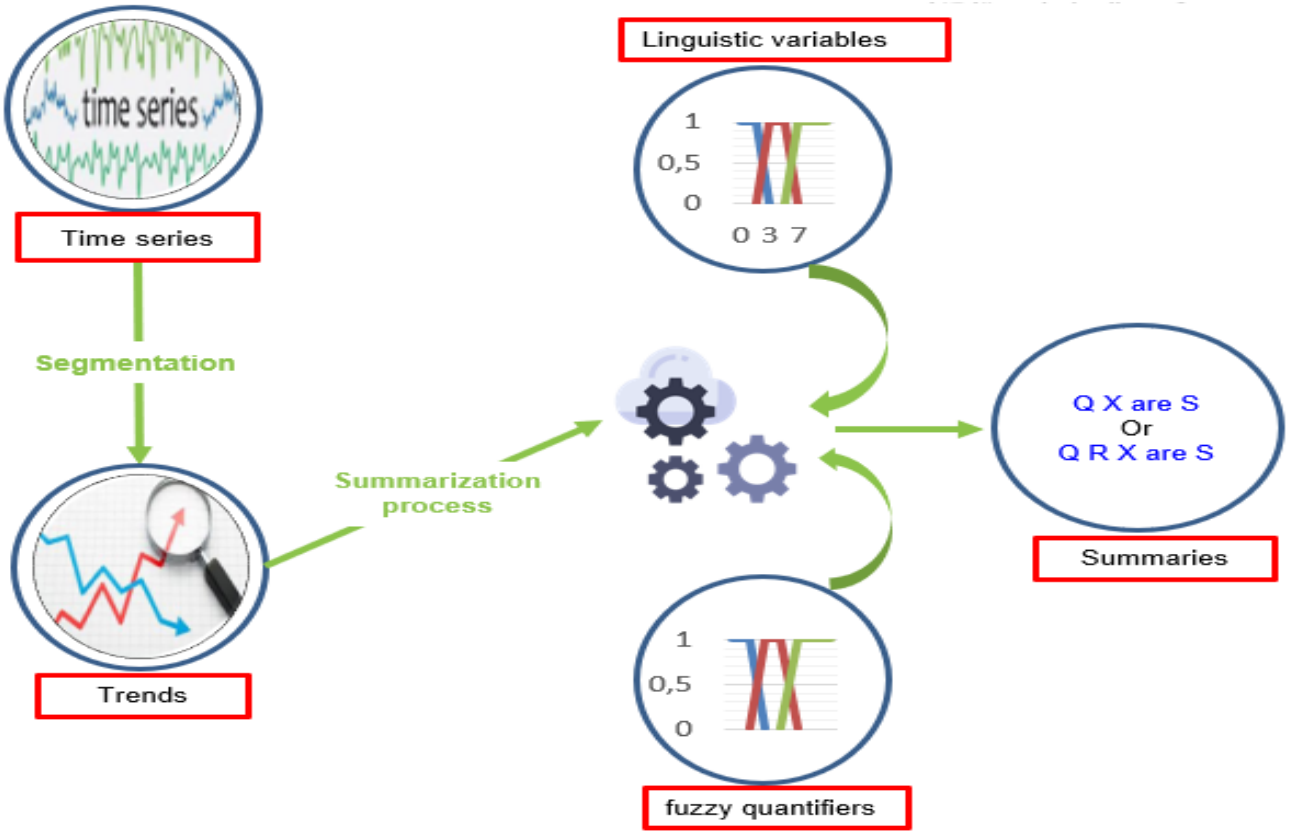


FIGURE 4.3 – Processus de résumé des séries temporelles

4.3.3.1 Caractéristiques dynamiques des tendances

Comme mentionné précédemment, pour générer les résumés linguistiques d'une sérié temporelle, il faut d'abord identifier les tendances de la série et étudier les caractéristiques liées à ces tendances. Dans [KWZ06a, KWZ10, KW09], les auteurs considèrent les trois aspects suivants :

- dynamique du changement
- durée du changement, et
- variabilité.

Selon [KWZ08] ces trois caractéristiques des tendances sont les plus simples et les plus intuitives, car elles concernent les aspects de ce qui se passe avec les données au fil du temps et qui peuvent être facilement compris par les experts du domaine. Il convient également de noter que ces caractéristiques sont d'une importance primordiale et de nombreux outils pour y faire face sont disponibles. Cependant, ces trois caractéristiques de base utilisées dans la suite ne sont clairement pas le seul choix et peuvent être complétées par d'autres aspects appropriés, selon les besoins de l'application considérée.

Dynamique du changement Sous le terme de dynamique du changement, nous comprenons la vitesse du changement. Elle peut être décrite par la pente d'une ligne droite représentant la tendance (segment). Nous attribuons une valeur unique d'angle caractérisant la dynamique de changement d'une tendance identifiée à l'aide de l'équation (4.2). Pour chaque tendance, nous avons le couple suivant $((t_{debut}, x_{debut}), (t_{fin}, x_{fin}))$ qui sera utilisé par la suite pour calculer l'angle de tendance (voir l'équation (4.2)).

$$angle = \arctan\left(\frac{x_{fin} - x_{debut}}{t_{fin} - t_{debut}}\right) \tag{4.2}$$

Pour quantifier la dynamique du changement, nous pouvons utiliser l'intervalle d'angles possibles $[-90°, 90°]$. Cependant, il pourrait être impossible, et non cohérent d'utiliser directement une telle échelle lors de la description des tendances. Par conséquent, nous pouvons utiliser une granulation floue afin de répondre aux besoins des utilisateurs et à la spécificité des tâches. L'utilisateur peut construire une échelle des termes linguistiques correspondant à diverses inclinaisons d'une ligne de tendance comme [fortement décroissante, décroissante, lentement décroissante, constante, lentement croissante, croissante, fortement croissante].
La figure 4.4 illustre les lignes possibles correspondant aux termes linguistiques particuliers.

En fait, chaque terme représente une valeur floue décrivant la direction de la tendance. Dans [Bat02], l'auteur présente de nombreuses méthodes de construction d'une telle granulation floue. L'utilisateur peut définir les fonctions d'appartenance de termes linguistiques particuliers en fonction du contexte de l'application.

Durée de changement La durée décrit la longueur d'une tendance unique $t = t_{fin} - t_{debut}$, conçue comme une variable linguistique dont la valeur linguistique peut être illustrée par une **tendance courte** définie comme un ensemble flou dont la fonction d'appartenance est donnée par l'équation 4.3. L'axe du temps est divisé en unités appropriées (segments de temps). Les définitions des termes linguistiques décrivant la durée dépendent clairement de la perspective ou du but assumé par l'utilisateur.

$$\mu_R(t) = \begin{cases} 1 & t \leq 1 \\ \frac{-1}{2}t + \frac{3}{2} & 1 \leq t \leq 3 \\ 0 & t \geq 3 \end{cases} \tag{4.3}$$

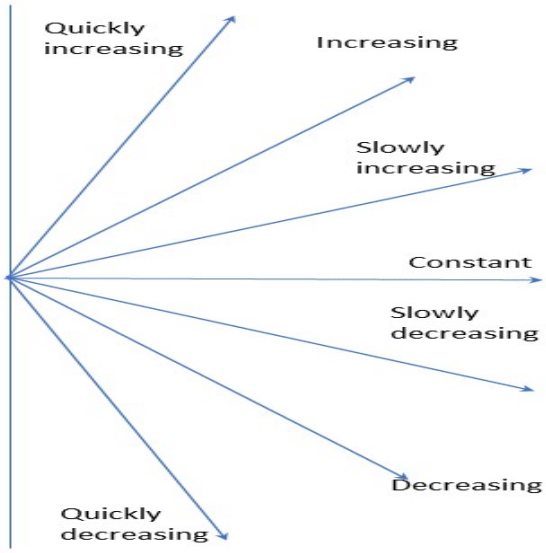


FIGURE 4.4 – Représentation des angles des tendances

Variabilité La mesure de la variabilité est une statistique récapitulative qui représente la quantité de dispersion dans un ensemble de données. Les mesures de la variabilité définissent à quelle distance les points de données ont tendance à tomber au centre. Nous parlons de variabilité dans le contexte d'une distribution de valeurs. Une faible dispersion indique que les points de données ont tendance à être regroupés étroitement autour du centre. Une dispersion élevée signifie qu'ils ont tendance à tomber plus loin.

Lorsqu'une distribution présente une variabilité plus faible, les valeurs d'un ensemble de données sont plus cohérentes. Cependant, lorsque la variabilité est plus élevée, les points de données sont plus dissemblables et les valeurs extrêmes deviennent plus probables. De nombreuses mesures statiques de la variabilité sont généralement utilisées :

— Le Rang : la mesure de la variabilité la plus simple à calculer et la plus simple à comprendre. Le rang d'un ensemble de données est la différence entre la valeur la plus grande et la plus petite de cet ensemble de données, ce qui la rend très sensible aux valeurs aberrantes.

— L'écart interquartile (IQR) : fait référence à la moitié médiane des données qui se situe entre les quartiles supérieur et inférieur. En d'autres termes, l'intervalle interquartile comprend les 50 % de points de données qui se situent entre Q_3 le troisième quartile et le premier quartile Q_1.

$$Q = Q_3 - Q1 \tag{4.4}$$

— La variance : la différence quadratique moyenne des valeurs par rapport à la moyenne. Contrairement aux mesures précédentes de la variabilité, la variance inclut toutes les valeurs du calcul en comparant chaque valeur à la moyenne.

$$V = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{4.5}$$

— L'écart type : la différence standard ou typique entre chaque point de données et la moyenne.

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{4.6}$$

Dans notre étude, nous avons retenu comme mesure de variabilité d'une tendance l'écart interquartile des données de la série temporelle recouverte par cette tendance [BHL20].

Exemple illustratif

Dans cet exemple, nous proposons une série temporelle modélisée avec 9 tendances, voir le Tableau 4.1. Pour extraire les résumés à partir de ses caractéristiques dynamiques, nous proposons d'utiliser les deux protoformes classiques "Q y sont S" et "Q R y sont S". Nous définissons le terme décroissant de vitesse du changement comme indiqué dans l'équation (4.7).

$$\mu_S(\alpha) = \begin{cases} 0 & \alpha \leq -65 \\ 0.066\alpha + 4.333 & -65 \leq \alpha \leq -50 \\ 1 & -50 \leq \alpha \leq -40 \\ -0.05\alpha - 1 & -40 \leq \alpha \leq -20 \\ 0 & \alpha \geq -20 \end{cases} \tag{4.7}$$

Pour modéliser la durée du changement, nous utilisons la valeur linguistique "courte" qui pourra être définie par l'équation (4.3) :

$$\mu_R(t) = \begin{cases} 1 & t \leq 1 \\ \frac{-1}{2}t + \frac{3}{2} & 1 \leq t \leq 3 \\ 0 & t \geq 3 \end{cases} \tag{4.8}$$

Nous définissons le quantificateur linguistique **la plupart** comme mentionné dans l'équation (4.9) :

$$\mu_Q(x) = \begin{cases} 1 & x \geq 0.8 \\ 2x - 0.6 & 0.3 \leq x \leq 0.8 \\ 0 & x < 0.3 \end{cases} \tag{4.9}$$

Nous considérons les résumés des tendances suivant :

La plupart des tendances sont décroissantes

La plupart des courtes tendances sont décroissantes

Le degré de vérité est calculé à l'aide des équations $T = \mu_Q[\frac{\sum_{i=1}^{n} \mu_S(y_i)}{n}]$ et $T = \mu_Q[\frac{\sum_{i=1}^{n} min(\mu_S(y_i).\mu_R(y_i))}{\sum_{i=1}^{n} \mu_R(y_i)}]$ pour le premier et le deuxième résumé respectivement.

*T(La plupart des tendances sont décroissantes)=**0.38***

*T(La plupart des courtes tendances sont décroissantes)=**0.85***

id	Dynamique de changement(angle en degré)	Durée(unité de temps)	Variabilité([0,1])
1	25	15	0.2
2	-45	1	0.3
3	75	2	0.8
4	-40	1	0.1
5	-55	1	0.7
6	50	2	0.3
7	-52	1	0.5
8	-37	2	0.9
9	15	5	0.0

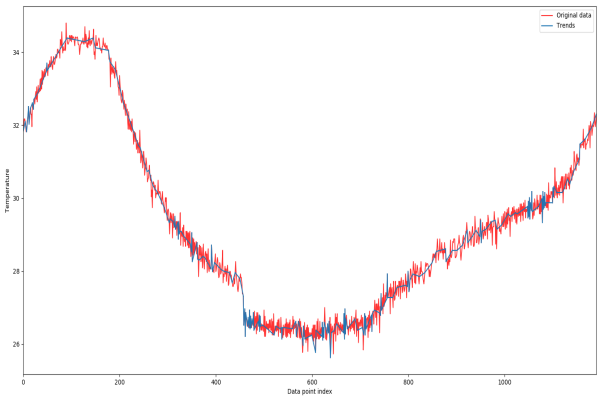
Tableau 4.1 – Exemple de caractéristiques dynamiques des tendances

4.4 Étude expérimentale

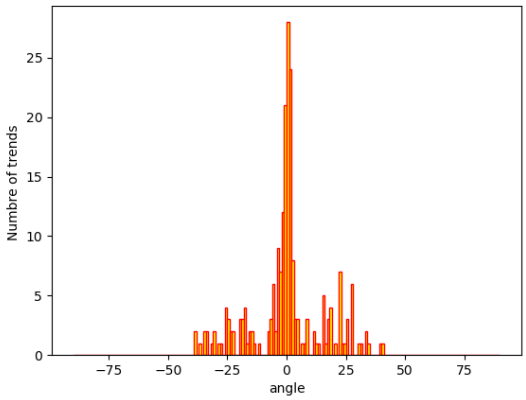
Le but de ce chapitre est de fournir un ensemble de résumés décrivant une série temporelle. Pour réaliser notre objectif, nous proposons de résumer les caractéristiques dynamiques des tendances de la série. Nous commençons par l'identification des ces tendances en utilisant une version modifiée et améliorée de l'algorithme Bottom-Up présenté dans la section précédente. Puis nous exploitons les caractéristiques dynamiques discutées avant pour dériver les résumés. Les données, sur lesquelles les tests sont réalisés, sont collectées à partir de plusieurs flux de données générés par de multiples capteurs (température, humidité, luminosité, CO2, énergie, ...) installés sur le campus de l'Université de Toulouse III, dans le cadre d'un projet NeOCampus soutenu par l'Université.
En utilisant l'algorithme Bottom up, nous avons obtenu 216 tendances (voir Figure 4.5a), les tendances les plus courtes n'ont pris qu'une unité de temps et les tendances les plus longues ont pris 16 unités. La figure 4.5b présente les pentes (angles) des tendances, tandis que l'histogramme 4.5c présente la durée des tendances. Pour calculer la variabilité des tendances, nous proposons d'utiliser l'écart interquartile. Pour chaque tendance, nous calculons l'interquartile des données incluses dans cette tendance. L'histogramme de la variabilité est présenté dans la figure 4.5d.
Différentes valeurs linguistiques permettant de modéliser les caractéristiques dynamiques des tendances sont utilisées pour extraire les résumés. Pour clarifier les résultats obtenus, nous décrivons ces valeurs linguistiques :

- Sept valeurs linguistiques sont utilisées pour décrire la dynamique du changement : rapidement décroissante, lentement décroissant, décroissante, constante, lentement croissante, croissante et fortement croissante.
- Trois étiquettes sont utilisées pour modéliser la variabilité : faible, modérée, haute.
- Pour la durée, nous utilisons trois valeurs linguistiques : courte, moyenne et longue.

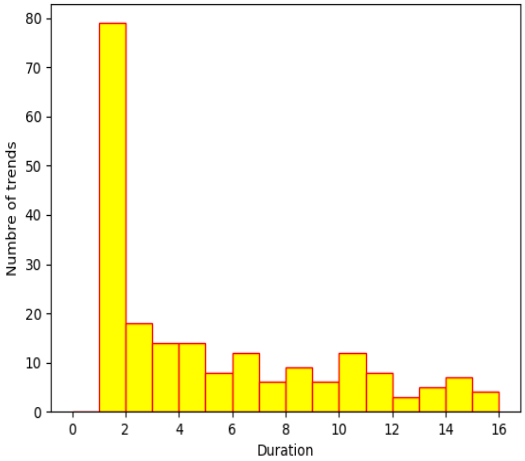
La figure 4.6 illustre les variables et les valeurs linguistiques utilisées pour décrire les caractéristiques des tendances.
Pour les quantificateurs linguistiques, nous avons proposée d'utiliser trois quantificateurs relatifs : la plupart, presque la moitie et peu. Ces quantificateurs sont illustrées dans le chapitre 3. Afin d'approuver la qualité des résumés obtenus pour le premier protoforme "Q y sont S", nous utilisons l'équation $T = \mu_Q[\frac{\sum_{i=1}^{n} \mu_S(y_i)}{n}]$. Quelques résumés intéressants sont présentés dans le tableau 4.2.



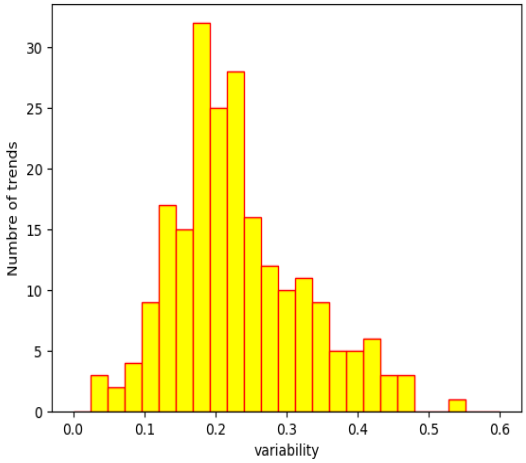
(a)



(b)



(c)



(d)

FIGURE 4.5 – Caractéristiques dynamiques des tendances (a) Création des tendances (b) Histogramme de la dynamique du changement (c) Histogramme de la durée de changement (d) Histogramme de la variabilité

Le tableau 4.3 montre les résumés linguistiques obtenus en utilisant le deuxième protoforme "Q R y sont S". Le degré de vérité est calculé selon l'équation :

$$T = \mu_Q[\frac{\sum_{i=1}^{n} min(\mu_S(y_i).\mu_R(y_i))}{\sum_{i=1}^{n} \mu_R(y_i)}] \tag{4.10}$$

Les autres mesures de la qualité discutées dans le chapitre 3 sont également utilisées pour évaluer les résumés obtenus.
Comme on peut le voir, les résumés linguistiques obtenus et leurs mesures de qualité associées sont intéressants et utiles pour le décideur, ils donnent une vue globale sur la variation des séries temporelles étudiées sans aller dans les détails qui pourraient dépasser le cognitif, la capacité de perception et de compréhension d'un utilisateur non expert.

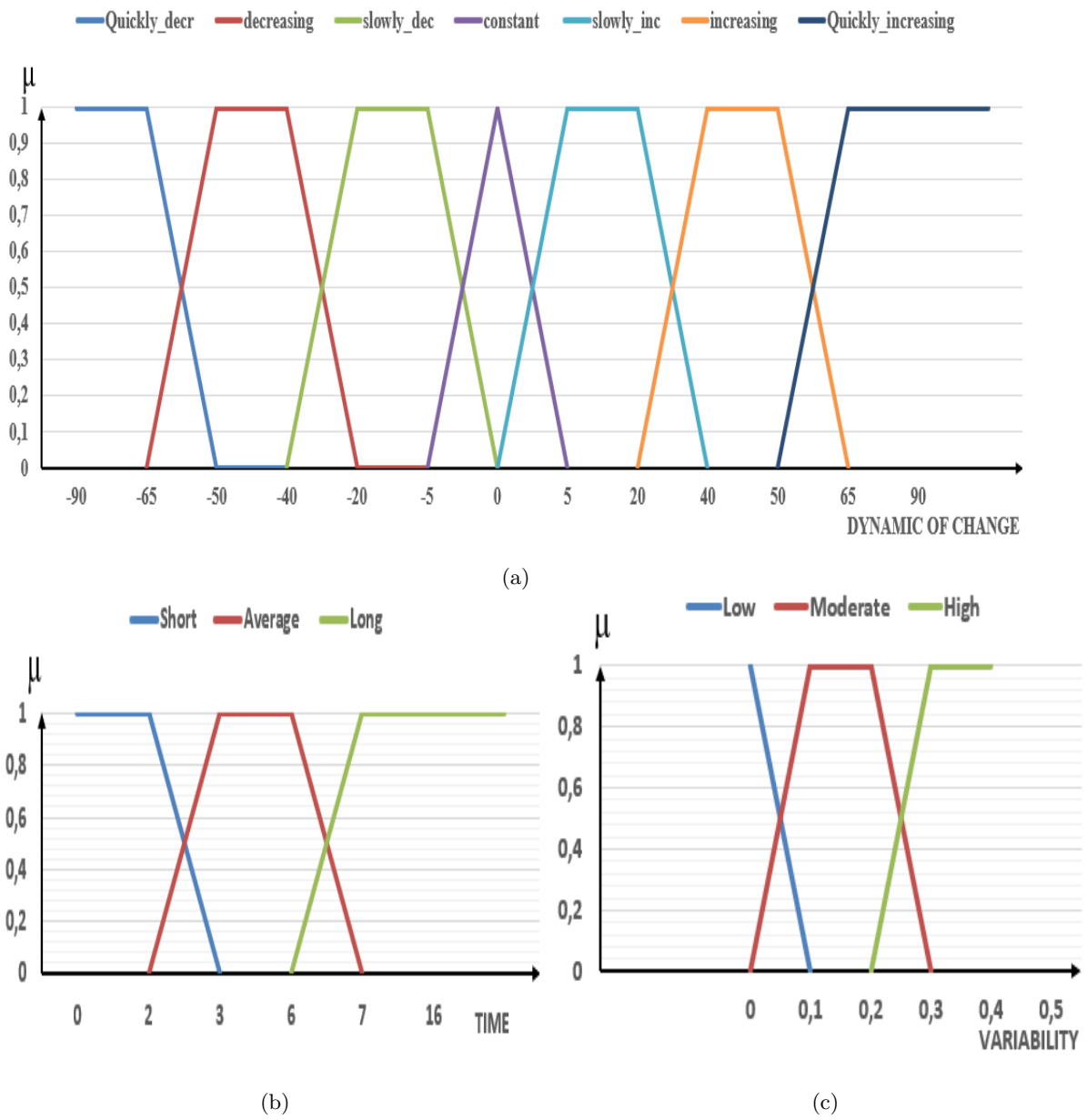


FIGURE 4.6 – Ensembles flous caractérisant les caractère dynamique des tendances (a) la dynamique du changement (b) la durée (c) la variabilité

Résumé	**Degré de vérité**
La plupart des tendances sont constantes	0.398
La plupart des tendances sont croissantes	0
Presque la moitié des tendances sont constantes	0.159
La plupart des tendances ont une faible durée	0.745
Presque la moitié des tendances ont une long durée	0.02
La plupart des tendances ont une haute variabilité	0.205
Presque la moité des tendances ont une variabilité modérée	0.64

TABLEAU 4.2 – Résumés linguistiques de la série temporelle avec le protoforme classique Q y sont S

Résumé	**Degré de vérité**	**Imprécision**	**Couverture**	**pertinence**
La plupart des tendances à faible variabilité sont croissantes	0.851	0.299	0.636	0.145
La plupart des tendances de variabilité modérée sont constantes	0.218	0.264	0.520	0.562
La plupart des tendances à haute variabilité sont constantes	0.245	0.264	0.539	0.91
La plupart des tendances courtes sont décroissantes	0.157	0.312	0.484	0.012
La plupart des tendances longues sont constantes	0.923	0.264	1	0.625
Presque la moitié des tendances courtes sont croissantes	0.401	0.299	0.49	0.04
La plupart des tendances à moyenne durée ont une variabilité modérée	0.751	0.115	0.824	0.042
Presque la moitié des longues tendances ont une faible variabilité	0.920	0.774	0.0446	0.1

Tableau 4.3 – Mesures de qualité pour le protoforme Q R y sont S

4.5 Conclusion

La gestion des séries temporelles à grande échelle a attiré l'attention de nombreux chercheurs de la communauté des bases de données. Les solutions actuelles se concentrent principalement sur la version statique du problème où les données de séries temporelles sont déjà stockées dans la base de données et sont disponibles pour un traitement ultérieur. Cependant, dans de nombreuses applications réelles impliquant des données de séries chronologiques, telles que les flux de données, la surveillance du réseau, les données sont modélisées sous forme de séries temporelles dynamiques, qui sont généralement de nature continue et illimitée.
Dans ce travail, nous avons proposé une approche hybride pour résumer linguistiquement les séries temporelles. La première étape de l'approche proposée consiste à segmenter la série chronologique en un ensemble de tendances à l'aide de l'algorithme Bottom-Up. Afin de caractériser ces tendances, nous les avons associées à un ensemble de caractéristiques telles que la dynamique du changement, la durée et la variabilité. ces propriétés sont exprimées comme des variable linguistiques où chaque variable est modélisée par un ensemble de prédicats flous. Ensuite, nous avons utilisé le calcul de Zadeh des propositions linguistiquement quantifiées pour évaluer le résumé linguistique lié aux tendances. Nous avons utilisé les différentes mesures de qualité pour prouver la qualité des résumés obtenus. L'évaluation expérimentale menée sur des données réelles, montre des résultats intéressants et qui semblent très prometteurs.
Cependant, Cette technique produit un nombre important de résumés avec des degrés de vérité qui peuvent être parfois faibles et non significatifs. Pour surmonter cet obstacle, nous envisageons d'utiliser une classe d'algorithmes génétiques, pour la sélection des meilleurs résumés, parmi un ensemble important de résumés, dans le cas d'un simple critère de sélection présenté par le degré de vérité du protoforme "Q y sont S". Dans le cas de multi critères décrits par le protoforme "Q R y sont S" où plusieurs mesures de qualité sont définies pour évaluer le résumé, nous proposons d'utiliser l'algorithme génétique multi-objectif NSGA II. Cette étude fera l'objet du prochain chapitre.

Chapitre 5

Algorithmes génétiques multiobjectif au service des résumés linguistiques

"Logic is a wonderful thing but doesn't always beat actual thought."

— Terry Pratchett, *The last continent*

Sommaire

5.1 Introduction **79**
5.2 Principe des algorithmes génétiques **79**
5.2.1 Codage d'individu 80
5.2.2 Fonction objectif 81
5.2.3 Taille de population 81
5.2.4 Opérateurs génétiques 81
5.2.5 Paramètres de dimensionnement 83
5.2.6 Discussion 84
5.3 Optimisation multi-objectif **84**
5.3.1 Problème multi-objectif 84
5.3.2 Panorama des systèmes d'optimisation multi critères 85
5.3.3 Méta-heuristiques pour l'optimisation multi-objectif 85
5.3.4 Notion de dominance 86
5.3.5 Frontière de Pareto 86
5.4 Algorithme génétique multi objectif NSGA II **87**
5.5 Étude expérimentale **89**
5.5.1 Modèle de l'algorithme génétique 89
5.5.2 Sélection multi-critère basée sur NSGA II 91
5.6 Conclusion **94**

5.1 Introduction

La technique de résumé proposée dans le chapitre précèdent permet de générer des résumés linguistiques compréhensibles et cohérents. Cependant, cette méthode est très fastidieuse lorsque le nombre de résumeur devient très important. À cet effet, le processus d'élaboration des résumés peut être considéré comme un problème d'optimisation ; dans lequel les meilleurs résumés d'un large éventail de candidats sont sélectionnés. Dans la littérature, plusieurs philosophies de méta heuristiques sont proposées pour améliorer la solution d'un problème d'optimisation, tandis que dans le cadre de résumés linguistiques, les chercheurs se sont concentrés sur l'utilisation d'une classe d'algorithmes génétiques. Un des premiers travaux dans ce contexte, nous citons [KWZ06b] où les auteurs ont discute l'extraction des résumés linguistiques de séries temporelles ; chaque résumé est considéré comme un chromosome et le degré de vérité est supposé être la fonction objectif utilisée pour évaluer le chromosome. Dans [COMST11a, COMST11b, AYA+17], les auteurs ont cherché à dériver des résumés linguistiques sur des données représentant des informations de patients au cours d'une année donnée, ils ont défini chaque résumé comme un gène. Ce concept est également utilisé dans [DDBK+13], Aussi, dans [DDMBPM14, DDBK15], les auteurs introduisent deux opérateurs spécifiques (nettoyage et amélioration des propositions) afin de garantir un résumé de haut niveau de qualité et d'assurer une diversité dans la prochaine génération (itération).

L'objectif de ce chapitre est de proposer une nouvelle approche facilement réalisable, pour produire "un ensemble des meilleurs résumés" des tendances associées à des séries temporelles. Ces dernières sont caractérisées par trois aspects importants : la dynamique du changement, la durée et la variabilité où chaque caractéristique est associée à un ensemble de prédicats flous. Pour atteindre cet objectif nous utilisons l'algorithme génétique GA introduit dans [H+92], comme première contribution où le degré de vérité basé sur le calcul d'une proposition linguistique quantifiée (résumé linguistique) [Zad83] est considéré comme la fonction objectif permettant d'évaluer la qualité des résumés produits.
Afin d'optimiser les différentes mesures de qualité d'une série temporelle telles que le degré de couverture, le degré d'imprécision, le degré de pertinence (discutées dans le chapitre 3), nous proposons d'utiliser l'algorithme Non dominated Sorting Genetic Algorithm II (NSGA II) développe par [DPAM02] qui est un algorithme d'optimisation multi-objectif permettant de résoudre des problèmes multi-critères généralement contradictoires. Nous validons notre approche en menant un ensemble d'expérimentations sur des données collectées du campus intelligent de l'université de Toulouse III [iri17].

5.2 Principe des algorithmes génétiques

Un algorithme génétique (GA) est un concept introduit pour la première fois par Holland [H+92] et décrit dans [Dav91]. C'est un algorithme évolutif qui utilise le principe de la sélection naturelle pour trouver la meilleure solution à un problème donné ; il s'inspire du mécanisme de la biologie et de la génétique. L'idée de base d'un algorithme génétique est de générer une population aléatoire d'individus ou de chromosomes ; chaque individu possède un ensemble de propriétés dites gènes et représente une solution au problème considéré. Pour évaluer la solution candidate, une fonction objectif où une fonction de fitness est associée à chaque individu. Elle détermine quel chromosome sera participé à la création de la prochaine génération.

Un GA commence avec une population de chromosomes générés aléatoirement, chaque chromosome est considéré comme une solution proposée au problème traité. Le GA progresse vers les meilleurs chromosomes en appliquant des opérateurs génétiques. La population évolue sous

forme de sélection naturelle. Au cours d'itérations successives, appelées générations, une nouvelle population de chromosomes se forme à l'aide d'un mécanisme de sélection et d'opérateurs génétiques spécifiques tels que les croisements et les mutations. Voir Figure 5.1. Une fonction d'évaluation doit être conçue pour chaque problème à résoudre. Étant donné un chromosome particulier (une solution possible), la fonction d'évaluation renvoie une seule valeur numérique, qui est supposée être proportionnelle à l'utilité ou à l'adaptation de la solution représentée par ce chromosome.
La nouvelle génération de population est créée à l'aide d'opérateurs évolutifs simples : sélection, croisement et mutation. L'opérateur de sélection, ou de reproduction, est un processus basé sur la fonction de remise en forme qui indique les individus qui participeront à la création de la nouvelle génération. L'opérateur de croisement est un processus qui combine les gènes de deux individus, appelés parents, afin de produire un chromosome portant des gènes des deux parents. L'opérateur de mutation permet de changer aléatoirement les gènes dans chaque chromosome.

Un ensemble de paramètres est utilisé pour concevoir le modèle d'algorithme génétique : la taille de la population, la probabilité de mutation et la probabilité de croisement. En plus de ces paramètres, il faut choisir avec soin et en fonction du problème à résoudre le modèle de représentation du chromosome (le codage), la méthode de sélection (sélection de la roulette, sélection élitiste, sélection de rang ...), et celle du croisement (croisement à un point, croisement à deux points, croisement uniforme). Toutes ces notions sont discutées dans ce qui suit.

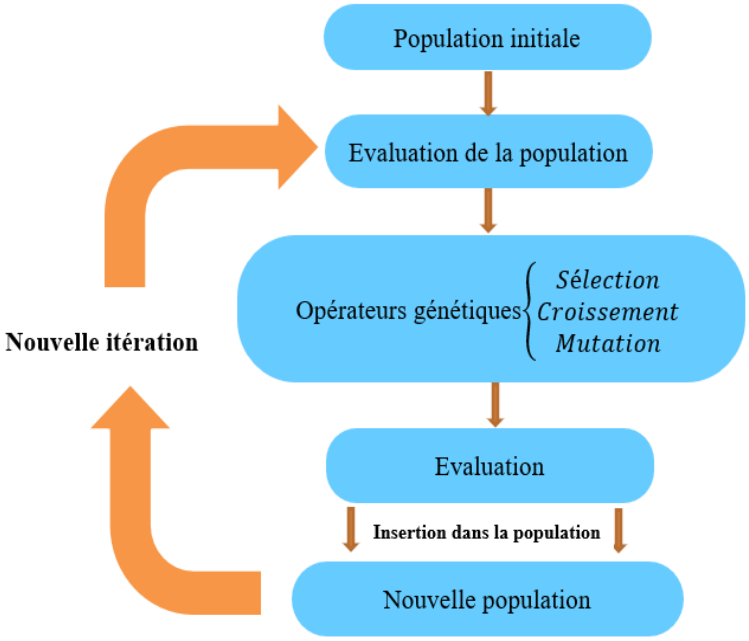


FIGURE 5.1 – Principe de l'algorithme génétique

5.2.1 Codage d'individu

Avant d'expliquer les opérateurs génétiques, il est nécessaire de définir la notion de codage qui permet de présenter les individus de la population. La procédure de codage est une étape cruciale dans l'algorithme génétique, il consiste à coder chaque chromosome comme un ensemble de bits où chaque séquence représente un gène (paramètre du chromosome).
Exemple : soit un chromosome constitué de trois gènes, chaque gène est codé sur 4 bits :

$$x_1 = 1011$$

$$x_2 = 0101$$
$$x_3 = 0010$$

Le chromosome est représenté par :

$$x = \{\underbrace{1011}_{x_1} | \underbrace{0101}_{x_2} | \underbrace{0010}_{x_3}\}$$

Le codage binaire est la première méthode utilisée pour représenter les individus, car il possède un certain nombre d'avantages tels que, le passage simple d'une représentation à une autre, il facilite les opérations génétiques comme le croisement et la mutation. Cependant, il existe d'autres techniques de codage comme le codage réel qui transforme les bits en nombres réels. Le choix de type de codage dépend donc du contexte de l'application.

5.2.2 Fonction objectif

La fonction d'évaluation, la fonction à optimiser, la fonction d'adaptation ou fitness est une formule mathématique permettant d'associer une valeur à chaque individu, dans le but de comparer les individus entre eux et sélectionner le meilleur. Le choix de la fonction objectif dépend du problème traité (on cherche à minimiser le coût ou maximiser les performances) et du nombre des paramètres.

5.2.3 Taille de population

La taille de la population est le nombre de chromosomes présents dans une population. Des tailles de population plus importantes augmentent la quantité de variation présente dans la population (ou la diversité de la population). De plus, lorsque la taille de la population est trop importante, l'utilisateur a tendance à réduire le nombre de générations afin de réduire l'effort de calcul, puisque l'effort de calcul dépend à la fois de la taille de la population et du nombre de générations. La réduction du nombre de générations réduit la qualité globale de la solution. D'un autre côté, une petite taille de population peut amener l'GA à converger prématurément vers une solution optimale [GH88].

5.2.4 Opérateurs génétiques

Pour passer d'une génération à une autre, l'algorithme génétique utilise un ensemble d'opérations inspirés de la génétique et de la biologie permettant de produire de nouveaux individus et assurer la diversité dans la nouvelle génération. En plus, ces opérations, décrites dans l'algorithme 5, ont pour but de converger vers la solution optimale d'un problème d'optimisation. Les opérateurs génétiques (sélection, croisement et mutation) sont présentées dans ce qui suit :

Algorithm 5 Principe de l'algorithme génétique

Input: $p = initial_population$
$evaluate(p)$
while $conditions\ non\ satisfied$ **do**
  $select(p)$
  $crossover(p)$
  $mutation(p)$
  $evaluate(p)$
end

5.2.4.1 Sélection

La sélection est une opération permettant d'identifier les chromosomes participant à la création de la nouvelle génération en se basant sur la fonction objectif. La sélection est l'étape la plus importante dans les algorithmes génétiques, car elle affecte considérablement la convergence et elle permet aux individus de survivre et de se reproduire. De nombreuses techniques peuvent être utilisées pour la sélection des meilleurs individus :

1. **Sélection par roue de loterie :** la méthode de sélection la plus populaire. L'idée est que chaque individu a une probabilité d'être sélectionné, cette probabilité est proportionnelle à la valeur de la fonction objectif de l'individu. elle peut être brièvement décrite comme suit : considérons N individus, chacun caractérisé par son aptitude $f_i > 0$ (i = 1, 2,..., N). La probabilité de sélection du i-ème individu est donc donnée comme :

$$p_i = \frac{f_i}{\sum_{i=1}^{N}} \tag{5.1}$$

   Imaginons une roulette avec des secteurs de taille proportionnelle à f_i (i = 1, 2,..., N). La sélection d'un individu équivaut alors à choisir au hasard un point de la roue et à localiser le secteur correspondant (Figure 5.2).

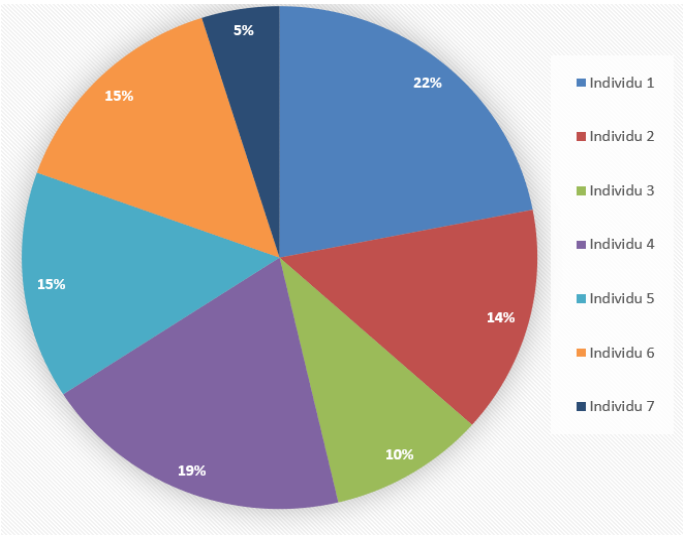


FIGURE 5.2 – Sélection par roue de loterie

2. **Sélection par tournoi :** la sélection par tournoi est également une stratégie de sélection qui sélectionne les individus en fonction de leurs valeurs de la fonction objectif. L'idée de base de cette stratégie est de sélectionner l'individu ayant la valeur de fonction objectif la plus élevée à partir d'un certain nombre d'individus. Dans la sélection des tournois, il n'y a pas de calcul arithmétique basé sur la valeur de fitness, mais seulement une comparaison entre les individus par valeur de fitness. Le nombre d'individus participant au tournoi est appelé taille du tournoi. La sélection s'effectue en suivant deux étapes : la première consiste à sélectionner au hasard K individus de la population pour participer au tournoi. Puis parmi ces individus, nous choisissons les individus qui ont les valeurs de fitness les plus élevées. Ensuite, les individus élus sont copiés dans la prochaine génération.
3. **Élitisme :** où la sélection par rang, cette technique permet de sélectionner les meilleurs individus (ceux qui ont les valeurs de fonction d'adaptation les plus fortes) et de négliger les individus à faible fonction objectif.
4. **Sélection uniforme :** permet de tirer aléatoirement les individus parmi la population. Cette technique s'effectue sans l'intervention de la fonction objectif, tous les individus ayant la même probabilité à être sélectionnés.

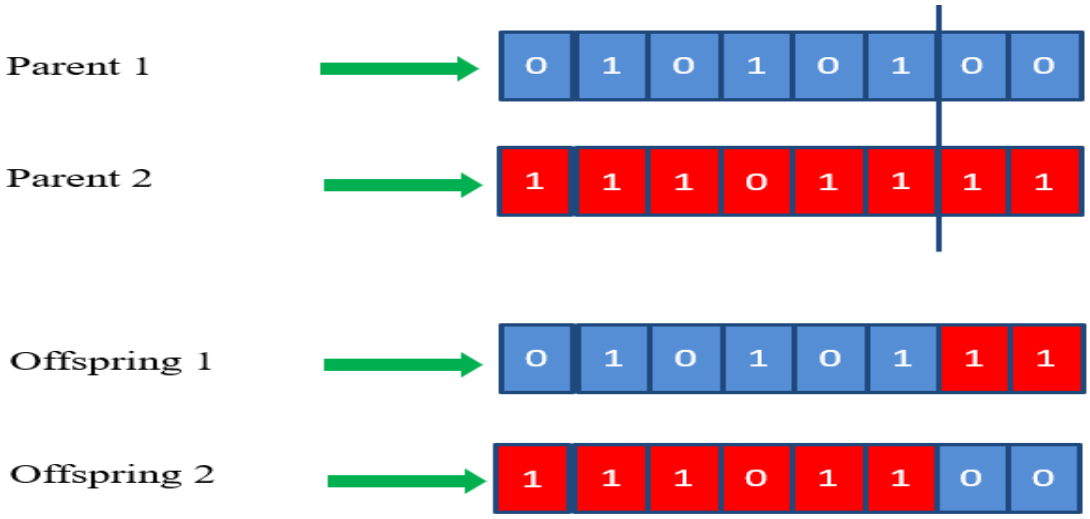


FIGURE 5.3 – Croisement en un point

5.2.4.2 Croisement

Le croisement où l'opération de reproduction permet d'enrichir la diversité dans la population, et la génération de nouveaux individus en assurant l'échange entre les individus sélectionnés nommés parents pour produire des individus dits enfants. Ces derniers héritent certaines caractéristiques de chaque parent dont l'objectif est d'améliorer les performances des individus participant à la nouvelle génération. De nombreuses techniques de croisement selon le type de codage sont proposées afin d'assurer la meilleure reproduction.

1. **Croisement simple :** le choix de point de croisement est effectué d'une manière aléatoire. par la suite, les gènes des individus parents sont échangés entre eux et recombinés au point de croisement (voir la figure 5.3).
2. **Croisement double points :** le croisement est effectué en deux points choisis aléatoirement. Les individus enfants héritent les gènes des parents selon les deux points de croisement.
3. **Croisement uniforme :** consiste à définir une séquence de bits ayant la même longueur de chromosomes, nommée masque, permettant de définir les positions où l'enfant doit hériter du premier parent ou du deuxième parent.

5.2.4.3 Mutation

Cet opérateur a pour but de changer et de modifier aléatoirement une partie du chromosome avec une probabilité définie. Le taux de probabilité est généralement faible, il est compris entre 0.01 et 0.001, ce qui rend l'opération très rare. Voir Figure 5.4.
Le but de la mutation est d'explorer tout l'espace de recherche, d'assurer la diversité et la convergence vers la meilleure solution.

5.2.5 Paramètres de dimensionnement

En plus de la taille de population, d'autre paramètres influencent le résultat final d'un algorithme génétique. Ces paramètres sont liés principalement au critère d'arrêt et à la probabilité d'application des opérateurs génétiques (croisement et mutation). Le critère d'arrêt varie d'une application à une autre, il peut dépendre du nombre maximum de génération ou l'atteinte de la valeur optimale, il peut aussi être lié à limitation d'utilisation de CPU, comme il peut arrêter le programme si les générations ne subissent à aucun changement.

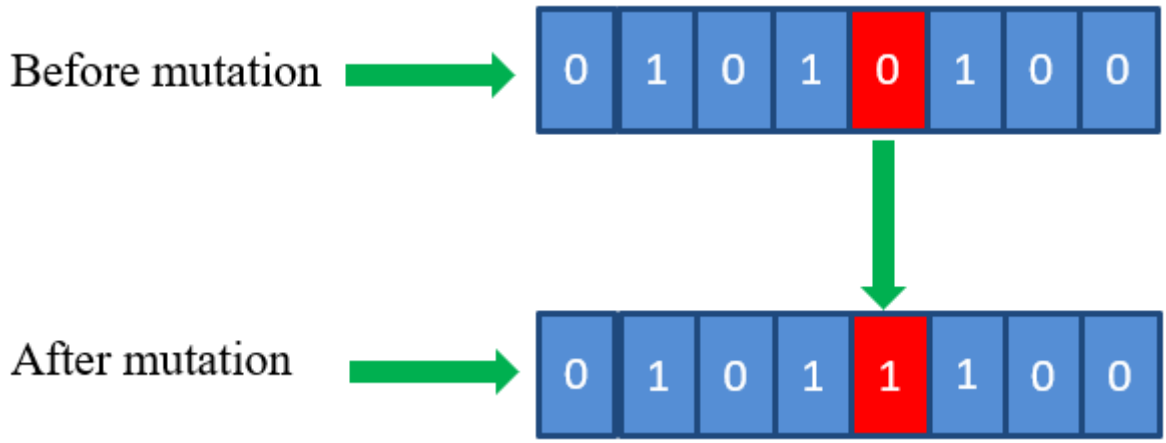


FIGURE 5.4 – Exemple de mutation

Le choix de taux d'application des opérateurs génétiques est critique, il affecte directement la diversité dans les nouvelles générations et la convergence vers les solutions optimales. Généralement la probabilité de mutation est très faible par rapport à la probabilité de croisement. Cette dernière varie entre 70% et 95%, tandis que le taux de mutation est entre 0.01% et 1% [H+92].

5.2.6 Discussion

Parmi les algorithmes d'optimisation, les algorithmes génétiques ont prouvé leur puissance de résoudre les problèmes du monde réel. Ils sont largement applicables dans de nombreux domaines comme la médecine, la biologie et l'informatique. Par leur nature évolutionnaire, ces algorithmes ne demandent pas beaucoup de connaissances mathématiques sur le problème à optimiser. En plus, ils offrent une flexibilité leur permettant d'être combinés avec d'autres algorithmes d'inspirant de l'intelligence et du comportement humains.
Malheureusement, ces algorithmes ne garantissent pas la convergence et l'obtention de la meilleure solution et leur utilisation est généralement coûteuse en terme de temps de calcul parce qu'ils explorent toute l'espace de recherche qui possède un nombre important de solutions. Il convient de noter que les algorithmes génétiques traitent les problèmes mono-objectif, c'est-à-dire la fonction d'évaluation à affecter a un seul objectif. Prenons l'exemple de notre cas d'étude, l'évaluation du protoforme "Q y sont S" où l'objectif est d'optimiser le degré de vérité de la déclaration quantifiée. Cependant, dans la plupart des cas, le modèle constitué cherche à optimiser plusieurs critères simultanément pour aboutir à la meilleure solution.
La notion de multi-objectif ouvre la porte à des nouveaux axes de recherche qui consiste à trouver un compromis entre les objectifs qui sont dans la plupart du temps contradictoires où l'amélioration d'un objectif provoque la détérioration de l'autre. Dans ce qui suit, nous discutons l'aspect lié à l'optimisation multi critère.

5.3 Optimisation multi-objectif

5.3.1 Problème multi-objectif

Contrairement au problème d'optimisation à simple objectif, l'optimisation multi-objectif cherche à trouver une solution adéquate permettant d'optimiser un ensemble de fonctions objectif généralement contradictoire, en respectant un certain nombre de contraintes liées au problème à résoudre. Un problème multi objectif peut être formulé à l'aide de l'équation (5.2) :

$$f(x) = [f_1(x), f_2(x), \cdots, f_m(x)] \tag{5.2}$$

f représente l'ensemble des fonctions objectif, m fait référence au nombre des fonctions objectif et x représente un individu de la population.

5.3.2 Panorama des systèmes d'optimisation multi critères

Récemment, plusieurs travaux traitant les algorithmes génétiques se sont focalisés sur l'aspect multi-objectif. Dans la plupart des situations réelles, nous sommes confrontés à des problèmes d'optimisation où , généralement, les objectifs sont en conflit les uns avec les autres tels que l'augmentation des performances et la réduction du coût [KCS06]. L'amélioration d'un objectif particulier peut conduire à des valeurs inacceptables pour l'autre objectif. La solution appropriée est d'avoir un compromis entre les objectifs, ce qui signifie trouver une solution qui satisfait les objectifs à un niveau acceptable sans être dominée par d'autres solutions. Un algorithme génétique peut être exploité en optimisation multi-objectifs. Plusieurs modèles de l'GA multi-objectif ont été proposés dans la littérature, MOGA [FF$^+$93], NSGA [SD94], SPEA [ZT99], NSGA II [DPAM02], NPGA [HNG94], PESA [CKO00].
Une étude comparative entre les techniques d'optimisation élitistes les plus populaires : NSGA II, SPEA et PESA, a été présentée dans [KYD03, SS08], les trois méthodes étudiées propagent la notion de dominance dans le sens de Pareto pour conserver les solutions non dominées par les autres solutions. Les résultats de la comparaison montrent qu'aucune méthode ne domine les autres dans le sens de Pareto. Ils montrent aussi que NSGA II et SPEA offrent les mêmes performances de convergence vers la solution optimale et l'algorithme NSGA II est plus rapide que les autres. En plus, on prétend que cette technique a surpassé PAES et SPEA en terme de recherche d'un ensemble diversifié de solutions. En se basant sur cette étude, nous proposons d'exploiter l'algorithme NSGA II et le combiner avec l'algorithme de résumé discuté dans le chapitre 4 afin d'obtenir l'ensemble des meilleurs résumés linguistiques en utilisant les mesures de qualité comme critères de sélection.

5.3.3 Méta-heuristiques pour l'optimisation multi-objectif

Afin de traiter un problème multi-objectif, [Ber01] propose d'utiliser trois méthodes adaptées aux problèmes à optimiser :

1. **Méthodes agrégées :** consistent à transformer les ensembles des objectifs en un simple objectif en exploitant la fonction d'utilité qui regroupe l'ensemble des fonctions objectifs. Ces méthodes sont considérées comme des techniques à priori, c'est-à-dire la définition des préférences des objectifs est effectuée avant le début de processus de recherche de la solution optimale.
2. **Méthodes non agrégées et de type non Pareto** consistent à traiter séparément les objectifs, parmi ces techniques, deux sont les plus utilisées. Vector Evaluated Genetic Algorithm (VEGA) fonctionne comme un algorithme génétique traditionnel, elle traite chaque objectif indépendamment des autres. La deuxième technique lexicographique traite les fonctions objectifs d'une manière séquentielle en respectant l'ordre d'importance.
3. **Méthodes basées sur le principe de Pareto :** contrairement aux méthodes agrégées et non agrégées, les techniques de Pareto sont des méthodes à posteriori reposant sur l'utilisation de la notion de dominance au sens de Pareto [Par97] et permettant de traiter simultanément les objectifs pour fournir les solutions optimales en cherchant un meilleur compromis entre les objectifs.

Les techniques d'optimisation multi-objectif s'inspirent généralement des comportements observés en biologie, elles s'appuient sur le caractère stochastique qui autorise le mécanisme du choix aléatoire des solutions pour assurer la diversité et l'exploration d'espace de recherche. En plus de ces caractéristiques, les méthodes heuristiques sont des méthodes itératives, approchées et

directes. Le premier caractère fait référence au processus répétitif jusqu'à la satisfaction des critères d'arrêt. La notion approchée se réfère à l'incertitude de la convergence et l'obtention de la solution optimale. Le dernier caractère "direct" sert à associer une valeur des fonctions objectifs à chaque individu de la population.
Comme mentionné précédemment, nous nous somme intéressés à la méthode NSGA II qui est une méthode fondée sur le principe de la dominance au sens de Pareto. Avant d'expliquer le principe de fonctionnement de NSGA II, nous proposons d'étudier la notion de dominance et l'optimalilté de Pareto.

5.3.4 Notion de dominance

Dans les problèmes d'optimisation multi-objectif, la bonne solution est déterminée par la dominance proposée par Pareto [Par97]. Étant donné deux objectifs, une solution non dominée est lorsqu'aucune des solutions n'est meilleure que l'autre par rapport à ces deux objectifs.
Lorsque la solution a n'est pas pire que la solution b dans tous les objectifs et que la solution a est strictement meilleure que b dans au moins un objectif, alors la solution a domine la solution b.

$$a \prec b \text{ (a domine b)} \iff f(a) < f(b)$$

$$a \preceq b \text{ (a domine faiblement b)} \iff f(a) \leq f(b)$$

$$a \sim b \text{ (a est indifférent pour b)} \iff f(a) \not\leq f(b) \wedge f(b) \not\leq f(a)$$

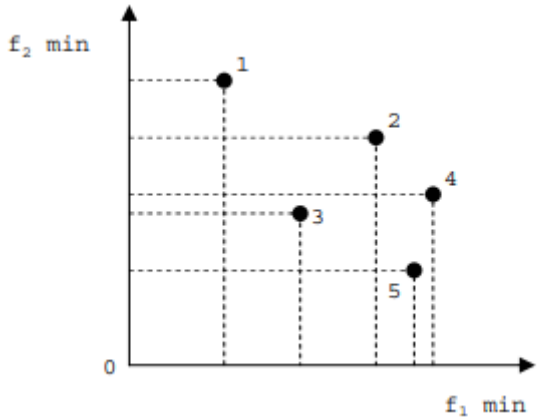


FIGURE 5.5 – Exemple de dominance

Dans l'exemple montré par la figure 5.5, nous cherchons à minimiser les deux fonction objectifs, les point 1, 3 et 5 ne sont dominés par aucun autre point, alors que le point 2 est dominé par le point 1, et le point 4 est dominé par 3 et 5.

5.3.5 Frontière de Pareto

Le frontière de Pareto décrit l'ensemble des points non dominés, c'est à dire l'ensemble des solution optimales. La figure 5.6 illustre les frontières de Pareto obtenus pour plusieurs scénarios ; maximisation des deux fonction objectifs, minimisations des deux objectifs, maximisation d'un objectif et la minimisation de l'autre.
La détermination des solutions optimales d'un problème multi-objectif consiste à définir la frontière de Pareto.

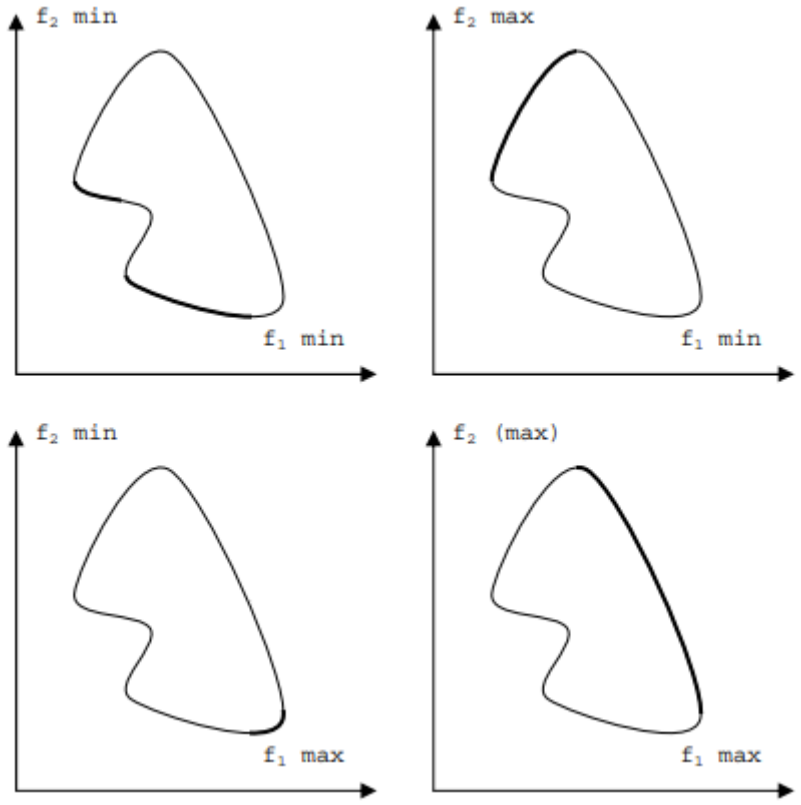


FIGURE 5.6 – Frontière de Pareto

5.4 Algorithme génétique multi objectif NSGA II

Non dominated Sorting Genetic Algorithm II (NSGA II) est l'un des algorithmes génétiques multi-objectifs les plus connus, il est proposé par [DPAM02]. NSGA II utilise la notion de dominance de Pareto pour classer les solutions et définir des stratégies de sélection (reproduction ou survie). En général, il est considéré comme un algorithme élitiste. Il garde les meilleures solutions dans la population au fil des générations ; ces solutions participent au processus de reproduction. Cependant, le nombre de solutions non dominées peut très vite augmenter pour des problèmes à plusieurs objectifs. Voir la Figure 5.7.
Afin de surmonter ce problème, NSGA II utilise un mécanisme de préservation de la diversité. À chaque génération, la population des parents et celle des enfants sont fusionnées et classées en plusieurs fronts de Pareto. La population de la prochaine génération se forme en choisissant des solutions dans ces fronts de Pareto en commençant par la première. Lorsque la taille du front à utiliser est supérieure au nombre de places restant à voir dans la population future, les solutions sont choisies en fonction de leurs valeurs de distance d'encombrement (crowding). Elle s'agit d'un indicateur qui calcule la distance moyenne, sur l'ensemble des objectifs, entre une solution donnée et ses voisins directs dans l'espace des résultats (l'espace des objectifs) comme indiqué dans la formule (5.3), où N indique la taille de la population et M le nombre d'objectifs.

$$CD_i = \begin{cases} \sum_{m=1}^{M} \frac{f_m(x_{i+1}) - f_m(x_{i-1})}{f_m(x_{max}) - f_m(x_{min})} & \text{for } i = 2, ..N-1 \\ \infty & \text{for } i = 1 \text{ and } i = N \end{cases} \tag{5.3}$$

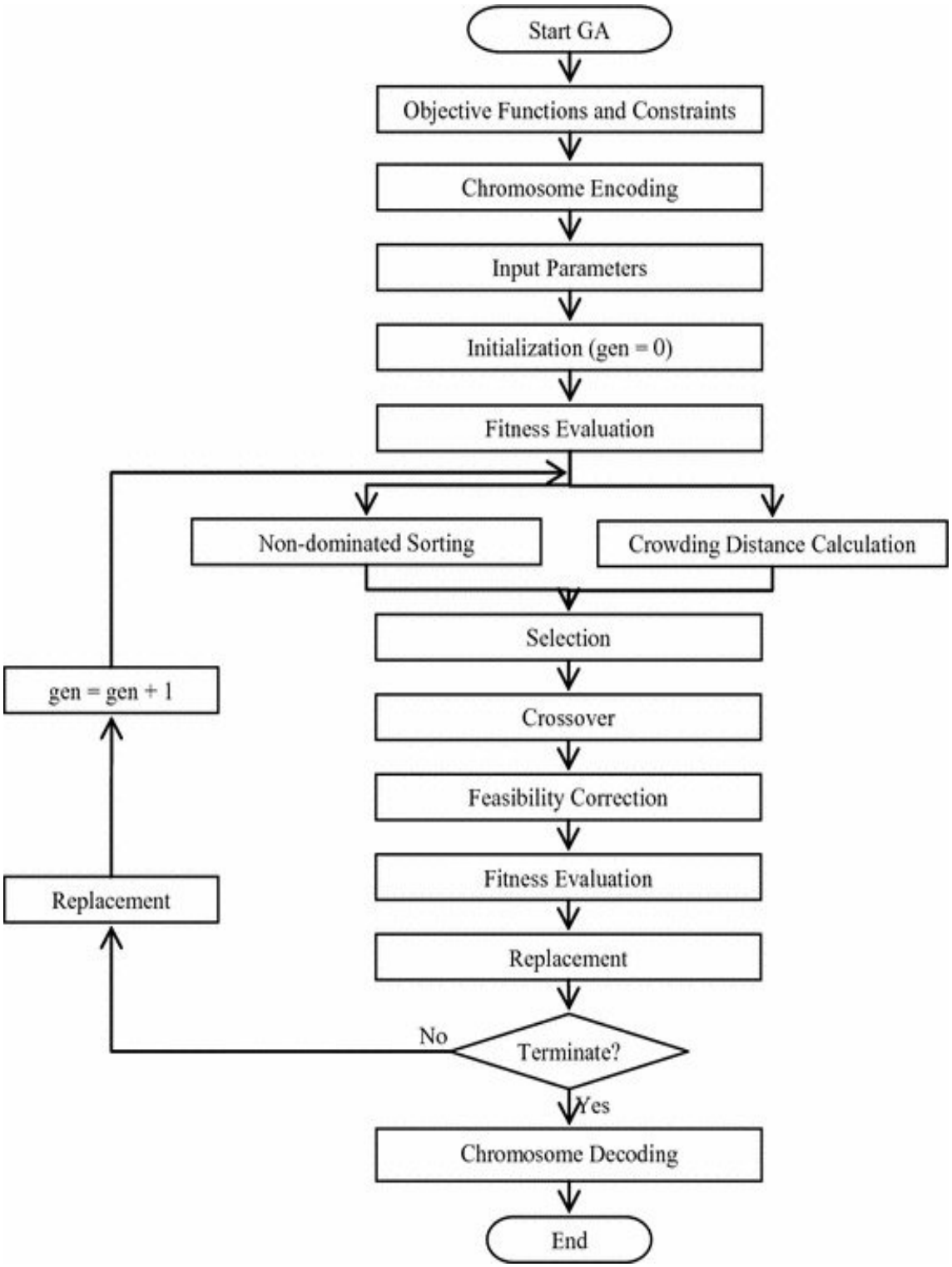


FIGURE 5.7 – Principe de fonctionnement de NSGA II

5.5 Étude expérimentale

L'objectif de cette étude est de proposer une méthode hybride permettant de combiner de nombreux aspects :

— le résumé linguistique basé sur la logique floue et qui s'inspire du raisonnement humain,

— l'identification des caractéristiques dynamiques des séries temporelle (dynamique du changement, durée et la variabilité) en se basant sur la segmentation linéaire par morceau présenté par l'algorithme Bottom-Up,

— l'optimisation mono objectif et multi objectif basée sur l'exploitation de l'algorithme génétique et le NSGA II respectivement.

Afin de concevoir et de produire un ensemble des meilleurs résumés dérivés à partir des tendances des séries temporelles sous la forme du protoforme **Q R y sont S**.

5.5.1 Modèle de l'algorithme génétique

Dans le travail actuel, nous utilisons l'algorithme Bottom-Up proposé [BHL20] pour segmenter les séries temporelles et générer des tendances. Les données sur lesquelles les tests sont réalisés sont collectées à partir de plusieurs flux de données générés par de multiples capteurs installés sur le campus de l'Université de Toulouse III. Les tendances obtenues, en utilisant l'algorithme de segmentation linéaire par morceaux "Bottom-Up" présenté par l'algorithme 4, sont associées à un ensemble de caractéristiques : dynamique du changement, variabilité et durée. Les résultats de cette segmentation sont présentés dans la figure 4.5 dans la page 74.
Notre première contribution consiste à créer un ensemble de meilleurs résumés dans le second protoforme *"Q R y are S"*, en utilisant l'algorithme génétique avec les opérateurs traditionnels : sélection, croisement et mutation. Pour ce faire, nous supposons que la population initiale est un ensemble de résumés et que chaque résumé représente un individu. Un résumé peut être codé sous forme binaire avec une longueur de 8 bits (longueur du chromosome) comme montre la figure 5.8.
Le choix de la longueur du chromosome dépend du nombre de quantificateurs et de valeurs linguistiques utilisés. Dans notre cas, nous avons utilisé trois quantificateurs linguistiques : la plupart, presque la moitié et peu. Ces trois peuvent être codés sur deux bits, En plus, pour les deux variables linguistiques durée et variabilité nous proposons d'utiliser trois modalités floues chacune codée sur deux bits. Enfin, nous consacrons deux bits pour positionner les variables linguistiques. Par exemple, le résumé *"la plupart des tendances courtes ont une faible variabilité"* peut être codé comme 10 0 00 1 00.

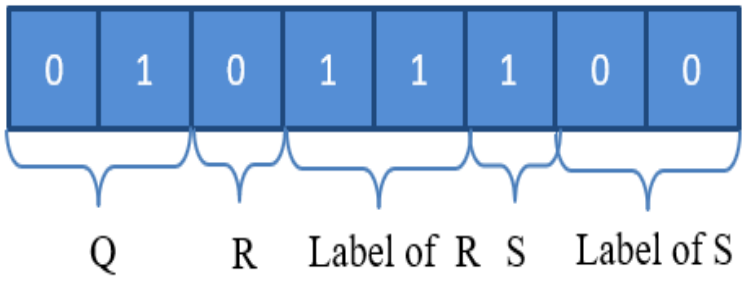


FIGURE 5.8 – Exemple d'un individu

La population initiale est générée aléatoirement et la fonction de fitness associée à chaque individu est calculée en se basant sur le calcul de degré de vérité d'une proposition quantifiée

proposé par [Zad83]. La fonction d'évaluation est donnée par la formule :

$$T = \mu_Q[\frac{\sum_{i=1}^{n} min(\mu_S(y_i).\mu_R(y_i))}{\sum_{i=1}^{n} \mu_R(y_i)}] \tag{5.4}$$

Pour sélectionner les individus qui participeront à la création de la nouvelle génération, nous appliquons la sélection par roue de loterie, où la valeur de la fonction d'évaluation est utilisée pour associer une probabilité de sélection à chaque individu. La probabilité d'être sélectionné pour un individu i est donnée par l'équation (5.5) où f_i est la valeur de fitness de l'individu i, et N est la taille de la population.

$$p_i = \frac{f_i}{\sum_{i=1}^{N} f_i} \tag{5.5}$$

L'opérateur de croisement à point unique est appliqué pour générer les individus enfants, le point de croisement est choisi au hasard. Ensuite, les deux individus générés à la suite du l'opérateur de croisement sont mutés en fonction du paramètre de taux de mutation. Cette procédure est répétée jusqu'à la satisfaction des critères finaux proposés par l'utilisateur ; ici nous avons utilisé le nombre maximum d'itérations comme critère d'arrêt.
La figure 5.9 illustre la variation du temps d'exécution en fonction des paramètres du modèle. La figure 5.9a montre l'effet de la variation du nombre d'itérations sur le temps d'exécution en fixant le taux de mutation, tandis que la figure 5.9b décrit la variation du temps d'exécution en fonction du changement du taux de mutation. La figure 5.9c représente l'évolution de la fonction d'évaluation au cours des générations.
Comme cas d'étude, nous supposons le taux de mutation = 0.1, la taille de la population initiale = 15 et le nombre maximum d'itérations = 100. Pour cet ensemble de paramètres de l'algorithme génétique, nous obtenons un ensemble des meilleurs résumés décrits dans le tableau 5.1.

Résumé linguistique	**Degré de vérité**
Presque la moitié des tendances avec une variabilité modérée sont courtes	0.944
Presque la moitié des tendances moyennes ont une forte variabilité	0.939
Presque la moitié des longue tendances ont une haute variabilité	0.873
Presque la moitié des tendances à forte variabilité sont courtes	0.837
La plupart des tendances moyennes ont une variabilité modérée	0.753
La plupart des tendances moyennes ont une variabilité modérée	0.704
La plupart des tendances à faible variabilité sont courtes	0.7

Tableau 5.1 – Résumés linguistiques obtenus en utilisant l'algorithme génétique

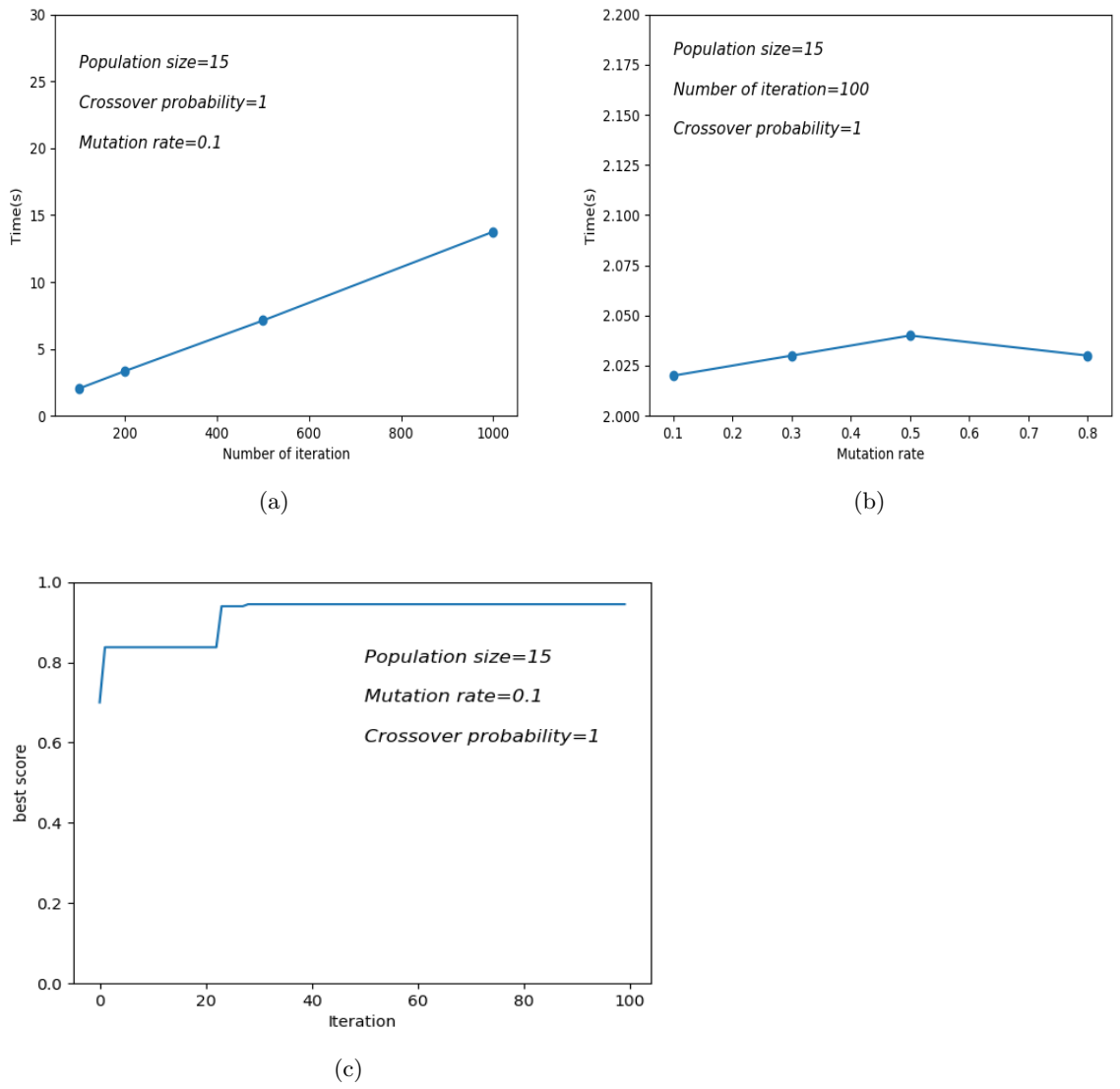


FIGURE 5.9 – Évolution de l'algorithme génétique (a) Temps d'exécution en fonction du nombre d'itérations (b) Temps d'exécution en fonction du taux de mutation (c) Taux de réussite par rapport au nombre d'itérations

5.5.2 Sélection multi-critère basée sur NSGA II

Il est clair que le degré de vérité est le critère le plus important utilisé pour évaluer le résumé. Pour cette raison, nous utilisons l'algorithme génétique pour choisir l'ensemble des meilleurs résumés. Cependant, il existe plusieurs critères utilisés pour décrire la qualité d'un résumé comme mentionné dans le chapitre 3, ce qui nous a incité à utiliser l'algorithme génétique multi-objectif NSGA II.

Les critères utilisés dans cette section sont : le degré de vérité, le degré de pertinence, le degré de couverture et le degré d'imprécision. Ils sont considérés comme des objectifs qu'on cherche à optimiser afin d'extraire des résumés linguistiques permettant de maximiser les quatre critères.

En plus des opérations traditionnelles, NSGA II utilise la notion de non-dominance au sens de Pareto pour choisir les individus qui participeront à la nouvelle génération. Si le nombre d'individus dans le front de Pareto dépasse la taille de la population initiale, NSGA II utilise la distance d'encombrement introduite dans l'équation (5.3) pour réduire le nombre de participants.

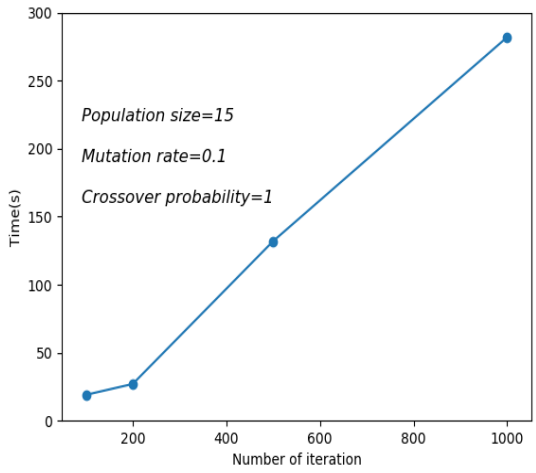
Les résultats de nos expérimentations sont reportés dans la figure 5.10

— la figure 5.10a représente la variation du temps d'exécution avec le nombre d'itérations,

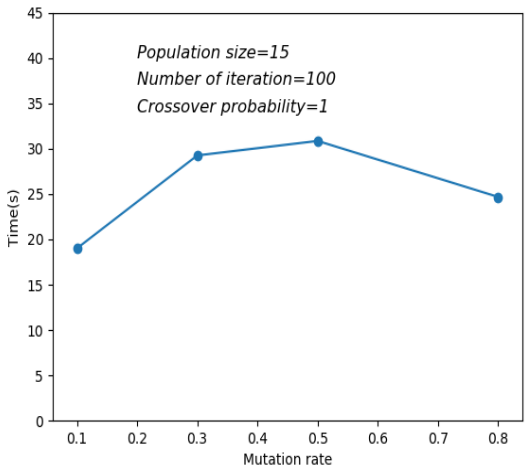
— la figure 5.10b représente le temps d'exécution en faisant varier la probabilité de mutation,

— la figure 5.10c représente le front de Pareto final où le taux de mutation = 0,1 et le nombre d'itérations = 100.

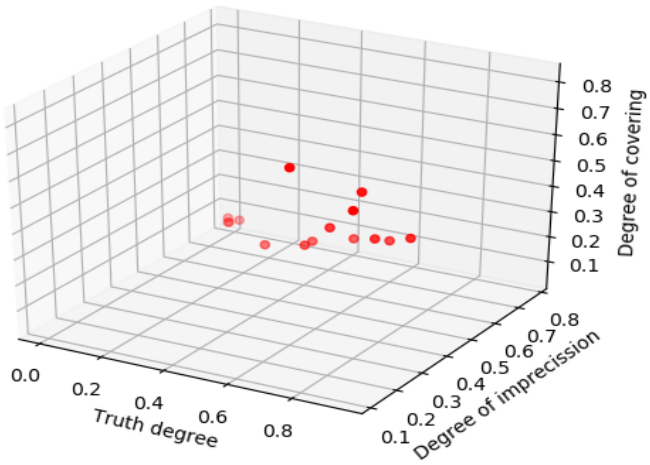
Dans le tableau 5.2, nous donnons l'ensemble des meilleures solutions obtenues avec différents critères d'optimisation en utilisant NSGA II (avec 100 itérations et une probabilité de mutation = 0,1).



(a)



(b)



(c)

FIGURE 5.10 – Évolution de l'algorithme génétique multi-objectif (a) Temps d'exécution en fonction du nombre d'itérations (b) Temps d'exécution en fonction du taux de mutation (c) Front de Pareto final

Résumé	**Degré de vérité**	**Imprécision**	**Couverture**	**Pertinence**
Presque la moitié des tendances à variabilité modérée sont courte	0.944	0.329	0.473	0.024
Presque la moitié des tendances avec durée moyenne ont une variabilité haute	0.939	0.115	0.701	0.0.080
Presque la moitié des tendances longues ont une forte variabilité	0.873	0.230	0.686	0.090
Presque la moitié des tendances à forte variabilité sont courtes	0.837	0.329	0.445	0.003
La plupart des tendances moyennes ont une variabilité modérée	0.753	0.115	0.812	0.030
La plupart des tendances à faible variabilité sont courtes	0.7	0.329	0.454	0.005

Tableau 5.2 – Résumés linguistique en utilisant NSGA II

5.6 Conclusion

Dans ce chapitre, nous avons proposé d'utiliser le concept d'algorithmes génétiques pour améliorer la qualité des résumés linguistiques des séries temporelles. L'objectif principal est de générer *un ensemble des meilleurs résumés* des caractéristiques dynamiques des tendances associées à la série temporelles, en utilisant tout d'abord l'algorithme génétique mono objectif, où la fonction d'évaluation représente le degré de vérité d'un proposition linguistique quantifiée sous le protoforme basic "Q R y sont S". Ensuite, nous avons utilisé l'algorithme génétique multi-objectif NSGA II, où la fonction de fitness proposée inclut les différentes mesures de qualité afin de garantir une haute qualité des résumés obtenus.
Pour atteindre cet objectif, nous avons commencé par la description du concept de l'algorithme génétique, puis nous avons détaillé les opérateurs génétiques permettant d'assurer la diversité dans la population et ensuite nous avons discuté les avantages et les inconvénients de l'utilisation de tels algorithmes. La deuxième partie du chapitre a été consacrée à élucider le but de l'utilisation de l'algorithme génétique multi-objectif, plus précisément nous avons décrit l'algorithme génétique multi-objectifs NSGA II.
Les deux méthodes méta heuristiques sont exploitées afin d'extraire un ensemble de meilleurs résumés décrivant les caractéristiques dynamiques des tendances d'une série temporelle. Les données sur laquelle les deux méthodes sont implémentées, sont collectées à partir des capteurs installés dans le campus intelligent NeOCampus.

Conclusion et perspectives

Conclusion

À l'ère du numérique et en raison du progrès et du développement importants et sans précédents des technologies à différentes échelles, nous sommes quotidiennement confrontés à un énorme volume de données provenant de nombreuses sources telles que les réseaux de capteurs, les stockages dans le cloud, les réseaux sociaux, etc. Bien que ce volume gigantesque de données puisse être vraiment utile pour les décideurs particuliers et les décideurs dans les entreprises, il pourrait également être problématique. En effet, un grand volume de données a ses propres lacunes : Il a besoin de gros espace de stockage, il rend les opérations de traitement et de récupération difficiles et très coûteuse. Une solution pour surmonter ces problèmes consiste à réduire le volume de ces données sous une forme plus compacte mais informative. Une technique de réduction de données est d'appliquer des méthodes de résumés de données.

Dans le cadre du travail mené dans cette thèse, nous avons contribué à la construction de résumés de données en s'appuyant sur des techniques issues de l'intelligence computationnelle qui reflètent fidèlement le raisonnement et le comportement humains. La première méthode de construction de résumés consiste à utiliser les propositions linguistiques quantifiées, nommées résumés linguistiques. La deuxième méthode concerne l'exploitation du concept de la valeur typique pour fournir un résumé d'un ensemble de données. L'étude comparative, menée sur les deux approches, nous a permis de constater que le résumé linguistique possède des performances lui permettant de s'adapter au mieux dans le contexte des données massives et volumineuses.

L'approche pour les résumés linguistiques a été appliquée dans un premier temps sur des données à caractère statique. Puis, nous avons considéré des données présentant une spécificité temporelle, nommée séries temporelles. Nous avons ainsi proposé de résumer les propriétés des tendances caractérisant les séries temporelles. Nous avons commencé par l'exploitation de l'algorithme de segmentation linéaire par morceaux "Bottom-Up" afin d'obtenir les segments des série temporelles. Après, nous avons associé à chaque segment un ensemble de propriétés : la dynamique du changement, la durée et la variabilité. Ces caractéristiques sont traitées comme des variables linguistiques où chaque variable est caractérisée par un ensemble de valeurs linguistiques modélisées par des ensembles flous.

Le processus de génération de résumés linguistiques pour un ensemble de données numériques, peut être considéré comme un problème d'optimisation. Il s'agit de sélectionner les meilleurs résumés parmi un grand ensemble de candidats où la fonction objectif de base est supposée être le degré de validité associée à chaque résumé.

Pour traiter ce problème, nous avons proposé d'utiliser des heuristiques basées sur une procédure d'amélioration des solutions, plus spécifiquement, les algorithmes génétiques. Pour aller plus loin dans notre réflexion, nous avons proposé d'évaluer le résumé non seulement sur la base du seul degré de vérité mais avec de multiples critères qui rend l'algorithme plus efficace tels que le degré de couverture, le degré d'imprécision, etc. Cette nature multiobjectif du problème à nécessité l'utilisation de l'algorithme génétique multi objectifs NSGA II.

Toutes nos propositions ont été validées en menant un ensemble d'expérimentations sur des données réelles collectées dans le cadre du projet ADSB de l'ENSMA et du projet NeOCampus

de l'université de Toulouse III.

Perspectives

L'ensemble des contributions proposées dans le cadre de cette thèse ouvre la voie vers de nouveaux champs d'études permettant ainsi la poursuite de nos travaux de recherche. Les perspectives détaillées dans ce qui suit s'articulent autour de trois axes principaux : l'identification de classes de requêtes pertinentes, l'utilisation des autres méthodes d'optimisation, et enfin l'exploitation des résumés linguistiques dans des domaines d'applications critiques et d'actualité.

Typologie de requêtes

Le résumé linguistique des données peut être utilisé pour répondre à des requêtes floues bien particulières. Nous envisageons d'implémenter l'algorithme de construction des résumés linguistiques dans les systèmes de gestion des flux de données et d'identifier une typologie de requêtes pertinentes qui pourraient être évaluer sur ces structures de résumé.

Méthodes d'optimisation multicritères

Pour améliorer la qualité des résumés obtenus nous avons proposé dans notre étude l'utilisation de l'algorithme génétique dans le cas de résumé simple critère (évaluation de degré de vérité), et l'algorithme NSGA II dans le cas d'optimisation multi critères. Ces deux techniques sont permettent l'amélioration de la solution. Cependant, il existe des autres méthodes d'optimisation comme Ant colony optimisation (ACO) [DBS06] qui s'inspire du comportement des fourmis et qui cherche les chemins optimaux pour trouver la meilleure solution.

Résumés linguistiques et maisons intelligentes

Le passage aux villes intelligentes, puis aux villes cognitives, devrait suivre les besoins des citoyens, et pas seulement l'utilisation efficace des ressources. Les citoyens souhaitent coopérer à la prise de décision (ou au vote) et être informés des divers développements dans ces villes, de préférence de manière compréhensible et intelligibles. L'exploitation des résumés linguistiques à des fins d'information et de production de l'information envers les utilisateurs les encourageraient à participer davantage à la maintenance et la préservation de la qualité de vie dans ces villes. D'autres acteurs des villes (répartiteurs, planificateurs, spécialistes du marketing, autorités locales, journalistes) peuvent également bénéficier de cette approche.
Des exemples comme l'information citoyens, gestion d'enquêtes, explication de l'évolution de la pollution et du trafic, analyse des activités touristiques, pourraient également bénéficier de l'approche des résumés linguistiques. De cette manière, les parties prenantes sont informées de manière concise de la situation et des tendances [HBH18, Hud19].
Le concept de résumés linguistiques serait d'un intérêt importante permettant aux citoyens de s'adapter avec les villes intelligentes, de mieux comprendre le mode de vie dans ces villes, de se préparer aux évènements prédicatifs et de participer activement au développement de la civilisation dans ces villes.

Aéronautique et résumés linguistiques

Le but de tous les recherches dans le domaine de l'aéronautique est de maintenir l'aéronef dans les bonnes conditions de navigabilité. Afin de réaliser cet objectif plusieurs systèmes sont placés à bord des aéronefs et des aérodromes. Ces systèmes génèrent une quantité énorme des données dont le résume de données peut être exploité à des fins multiples : La maintenance pourrait anticiper le besoin de remplacement de certaines parties de l'avion ; la congestion du trafic aérien

pourrait être réduite ; les trajets de vol pourraient être modifiés bien avant le décollage pour éviter les tempêtes ; les systèmes pourraient seconder les pilotes en gérant des tâches de cockpit routinières. En plus de ces tâches, les résume linguistiques peuvent être exploités autrement par les décideurs finaux. Ils permettent aux compagnies aériennes non seulement d'assurer la sécurité, mais aussi de réaliser des économies de coûts potentiels. Cela serait possible par l'utilisation de ces résumés pour l'extraction d'informations concises et utiles à partir des données provenant de multiples capteurs positionnés dans différents endroits dans la gestion des vols d'avions.

Bibliographie

[ACC+03] Daniel J Abadi, Don Carney, Ugur Cetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Michael Stonebraker, Nesime Tatbul, and Stan Zdonik. Aurora : a new model and architecture for data stream management. *the VLDB Journal*, 12(2) :120–139, 2003. (Cité en page 17)

[ads16] `https://forge.ia.ensma.fr/projects/ads-b/wiki`, 2016. (Cité en page 2), (Cité en page 39), (Cité en page 52)

[Ahm19a] Mohiuddin Ahmed. Data summarization : a survey. *Knowledge and Information Systems*, 58(2) :249–273, 2019. (Cité en page 1), (Cité en page 23), (Cité en page 25), (Cité en page 33)

[Ahm19b] Mohiuddin Ahmed. Intelligent big data summarization for rare anomaly detection. *IEEE Access*, 7 :68669–68677, 2019. (Cité en page 23)

[ALBM+13] Rui Jorge Almeida, Marie-Jeanne Lesot, Bernadette Bouchon-Meunier, Uzay Kaymak, and Gilles Moyse. Linguistic summaries of categorical time series for septic shock patient data. In *2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE, 2013. (Cité en page 40)

[AT13] Luis Arguelles and Gracian Trivino. I-struve : Automatic linguistic descriptions of visual double stars. *Engineering Applications of Artificial Intelligence*, 26(9) :2083–2092, 2013. (Cité en page 40)

[AYA+17] Tunahan Altintop, Ronald R Yager, Diyar Akay, Fatih Emre Boran, and Muhammet Ünal. Fuzzy linguistic summarization with genetic algorithm : An application with operational and financial healthcare data. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 25(04) :599–620, 2017. (Cité en page 79)

[Bat02] Ildar Batyrshin. On granular derivatives and the solution of a granular initial value problem. *International Journal of Applied Mathematics and Computer Science*, 12 :403–410, 2002. (Cité en page 69)

[BAY16] Fatih Emre Boran, Diyar Akay, and Ronald R Yager. An overview of methods for linguistic summarization with fuzzy sets. *Expert Systems with Applications*, 61 :356–377, 2016. (Cité en page 39)

[BBC+04] Hari Balakrishnan, Magdalena Balazinska, Don Carney, Uğur Çetintemel, Mitch Cherniack, Christian Convey, Eddie Galvez, Jon Salz, Michael Stonebraker, Nesime Tatbul, et al. Retrospective on aurora. *The VLDB Journal*, 13(4) :370–383, 2004. (Cité en page 17)

[BBD+02] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–16, 2002. (Cité en page xiii), (Cité en page 18)

[Ber01] Alain Berro. *Optimisation multiobjectifs et stratégies d'évolution en environnement dynamique*. PhD thesis, ANRT [diff.], 2001. (Cité en page 85)

[BHL20] Khedidja Boulanouar, Allel Hadjali, and Mohand Lagha. A hybrid approach for linguistic summarization of time series. In *2020 International Conference on Data Analytics for Business and Industry : Way Towards a Sustainable Economy (ICDABI)*, pages 1–5. IEEE, 2020. (Cité en page 71), (Cité en page 89)

[BMM03] Bernadette Bouchon-Meunier and Christophe Marsala. *Logique floue : principes, aide à la décision*. Hermes-Lavoisier, 2003. (Cité en page 10)

[BMM12] Bernadette Bouchon-Meunier and Gilles Moyse. Fuzzy linguistic summaries : Where are we, where can we go ? In *2012 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pages 1–8. IEEE, 2012. (Cité en page 39)

[CBMB99] Purificación Cariñena, Alberto Bugarín, Manuel Mucientes, and Senén Barro. A language for expressing expert knowledge using fuzzy temporal rules. 1999. (Cité en page 2), (Cité en page 63)

[CBMB00] Purificación Cariñena, Alberto Bugarn, Manuel Mucientes, and Senén Barro. A language for expressing fuzzy temporal rules. *Mathware and Soft Computing*, 7(2-3) :213–227, 2000. (Cité en page 2), (Cité en page 63)

[CCD+03] Sirish Chandrasekaran, Owen Cooper, Amol Deshpande, Michael J Franklin, Joseph M Hellerstein, Wei Hong, Sailesh Krishnamurthy, Samuel R Madden, Fred Reiss, and Mehul A Shah. Telegraphcq : continuous dataflow processing. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 668–668, 2003. (Cité en page 17)

[CFLN02] Fu-lai Chung, Tak-chung Fu, Robert Luk, and Vincent Ng. Evolutionary time series segmentation for stock data mining. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 83–90. IEEE, 2002. (Cité en page 61)

[CFNL04] Fu-Lai Chung, Tak-Chung Fu, Vincent Ng, and Robert WP Luk. An evolutionary approach to pattern-based time series segmentation. *IEEE transactions on evolutionary computation*, 8(5) :471–489, 2004. (Cité en page 63)

[CH98] Colin L Carter and Howard J Hamilton. Efficient attribute-oriented generalization for knowledge discovery from large databases. *IEEE Transactions on knowledge and data engineering*, 10(2) :193–208, 1998. (Cité en page 41)

[Chi09] Raja Chiky. *Résumé de flux de données ditribués*. PhD thesis, 2009. (Cité en page 14), (Cité en page 25)

[CKO00] David W Corne, Joshua D Knowles, and Martin J Oates. The pareto envelope-based selection algorithm for multiobjective optimization. In *International conference on parallel problem solving from nature*, pages 839–848. Springer, 2000. (Cité en page 85)

[CMPV99] Juan C. Cubero, Juan Miguel Medina, Olga Pons, and Maria-Ampora Vila. Data summarization in relational databases through fuzzy dependencies. *Information Sciences*, 121(3-4) :233–270, 1999. (Cité en page 41)

[Coc07] William G Cochran. *Sampling techniques.* John Wiley & Sons, 2007. (Cité en page 31), (Cité en page 32)

[COMS09] Rita Castillo-Ortega, Nicolás Marín, and Daniel Sánchez. Fuzzy quantification-based linguistic summaries in data cubes with hierarchical fuzzy partition of time dimension. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 578–585. Springer, 2009. (Cité en page 1), (Cité en page 47), (Cité en page 62)

[COMS11] Rita Castillo-Ortega, Nicolas Mann, and Daniel Sánchez. Linguistic local change comparison of time series. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 2909–2915. IEEE, 2011. (Cité en page 63)

[COMST11a] Rita Castillo Ortega, Nicolás Marín, Daniel Sánchez, and Andrea GB Tettamanzi. Linguistic summarization of time series data using genetic algorithms. In *EUSFLAT*, volume 1, pages 416–423. Atlantis Press, 2011. (Cité en page 79)

[COMST11b] Rita Castillo-Ortega, Nicolás Marín, Daniel Sánchez, and Andrea GB Tettamanzi. A multi-objective memetic algorithm for the linguistic summarization of time series. In *Proceedings of the 13th annual conference companion on genetic and evolutionary computation*, pages 171–172, 2011. (Cité en page 79)

[COMST12] Rita Castillo-Ortega, Nicolás Marín, Daniel Sánchez, and Andrea GB Tettamanzi. Quality assessment in linguistic summaries of data. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 285–294. Springer, 2012. (Cité en page 47)

[Cse08] Baptiste Csernel. *Résumé généraliste de flux de données.* PhD thesis, Paris, ENST, 2008. (Cité en page 14), (Cité en page 17), (Cité en page 25)

[dat] `https://www.aerospacemanufacturinganddesign.com/article/millions-of-data-points-flying-part2-121914/`. (Cité en page 24)

[Dav91] Lawrence Davis. Handbook of genetic algorithms. 1991. (Cité en page 79)

[DBS06] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4) :28–39, 2006. (Cité en page 96)

[DDBK+13] Carlos A Donis-Diaz, Rafael Bello, Janusz Kacprzyk, et al. Linguistic data summarization using an enhanced genetic algorithm. *Czasopismo Techniczne*, 2013(Automatyka Zeszyt 2 AC (10) 2013) :3–12, 2013. (Cité en page 79)

[DDBK15] Carlos A Donis-Díaz, Rafael Bello, and Janusz Kacprzyk. Using ant colony optimization and genetic algorithms for the linguistic summarization of creep data. In *Intelligent Systems' 2014*, pages 81–92. Springer, 2015. (Cité en page 79)

[DDMBPM14] CA Donis-Diaz, AG Muro, R Bello-Pérez, and Eduardo Valencia Morales. A hybrid model of genetic algorithm with local search to discover linguistic data summaries from creep data. *Expert systems with applications*, 41(4) :2035–2042, 2014. (Cité en page 79)

[DP93] Didier Dubois and Henri Prade. On data summarization with fuzzy sets. In *Fifth IFSA Congress*, page 465, 1993. (Cité en page 47), (Cité en page 51)

[DPAM02] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm : Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2) :182–197, 2002. (Cité en page 2), (Cité en page 79), (Cité en page 85), (Cité en page 87)

[DPR98] D Dubois, H Prade, and E Rannou. An improved method for finding typical values. In *IPMU : information processing and management of uncertainty in knowledge-based systems (Paris, 6-10 July 1998)*, pages 1830–1837, 1998. (Cité en page 48), (Cité en page 50)

[DRSV14] Miguel Delgado, M Dolores Ruiz, Daniel Sánchez, and M Amparo Vila. Fuzzy quantification : a state of the art. *Fuzzy Sets and Systems*, 242 :1–30, 2014. (Cité en page 39)

[EKS+96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996. (Cité en page 29)

[FCNL01] Tak-chung Fu, Fu-lai Chung, Vincent Ng, and Robert Luk. Evolutionary segmentation of financial time series into subsequences. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)*, volume 1, pages 426–430. IEEE, 2001. (Cité en page 63)

[Feu10] George Feuerlicht. Database trends and directions : Current challenges and opportunities. In *DATESO*, pages 163–174. Citeseer, 2010. (Cité en page 12), (Cité en page 13)

[FF+93] Carlos M Fonseca, Peter J Fleming, et al. Genetic algorithms for multiobjective optimization : Formulationdiscussion and generalization. In *Icga*, volume 93, pages 416–423. Citeseer, 1993. (Cité en page 85)

[For65] Edward Forgey. Cluster analysis of multivariate data : Efficiency vs. interpretability of classification. *Biometrics*, 21(3) :768–769, 1965. (Cité en page 26)

[FSGM+] Min Fangy, Narayanan Shivakumar, Hector Garcia-Molina, Rajeev Motwani, et al. Computing iceberg queries e ciently. (Cité en page 32)

[Gab11] Nesrine Gabsi. *Extension et interrogation de résumés de flux de données*. PhD thesis, 2011. (Cité en page 23)

[GH88] David E Goldberg and John Henry Holland. Genetic algorithms and machine learning. 1988. (Cité en page 81)

[GÖ03] Lukasz Golab and M Tamer Özsu. Data stream management issues–a survey. Technical report, Technical Report, Apr. 2003. db. uwaterloo. ca/˜ ddbms/publications/stream . . ., 2003. (Cité en page xiii), (Cité en page 18)

[God99] Jelena Godjevac. *Idées nettes sur la logique floue*. PPUR presses polytechniques, 1999. (Cité en page 10)

[GRS00] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock : A robust clustering algorithm for categorical attributes. *Information systems*, 25(5) :345–366, 2000. (Cité en page 26)

[GRS01] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure : an efficient clustering algorithm for large databases. *Information systems*, 26(1) :35–58, 2001. (Cité en page 26)

[GSW04] Sudipto Guha, Kyuseok Shim, and Jungchul Woo. Rehist : Relative error histogram construction algorithms. In *VLDB*, volume 4, pages 300–311, 2004. (Cité en page 32)

[H+92] John Henry Holland et al. *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992. (Cité en page 79), (Cité en page 84)

[HBH18] Miroslav Hudec, Erika Bednárová, and Andreas Holzinger. Augmenting statistical data dissemination by short quantified sentences of natural language. *Journal of Official Statistics*, 34(4) :981–1010, 2018. (Cité en page 96)

[HNG94] Jeffrey Horn, Nicholas Nafpliotis, and David E Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence*, pages 82–87. Ieee, 1994. (Cité en page 85)

[HTG+15] ZR Hesabi, Zahir Tari, A Goscinski, Adil Fahad, Ibrahim Khalil, and Carlos Queiroz. Data summarization techniques for big data—a survey. In *Handbook on Data Centers*, pages 1109–1152. Springer, 2015. (Cité en page xiii), (Cité en page 25), (Cité en page 27), (Cité en page 30), (Cité en page 33)

[Hud16] Miroslav Hudec. Fuzziness in information systems. *Springer International Publishing*, 2016. (Cité en page 10), (Cité en page 23), (Cité en page 39), (Cité en page 40), (Cité en page 42), (Cité en page 47)

[Hud19] Miroslav Hudec. Possibilities for linguistic summaries in cognitive cities. In *Designing Cognitive Cities*, pages 47–84. Springer, 2019. (Cité en page 96)

[IP95] Yannis E Ioannidis and Viswanath Poosala. Balancing histogram optimality and practicality for query result size estimation. *Acm Sigmod Record*, 24(2) :233–244, 1995. (Cité en page 32)

[iri17] `https://www.irit.fr/neocampus/fr/`, 2017. (Cité en page 2), (Cité en page 39), (Cité en page 52), (Cité en page 53), (Cité en page 61), (Cité en page 79)

[JMF99] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering : a review. *ACM computing surveys (CSUR)*, 31(3) :264–323, 1999. (Cité en page 26)

[JPK+19] Akshay Jain, Mihail Popescu, James Keller, Marilyn Rantz, and Brianna Markway. Linguistic summarization of in-home sensor data. *Journal of biomedical informatics*, 96 :103240, 2019. (Cité en page 24)

[KAM20] Boulanouar Khedidja, Hadjali Allel, and Lagha Mohand. Data summarization for sensor data management : Towards computational-intelligence-based approaches. *International Journal of Computing and Digital Systems*, 9(5) :825–833, 2020. (Cité en page 53)

[KB05] Andrew Kusiak and Alex Burns. Mining temporal data : A coal-fired boiler case study. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 953–958. Springer, 2005. (Cité en page 62)

[KCC+03] Sailesh Krishnamurthy, Sirish Chandrasekaran, Owen Cooper, Amol Deshpande, Michael J. Franklin, Joseph M. Hellerstein, Wei Hong, Samuel Madden, Frederick

Reiss, and Mehul A. Shah. Telegraphcq : An architectural status report. *IEEE Data Eng. Bull.*, 26(1) :11–18, 2003. (Cité en page 17)

[KCHP01] Eamonn Keogh, Selina Chu, David Hart, and M Pazani. An online algorithm for segmenting time series. In *Proceedings 2001 IEEE international conference on data mining*, pages 289–296. IEEE, 2001. (Cité en page 2), (Cité en page 63), (Cité en page 66)

[KCHP04] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. Segmenting time series : A survey and novel approach. In *Data mining in time series databases*, pages 1–21. World Scientific, 2004. (Cité en page 2), (Cité en page 64)

[KCS06] Abdullah Konak, David W Coit, and Alice E Smith. Multi-objective optimization using genetic algorithms : A tutorial. *Reliability Engineering & System Safety*, 91(9) :992–1007, 2006. (Cité en page 85)

[Koo81] Robert Philip Kooi. The optimization of queries in relational databases. 1981. (Cité en page 32)

[KR09] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data : an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009. (Cité en page 26), (Cité en page 28)

[KW09] Janusz Kacprzyk and Anna Wilbik. Towards an efficient generation of linguistic summaries of time series using a degree of focus. In *NAFIPS 2009-2009 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 1–6. IEEE, 2009. (Cité en page 39), (Cité en page 69)

[KWZ06a] Janusz Kacprzyk, Anna Wilbik, and Slawomir Zadrozny. Linguistic summarization of trends : a fuzzy logic based approach. In *Proceedings of the 11th International Conference Information Processing and Management of Uncertainty in Knowledge-based Systems*, pages 2166–2172, 2006. (Cité en page 2), (Cité en page 39), (Cité en page 45), (Cité en page 67), (Cité en page 69)

[KWZ06b] Janusz Kacprzyk, Anna Wilbik, and Slawomir Zadrozny. Using a genetic algorithm to derive a linguistic summary of trends in numerical time series. In *2006 International Symposium on Evolving Fuzzy Systems*, pages 137–142. IEEE, 2006. (Cité en page 2), (Cité en page 79)

[KWZ08] Janusz Kacprzyk, Anna Wilbik, and S Zadrożny. Linguistic summarization of time series using a fuzzy quantifier driven aggregation. *Fuzzy Sets and Systems*, 159(12) :1485–1499, 2008. (Cité en page 39), (Cité en page 40), (Cité en page 63), (Cité en page 69)

[KWZ10] Janusz Kacprzyk, Anna Wilbik, and Sł awomir Zadrożny. An approach to the linguistic summarization of time series using a fuzzy quantifier driven aggregation. *International Journal of Intelligent Systems*, 25(5) :411–439, 2010. (Cité en page 2), (Cité en page 39), (Cité en page 69)

[KY01] Janusz Kacprzyk and Ronald R Yager. Linguistic summaries of data using fuzzy logic. *International Journal of General System*, 30(2) :133–154, 2001. (Cité en page 47)

[KYD03] Vineet Khare, Xin Yao, and Kalyanmoy Deb. Performance scaling of multi-objective evolutionary algorithms. In *International conference on evolutionary multi-criterion optimization*, pages 376–390. Springer, 2003. (Cité en page 85)

[KYZ00] Janusz Kacprzyk, Ronald R Yager, and S Zadrożny. A fuzzy logic based approach to linguistic summaries of databases. *International Journal of Applied Mathematics and Computer Science*, 10(4) :813–834, 2000. (Cité en page 1), (Cité en page 39), (Cité en page 44)

[KYZ02] Janusz Kacprzyk, Ronald R Yager, and Slawomir Zadrozny. Fuzzy linguistic summaries of databases for an efficient business data analysis and decision support. In *Knowledge discovery for business information systems*, pages 129–152. Springer, 2002. (Cité en page 1), (Cité en page 39), (Cité en page 47)

[KZ05] Janusz Kacprzyk and Sławomir Zadrożny. Linguistic database summaries and their protoforms : towards natural language based knowledge discovery tools. *Information Sciences*, 173(4) :281–304, 2005. (Cité en page 40), (Cité en page 41), (Cité en page 45), (Cité en page 46)

[KZ12] Janusz Kacprzyk and Slawomir Zadrozny. Protoforms of linguistic database summaries as a human consistent tool for using natural language in data mining. In *Software and Intelligent Sciences : New Transdisciplinary Findings*, pages 157–168. IGI Global, 2012. (Cité en page 42), (Cité en page 44)

[Lié95] Ludovic Liétard. *Contribution à l'interrogation flexible de bases de données : Etude des propositions quantifiées floues*. PhD thesis, Rennes 1, 1995. (Cité en page 42), (Cité en page 44)

[LMBM16] Marie-Jeanne Lesot, Gilles Moyse, and Bernadette Bouchon-Meunier. Interpretability of fuzzy linguistic summaries. *Fuzzy Sets and Systems*, 292 :307–317, 2016. (Cité en page 40)

[LMS14] Miodrag Lovrić, Marina Milanović, and Milan Stamenković. Algoritmic methods for segmentation of time series : An overview. *Journal of Contemporary Economic and Business Issues*, 1(1) :31–53, 2014. (Cité en page 66)

[M+67] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967. (Cité en page 26)

[Maz07] M Guy Mazaré. *Gestion à grande échelle de données de capteurs hétérogènes*. PhD thesis, Institut National Polytechnique de Grenoble, 2007. (Cité en page 24)

[MD88] M Muralikrishna and David J DeWitt. Equi-depth multidimensional histograms. In *Proceedings of the 1988 ACM SIGMOD international conference on Management of data*, pages 28–36, 1988. (Cité en page 32)

[ML16] Gilles Moyse and Marie-Jeanne Lesot. Linguistic summaries of locally periodic time series. *Fuzzy Sets and Systems*, 285 :94–117, 2016. (Cité en page 41)

[MLBM13] Gilles Moyse, Marie-Jeanne Lesot, and Bernadette Bouchon-Meunier. Linguistic summaries for periodicity detection based on mathematical morphology. In *2013 IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, pages 106–113. IEEE, 2013. (Cité en page 41)

[Moy16] Gilles Moyse. *Résumés linguistiques de données numériques : interprétabilité et périodicité de séries*. PhD thesis, 2016. (Cité en page 46), (Cité en page 61)

[MR19] Alessandro Margara and Tilmann Rabl. Definition of data streams., 2019. (Cité en page 15)

[NH02] Raymond T. Ng and Jiawei Han. Clarans : A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, 14(5) :1003–1016, 2002. (Cité en page 28)

[NP15] Vilém Novák and Irina Perfilieva. Time series mining by fuzzy natural logic and f-transform. In *2015 48th Hawaii International Conference on System Sciences*, pages 1493–1502. IEEE, 2015. (Cité en page 2), (Cité en page 39)

[OF06] Umit Y Ogras and Hakan Ferhatosmanoglu. Online summarization of dynamic time series data. *The VLDB Journal*, 15(1) :84–98, 2006. (Cité en page 62)

[Par97] Vilfredo Pareto. The new theories of economics. *Journal of political economy*, 5(4) :485–502, 1897. (Cité en page 2), (Cité en page 85), (Cité en page 86)

[PLS11] M Parimala, Daphne Lopez, and NC Senthilkumar. A survey on density based clustering algorithms for mining large spatial databases. *International Journal of Advanced Science and Technology*, 31(1) :59–66, 2011. (Cité en page 29)

[Ras01] Guillaume Raschia. *SAINTETIQ : une approche floue pour la génération de résumés à partir de bases de données relationnelles*. PhD thesis, Nantes, 2001. (Cité en page 41)

[RG99] Raghu Ramakrishnan and Johannes Gehrke. Introduction to database systems. *Database Management Systems*, 1999. (Cité en page 13), (Cité en page 14)

[RM02] Guillaume Raschia and Noureddine Mouaddib. Saintetiq : a fuzzy set-based approach to database summarization. *Fuzzy sets and systems*, 129(2) :137–162, 2002. (Cité en page 41)

[SAWH14] Ali Seyed Shirkhorshidi, Saeed Aghabozorgi, Teh Ying Wah, and Tutut Herawan. Big data clustering : a review. In *International conference on computational science and its applications*, pages 707–720. Springer, 2014. (Cité en page 26)

[SÇZ05] Michael Stonebraker, Uğur Çetintemel, and Stan Zdonik. The 8 requirements of real-time stream processing. *ACM Sigmod Record*, 34(4) :42–47, 2005. (Cité en page 17)

[SD94] Nidamarthi Srinivas and Kalyanmoy Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3) :221–248, 1994. (Cité en page 85)

[SDDS96] Eric J Stollnitz, Tony D DeRose, Anthony D DeRose, and David H Salesin. *Wavelets for computer graphics : theory and applications*. Morgan Kaufmann, 1996. (Cité en page 32)

[SG80] Jack Sklansky and Victor Gonzalez. Fast polygonal approximation of digitized curves. *Pattern recognition*, 12(5) :327–331, 1980. (Cité en page 2), (Cité en page 63)

[SS08] A Slowik and J Slowik. Multi-objective optimization of surface grinding process with the use of evolutionary algorithm with remembered pareto set. *The International Journal of Advanced Manufacturing Technology*, 37(7-8) :657–669, 2008. (Cité en page 85)

[TC87] Masaki Togai and Stephen Chiu. A fuzzy logic chip and a fuzzy inference accelerator for real-time approximate reasoning. In *Proc. of 17th. International Symposium of Multiple Valued Logic*, pages 25–29, 1987. (Cité en page 9)

[vdHT09] Albert van der Heide and Gracián Triviño. Automatically generated linguistic summaries of energy consumption data. In *2009 Ninth international conference on intelligent systems design and applications*, pages 553–559. IEEE, 2009. (Cité en page 63)

[Vit85] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1) :37–57, 1985. (Cité en page 31)

[VM02] Jeroen K Vermunt and Jay Magidson. Latent class cluster analysis. *Applied latent class analysis*, 11(89-106) :60, 2002. (Cité en page 26)

[Wei05] Gary M Weiss. Data mining in telecommunications. In *Data Mining and Knowledge Discovery Handbook*, pages 1189–1201. Springer, 2005. (Cité en page 24)

[WKA11] Anna Wilbik, James M Keller, and Gregory Lynn Alexander. Linguistic summarization of sensor data for eldercare. In *2011 IEEE International Conference on Systems, Man, and Cybernetics*, pages 2595–2599. IEEE, 2011. (Cité en page 24)

[XEKS98] Xiaowei Xu, Martin Ester, H-P Kriegel, and Jörg Sander. A distribution-based clustering algorithm for mining in large spatial databases. In *Proceedings 14th International Conference on Data Engineering*, pages 324–331. IEEE, 1998. (Cité en page 29)

[Yag82] Ronald R Yager. A new approach to the summarization of data. *Information Sciences*, 28(1) :69–86, 1982. (Cité en page 1), (Cité en page 39), (Cité en page 40), (Cité en page 41), (Cité en page 45), (Cité en page 46)

[Yag88] Ronald R Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on systems, Man, and Cybernetics*, 18(1) :183–190, 1988. (Cité en page 1), (Cité en page 43), (Cité en page 47)

[Yag97] Ronald R Yager. A note on a fuzzy measure of typicality. *International journal of intelligent systems*, 12(3) :233–249, 1997. (Cité en page 48)

[Yag21] Ronald R Yager. An introduction to linguistic summaries. In *Fuzzy Approaches for Soft Computing and Approximate Reasoning : Theories and Applications*, pages 151–162. Springer, 2021. (Cité en page 39)

[Zad65] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3) :338–353, 1965. (Cité en page 1), (Cité en page 9)

[Zad83] Lotfi A Zadeh. A computational approach to fuzzy quantifiers in natural languages. In *Computational linguistics*, pages 149–184. Elsevier, 1983. (Cité en page 42), (Cité en page 43), (Cité en page 53), (Cité en page 79), (Cité en page 90)

[Zad96] Lotfi A Zadeh. A note on prototype theory and fuzzy sets. In *Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems : Selected Papers by Lotfi A Zadeh*, pages 587–593. World Scientific, 1996. (Cité en page 48)

[ZRH+16] Hao Zhuang, Rameez Rahman, Xia Hu, Tian Guo, Pan Hui, and Karl Aberer. Data summarization with social contexts. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 397–406, 2016. (Cité en page 24)

[ZRL96] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch : an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2) :103–114, 1996. (Cité en page 26)

[ZT99] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms : a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4) :257–271, 1999. (Cité en page 85)

Annexe A

Acronymes

ACO *Ant colony optimisation*
ADSB *Automatic Dependent Surveillance-Broadcast*
ALT *Altitude*
BIRCH *Balanced Iterative Reducing and Clustering using Hierarchies*
CF Tree *Clustering Feature Tree*
CLARA *Clustering LARge Applications*
CLARANS *Clustering Large Applications based upon Randomized Search*
CURE *Clustering Using REpresentatives*
DBCLASD *Distribution Based Clustering of Large Spatial Databases*
DBSCAN *Density-Based Spatial Clustering of Applications with Noise*
DCL *Data Control Language*
DDL *Data Definition Language*
DML *Data manipulation Language*
GA *Genetic Algorithm*
GPS *Global Position System*
GS *Ground Speed*
IATA *International Air Transport Association*
ICAO *International Civil Aviation Organization*
IBM *International Bussiness Machine*
IoT *Internet of Things*
MOGA *Multi Objective Genetic Algorithm*
NPGA *Niched Pareto Genetic Algorithm*
NSGA *Non dominated Sorting Genetic Algorithm*
NSGA II *Non dominated Sorting Genetic Algorithm II*
OWA *Ordered weighted averaging*
PAM *Partitioning Around Medoid*
PESA *Pareto Envelope-based Selection Algorithm*
RLF *Résumé Linguistique Flou*
ROKE *RObust Clustering using linKs*
SGBD *Système de Gestion de Base de Données*
SGFD *Système de Gestion du flux de Données*

SPEA *Strength Pareto Evolutionary Algorithm*

SQL *Structured Query Language*

t.m.f *triangular membership function*

VEGA *Vector Evaluated Genetic Algorithm*

Résumé

Dans de nombreuses applications modernes (issues des domaines scientifiques, du transport, de l'énergie, de l'environnement, etc.), les données représentent une matière première et des produits à forte valeur ajoutée pour la prise de décision. Les déluges des données générées par ces applications font que certains paradigmes classiques de traitement ne répondent plus, d'une manière complètement pertinente, à certaines situations de prise de décision. Ainsi, un regain d'intérêt (des chercheurs) pour certaines approches de traitement de données est observé. Récemment, l'approche utilisant le principe de réduction de données a suscité un réel engouement. Le principe de cette approche est de réduire le volume de données en entrée du processus de traitement. Cette approche permet, notamment, de garantir une exploitation de données moins coûteuse (en termes de calcul et de temps) et d'obtenir des réponses approximatives ou juste certaines tendances des données. Ce qui est, particulièrement, désirable dans des contextes où une réponse approximative est plutôt souhaitable et apporte suffisamment d'informations pour être acceptable.

Il existe de nombreuses techniques de réduction du volume des données, dont les structures de résumé de données (ou synopsis) font partie. Dans le cadre de cette thèse, nous nous sommes intéressés à une famille de structures de résumé issues du domaine de l'intelligence computationnelle. Ces structures (comme les quantificateurs mathématiques non classiques, la typicité, les labels/motifs linguistiques, etc.) se distinguent par deux particularités : (i) l'intelligibilité des résumés construits et ; (ii) la génération des résumés qui décrivent les données à des niveaux d'abstraction différents. Les données cibles sont des données réelles provenant de multi-capteurs concernant (i) des vols d'aéronefs collectées dans le cadre du projet ADSB et (ii) des Smart Cities dans le contexte du projet neOCampus.

Dans la première contribution de la thèse, nous avons proposé une méthode d'extraction de résumé de données en utilisant (i) les quantificateurs non classiques et (ii) la notion de typicité. Des mesures pour caractériser les propriétés des résumés construits (véracité, représentativité, imprécision, etc.) sont également définies sachant que ces propriétés évoluent d'une manière contradictoire. Puis, nous avons analysé les différentes manières d'exploiter chacun des résumés à des fins de prise de décision.

Dans un second temps, nous nous sommes intéressés à l'étude de certaines caractéristiques des tendances des données (issues de capteurs ou de séries temporelles) comme le changement dynamique, la durée et la variabilité. Cette étude nous a permis de sélectionner le meilleur résumé parmi les résumés construits sur la base des quantificateurs non classiques. Cette sélection est formalisé sous forme d'un problème d'optimisation multi-objectif. L'approche de résolution proposée utilise un algorithme génétique convenablement choisi. Enfin, une série d'expérimentations ont été menées, sur des données réelles, pour valider et comparer toutes nos propositions.

Mots-clés : Données capteurs, Exploitation et analyse, Réduction de données, Résumés linguistiques, Quantificateurs mathématiques, Typicité, Algorithmes génétiques.

Abstract

In many modern applications (stemming from scientific fields, transport, energy, environment, etc.), data represent a raw material and a product with high added value for decision-making. The deluge of data generated by these applications makes some classic processing paradigms no longer completely relevant way to some decision-making situations. Thus, a renewed interest (of researchers) for some data processing approaches is observed. Recently, the approach using the principle of data reduction has aroused a real enthusiasm. The principle of this approach is to reduce the amount of data in imput of the processing process. This approach allows a less expensive data exploitation (in terms of calculation and time) and to obtain approximate answers or just some trends of the target data. This is particularly desirable in contexts where an approximate answer is rather desirable and provides enough information to be acceptable.

There are many techniques for reducing data volume, of which data summary structures(or synopsis) are part of these techniques. As part of this thesis, we are interested in a family of summary structures borrowed from the field of computational intelligence. These structures (such as non-classical mathematical quantifiers, typicity, labels / linguistic patterns, etc.) have two interesting features : (i) the intelligibility of the summaries constructed and ; (ii) the generation of summaries that describe the data at different levels of abstraction. The target data are real data coming from multi-sensors in (i) aircraft flights collected within the framework of the ADSB project and (ii) Smart Cities within the context of the neOCampus project.

As first contribution of the thesis, we proposed a method for summary extracting using (i) non-classical quantifiers and (ii) the notion of typicity. Measures to characterize the properties of the constructed summaries (veracity, representativeness, imprecision, etc.) are also defined knowing that these properties evolve in a contradictory way. Then, we analyzed the different ways to use each of the summaries for the decision-making purpose.

Secondly, we were interested in the study of certain characteristics of data trends (in sensor data or time series) such as dynamic change, duration and variability. This study allowed us to select the best summary among the summaries constructed using the non-classical quantifiers. This selection is formalized as a multi-objective optimization problem. The proposed resolution approach uses a genetic algorithm suitably chosen. Finally, a set of experiments were carried out on real data to validate and compare all our proposal.

Keywords : Sensor data, Exploitation and analysis, Data reduction, Linguistic summaries, Mathematical quantifiers, Typicity, Genetic algorithms.

Secteur de recherche : Informatique et applications
Secteur de recherche : Avionique

Laboratoire d'Informatique et d'Automatique pour les Systèmes
École Nationale Supérieure de Mécanique et d'Aérotechnique
Laboratoire des Sciences Aéronautiques
Institut d'Aéronautique et des Études Spatiales