Evidential clustering for trajectory analysis

Armel Soubeiga

Université Clermont Auvergne

Ecole Doctorale Sciences pour l'ingénieur
SPI UCA n°70

Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes
LIMOS - UMR 6168

Par Armel SOUBEIGA

Evidential Clustering for Trajectory Analysis

Classification automatique évidentielle pour l'analyse des trajectoires

Thèse de doctorat en Informatique

— spécialité Intelligence Artificielle —

Sous la direction de Jonas KOKO,

Et des Co-directeurs Violaine ANTOINE et Sylvain MORENO

Présentée et soutenue publiquement le 7 Avril 2025.

Devant un jury composé de :

Vincent BARRA, Prof.	Université Clermont Auvergne	Président
Sébastien DESTERCKE, DR-CNRS	Université de Technologie de Compiègne	Rapporteur
Lina F. SOUALMIA, Prof.	Université Rouen Normandie	Rapporteuse
Engelbert MEPHU NGUIFO, Prof.	Université Clermont Auvergne	Examinateur
Nicolas SUTTON-CHARANI, MCF	IMT Mines Alès	Examinateur
Thomas GUYET, CR-HDR	Inria Lyon	Examinateur
Violaine ANTOINE, MCF-HDR	Université Clermont Auvergne	Co-Directrice de thèse
Jonas Koko, MCF-HDR	Université Clermont Auvergne	Directeur de thèse

La recherche doit avant tout être un jeu et un plaisir, car toute certitude est par essence contradictoire avec la philosophie de la recherche.

Pierre Joliot

Acknowledgements

First off, I want to thank my thesis supervisors, Dr. Antoine Violaine, Prof. Sylvain Moreno, and Dr. Jonas Koko. Thanks for guiding, supporting, and backing me up with your scientific culture and rigor, your understanding, and your dedication. My deep gratitude to Dr. Violaine — working with you was a privilege, and I learned so much.

I sincerely thank Dr. Thomas Guyet for his valuable collaboration and mentorship, which greatly enriched my work. Many thanks to Dr. Issam Falih for his support and insightful advice.

I extend my heartfelt thanks to the members of the jury, to the reviewers, Professors Lina Soualmia and Sébastien Destercke, for their significant contribution to the evaluation of my thesis. To all other jury members, the examiners, Drs. Nicolas Sutton-Charani and Thomas Guyet, as well as Professors Engelbert Mephu Nguifo and Vincent Barra.

I would also like to thank my collaborators from the eDOL project, especially the members of the Analgesia Institute, Nicolas Kerckhove and Alice Corteval, as well as those from LIMOS, including Bastien Doreau, for their contributions that enriched my work and broadened my perspective in this field.

Thank you to the LIMOS laboratory for hosting me during my thesis. Thanks to my doctoral colleagues and friends at LIMOS, whose presence and sharing of ideas created a stimulating and dynamic environment. Thanks to the administrative staff of LIMOS and especially to Beatrice Bourdieu.

Finally, my heartfelt thanks to my family for their support and encouragement. I am particularly grateful to my wife Ingrid for her precious support, and to our princess, Manuela, whose birth during these years of thesis has been more challenging and motivating.

Abstract

Clustering is a fundamental task in data analysis, aimed at grouping objects into clusters based on similarity. Traditional methods such as the k-means algorithm have dominated this field for decades, providing a simple and efficient approach to partitioning data. However, hard clustering techniques like k-means assign each object to a single cluster, which can be limiting in applications where uncertainty or overlap between clusters must be accounted. Soft clustering methods address this limitation by allowing partial memberships to multiple clusters. Among soft clustering methods, evidential clustering, based on the concept of the Dempster-Shafer theory, has received a lot of attention for its ability to deal with this problem.
The objective of the research is to study and propose social trajectory clustering methods that provide an effective representation of uncertainty and imprecision. Trajectory is the movement or event dynamics of an object, like animals, humans, vehicles and natural phenomena. Trajectory analysis and clustering are essential to learn the dynamic or movement patterns of these objects.

In this thesis, we contribute to the research objective for evidential clustering of trajectory analysis. We present a comprehensive review and comparative analysis of clustering methods adapted to multidimensional sequential trajectories (MST). We examined various approaches, including feature-based, raw-data-based and model-based methods, focusing on their application to categorical or discrete longitudinal or time series data. We also explore new frameworks, including the use of traditional evidential c-means (ECM) algorithms combined with feature extraction and unsupervised feature selection methods. We proposed new clustering algorithms based on belief functions for MST clustering or, more generally, adapted to clustering of multidimensional trajectory data, such as multi-view evidential c-medoids (MECMdd) and soft evidential c-means (Soft-ECM). The effectiveness of the proposed algorithms is estimated on different synthetics and real datasets. Experiments show that our proposed algorithms effectively improve the traditional evidential partitioning methods and detect imbalanced or arbitrary trjectories clusters, and characterizes the uncertainty and imprecision between these clusters.

Upon completion of these works, we have applied our frameworks and algorithms to a real-life case study. We studied the clustering and the identification of typologies of chronic pain trajectories through the application of the different contributions to the eDOL project data. Afterward, we contributed to open-source with the implementation of the evclust Python library providing several state-of-the-art evidential clustering algorithms, in a user-friendly environment.

Keywords: Trajectory clustering – Evidential clustering – Multi-view relational clustering – Categorical and discrete time series – Uncertainty and imprecision trajectory – Belief functions.

Résumé

La classification automatique est une tâche fondamentale en analyse de données, visant à regrouper des objets en classes en fonction de leur similarité. Les méthodes traditionnelles, telles que l'algorithme k-means, ont longtemps constitué une référence dans ce domaine, en offrant une approche simple et efficace pour partitionner les données. Cependant, les techniques de regroupement dur, comme les k-means, attribuent chaque objet à une seule classe, ce qui peut être limitant dans les applications où il est nécessaire de prendre en compte l'incertitude ou le chevauchement entre les classes. Les méthodes de regroupement souple répondent à cette limitation en permettant des appartenances partielles à plusieurs classes. Parmi ces méthodes, le regroupement basé sur les fonctions de croyance, fondé sur la théorie de Dempster-Shafer, a suscité beaucoup d'attention pour sa capacité à traiter ce type de problème.

L'objectif de cette recherche est de proposer des méthodes de regroupement des trajectoires en science sociales et permettant une représentation efficace de l'incertitude et de l'imprécision. Une trajectoire correspondant au mouvement ou à la dynamique des événements d'un objet, comme des animaux, des humains, des véhicules ou des phénomènes naturels. L'analyse et le regroupement des trajectoires sont essentiels pour comprendre les modèles dynamiques ou de mouvement de ces objets.

Dans cette thèse, nous contribuons à cet objectif en développant des méthodes de regroupement basées sur les fonctions de croyance pour l'analyse des trajectoires. Nous présentons une revue complète et une analyse comparative des méthodes de regroupement adaptées aux trajectoires séquentielles multidimensionnelles (TSM). Nous avons examiné diverses approches, y compris les méthodes basées sur des caractéristiques, sur des données brutes et sur des modèles, en mettant l'accent sur leur application aux données longitudinales ou aux séries temporelles catégoriques ou discrètes. Nous explorons également de nouveaux cadres méthodologiques, notamment l'utilisation de l'algorithme c-moyennes évidentielle (ECM) traditionnel combiné à des méthodes d'extraction et de sélection non supervisées des caractéristiques. Nous avons proposé de nouveaux algorithmes de regroupement basés sur les fonctions de croyance pour le clustering des TSM ou, plus généralement, adaptés au clustering des données de trajectoires multidimensionnelles, tels que le multi-view évidentielle c-medoids (MECMdd) et le soft évidentielle c-means (Soft-ECM). L'efficacité des algorithmes proposés a été évaluée sur différents ensembles de données synthétiques et réels. Les expérimentations montrent que nos algorithmes améliorent efficacement les méthodes traditionnelles de partitionnement basé sur les fonctions de croyance, détectent des classes de trajectoires déséquilibrées ou arbitraires, et caractérisent l'incertitude et l'imprécision entre ces classes.

A l'issue de ces travaux, nous avons appliqué nos approches et algorithmes à une étude de cas réelle. Nous avons étudié le regroupement et l'identification des typologies des trajectoires

de soins de la douleur chronique en appliquant les différentes contributions aux données du projet eDOL. Enfin, nous avons contribué à l'open source en développant la bibliothèque Python evclust, qui fournit plusieurs algorithmes de classification automatique basés sur les fonctions de croyance.

Mots clés: Classification automatique de trajectoires – Classification automatique évidentielle – Regroupement relationnel multi-vues – Séries temporelles catégorielles et discrètes – Incertitude et imprécision des trajectoires – Fonctions de croyance.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations and Notations

In the following, a list as exhaustive as possible of basic abbreviations and notations used in this thesis:

Basic & general.

Notation	Description
$\Omega = \{\omega_1, \ldots, \omega_c\}$	c set of singleton solutions with grouped data.
$\mathbf{X} = (\mathbf{x}_i)$	The set of objects of dataset when object-based partition.
$\mathbf{E} = (\mathbf{e}_i)$	The set of objects of dataset when Relational-based partition.
n	The number of objects in **X** or **E**.
d_{ij}	Euclidean distance between objects x_i and x_j.
A_j	Singleton cluster or meta-cluster.
$\|A_j\|$	The cardinality of A_j, i.e. number of elements in A.
$\mathbf{M}$	Credal partition matrix.
m_{ij}	Belief mass of x_i to A_j.
$\mathbf{V}$	Cluster prototypes (barycenters or medoids) matrix.
$v_j, \overline{\boldsymbol{v}}_j$	Barycenters of singleton and meta-cluster.
α	Penalization degree.
β	Weighting exponent.
δ	Outlier threshold of $\emptyset$.
$\mathcal{M}_{hp}$	Hard partition.
$\mathcal{M}_{sp}$	Soft partition.
$U(\mathbf{f})$	utility of feature F.
$U(\mathbf{S})$	utility of feature similarity matrix S.
$SP_{score}(F_i)$	Spectral relevance of feature F_i.
$LS_{score}(F_i)$	Laplacian Score of F_i.
$\mathscr{G} = (G_1, \ldots, G_K)$	k groups or k hard partitions.
2^Ω	The power set of the frame of discernment Ω.
$Bel(.)$	The belief function.
$Pl(.)$	The plausibility function.
$BetP(.)$	The pignistic probability function.
$p_m(.)$	The plausibility probability function.
$A^*(m)$	The largest mass function.
$\oplus$	The resulting operation of Dempster's rule of combination.
$N^\star$, NS	The standardized non-specificity.

Related to MECMdd.

Notation	Description
$\boldsymbol{\mathcal{R}} = [\tau(e_i, e_j)]$	Relational or proximity dataset.
p	The number of pairwise dissimilarity matrix (view).
τ_{ij}	Dissimilarity between objects e_i and e_j.
τ_{ijl}	Dissimilarity between objects e_i and cluster A_j in matrix l.
$v_k^{\Omega}, v_j^{2^{\Omega}}$	Medoids of singleton cluster and meta-cluster.
$\boldsymbol{\Lambda}, \boldsymbol{\lambda}$	View weights matrix or vector.
λ_{jl}	View of l^{th} matrix in cluster A_j.
s	Parameter of view weight.
δ_l	Outlier threshold of $\emptyset$ matrix l.
η	Penalization of outliers from the possible medoids.
γ	Balance of the contribution for imprecise classes.

Related to SoftECM.

Notation	Description
$\mathbf{v}_{\{\omega_k\}}, \mathbf{v}_A$	Centroids of singleton ($k \in [c]$) and meta-cluster ($A \subseteq \Omega$).
$d(\mathbf{x}_i, \mathbf{v}_A)$	Semi-metric dissimilarity measure between $\mathbf{x}_i$ and $\mathbf{v}_A$.
$d(\mathbf{v}_{\{\omega_k\}}, \mathbf{x})$	Semi-metric dissimilarity measure between $\mathbf{v}_{\{\omega_k\}}$ and $\mathbf{x}$.
$d(\mathbf{v}_{\{\omega_k\}}, \mathbf{v}_A)$	Semi-metric dissimilarity measure between $\mathbf{v}_{\{\omega_k\}}$ and $\mathbf{v}_A$.
λ	The weighting factor of the distances among singleton cluste and meta-cluster centers.

Acronyms & Nomenclature.

Notation	Description
ASW	Average silhouette width.
Cclust	Combination of clusters.
Ccost	Combination of costs.
Cdist	Combination of distance.
Cstat	Combination of states.
Codist	Distance collaborative clustering.
CRI	Credal rand index.
ECM	Evidential C-Means.
GIMSA	Globally Interdependent Multiple Sequence Analysis.
MECMdd-GWL	Multi-view evidential c-Medoid global relevance weight.
MECMdd-RWL	Multi-view evidential c-Medoid local relevance weight.
MST	Multidimensionnal sequential trajectory.
RI	Rand Index.
Soft-ECM	Soft Evidential C-Means.

Part I

Context

Chapter 1

Introduction

Abstract – In this chapter, we first state the motivation for the research of this thesis, i.e., representation of uncertainty and imprecision for trajectoires clustering with the theory of belief functions. After that, the main research objectives of this thesis are discussed. Then, we highlight the contributions of this thesis. Finally, the structure of this thesis is presented in detail.

Contents

1.1 Motivations

Trajectories are essential data in numerous fields, representing the dynamic evolution of individuals, objects or systems in space, time or sequential events [2, 3]. Trajectories provide a structured way to study a variety of phenomena, such as the paths of satellites, vehicle movements, cyclone paths, or web clickstream. They are also essential for understanding trajectories from consumption behaviour, health pathways, or broader life-course [4]. These different applications show that trajectory data are omnipresent and useful for understanding how objects evolve and change, whether over time or through sequences of events [5]. In the social sciences, trajectory analysis has become an indispensable tool for exploring how individuals and groups evolve through different states over time. We are particularly interested in trajectories that describe ordered changes through distinct categorical, or discrete states, often referred to sequential trajectories. For example, researchers study sequences of educational steps, career progression or family dynamics to understand the interaction between personal choices, structural opportunities and social influences [6]. These analyses, often based on longitudinal, panel or time series data, highlight the orderly progression of events and the factors that shape these pathways [7, 8].

In healthcare, sequential trajectories—often referred to as medical trajectories or care pathways—represent the sequence of events experienced by patients over time, such as diagnoses, treatments, hospitalizations, or follow-ups [9, 10, 11]. The study of such trajectories is crucial for understanding patient pathways and optimizing healthcare delivery. Medical trajectories are inherently characterized by categorical or discrete data, representing transitions between well-defined care states [12, 13].

Despite their importance, analyzing sequential trajectories in healthcare presents unique challenges, particularly due to the uncertainty and imprecision inherent in the data. Health trajectories are often influenced by incomplete medical records, variability across individuals and contexts, and the inherent complexity of human behavior and healthcare systems [14, 15, 16]. Traditional clustering methods, which rely on deterministic or probabilistic assumptions, struggle to adequately model this uncertainty. For instance, a patient's pathway can include overlapping or ambiguous states—such as receiving outpatient care and undergoing rehabilitation simultaneously—or may not follow predefined sequences, featuring concurrent diagnoses, delayed treatments, or even outliers that deviate from established patterns [14].

Evidential clustering, based on the theory of belief functions, offers a robust alternative for tackling these challenges [17, 18]. Unlike conventional clustering approaches, evidential clustering models partial knowledge by allocating degrees of belief to multiple clusters simultaneously, including the possibility of imprecise or intermediate states. The concept of credal partitions, allows for a nuanced representation of an object's association with clusters, accommodating partial or conflicting information [19]. This flexibility makes evidential clustering particularly well-suited for analyzing healthcare trajectories, where uncertainty is not an exception but a defining characteristic.

The main motivation for this thesis is to use evidential clustering to study healthcare trajectories more effectively. These trajectories often involve categorical data, uncertainty,

and imprecision. By creating new clustering methods that handle these challenges, we aim to gain deeper insights into patient pathways, helping both researchers and healthcare practitioners make better decisions in real-world settings for personalized medicine.

1.2 Research objectives

In this thesis, we aim to address several major research problems related to the clustering of healthcare trajectories and social science trajectories in general. Indeed, trajectories are complex data in machine learning, and even more so when they are sequential trajectories characterized by categorical or discrete event sequences based on longitudinal or time series data. Moreover, in the health and social sciences, trajectory data tend to be subjective, often ambiguous and imprecise due to their dependence on human factors.

To achieve this goal, we first review the origin and evolution of clustering techniques applied to medical and social trajectories. A critical understanding of these methods, their applicability to multidimensional trajectories, and their comparative effectiveness is essential to establish a robust foundation for advancing trajectory clustering approaches. Next, we explore the uses and adaptation of conventional clustering techniques, intially designed for cross-sectional data, to the context of evidential trajectory clustering. These adaptations aim to take into account temporal dynamics and uncertainty inherent in sequential trajectories, particularly in applications like chronic pain trajectories. Finally, this thesis addresses the challenge of handling uncertainty in clustering multidimensional trajectories by proposing evidential clustering algorithms to model these complex data.

1.3 Contributions

In order to address the stated research problems identified in this thesis, we have proposed several contributions in the field of evidential clustering for health trajectory analysis. The following subsections present a concise overview and a summary of these contributions, followed by publications from our works related of these contributions.

Thesis contributions

The first contribution consists of a comprehensive review and comparative analysis of clustering methods adapted to multidimensional sequential trajectories [b],[c], [e], [k]. We examined various approaches, including feature-based, raw-data-based and model-based methods, focusing on their application to categorical or discrete longitudinal data. This analysis highlighted the strengths and limitations of additive and combinatorial sequential clustering techniques in the literature. In addition, we highlighted the importance of considering uncertainty in trajectory data, which often arises from real-world variability and measurement noise.

We then proposed a feature-extraction-based clustering framework grounded in evidential theory for time series data [a],[c],[d], [i]. We begin by extracting and selecting relevant attributes from chronic pain trajectories, followed by the application of an evidential

c-means algorithm. This framework enhances the interpretability of clustering results and facilitates the use of traditional clustering algorithms.

The third contribution introduces a row-data-based multi-view evidential c-medoids (MECMdd) method designed to handle scenarios where different types of proximity measures for complex data, such multivariate time series trajectories [c], [g], [j]. By assigning local or global weights to each view or each dimension of the series, this algorithm facilitates collaborative learning of multidimensional sequential trajectory data and produces credal partitions to reflect imprecision in cluster assignments, with interpretability of clustering results.

We further developed a row-data-based soft-evidential c-means (Soft-ECM) algorithm aimed at capturing uncertainty in non-Euclidean spaces [c], [f], [h]. This algorithm reformulated classical evidential clustering so that numerical, categorical, and time series data could be used directly. Soft-ECM also overcomes limitations of existing methods by coherently defining imprecise cluster prototypes and facilitates the clustering of the uncertain multidimensional sequential trajectory data.

Finally, we implemented the evclust Python library to provide a user-friendly environment for conducting evidential clustering [l]. This toolbox state-of-the-art evidential clustering algorithms, along with convenient utilities for visualizing and interpreting uncertain cluster memberships.

Thesis publications

Workshops

[a] **Armel Soubeiga**, Violaine Antoine, Alice Corteval, Nicolas Kerckhove, Sylvain Moreno, and Issam Falih. Clustering and Interpretation of Time-series Trajectories of chronic pain using Evidential c-means. *16th International Conference of the ERCIM WG on Computational and Methodological Statistics, CFE-CMStatistics 2023.*

[b] **Armel Soubeiga**, Violaine Antoine, and Sylvain Moreno. Clustering flou de séries temporelles discrètes pour la modélisation des trajectoires de soins de la douleur chronique. *Workshop : IA pour la santé, ECG 2024.*

[c] Bastien Doreau, **Armel Soubeiga**, Violaine Antoine. Retour d'expérience de traitement de données de santé cadre du projet eDOL. *Atelier Traitement Informatique des Données de Santé - TIDS (2024).*

National conferences

[d] **Armel Soubeiga**, Jessem Ettaghouti, Violaine Antoine, Alice Corteval, Nicolas Kerckhove, and Sylvain Moreno. Classification automatique de séries chronologiques de patients souffrant de douleurs chroniques. *Revue des Nouvelles Technologies de l'Information, Extraction et Gestion des Connaissances, RNTI-E-39:651–652, 2023.*

[e] **Armel Soubeiga**, Violaine Antoine and Sylvain Moreno. Clustering multi-relationnel flou des trajectoires de la douleur chronique. *Rencontres francophones sur la Logique Floue et ses Applications - LFA. Pp. 133-140, Brest, France, 2024.* **Best doctoral paper award**.

[f] **Armel Soubeiga**, Thomas Guyet, and Violaine Antoine. Soft-ECM : une extension de l'algorithme EvidentielC-Means pour des données complexes. *Revue des Nouvelles Technologies de l'Information, Extraction et Gestion des Connaissances, RNTI-E-41, pp. 319-326, 2025.*

International conferences

[g] **Armel Soubeiga**, Violaine Antoine and Sylvain Moreno. Multi-View Relational Evidential C-Medoid Clustering with Adaptive Weighted. *2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA). San Diego, CA, USA, 2024, pp. 1-10, doi: 10.1109/DSAA61799.2024.10722777.*

[h] **Armel Soubeiga**, Thomas Guyet, and Violaine Antoine. Soft-ECM: An extension of Evidential C-Means for complex data. *Accepted to FUZZ-IEEE, 2025.*

Journals

[i] **Armel Soubeiga** , Violaine Antoine, Alice Corteval, Nicolas Kerckhove, Sylvain Moreno, Issam Falih and Jules Phalip. Clustering and Interpretation of Time-series Trajectories of chronic pain using Evidential c-means. *Expert Systems With Applications, Vol 260, pp 125369, 2025*

[j] **Armel Soubeiga**, Violaine Antoine, Sylvain Moreno, and Jonas Koko. Evidential clustering with view-weight learning for proximity data.*Preprint submitted to Neurocomputing – under review.*

[k] **Armel Soubeiga**, Violaine Antoine, and Sylvain Moreno. Comparative analysis of multidimensional sequential trajectories clustering methods. *HAL preprint. pp 1-28. 2025. <hal-05061916>.*

Technical report

[l] **Armel Soubeiga**, Violaine Antoine. evclust: Python library for evidential clustering. *arXiv preprint*, arXiv:2502.06587, 2025.

1.4 Structure of the thesis

The manuscript is divided into seven chapters, organized into three main parts. The first part, chapter 1 and chapter 2, provides the context and motivation of the thesis. It introduces some related preliminary knowledge, including the theory of belief functions and other uncertainty theories. The concepts and definitions of multidimensional trajectories applied to the medical domain are also presented. The second part, chapter 3 to chapter 7, presents the core contributions. The chapter 3 reviews and compares clustering methods for multidimensional trajectories. The subsequent chapters introduce, novel evidential clustering algorithms

for multidimensional categorical trajectories, and the development of evclust, a Python framework for evidential clustering. The final part concludes the thesis, summarizing the key contributions in chapter I, and future research directions in chapter II. The chapter III presents an extended summary of the works in French.

Chapter 2

Background

Abstract – In this chapter, we introduce the main background concepts and research context for this thesis. We begin by describing the eDOL project, an innovative platform that studies chronic pain through long-term data collection, data used in the application of our work. Next, we discuss the notion of health trajectories, highlighting the importance of modeling sequential data, such as eDOL data, for clustering chronic pain trajectories. We then provide an overview of clustering methods and algorithms, covering fuzzy, possibilistic, rough, and evidential approaches. Each of these handles uncertainty differently, and we briefly outline their principles, and typical applications. Overall, this chapter provides the foundation for the rest of the thesis, drawing together both the practical context of chronic pain research and the theoretical tools needed for clustering.

Contents

2.1 eDOL project

This section presents the background of the eDOL project, an innovative initiative to study chronic pain through longitudinal follow-up, as described in Section 2.1.1. Section 2.1.2 gives an overview of the eDOL platform, designed to address the project's objectives, highlighting the types of data collected, in Section 2.1.3.1 and presenting the subsets of data that are particularly relevant to our research objectives, in Section 2.1.3.2.

2.1.1 The project context

Chronic pain affects around 30% of the general population, and is one of the main causes of disability, particularly among the older people [20, 21]. In addition to its socio-economic costs, it has a major impact on professional life, with 60% of patients less able to work and 20% losing their jobs [22]. At the same time, existing treatments, which are often old and not very effective, have not progressed much, despite extensive research [23]. This situation illustrates the need for innovation, both in the development of new therapies and in patient characterization to adapt treatments more effectively [24, 25]. One promising way is the use of digital tools, such as smartphone applications and online platforms, offering a new perspective by enabling daily monitoring of pain and treatment under real-life conditions [26]. These tools contribute to a better understanding of patients' daily experiences and facilitate informed decision-making for physicians, by providing a more detailed characterization of patients, integrating biomedical, psychological, and social dimensions [26]. Thus, eHealth technologies emerge as a promising solution to enhance treatment adherence and encourage patients to take an active role in their care management [27].

eDOL is an eHealth tool (combining a smartphone application and a web platform), created by the Institut Analgesia[1] with the aim of establishing an observational cohort of patients with chronic pain in France. This cohort is intended for both healthcare and epidemiological research purposes. Initiated in 2017, the project has gone through several versions (3rd version to day). Efforts are currently underway to link the collected data with social security data via the Health Data Hub (HDH). Figure 2.1 summarizes the progress of the project.

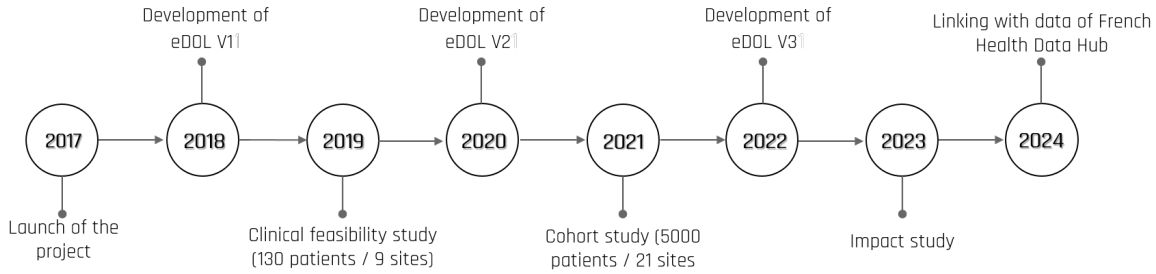


Figure 2.1: Background of the eDOL project.

Participation in the study was offered to patients with non-cancer chronic pain who regularly used a smartphone and were being followed in a specialized pain clinic. Eligible participants

[1] https://www.institut-analgesia.org/

included adults ($\geq$ 18 years old) who could read and understand French and provide consent for the study. During an initial inclusion visit, physicians introduced the study, verified eligibility, explained the eDOL tool, and provided a brief training document. The smartphone application link was provided to patients via email by their physicians, after initial visit. Those agreeing to participate downloaded the eDOL smartphone application, completed questionnaires via the smartphone application, and confirmed consent for the use of their medical data. patient could withdraw his consent at any time by notifying the study sponsor. Participants completed various questionnaires and evaluations via the eDOL application over a two-week period (for an initial characterization of the patient), then repeatedly for those wishing to continue using the application (at weekly, quarterly or six-monthly intervals, depending on the questionnaires).

2.1.2 eDOL application

Data were collected using the eDOL platform, which includes a smartphone application for patients to complete self-questionnaires and perform evaluations for semiological follow-up, and a web interface for physicians, offering them a visualization of data summaries provided by their patients for clinical and therapeutic monitoring.

Figure 2.2: eDOL platform: an easy-to-use interface, a chatbot for day-to-day assistance and clinical and therapeutic monitoring.

The eDOL tool, developed by Analgesia in partnership with BePatient (2020-2023) then Agaetis (from 2024), has benefited from collaboration with LIMOS to create a research database and web interface enabling efficient data processing by researchers [c]. This circuit, illustrated in Figure 2.3, is based on a web application developed with the Django framework (Python), offering a powerful, scalable solution. It allows data storage in a PostgreSQL database, interactive consulting, filtering and downloading, as well as the creation of customized visualizations. Health data, which is particularly sensitive, is subject to strict constraints, requiring it to be hosted by approved health data hosting establishments. In this context, Analgesia, via the CHU (University Hospital Center) has authorized LIMOS to access the data only after anonymization by an authorized partner (in this case Agaetis), which replaces the identification information with a secret UUID [c]. To ensure security and compliance with the RGPD (General Data Protection Regulation), Analgesia, in collaboration with its partners (BePatient, Agaetis and LIMOS), carried out impact analyses via CNIL's PIA

software [2], resulting in documents describing security measures and processes to manage risks.

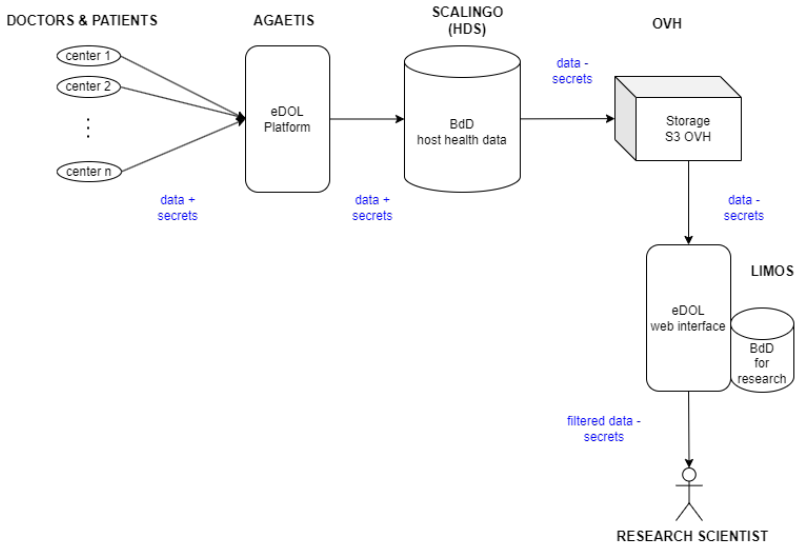


Figure 2.3: eDOL data flow diagram.

2.1.3 eDOL Data

2.1.3.1 Available data

The data collected via the eDOL smartphone application are intended to answer clinical and research objectives. These data can be organized into three main categories: Clinical data, Socio-demographic data and Barometric data. Clinical and socio-demographic data were mainly collected via questionnaires administered once or at regular intervals, while barometric data were collected on a weekly basis. Data collection was carried out either by patient self-reporting in response to questionnaires, or by entries made by the physician during a medical visit (Figure 2.4).

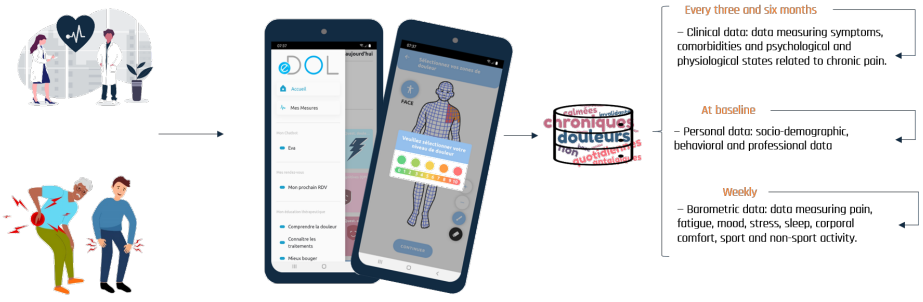


Figure 2.4: Illustration of eDOL data collection approach.

Clinical data [28], is collected by physicians during consultations using specific forms available on the eDOL web platform. This data includes medical information relating to the

[2] https://www.cnil.fr

patient's medical history and psychiatric, diagnostic results and prescribed treatments. This information is collected through periodic self-questionnaires completed every three or six months. Furthermore, physicians can also activate additional specific self-questionnaires, which are completed by patients, to deepen diagnoses or evaluate the effects of treatments. Finally, specific treatment-related information, such as adjustments or side-effects, is recorded by physicians on dedicated forms during consultations.

Socio-demographic and general data [28], are collected at patient inclusion in the study, and are completed both by patients via the smartphone application (personal information) and by physicians via the web platform (inclusion questionnaire). Personal data include information such as age, gender, lifestyle (such as smoking or alcohol consumption), employment status, and elements related to the psychosocial impact of pain. Other data include the patient's pain status, and different dignostics on pain.

Barometric data [28], collected via the smartphone application, are weekly assessments that measure changes in patients' emotional or physical states. They include variables such as pain, sleep quality, mood, fatigue, body comfort, stress and physical activity. This standardized data, collected on a numerical scale, enables continuous, detailed monitoring. These barometers are completed by patients, either manually (in version v1 of the tool), or with the assistance of a chatbot (from 2020 in version v2).

Questionnaires sets were chosen for their relevance to the selected barometer classifications and their ability to effectively monitor the impact of chronic pain on patients' quality of life in a real-life setting. While there are other classification systems such as AAPT, these do not correspond to real-time monitoring elements of the impact of pain on patients' quality of life, but rather to elements for classifying the type of chronic pain and its repercussions. These elements are not suitable for real-time weekly monitoring, but rather for longer-term follow-up, particularly to assess the emergence of risk factors or comorbidities. The barometer classifications of eDOL are based on what clinicians traditionally evaluate in terms of the impact of chronic pain on patients' quality of life, following recommendations from a group of pain physician experts from the ePAIN group.

2.1.3.2 Interesting data for our research

In this thesis, we focused mainly on barometric data, which have a central role in the identification of care trajectory typologies for patients with chronic pain. These data, composed of eight attributes (pain, fatigue, mood, stress, sleep, body comfort, sporting and non-sporting activity) measured weekly, allow measurement of pain intensity and its impact on daily life. Socio-demographic and clinical data are used to explain clustering results and to identify and characterize patient profiles.

Barometric data, shown in Table 2.1, present intrinsic challenges related to their temporal and multidimensional nature. Firstly, each barometric attribute defines an univariate time series, whereas a patient's pain trajectory defined by all eight barometers constitutes a multivariate time series. In addition, barometers can be treated and/or interpreted according to different paradigms, such as discrete time series (DTS), categorical time series (CTS) or continuous/numerical time series (CNTS). Each format presents unique challenges. Another

important aspect is the subjectivity associated with these data. The scores assigned by patients reflect their individual perception of pain, influenced by personal and contextual factors and associated conditions, which introduces uncertainty into the measurements. This subjectivity, amplified by the context of real-life data collection, directly impacts the quality of analyses and complicates the interpretation of results. Moreover, these time series are frequently irregular, as some patients do not use the smartphone application constantly. This discontinuity presents an additional challenge for modelling, particularly as patients have different basic length sequences, as they are not all included in the study at the same time. Finally, the integration of barometers with clinical and socio-demographic data, used to characterize patient profiles, adds a further layer of complexity. This involves combining static information (initial profile) with dynamic time series (clinical data), while maintaining robustness in the presence of missing and heterogeneity data.

Table 2.1: Illustration of attributes corresponding to barometers. In the Polarity column, positive means symptom improvement and negative means worsening of symptoms.

Features Name	Features Description	Scale	Polarity
Sleep	Average sleep quality in the past week	[0 10]	high value is positive
Mood	Average mood in the past week	[0 10]	high value is positive
Fatigue	Average fatigue in the past week	[0 10]	high value is negative
Body comfort	Body comfort quality in the past week	[0 10]	high value is positive
Stress	Average stress level in the past week	[0 10]	high value is negative
Pain	Average pain in the past week	[0 10]	high value is negative
Sports activities	Intense physical activity (number of hours per week)	[0 10]	high value is positive
Non-sports activities	Moderate physical activity (number of hours per week), including walking, daily housework, etc	[0 10]	high value is positive

In summary, the eDOL data are particular due to their temporal, sequential, longitudinal, and particularly subjective nature. They can be analyzed as categorical, discrete, or continuous time series data. Since they are also secondary data (clinical term), the series do not start at the same dates, meaning that they are of different (irregular) lengths. Regarding missing data, while they can be imputed, they may also carry medical significance, such as indicating patient non-adherence. All these characteristics add further complexity to modeling these data.

2.2 Health trajectory

The concept of care pathways or health trajectories is increasingly being used to enhance the quality of care and to optimize the use of resources for health care. Health trajectories describe the evolution of an individual's health states or events over time. These trajectories are constructed from longitudinal or time-series data, which consist of repeated observations of health-related features collected over a defined period [29]. In medical fields, these series

of repeated observations are often categorical or discrete in nature. This characteristic enables us to consider health trajectories, or care trajectories, as categorical or discrete sequences, thus qualifying them as sequential trajectories [30, 31]. Thus, sequential trajectory analysis concerns the analysis of sets of categorical or discrete sequences that generally describe longitudinal or temporal data. The sequences analyzed are encoded representations, for example, of individual care trajectories such as consultations, transitions between outpatient and inpatient care, drugs consumed, but they can also describe daily or weekly time use, or represent observed or self-reported health outcomes.

In the field of epidemiological research, the use of secondary data, as administrative health databases or eDOL data studied here, has now become a widely used strategy thanks to the always greater reliability of the detection methodologies adopted, which involve the acquisition of high quality data [31]. They also offer an ever increasing amount of useful information that allow to conduct epidemiological studies with less resources and cost and time savings. Although they offer a growing amount of useful information, to date, studies focusing on individual pathways have mainly remained descriptive, without taking into account the possible evolution of care consumption over time. In a real-world setting, classical statistical tools resulted to be insufficient to adequately consider a phenomenon with such high variability and multidimensional, and has to be integrated with novel data mining techniques suitable of identifying patterns in complex data structures [29]. Although in recent years some methodologies were proposed to tackle this kind of information [**?**], they are not always suitable to evaluate complex and multidimensional sequential or longitudinal patterns of care and to be implemented on real-world data.

Sequential trajectory clustering, in healthcare in particular, is becoming a topic of growing interest due to its potential to address the limitations of statistical group analysis approaches, for personalized medicine [32]. Nguena Nguefack and al. [33], examined different trajectory modeling techniques in epidemiology, highlighting the importance of clustering as a non-parametric approach for identifying care typologies and characterizing patient profiles. This enables healthcare professionals to design targeted interventions and optimize care pathways. However, the complexity of multidimensional trajectories often demands advanced computational techniques to process and analyze the data efficiently [34], requiring further research.

2.3 Clustering approaches and algorithms

This section provides an organized overview of clustering approaches and algorithms, starting with a taxonomy of clustering methods in 2.3.1. It explores key partitioning frameworks, including fuzzy in 2.3.2, possibilistic in 2.3.3, rough in 2.3.4, and evidential in 2.3.5, highlighting their principles, applications, and algorithms.

2.3.1 Taxonomy for clustering

Clustering – the art and science of manipulating data without any supervision – has attracted substantial attention in artificial intelligence and machine learning communities [35], and has successfully permeated in a wide range of fields [36]. The goal of clustering is to

identify structure in a data set without any prior information by objectively organizing data into homogeneous groups where the within-group-object similarity is minimized and the between-group-object dissimilarity is maximized. Clustering is necessary when no labeled data are available regardless of whether the data are binary, categorical, numerical, interval, ordinal, relational, textual, spatial, temporal, spatio-temporal, image, multimedia, or mixtures of the above data types.
Based on the fundamental characteristics and approaches to grouping data, several different clustering strategies have been proposed, each of which uses a different inclusion principle. Fraley and Raftery [37] suggested dividing the clustering approaches into two different groups: partition based methods, and hierarchical based methods. Saxena and al. [38] suggested the following three additional categories, as density-based methods, model-based methods and grid-based methods. Zhou and al. [39], extends and broadens the categorization by adding families, spectral based methods, and neural networks based methods. A summary of these approaches can be found in Table 2.2.

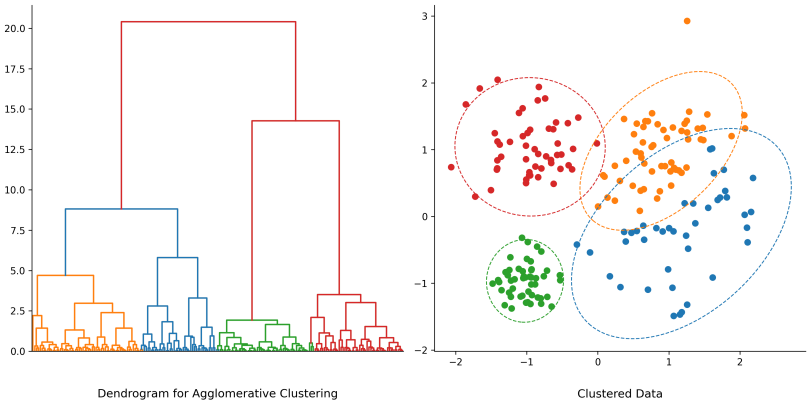
Table 2.2: Algorithms classified by underlying principles and characteristics.

Category	**Benefits**	**Limitations**	**Popular Algorithms**
Partition-Based Clustering	Efficient (low time complexity). Simple to implement.	Sensitive to outliers. Drawn to local optima. Requires predefined number of clusters.	K-Means[40], PAM[41], CLARA[41]
Hierarchical Clustering	Does not require predefined cluster number. Dendrogram provides hierarchy.	High time complexity. Sensitive to noise/outliers.	Agglomerative Clustering[42], Complete Linkage[43], Divisive Clustering[44]
Density-Based Clustering	Suitable for arbitrary shapes. Efficient for high clustering and dense data.	Poor performance for uneven densities. Inefficient for large datasets.	DBSCAN[45], HDBSCAN[46], Mean-Shift[47]
Grid-Based Clustering	Fast. Handles arbitrary cluster shapes.	Sensitive to grid granularity.	CLIQUE[48], ST-DBSCAN[49], Grid-DBSCAN[45]
Model-Based Clustering	Handles non-spherical clusters. Membership probabilities.	High time complexity. Strong assumptions about data.	GMM[50], HMM[51], LDA[52], EM[53]
Spectral Clustering	Detects clusters with complex shapes. Effective for small to mid-sized datasets.	Computationally expensive. Requires predefined cluster number.	Normalized Cuts[54], Spectral Clustering[54]
Neural Network-Based Clustering	Handles high-dimensional data. Learns complex structures.	Requires large datasets and computational resources.	SOM[55], DEC[56]

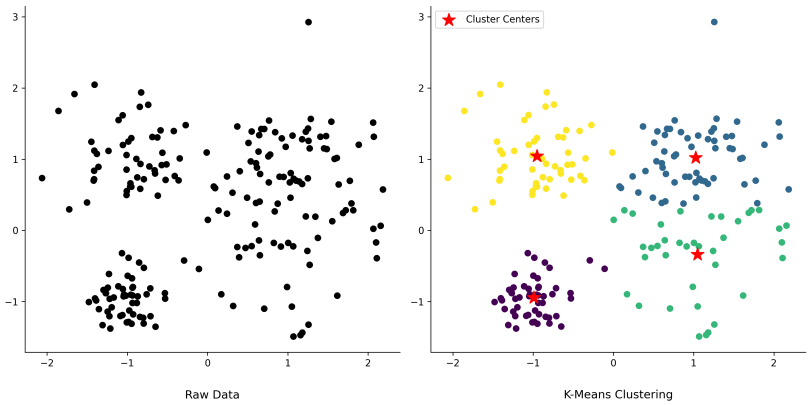
Among these approaches, the most popular and widely used clustering techniques are hierarchical and partitioning methods [36]. Their fundamental characteristics, underlying

principles, and approaches to grouping data are defined and discussed below. For other techniques, these aspects are detailed further in Appendix A.

Hierarchical-Based Clustering(e.g. Figure 2.5a) – The concept of hierarchical clustering lies in the construction and analysis of a dendrogram. A dendrogram is a tree-like structure that contains the relationship between all the data points in the system. There is no need to specify a predefined number of clusters. A horizontal cut in the dendrogram defines clusters at the highest level, with each child branch representing a distinct cluster. However, hierarchical clustering has a high time complexity for large datasets and sensitive to noise and outliers, which can lead to sub-optimal results if not properly managed.
Partition-Based Clustering (e.g. Figure 2.5b) – Partition-based clustering algorithms partition data into non-overlapping clusters in theory. The basic idea of these clustering algorithms is to consider the center of data points as the center of the corresponding cluster, which generally has low time complexity and high computing efficiency. However, they are not well-suited for non-convex data, can be sensitive to outliers, can be easily drawn into local optima, and require a predefined number of clusters, which can impact the performance of clustering results.



(a) Hierarchical Clustering: the dendrogram on the left reflects object relationships, while the right shows clusters resulting the dendrogram.



(b) Partition-Based Clustering: the original data on the left is clustered using K-Means, with results displayed on the right.

Figure 2.5: Illustration of partition-based and hierarchical-based clustering approaches.

There are two common representations of the datasets upon which partitioning methods can be based, feature data (vector data or object-base) and relational data (proximity data or distance-based) [57, 58, 59, 60, 61, 62]. When each object is described by a vector of quantitative or qualitative values, the set of vectors describing the objects is called a feature data. Alternatively, when each pair of objects is represented by a relationship, then it is called relational data. The most common case of relational data is when one has (a matrix of) dissimilarity data, say $\boldsymbol{\mathcal{R}} = [\tau_{ij}]$, where τ_{ij} is the pairwise dissimilarity (often a distance) between objects i and j. Relational clustering is particularly useful when computing the distance between objects is complex, when the distance measure cannot be expressed in a simple mathematical way, or when clusters of similar objects cannot be efficiently represented by a simple barycenter. These two approaches differ in how they represent and process data.

Definition 2.3.1 (Object-Based Partitioning)

Let $\mathbf{X} = (\mathbf{x}_i)$ be a dataset where each $\mathbf{x}_i \in \mathbb{R}^d$ is a vector in a metric space. Object-based partitioning involves dividing this dataset into c clusters $\{C_1, C_2, \ldots, C_k\}$ (with $k \leq n$), such that each $\mathbf{x}_i$ belongs to exactly one cluster C_j.

Definition 2.3.1 (Relational-Based Partitioning)

Let $\mathbf{E} = (\mathbf{e}_i)$ be a dataset and $\boldsymbol{\mathcal{R}} = [\tau_{ij}]$ a dissimilarity matrix where τ_{ij} represents the dissimilarity between $\mathbf{e}_i$ and $\mathbf{e}_j$. Relational partitioning involves dividing this dataset into c clusters $\{C_1, C_2, \ldots, C_k\}$ (with $k \leq n$), using the values in $\boldsymbol{\mathcal{R}}$ to determine the relationships between objects.

In object-based partitioning, data must be available as vectors in a metric space. In relational partitioning, only the relationships between objects, represented by a dissimilarity matrix $\boldsymbol{\mathcal{R}}$, are needed. In object-based partitioning, distances are calculated directly in the data space $\mathbb{R}^d$. In relational approaches, distances are provided by $\boldsymbol{\mathcal{R}}$, and positioning in an explicit space is not required. Relational methods are more flexible as they can handle complex data. However, they may be limited by the quality and properties of the matrix $\boldsymbol{\mathcal{R}}$. Three main methods fall under relational partitioning, each with distinct mathematical formulations.

Relational Means-based [57, 59, 63] – Relational Means-based clustering extends the classical concept of clustering to relational data by minimizing the total dissimilarity between objects and the barycenter of their respective clusters. Given the relational data as a dissimilarity matrix $\boldsymbol{\mathcal{R}} = [\tau_{ij}]$, the goal is to partition the dataset into k clusters such that the following objective function is minimized:

$$J(\mathcal{V}) = \sum_{j=1}^{k} \sum_{e_i \in C_j} \tau_{ij}, \tag{2.1}$$

where $\mathcal{V} = \{v_1, v_2, \ldots, v_k\}$ is the set of barycenters, C_j represents the set of objects in cluster j, and $\tau_{ij} = \|v_j - e_i\|^2$ is the distance between object e_i and the barycenter v_j

of cluster j expressed in terms of $\boldsymbol{\mathcal{R}}$ (necessarily be defined in a Eucludian space). The barycenter v_j of a cluster C_j is defined as the mean of the total dissimilarity to all other objects in the cluster:

$$v_j = \frac{1}{|C_j|} \sum_{e_i \in C_j} e_i. \tag{2.2}$$

Relational Medoids-based [41, 64, 62] – Relational Medoids-based clustering selects representative objects, called medoids, as cluster centers. The objective is to minimize the total dissimilarity between objects and their closest medoids. The optimization problem is formulated as:

$$J(\mathcal{V}) = \sum_{j=1}^{k} \sum_{e_i \in C_j} \tau_{ij}, \tag{2.3}$$

where $\mathcal{V} = \{v_1, v_2, \ldots, v_k\}$ is the set of medoids, with $v_j \in \{e_1, e_2, \ldots, e_n\}$, C_j represents the set of objects assigned to cluster j, and τ_{ij} is the dissimilarity between object e_i and the medoid v_j of cluster j. Unlike the means-based approach, the medoid v_j of a cluster C_j is defined as the objet that minimizes the total dissimilarity $\boldsymbol{\mathcal{R}}$(not need satisfy the axioms of a distance) to all other objects in the cluster:

$$v_j = \arg\min_{e_j \in \mathbf{E}} \sum_{e_i \in C_j} \tau_{ij}. \tag{2.4}$$

Relational Multidimensional Scaling (MDS) based [65, 66, 67] – MDS-based aims to embed relational data into a low-dimensional Euclidean space $\mathbb{R}^p$ while preserving pairwise dissimilarities as much as possible. Given the dissimilarity matrix $\boldsymbol{\mathcal{R}} = [\tau_{ij}]$, the optimization problem is:

$$\min_{\mathbf{Y}} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\|\mathbf{y}_i - \mathbf{y}_j\| - \tau_{ij} \right)^2, \tag{2.5}$$

where $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]^\top$ represents the coordinates of the n objects in $\mathbb{R}^p$, $\|\mathbf{y}_i - \mathbf{y}_j\|$ is the Euclidean distance between $\mathbf{y}_i$ and $\mathbf{y}_j$ in the embedded space, and τ_{ij} is the dissimilarity between e_i and e_j in the original relational data and need not satisfy the axioms of a distance. Based to $\mathbf{Y}$, two main approaches can be used for clustering. The first approach applies a traditional clustering algorithm directly to the coordinates $\mathbf{Y}$. The second relies on optimizing a partitioning function that directly adjusts a partition to best preserve $\boldsymbol{\mathcal{R}}$.

Partitioning methods (object-based or relational-based) can be broadly categorized into hard partitioning and soft partitioning approaches [1, 36, 41, 68]. In a hard partition, each object of the dataset is assigned to one and only one cluster. This ensures that clusters are disjoint, and every object is uniquely associated with a single cluster. Among the popular algorithms for hard clustering, the k-means algorithm [40] is widely used for its simplicity and efficiency in handling object-based data, while the Partitioning Around Medoids (PAM) algorithm [41] is particularly effective for clustering relational data. Soft partitioning generates a partition where each object has a degree of membership in multiple clusters, reflecting the uncertainty

or imprecision in the assignment process. Unlike hard partitioning, soft partitioning allows objects to belong to more than one cluster simultaneously. For instance, fuzzy c-means [69] and fuzzy c-medoids [70] assign membership values based on the proximity of objects to cluster centers, while possibilistic variants such as possibilistic c-means [71] and possibilistic c-medoids [70] focus on reducing sensitivity to noise and outliers. Additionally, rough c-means and rough c-medoids [36] leverage boundary regions to handle overlapping clusters, and evidential approaches like the evidential c-means [1] and c-medoids [68] integrate belief functions to model and quantify uncertainty more robustly. Hard partitioning is suitable for cases where precise and unique assignments are required, whereas soft partitioning provides flexibility in representing overlapping or uncertain clusters. The choice between these approaches depends on the nature of the data and the problem context. The following definitions of object-based partitions can be extended and/or derived for definitions of relation-based partitions.

Definition 2.3.2 (Hard Partition)

Let $\mathbf{X} = (\mathbf{x}_i)$ be a finite dataset and $k \in \mathbb{N}$ such that $2 \leq k < n$. A hard k-partition of X is defined as a boolean matrix $\mathbf{H} \in \mathbb{R}^{n \times c}$, where each element $h_{ij} \in \{0, 1\}$ indicates the membership of x_i in cluster j. The matrix $\mathbf{H}$ satisfies the following constraints:

$$\sum_{i=1}^{n} h_{ij} > 0, \quad \forall j \in \{1, \ldots, k\}, \tag{2.6}$$

where (2.6) ensures that no cluster is empty, and $\sum_{j=1}^{c} h_{ij} = 1, \forall i$, in (2.7) ensures that each object belongs to exactly one cluster. The set of all hard c-partitions is denoted by:

$$\mathcal{M}_{hp} = \left\{ \mathbf{H} \in \mathbb{R}^{n \times k} \mid h_{ij} \in \{0, 1\}, \sum_{j=1}^{k} h_{ij} = 1, \sum_{i=1}^{n} h_{ij} > 0 \right\}. \tag{2.7}$$

Where $\mathbf{H}$ is a hard partition and $\mathcal{M}_{hp}$ is the set of all hard partitions matrices.

Definition 2.3.2 (Soft Partition)

Let $\mathbf{X} = (\mathbf{x}_i)$ be a finite dataset and $c \in \mathbb{N}$ such that $2 \leq c < n$. A soft c-partition is defined by a membership matrix $\mathbf{S} \in \mathbb{R}^{n \times c}$, where $s_{ij} \in [0, 1]$ represents the degree of membership of x_i in cluster j. The matrix $\mathbf{S}$ satisfies the following constraints:

$$\sum_{i=1}^{n} s_{ij} > 0, \quad \forall j \in \{1, \ldots, c\}, \tag{2.8}$$

where (2.8) ensures that no cluster is empty. The set of all soft c-partitions is:

$$\mathcal{M}_{sp} = \left\{ \mathbf{S} \in \mathbb{R}^{n \times c} \mid s_{ij} \in [0, 1], \sum_{i=1}^{n} s_{ij} > 0 \right\}. \tag{2.9}$$

Where $\mathbf{S}$ is a soft partition and $\mathcal{M}_{sp}$ is the set of all soft partitions matrices.

While hard and soft partitioning methods have been extensively developed for single-view data, they may not suffice in many real-world applications where data is naturally represented in multiple views or modalities. For instance, an event can be captured through text, audio, images, and video. In such cases, relying solely on single-view clustering methods like k-means or fuzzy c-means may fail to fully capture the underlying structures in the data, as these methods do not consider the complementary information provided by multiple views. Multi-view clustering addresses this limitation by integrating information from diverse views to produce more accurate and comprehensive clustering results. Early approaches often concatenated features from all views into a single space, applying classical single-view methods. However, this approach overlooks the distinct characteristics of each view and often leads to suboptimal performance. Instead, modern multi-view clustering methods leverage strategies such as view-specific weights, collaborative learning, and feature weighting to enhance clustering performance while preserving the unique contributions of each view. These methods have been successfully applied to both hard partitioning frameworks, such as multi-view k-means [72, 73, 74, 71] and its variants, and soft partitioning frameworks [70, 75, 76, 77, 36], including fuzzy, possibilistic and evidential clustering approaches, reflecting their versatility and effectiveness in real-world scenarios.

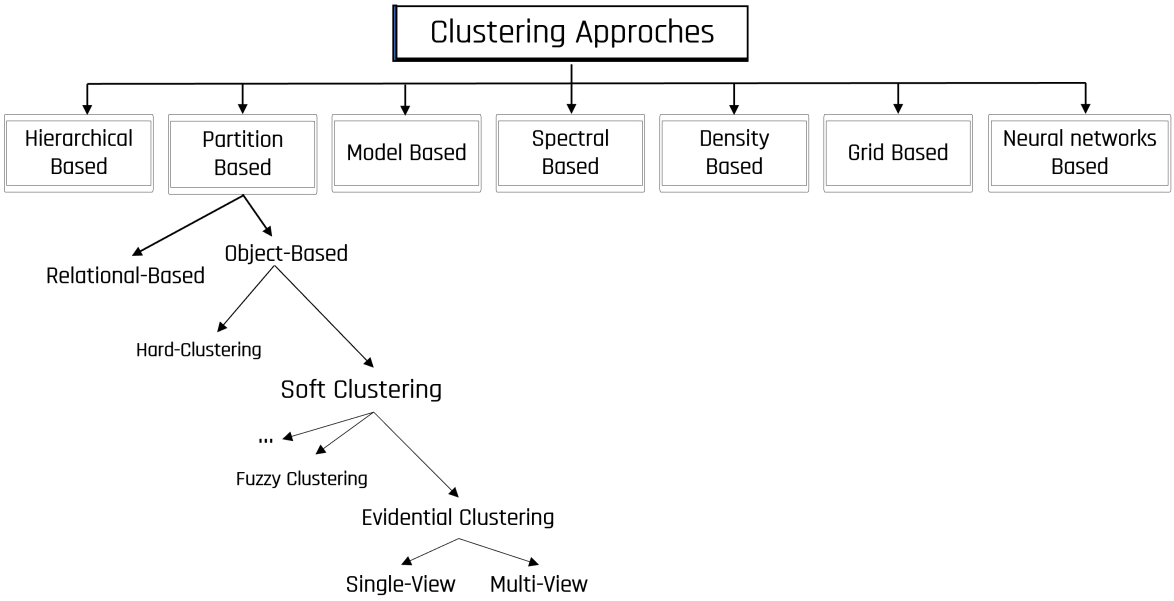


Figure 2.6: The basic principles for choosing different clustering approches.

In the literature, numerous clustering algorithms have been proposed with various categorizations. In the preceding sections, we discussed the most popular approaches and algorithms. As summarized in Figure 2.6, these algorithms have been categorized based on the type of data they handle object or relational based as well as whether they are single-view or multi-view based, and on the assignment approach used for clustering, distinguishing between hard and soft partitions. Our thesis work has focused on these differentiations. However, the are other categorizations in the literature, such as those based on data volume (e.g., big data), data flow (e.g., streaming), application domains, and more [78, 79, 38, 80].

In the next state-of-the-art, instead of discussing general issues and providing an exhaustive survey of various clustering approaches, we concentrate mainly on issues and clustering approaches related to soft partition, with an interesting focus on credal partitioning.

2.3.2 Fuzzy partition

This section provides an overview of fuzzy partitioning, starting with the basic principles of fuzzy set theory in 2.3.2.1. It includes illustrative examples and examines the limitations of the approach in 2.3.2.2, followed by a summary of key algorithms based on fuzzy partitioning in 2.3.2.3.

2.3.2.1 Fuzzy Set Theory

The fuzzy set theory, introduced by Zadeh in [81], provides a framework for representing uncertainty and imprecision by allowing partial membership of elements in a set (fuzziness degree). Unlike classical set theory, where membership is binary (0 or 1), fuzzy sets define a membership function $\mu_A : X \to [0, 1]$ that assigns to each element $x \in X$ a degree of membership in the set A. This degree, $\mu_A(x)$, represents how strongly x belongs to A, with $\mu_A(x) = 1$ indicating full membership, $\mu_A(x) = 0$ indicating no membership, and intermediate values reflecting partial membership. A fuzzy set A in a universe X can be expressed as:

$$A = \{\mu_A(x)/x \mid x \in X\}, \tag{2.10}$$

where $\mu_A(x)$ is the membership degree of x in A. Classical (crisp) sets are a special case of fuzzy sets where $\mu_A(x) \in \{0, 1\}$. Key notions in fuzzy set theory include the support, kernel, and alpha-cuts. The support of a fuzzy set A, denoted $\text{Supp}(A)$, is the set of all elements with non-zero membership, Equation (2.11). The kernel of a fuzzy set A, denoted $\text{Kern}(A)$, is the set of all elements with full membership, Equation (2.12). An α-cut of a fuzzy set A, denoted $A_{\geq\alpha}$, is the set of elements with membership degrees greater than or equal to α, see Equation (2.13). For $\alpha = 1$, the α-cut reduces to the kernel, and for $\alpha = 0$, it corresponds to the support.

$$\text{Supp}(A) = \{x \in X \mid \mu_A(x) > 0\} \tag{2.11}$$

$$\text{Kern}(A) = \{x \in X \mid \mu_A(x) = 1\} \tag{2.12}$$

$$A_{\geq\alpha} = \{x \in X \mid \mu_A(x) \geq \alpha\} \tag{2.13}$$

Fuzzy set theory also extends the basic operations of classical set theory. For two fuzzy sets A and B in X, the complement, intersection, and union are defined as follows:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x), \tag{2.14}$$

$$\mu_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x)), \tag{2.15}$$

$$\mu_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x)). \tag{2.16}$$

These operations ensure that fuzzy sets can model complex systems where elements may partially belong to multiple categories. Fuzzy set theory has found applications in various

fields, including control systems, logic and approximate reasoning, decision-making, and pattern recognition, through fuzzy partitioning techniques, refer to [81] for detailed reviews on applications of the theory to other theories and real problems. In the following Section 2.3.2.2, we illustrate the use of fuzzy partitioning and highlight its limitations in practical.

2.3.2.2 Illustration and limites of fuzzy partition

We illustrate the concept of fuzzy partitioning with an example involving two clusters ω_1, and ω_2, representing happy facial, and sad facial emojis, respectively.

Definition 2.3.3 (Example for Fuzzy Partitioning)

Let us consider the set $\mathbf{X} = \{$ 😀, 🤣, 😂, 😞, 😭, 😩, 😌, 😳, 😟, 🐐, 🌴 $\}$. We aim to partition $\mathbf{X}$ into two classes $\Omega = \{\omega_1, \omega_2\}$, representing happy, and sad facial, respectively. For emojis clearly associated with one category, such as $\{$ 😀, 🤣, 😂 $\}$ for happy, and $\{$ 😞, 😭, 😩 $\}$ for sad, the fuzzy partition aligns closely with a crisp partition. The membership degrees are set to 1 for the corresponding class and 0 for others.

However, for emojis associated with multiple classes, such as $\{$ 😌, 😳, 😟 $\}$ (happy and sad), their membership degrees reflect this uncertainty. We can observe that the emoji $\{$ 😌 $\}$ is closer to happy with 0.7 fuzziness degree. Emojis that do not fit any category (e.g., $\{$ 🐐, 🌴 $\}$) are assigned ? to indicate undefined membership. These objects are atypical and cannot be modeled, but in practice, they are assigned with an equiprobable degree of membership. The resulting fuzzy partition is presented in Table 2.3 and is illustrated in Figure 2.7.

Emojis	ω_1	ω_2
😀	1	0
🤣	1	0
😂	1	0
😞	0	1
😭	0	1
😩	0	1
😌	0.7	0.3
😳	0.5	0.5
😟	0.5	0.5
🐐	?	?
🌴	?	?

Table 2.3: Fuzziness degree for partition into two classes.

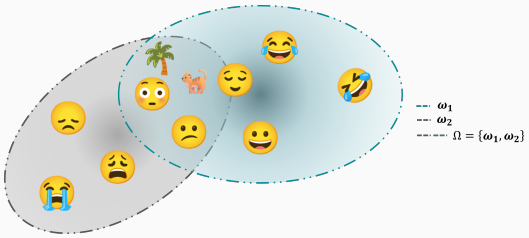


Figure 2.7: Illustration for partition into two classes.

In the context of clustering applications, the interpretation of fuzzy partition degrees can often be challenging to understand. Fuzzy partitioning is based on the premise of a closed universe, where the sum of membership degrees is constrained to one. This limitation

restricts the model's ability to account for atypical objects. Moreover, it does not provide a clear distinction between equiprobability and complete ignorance, and it is insufficient in managing imprecision.

2.3.2.3 Algorithms based on fuzzy partition

Various clustering algorithms can be employed to generate fuzzy partitions. Among these, the fuzzy c-means (FCM) algorithm proposed by Bezdek [69]. Numerous extensions of FCM have been proposed to handle the complexities of different data types, including relational data, for which the fuzzy c-medoids (FCMdd) algorithm [64] is well-suited. The soft partition **S** is replaced by a fuzzy partition **U** in definition 2.3.1. The objective of the FCM algorithm is to find the optimal fuzzy partition **U** and the cluster centers **V**, which minimize the intra-cluster distance. The optimization problem is formalized through the minimization of the following cost function:

$$J_{\text{FCM}}(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} d_{ij}^{2}, \tag{2.17}$$

$$\text{subject to} \quad \sum_{j=1}^{c} u_{ij} = 1 \quad \forall i, \tag{2.18}$$

$$u_{ij} \geq 0 \quad \forall i, j. \tag{2.19}$$

where $m > 1$ is the fuzzification parameter, which controls the degree of fuzziness in the partition, and d_{ij} represents the distance between the i-th data point x_i and the j-th cluster center v_j. The optimization of J_{FCM} is achieved through an iterative process alternating between updating **V** and **U** [69]. The iterative process stops when the objective function converges, described in the Algorithm 1.

Algorithm 1 Fuzzy c-means algorithm (FCM)

Input: $\mathbf{X} = (\mathbf{x}_i)$, number of clusters c, fuzzification $m > 1$, and the convergence threshold $\epsilon \geq 0$.

Initialization: Randomly initialize V_0.

$t \leftarrow 0$

repeat

  $t \leftarrow t + 1$

  Update membership degrees $\mathbf{U}_t = [u_{ij}]$ using Equation (11a) of [69].

  Update cluster centers $\mathbf{V}_t$ using Equation (11b) of [69].

until The prototypes remain unchanged based to ϵ;

Output: Optimal solutions **U**, **V**.

2.3.3 Possibilistic partition

The possibility theory, introduced by Dubois and Prade in [82], expands fuzzy set theory to model uncertainty. Unlike fuzzy theory, which quantifies uncertainty using additive measures, possibility theory employs a possibility distribution Π. For simplicity, assume that the universe of discourse Ω is a finite set. A possibility measure is a function Π from

2^{Ω} to $[0, 1]$. From this distribution, two measures are derived: the possibility measure Π and the necessity measure N. The possibility measure Π assesses how plausible a set $A \subseteq \Omega$ is, while the necessity measure N evaluates the certainty that A is true. These measures are related as follows:

$$\Pi(A) = \max_{\omega \in A} \Pi(\{\omega\}), \quad N(A) = 1 - \Pi(\bar{A}) \Rightarrow N(A) \leq \Pi(A) \quad \forall A, \tag{2.20}$$

where $\bar{A}$ denotes the complement of A in Ω. The measures also satisfy the properties:

$$\Pi(\Omega) = N(\Omega) = 1, \quad \Pi(\emptyset) = N(\emptyset) = 0, \tag{2.21}$$

$$\Pi(A \cup B) = \max(\Pi(A), \Pi(B)), \quad N(A \cap B) = \min(N(A), N(B)). \tag{2.22}$$

This theoretical foundation inspired the development of possibilistic clustering methods to address limitations in fuzzy partitioning, particularly for handling noise and outliers in datasets. Unlike fuzzy clustering methods, possibilistic approaches do not enforce the constraint 2.18. Instead, the degrees of membership indicate the degree of compatibility between data points and individual clusters. This property makes possibilistic clustering robust to atypics objets. The soft partition **S** is replaced by a possibilistic partition **P** in definition 2.3.1.

The possibilistic c-means (PCM) algorithm, introduced by Krishnapuram and Keller [83], was the first partioning approach based on the Possibility theory. Specifically, the objective function of PCM is given by:

$$J_{PCM}(\mathbf{P}, \mathbf{V}) = \sum_{i=1}^{n} \sum_{j=1}^{c} p_{ij}^{m} d_{ij}^{2} + \sum_{j=1}^{c} \eta_j \sum_{i=1}^{n} (1 - p_{ij})^{m}, \tag{2.23}$$

$$\text{subject to} \quad \sum_{j=1}^{c} p_{ij} \geq 0 \quad \forall i, \tag{2.24}$$

$$p_{ij} \geq 0 \quad \forall i, j. \tag{2.25}$$

where **P** is the possibilistic partition, **V** are the cluster centers, and d_{ij} represents the distance between the i-th data point x_i and the j-th cluster center v_j. The exponent $m > 1$ is the fuzzification parameter, and η_j are user-defined constants that control the scale of the membership degrees for each cluster. The terms $(1 - p_{ij})$ ensure that the trivial solution $p_{ij} = 0 \quad \forall i, j$ is avoided. The optimization rules for PCM alternate between optimizing the membership degrees **P** and the cluster centers **V** [83]. The choice of hyperparameters η_j is critical for balancing cluster coherence and outlier robustness. Despite this challenge, PCM has proven effective in various applications where classical fuzzy clustering methods fail [84, 70].
One application of PCM is the segmentation of fat and muscle from MRI images of the thigh. Barra and Boire, in [85], proposed a fast, unsupervised, three-dimensional method using a possibilistic clustering algorithm to extract muscle and fat volumes. This method has shown statistical concordance with validated methods and has proven effective in clinical, fitness and weight loss program contexts.

Possibilistic partitioning addresses some fuzzy clustering issues but has limitations. Atypical or outlier objects, assigned null possibility degrees, may be excluded from all clusters. Additionally, the lack of constraints on the sum of membership degrees causes clusters to behave almost independently, complicating the interpretation and control of the partitioning process. In the next section, we briefly review the Rough partition.

2.3.4 Rough partition

Rough set theory, introduced by Pawlak [86], provides a mathematical framework for handling uncertainty and vagueness by distinguishing between discernible and indiscernible objects. This approach uses two approximations to describe a set: the lower approximation and the upper approximation. Given a finite set $\mathbf{X}$ of objects and an indiscernibility relation R, the pair $(\mathbf{X}, R)$ forms a rough set space. For a subset $A \subseteq \mathbf{X}$, the R-lower approximation $\underline{R}(A)$ and the R-upper approximation $\overline{R}(A)$ are defined as:

$$\underline{R}(A) = \{x \in \mathbf{X} \mid [x]_R \subseteq A\}, \tag{2.26}$$

$$\overline{R}(A) = \{x \in \mathbf{X} \mid [x]_R \cap A \neq \emptyset\}. \tag{2.27}$$

The R-lower approximation $\underline{R}(A)$ contains all objects that certainly belong to A, while the R-upper approximation $\overline{R}(A)$ includes all objects that possibly belong to A. The boundary region of a set A with respect to the relation R denoted by $\widehat{R}(A)$ is defined as the difference between the upper approximation $\overline{R}(A)$ and the lower approximation $\underline{R}(A)$:

$$\widehat{R}(A) = \overline{R}(A) - \underline{R}(A). \tag{2.28}$$

The boundary region represents the area of uncertainty where objects cannot be definitively classified as members (A) or non-members of A (i.e. $\mathbf{X} \setminus A$). Indeed, for any object $x \in \mathbf{X}$ in the boundary region, its equivalence class $[x]_R$ intersects both A and its complement, i.e., $[x]_R \cap A \neq \emptyset$ and $[x]_R \cap (\mathbf{X} \setminus A) \neq \emptyset$. The soft partition $\mathbf{S}$ is replaced by a rough partition $\mathbf{R}$ in definition 2.3.1.

Based on rough set theory and c-partioning, Lingras and West [87, 88] introduced the Rough C-means (RCM) algorithm. In RCM, each cluster is characterized by its lower and upper approximations, enabling a more flexible representation of the clustering structure. This approach allows objects to partially belong to the boundary regions of clusters, reflecting uncertainty in their assignment. The objective function of the RCM algorithm is defined as:

$$J_{RCM} = \sum_{i=1}^{n} \sum_{j=1}^{c} \left[\frac{\gamma}{\underline{n}_j} \underline{\lambda}_{ij} d_{ij}^2 + \frac{1-\gamma}{\overline{n}_j} \overline{\lambda}_{ij} d_{ij}^2 \right], \tag{2.29}$$

$$\text{subject to} \quad \sum_{j=1}^{c} \underline{\lambda}_{ij} = 1 \vee \sum_{j=1}^{c} \overline{\lambda}_{ij} = 0, \tag{2.30}$$

$$\sum_{j=1}^{c} \underline{\lambda}_{ij} = 0 \vee \sum_{j=1}^{c} \underline{\lambda}_{ij} \geq 2, \tag{2.31}$$

$$\overline{\lambda}_{ij}, \underline{\lambda}_{ij} \in \{0, 1\} \quad \forall i, j. \tag{2.32}$$

where $\gamma \in [0, 1]$ is a fixed weight, $\underline{n}_j$ and $\overline{n}_j$ are the numbers of objects in ω_j and $\overline{\omega}_j$, respectively. Each cluster ω_j is express a lower approximation $\underline{\omega}_j$ and an upper approximation $\overline{\omega}_j$. **V** represents the set of cluster centers, and d_{ij} corresponds to the distance between object x_i and the center v_j of the cluster ω_j. The parameters $\underline{\lambda}_{ij}$ and $\overline{\lambda}_{ij}$ represent respectively the lower and upper membership bounds for the i-th object in the j-th cluster. The center v_j of cluster **X** is first evaluated and, next the lower $\underline{r}ij$ and upper $\overline{r}ij$ bounds are updated according to the distances between data points and cluster centers [89].
One example of the use of RCM in medical imaging is the segmentation of MRI images of the brain. In Study [90], Kumar and al. recently proposed a method for segmenting brain tissue using the RCM clustering algorithm. The results enable different brain structures, such as gray matter, white matter, cerebral spinal fluid and tumor tissue, to be distinguished efficiently, taking into account the uncertainty and imprecision in images.

RCM clustering handles outliers by placing them in the boundary region, reflecting their ambiguous membership. This approach, however, treats all boundary objects uniformly, which may not fully capture the nuances of uncertainty. To address these limitations, in the next section, Evidential Partitioning introduces more refined concepts through the Dempster-Shafer theory of belief functions. This method is more robust in handling outliers and structural uncertainty, providing a more nuanced representation of data uncertainty.

2.3.5 Evidential partition

This section introduces the foundational principles of evidential partitioning, grounded in the theory of belief functions in 2.3.5.1, and explores its applications in 2.3.5.2. We discuss illustrative examples and limitations of the approach in 2.3.5.3, followed by an overview of algorithms tailored for object-based data in 2.3.5.4 and relational data in 2.3.5.5

2.3.5.1 Theory of belief function

The theory of belief functions, also referred to as evidence theory or Dempster–Shafer theory (DST), is a generalization of the Bayesian theory of subjective probability, and a general robust mathematical framework for representing and managing uncertainty, with understood connections to traditional probability frameworks such as fuzzy, possibility and rough. DST was introduced by Shafer [91] based on the earlier work of Dempster [92].

The framework provides a means of representing data in the form of a mass function that quantifies our degree of belief in various propositions [93]. One of the major advantages of Evidence Theory over conventional probability is that it provides a straightforward way of quantifying ignorance and is therefore a suitable framework for handling outliers. In Dempster-Shafer theory, mass functions (also referred to basic belief assignment (BBA)) help us define a range of probabilities for any given proposition, creating lower and upper limits known as belief and plausibility, respectively. Instead of assigning an exact probability value, we use this range to reflect uncertainty, which can be more intuitive and flexible. This approach is practical when precise probabilities are difficult to determine, allowing us to express confidence in a more general way.

Let Ω be a finite set of mutually exclusive and exhaustive elements, called the frame of discernment and expressed as:

$$\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}. \tag{2.33}$$

The power set of Ω, denoted as 2^Ω, consists of all subsets of Ω, including the empty set $\emptyset$ and the full set Ω and is defined by (2.34). The mass function m is a mapping $m : 2^\Omega \to [0, 1]$ that satisfies the (2.35) conditions.

$$2^\Omega = \{\emptyset, \{\omega_1\}, \{\omega_2\}, \ldots, \{\omega_1, \omega_2\}, \ldots, \Omega\}. \tag{2.34}$$

$$m(\emptyset) = 0 \quad \text{and} \quad \sum_{A \subseteq \Omega} m(A) = 1. \tag{2.35}$$

The quantity $m(A)$ represents the degree of BBA to a subset $A \subseteq \Omega$, and A is called a *focal element*, if $m(A) > 0$. When m is defined such that all focal elements are singletons, i.e. the cardinality of A, $|A| = 1, \quad \forall A \subseteq \Omega$, it reduces to a Bayesian probability distribution. m is categorical if all belief is assigned to a single subset, i.e. $m(A) = 1, A \in 2^\Omega$. In cases where $m(\Omega) = 1$, it represents total ignorance, indicating a lack of information. From Equation 2.35, the mass assigned to the empty set can be positive, $m(\emptyset) > 0$, in a open-world assumption, and indicates unknown states outside the frame of discernment. The belief, and plausibility functions, defined in 2.3.5.1, are other equivalent representations of m.

Definition 2.3.4 (Belief and Plausibility functions)

For a given mass function m, the belief function Bel: $2^\Omega \to [0, 1]$ and the plausibility function Pl: $2^\Omega \to [0, 1]$ are defined for $A, B \in 2^\Omega$ as:

$$Bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B), \tag{2.36}$$

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \tag{2.37}$$

$$Pl(A) = 1 - Bel(\bar{A}). \tag{2.38}$$

The belief function $Bel(A)$ quantifies the total support for A, while the plausibility function $Pl(A)$ measures the extent to which A is plausible, given the evidence. These functions are dual and related by (2.38), where $\bar{A}$ is the complement of A. And conversely, for finite A, given the belief measure $Bel(B)$ for all subsets B of A, we can find the masses $m(A)$ with the following inverse function:

$$m(A) = \sum_{B \subseteq A, B \neq \emptyset} (-1)^{|A|-|B|} \text{Bel}(B). \tag{2.39}$$

The combination of mass functions has a critical role in uncertain information fusion. Dempster's combination rule, also known as the orthogonal combination rule, is a method for merging information sources within the framework of belief function theory. Depending on the nature of the information of the sources, the combination can be conjunctive when the sources are reliable and disjunctive when the sources are not reliable.

Definition 2.3.5 (Rule of Combination)

Let m_1 and m_2 be two mass functions from two different sources that are reliable and cognitively independent and defined on the same frame Ω.

The conjunctive combination of m_1 and m_2, denoted as $m_1 \cap m_2$, is defined as the unnormalized mass function

$$m_{1\cap 2}(A) = \sum_{B\cap C=A} m_1(B)m_2(C), \quad \forall A \subseteq \Omega. \tag{2.40}$$

The disjunctive combination of m_1 and m_2, denoted as $m_1 \cup m_2$, is defined as the unnormalized mass function

$$m_{1\cup 2}(A) = \sum_{B\cup C=A} m_1(B)m_2(C), \quad \forall A \subseteq \Omega. \tag{2.41}$$

Specifically, via (2.40, 2.41), the Dempster's rule of combination is globally calculated from the two sets of masses m_1 and m_2 in the following manner :

$$m_{1\oplus 2}(A) = \frac{1}{1-K} \sum_{B\cap C=A} m_1(B)m_2(C), \quad \forall \emptyset \neq A \subseteq \Omega, \tag{2.42}$$

where K is a measure of the amount of conflict between the two mass sets, defined as:

$$K = \sum_{B\cap C=\emptyset} m_1(B)m_2(C). \tag{2.43}$$

Dempster's rule is associative and commutative, ensuring a consistent fusion of evidence. However, it becomes inapplicable when the conflict K approaches 1, indicating contradictory evidence [94].

For decision-making purposes, mass functions are often transformed into probabilities using the pignistic transformation $BetP$ or the plausibility transformation p_m. The plausibility transformation [95] consists in normalizing the probability distribution plausibility function, resulting in (2.45). Alternatively, the pignistic transformation [96] distributes each normalized mass uniformly to the elements of A by using (2.44). Another approach is to approximate a mass function m as a set [97]. A simple choice is to select the focal set $A^*(m)$ with the largest mass, as defined by (2.46).

Definition 2.3.6 (Mass function transformation)

Given a mass function m, the different summarization of m are defined as:

$$BetP(\omega) = \sum_{\{\omega\in A\}} \frac{1}{|A|} \frac{m(A)}{(1-m(\emptyset))}, \quad \forall A \subseteq \Omega, \quad m(\emptyset) \neq 1. \tag{2.44}$$

$$p_m(\omega) = \frac{pl(\omega)}{\sum_{\omega'=1}^{c} pl(\omega')}, \quad \forall \omega \in \Omega. \tag{2.45}$$

$$A^*(m) = \underset{B \subseteq \Omega}{\operatorname{argmax}}\, m(B). \tag{2.46}$$

where $|A|$ denotes the cardinality of A.

2.3.5.2 Evidential partition

Let $\mathbf{X} = (\mathbf{x}_i)$ be a set of n objects. We assume that each object in $\mathbf{X}$ belongs to at most one cluster in a set $\Omega = \{\omega_1, \ldots, \omega_c\}$. Using the formalism recalled in Section 2.1, evidence about the cluster membership of each object x_i can be described by a mass function m_i defined on the frame on Ω. The soft partition $\mathbf{S}$ is replaced by a credal (or evidential) partition $\mathbf{M}$ in definition 2.3.1.

The concept of evidential partitions generalizes most of the clustering frameworks discussed [98]. Inspired from [99] , we have the relationship between different clusters, as shown in Figure 2.8. When all the mass functions m_i are certain, $\mathbf{M}$ becomes equivalent into a hard partition $\mathbf{H}$, as defined in 2.3.1, which corresponds to complete certainty regarding the membership of each object. In the case where the mass functions are Bayesian, $\mathbf{M}$ reduces to a fuzzy partition $\mathbf{U}$. The membership degree u_{ik} of an object i to a cluster k is then given by $u_{ik} = Bel_i(\{\omega_k\}) = Pl_i(\{\omega_k\}) \in [0, 1]$, where the condition $\sum_{k=1}^{c} u_{ik} = 1$ holds. When the mass functions m_i are consonant, they are equivalently represented by their contour functions pl_i. The values $pl_{ik} = pl_i(\omega_k)$ quantify the possibility that the object o_i belongs to the cluster ω_k, which is commonly used in possibilistic partition $\mathbf{P}$. Finally, when each mass function m_i is logical with a focal set $A_i \subseteq \Omega$, the evidential partition becomes equivalent to a rough partition $\mathbf{R}$. In this case, the lower approximation ω_k^l and the upper approximation ω_k^u of the cluster ω_k are defined as $\omega_k^l := \{i \in \mathbf{X} \mid A_i = \{\omega_k\}\}$ and $\omega_k^u := \{i \in \mathbf{X} \mid \omega_k \in A_i\}$.

The concepts of hard, fuzzy, possibilistic, rough, and evidential data reflect varying levels of uncertainty and representation. Hard data assumes deterministic values, while fuzzy data allows partial memberships quantified by $u_{ij} \in [0, 1]$. Possibilistic data further relaxes constraints by modeling compatibility rather than probabilities. Rough data captures uncertainty through lower and upper approximations, $\underline{R}(X)$ and $\overline{R}(X)$, and evidential data generalizes these ideas using mass functions $m : 2^{\Omega} \to [0, 1]$, offering a belief-based representation.

Each data type influences the corresponding partition structure, from exact assignments in hard partitions to flexible models in fuzzy, possibilistic, rough, and evidential partitions, which accommodate varying degrees of uncertainty and imprecision.

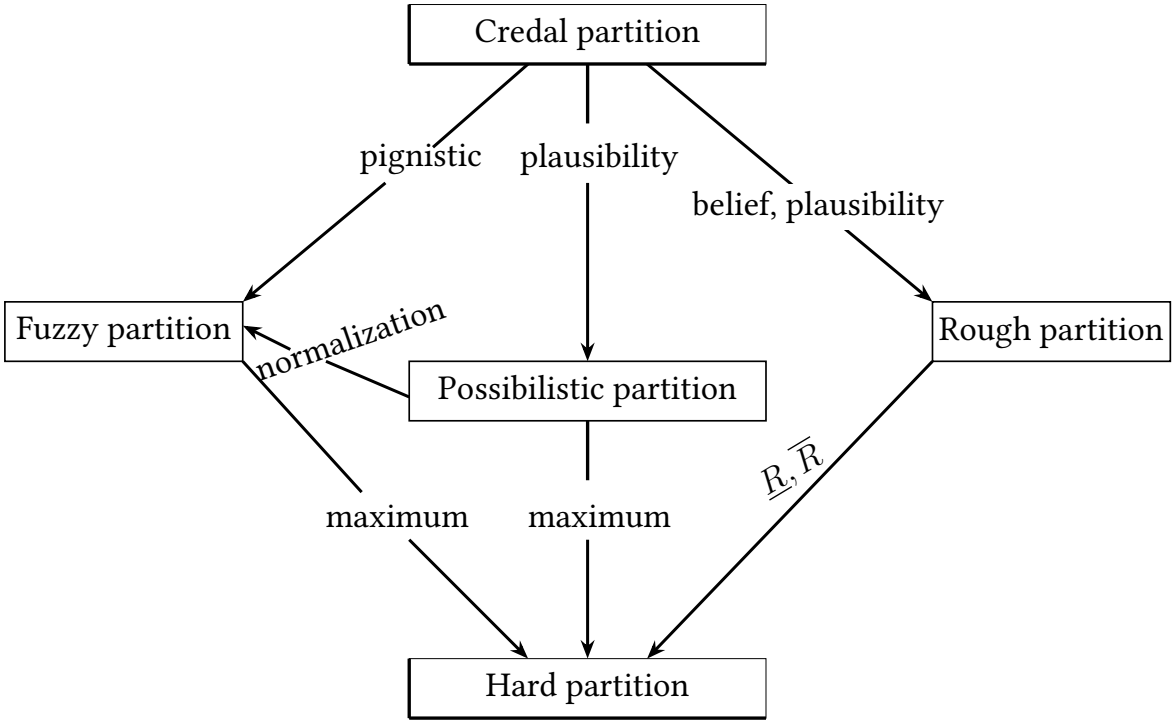


Figure 2.8: Relationship between hard, fuzzy, possibilistic, rough, and evidential clusters.

2.3.5.3 Illustration and limits of evidential partition

To illustrate the functioning of an evidential partition, we revisit the example of partitioning facial expressions, as discussed in Section 2.3.2.2. In this example, we consider two clusters, ω_1 and ω_2, representing happy and sad facial emojis, respectively. This demonstrate how the degree of belief is assigned to each cluster and how atypical and ambiguous facial expressions are treated.

Definition 2.3.3 (Example for Evidential Partitioning)

Consider the set $X = \{$ 😜, 🤣, 😂, 😞, 😭, 😩, 😔, 😳, 😕, 🐐, 🌴 $\}$. Our goal is to partition X into two classes $\Omega = \{\omega_1, \omega_2\}$, representing happy and sad emojis, respectively. Evidential partitioning introduces more flexibility by assigning belief masses not only to individual classes but also to subsets of Ω, enabling a richer representation of uncertainty.

For emojis clearly belonging to one class, such as $\{$ 😜, 🤣, 😂 $\}$ (happy) and $\{$ 😞, 😭, 😩 $\}$ (sad), the evidential partition aligns with crisp assignments. Their belief masses are allocated entirely to ω_1 or ω_2. For ambiguous emojis, such as $\{$ 😔, 😳, 😕 $\}$, the mass is distributed across ω_1, ω_2, and Ω to reflect uncertainty. For instance, 😔 has $m(\omega_1) = 0.1$, $m(\omega_2) = 0.2$, and $m(\Omega) = 0.7$.

Atypical emojis, like $\{$ 🐐, 🌴 $\}$, do not fit into any class. 🐐 is assigned a small belief mass to Ω, while 🌴 represents total ignorance with $m(\emptyset) = 1$. The

evidential partition offers a nuanced approach by allowing multi-level differentiation, as illustrated in Table 2.4 and Figure 2.9.

Emojis	$\emptyset$	ω_1	ω_2	Ω
🤪	0	1	0	0
🤣	0	1	0	0
😂	0	1	0	0
😞	0	0	1	0
😭	0	0	1	0
😩	0	0	1	0
😔	0	0.1	0.2	0.7
😳	0	0	0	1
😕	0	0	0	1
🐈	0.9	0	0	0.1
🌴	1	0	0	0

Table 2.4: Belief degree for partition into two classes.

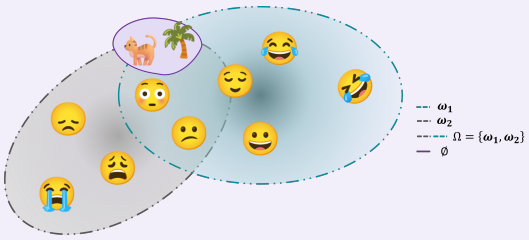


Figure 2.9: Illustration for partition into two classes.

The main limitation of evidential partitioning is its demand for large computational resources. The number of clusters can increase exponentially because of the power set defined in (2.34), to possible subsets of clusters. To manage this complexity, the size of the subsets can be restricted to a specific number.

2.3.5.4 Object-Based Evidential Partition Algorithms

Evidential partitioning algorithms extend the concepts of clustering into the framework of evidence theory, allowing more nuanced representation of uncertainty. The Evidential c-means (ECM) [1] algorithm, introduced by Masson and Denœux, stands as a significant development in this field. ECM can be viewed as an extension of FCM and noise clustering (NC) [100], adapted to use mass functions for clustering in the evidential framework.

Let $\mathbf{X} = (\mathbf{x}_i)$ be a matrix of n individuals in $\mathbb{R}^{n \times q}$ to be classified into a set $\Omega = \{\omega_1, ..., \omega_c\}$, where c is the number of classes, and v_k in $\mathbb{R}^q$ is the prototype (or centroid) of class ω_k. Each subset $A_j \subseteq \Omega$ with a cardinality greater than 1 is also represented by a prototype $\overline{\boldsymbol{v}}_j$. In ECM, it allows the object to be in any singleton clusters and meta-clusters with different masses of beliefs m_i, which are subsets of the clustering space Ω. It incorporates a noise cluster, represented by the empty set $\emptyset$, to handle outliers. The objective function of ECM is defined as:

$$J_{\text{ECM}}(\mathbf{M}, \mathbf{V}) = \sum_{i=1}^{n} \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^{\beta} d_{ij}^2 + \sum_{i=1}^{n} \delta^2 m_{i\emptyset}^{\beta}, \tag{2.47}$$

subject to:

$$\sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij} + m_{i\emptyset} = 1, \quad \forall i = 1, \ldots, n, \tag{2.48}$$

$$m_{ij} \geq 0 \quad \forall i = \{1, \ldots, n\}, \forall A_j \subseteq \Omega, \tag{2.49}$$

$$\sum_{i=1}^{n} m_{ij} > 0 \quad \forall A_j \subseteq \Omega. \tag{2.50}$$

Here, d_{ij} represents the Euclidean distance between object e_i and the center of subset A_j, while $\delta > 0$ is a parameter controlling the influence of the noise cluster. The weight $|A_j|^{\alpha}$ penalizes allocations to larger subsets, and $\beta > 1$ is the fuzziness parameter, where higher values lead to more distributed mass functions. For meta-clusters (i.e., $|A_j| > 1$), the centers are defined as barycenters of their constituent singleton cluster centers, computed with Equation 2.51.

$$\overline{\boldsymbol{v}}_j = \frac{1}{|A_j|} \sum_{k=1}^{c} s_{jk} \boldsymbol{v}_k, \quad s_{jk} = \begin{cases} 1, & \text{if } \omega_k \in A_j, \\ 0, & \text{otherwise.} \end{cases} \tag{2.51}$$

Optimization of the ECM objective function alternates between updating the evidential partition $\mathbf{M}$ using Equations (2.47) and (2.50), and updating cluster centers based on Equation (2.51). These optimization steps are summarized in Algorithm 2.

Algorithm 2 Evidential c-means algorithm (ECM)

Input: Objet data $\mathbf{X} = (\mathbf{x}_i)$ in $\mathbb{R}^p$; number of clusters c; parameters $\alpha \geq 0$, $\beta > 1$, $\delta > 0$, and termination threshold ϵ.

Initialization: Randomly initialize $\mathbf{V}_0$, with c prototypes.

$t \leftarrow 0$

repeat

  $t \leftarrow t + 1$

  Compute $\mathbf{M}_t$ using Equations (18-19, 29-30) of [1].

  Compute $\mathbf{H}_t$ and $\mathbf{B}_t$ using Equations (36-37) of [1].

  Solve $\mathbf{H}_t \mathbf{V}_t = \mathbf{B}_t$.

until he prototypes remain unchanged based to ϵ;

Output: Optimal solutions $\mathbf{M}$, and $\mathbf{V}$.

Beyond ECM, several variants and extensions have been proposed. The Belief c-means (BCM) [101] and Credal c-means (CCM) [102] algorithms redefine the distance between objects and meta-clusters to address limitations in ECM. Constrained versions, such as CECM [103], integrate prior knowledge through constraints, while semi-supervised approaches as SECM [104] utilize labeled data to guide clustering. Recent advancements also explore evidential clustering for categorical datasets, as CatECM [105] or ECM+ [106] when Mahalanobis distances are used.

2.3.5.5 Relational-Based Evidential Partition Algorithms

Among the relational data-based clustering methods discussed in Section 2.3.1, the medoids-based approach is the most popular due to its flexibility in representing cluster

prototypes. In this context, the Evidential c-medoids (ECMdd) [107] algorithm stands out as a significant development for relational data clustering, building on the principles of evidence theory and credal partitioning, where the only required information is the pairwise dissimilarity matrix $\boldsymbol{\mathcal{R}} = [\tau(e_i, e_j)]$.

Let $\mathbf{E} = (\mathbf{e}_i)$ be a set of n objets in $\mathbb{R}^{n\times q}$ to be classified into a set $\Omega = \{\omega_1, ..., \omega_c\}$, where c is the number of classes, and where the only required information is the pairwise dissimilarity matrix $\boldsymbol{\mathcal{R}}$. In ECMdd, the credal partition $\mathbf{M}$ is computed by optimizing the same objective function :

$$J_{ECMdd}(\mathbf{M}, \mathbf{V}) = \sum_{i=1}^{n} \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^{\beta} \tau_{ij} + \sum_{i=1}^{n} \delta^2 m_{i\emptyset}^{\beta}, \tag{2.52}$$

subject to :

$$\sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij} + m_{i\emptyset} = 1, \quad \forall i = 1, \ldots, n, \tag{2.53}$$

$$m_{ij} \geq 0 \quad \forall i = \{1, \ldots, n\}, \forall A_j \subseteq \Omega, \tag{2.54}$$

$$\sum_{i=1}^{n} m_{ij} > 0 \quad \forall A_j \subseteq \Omega. \tag{2.55}$$

where τ_{ij} is the dissimilarity between e_i and the focal set A_j. The hyper-parameters α, β, δ are adjustable and have the same meanings as those in ECM, view in (2.47). Let $v_k^{\Omega} \in E$ such that $A_j = \{\omega_k\}$ (thus $|A_j| = 1$) be the medoid of a specific cluster, and let $v_j^{2^{\Omega}} \in E$ such that $A_j \subseteq \{\omega_1, \ldots, \omega_c\}$, $|A_j| > 1$ be an imprecise cluster. When $|A_j| = 1$, we have $v_j^{2^{\Omega}} \triangleq v_k^{\Omega}$ thus $\tau_{ij} = \tau(e_i, v_k^{\Omega})$. Otherwise, when $|A_j| > 1$, the dissimilarity τ_{ij} is defined as follows:

$$\tau_{ij} = \frac{\tau(e_i, v_j^{2^{\Omega}}) + \gamma \frac{1}{|A_j|} \sum_{\omega_k \in A_j} \tau(e_i, v_k^{\Omega})}{1 + \gamma}, \tag{2.56}$$

where γ quantifies the impact of uncertainty on the dissimilarity between objects and imprecise clusters.

The algorithm alternates between updating the credal partition $\mathbf{M}$ and determining the optimal medoids $\mathbf{V}$. Medoids for singleton clusters are chosen to minimize the total dissimilarity to assigned objects, while medoids for meta-clusters are determined by balancing the distribution of dissimilarities to the singleton medoids and the centrality of the object [107]. The ECMdd algorithm is summarized in Algorithm 3

Algorithm 3 Evidential c-medoids algorithm (ECMdd)

Input: Relational data $\boldsymbol{\mathcal{R}} = [\tau(e_i, e_j)]$ from $\mathbf{E} = (\mathbf{e}_i)$; number of clusters c; parameters α, $\beta > 1$, $\delta > 0$, $\eta > 0$, $\gamma \in [0, 1]$, and the convergence threshold $\epsilon \geq 0$.

Initialization: Randomly initialize c medoids $\mathbf{V}_0$.

$t \leftarrow 0$

repeat

  $t \leftarrow t + 1$

  Update the credal partition $\mathbf{M}_t$ using Equation (27) and (28) of [107].

  Update the prototypes $\mathbf{V}_t$ using Equation (29) of [107].

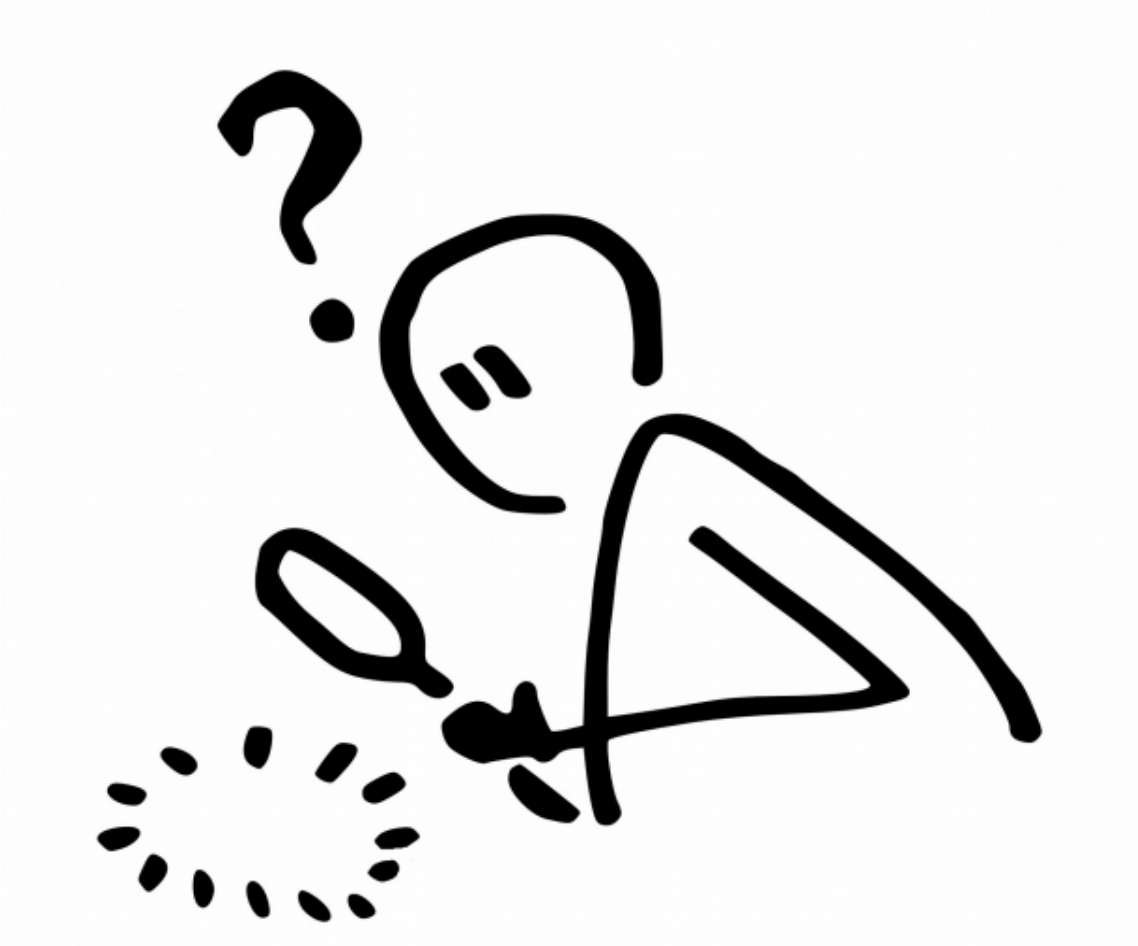
until The prototypes remain unchanged based to ϵ;

Output: Optimal solutions $\mathbf{M}$ and $\mathbf{V}$.

The ECMdd algorithm is closely related to other evidential clustering methods, such as EVCLUS [67], which based to MDS relational approch, and RECM [63], an extension of ECM related means-based relational approch. Other notable algorithms include Median ECM (MECM) [108], which also use medoid as clusters center.

Part II

Contributions



Chapter 3

Review and comparative analysis of multidimensional sequential trajectories clustering methods

Abstract – In this chapter, we review and compare clustering methods for multidimensional trajectories. We address the challenges posed by multidimensional trajectories characterized by discrete or categorical longitudinal or time series data. Our review encompasses existing methods for trajectory analysis, highlighting their applications and differences. The trajectories are investigated and categorized. The trajectory clustering techniques examined include feature-based, row-data-based, and model-based approaches. A comparative study of non-parametric sequential clustering approaches is conducted, focusing on multidimensional sequence dissimilarity analysis with both hard and soft clustering methods. Additive methods are found to be the most effective, despite their reliance on major assumptions about dimension interdependence. We provide a critical analysis of the differences between the studied approaches, and we proposed the exploration of a new multidimensional trajectory clustering approach in a collaborative paradigm.

Contents

3.1 Introduction

Over recent decades, the emergence of storage solutions has allowed scientists and companies to collect large quantities of individual data from large populations over long periods of time. Thus, longitudinal data and time series datasets have multiplied in the form of panels, biographical surveys, and studies measured according to time or age. The larger volume of data collected presents new opportunities and new challenges in analyzing these data. Trajectory analysis has become a major perspective in computational sciences for analyzing longitudinal data and time series at the individual level. Trajectory analysis allows us to study the evolution of markers such as behavior, event, or another measure of interest repeated over time. In practice, there are many methods for analyzing trajectories, and they can be applied to a variety of fields and themes. Depending on the research question and the data type, these methods can be modelling approaches or individual-centered statistical approaches, such as clustering. Considering the critical role of trajectory data mining in modern intelligent systems, this contribution surveys the development of trajectory clustering. Clustering methods offer a unique advantage in identifying hidden patterns and structures within complex datasets, particularly in trajectory analysis [109]. Trajectory clustering thus provides a detailed and understandable description of trajectory patterns between objects. Numerous researches have highlighted the importance of trajectory clustering across various fields. For instance, in healthcare and epidemiology, a study by S. Krishnagopal emphasized the role of trajectory clustering in identifying distinct sub-groups of Parkinson's disease based on disease progression [110]. Furthermore, recent work by J. Ruof and al. showed for example the effectiveness of trajectory clustering in revealing hidden patterns in urban mobility trajectories data [111].

The literature review on trajectory clustering studies reveals several definitions and formulations of trajectories, with no clear and comprehensive categorization of the different trajectory types. Moreover, while review works generally focus on temporal and spatial trajectories, few comparative methodological and empirical papers have been published describing non-parametric clustering on sequential trajectories [32]. Recently, K. Emery and A. Berchtold, conducted an empirical comparison of two methods for multidimensional sequential trajectories, noted multichannel sequence clustering, using the Swiss Household Panel [112]. These methods are sequential analyses by status combination and by cost combination, whereas the literature identifies five main methods [32]. In addition, their work highlights limitations such as selecting only complete sequences without missing data, relying on a single, relatively simple dataset, restricting the approach to one clustering algorithm, and using only edit distance-based measurement functions for computing sequence dissimilarities.

Our work extends their work by offering a comprehensive empirical comparison of the multidimensional sequential trajectories methods in the literature [32]. These methods include the relational clustering, a medoid-based procedure applicable to non-metric proximity data, and methods inspired by multidimensional scaling. In addition to the two approaches mentioned above [112], we also examine the distance combination approach [113], the cluster combination approach [113], and the globally interdependent multiple sequence analysis approach [14]. In relational clustering, the measure of dissimilarity between pairs of trajectories is crucial [114, 115]. We generate dissimilarity measures

using different sets of trajectories and a fixed dissimilarity function, or using a fixed set of variables and different dissimilarity functions, or even using different sets of trajectories and dissimilarity functions. In the literature, sequential clustering most often uses hard clustering algorithms as Ascending Hierarchical Classification (CAH) or Partitioning Around Medoids (PAM), both of which are relational clustering techniques using medoids. However, in many real-world applications, trajectory data can be ambiguous or uncertain, and in such cases, hard clustering may lead to insufficient accuracy. To address this issue and capture the degree of ambiguity, we add to the review two additional medoid-based soft relational partitioning algorithms, namely fuzzy c-medoid (FCMdd) [64] and evidential c-medoid (ECMdd) [62]. These variants allow expressing uncertainty and/or imprecision in clustering. The comparison is performed on different real datasets from various domains and simulated datasets were considered in this study. In addition, for complete comparison, we included large datasets characterized by large numbers of individuals, long-term trajectories, and high dimensions.

In this review, we provide a comprehensive overview of the major existing works in trajectory clustering and propose a contextual background for the diverse formulations and definitions of trajectories. Instead of citing works chronologically, we categorize these methods based on whether they work directly on the raw data, indirectly with features extracted from the raw data, or indirectly with models built from the raw data, with a focus on sequential trajectory clustering methods. We also offer a critical and empirical analysis of the most representative multidimensional sequential trajectory (MST) clustering methods, highlighting aspects of data, method process, application advantages, and limitations. Additionally, we introduce a new approach to MST clustering based on multi-view relational clustering. Finally, we discuss the current challenges in MST clustering and outline the main opportunities for future work.

The subsequent sections of this chapter are structured as follows. First, Section 3.2 presents a review of trajectory data and the main associated clustering methods. Next, Section 3.3 focuses on an extended analysis of clustering methods applied to sequential trajectories. Section 3.4 explores specific clustering methods for multidimensional sequential trajectories. In Section 3.5, an experimental comparison of multidimensional methods is conducted to evaluate their performance. Finally, the Section 3.6 discusses the results obtained.

3.2 Review of trajectory data and clustering methods

3.2.1 Diverse formulations of trajectory definitions

Trajectory data is commonly represented as continuous repeated-measure data, such as continuous time series or continuous longitudinal data. Spinsanti et al. [116] differentiate trajectory data based on sources as Global Positioning System (GPS), Global System for Mobile Communications (GSM), and geolocated social networks. Pelekis and Theodoris later added Radio-Frequency Identification (RFID) and WiFi based data. Trajectory data is generally examined in terms of the movements of people and objects, but can also include activities and events of objects such as animals, vehicles and satellites [117]. Robette Nicolas introduced social trajectories, used in fields such as health, psychology, and sociology

[32]. Trajectory data is part of contextual data representations, incorporating temporal or spatial properties. Although often treated as time series, trajectory data have more properties than simple time series. They are ordered, spatial (location, orientation, speed, direction), dynamic (constantly changing), multidimensional, heterogeneous (comprising multiple variables from diverse sources), and temporal (repeated measures). Considering these properties, we can define four main types of trajectory, as illustrated in Figure 3.1. Sequential trajectories involve the ordering of events or states. Temporal trajectories concern the tracking of data over time. Spatial trajectories refer to the tracking of data in space. Finally, spatio-temporal trajectories cover the tracking of data in both space and time.

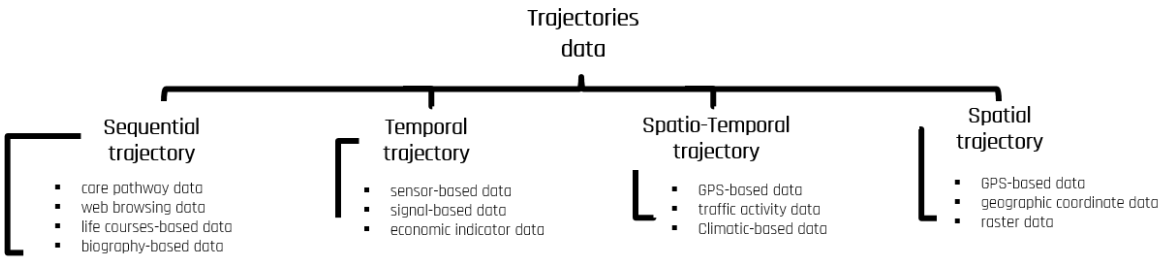


Figure 3.1: Possible classification of the different types of trajectories.

Sequential trajectory refers to the order of events or states over time. Sequential trajectories generally relate to categorical or discrete longitudinal data and categorical or discrete time series [32]. They are used in the human sciences and most often characterize human trajectories or life courses, such as care trajectories, consumption trajectories, professional trajectories, etc, [32]. In sequential trajectories, time may not always be explicitly defined, and the trajectory is constructed based on a sequence of categorical measurements. Since Andrew Abbott highlighted the importance of sequential methods for the social sciences, sequential trajectory analysis, or optimal correspondence analysis, has become popular in domains of sociology, economics, demography, political science and, more recently, health [118]. For instance, medical trajectory research defines the healthcare pathway as the sequence of events or care activities during a period of patient follow-up [33].

Temporal trajectory refers to the timeline of events of an object. Generally referred to as a movement trajectory, temporal trajectories are related to continuous time series data and often to continuous longitudinal data. Temporal trajectories clustering were the subject of the first trajectory analyses but today, the main thrust of temporal trajectory research is multidimensional [119].

Spatial trajectories generally refer to a sequence of observations or measurements related to the location of an object. These trajectories are often treated similarly to spatio-temporal trajectories, but a spatial trajectory is essentially a trace generated by a moving object in geographic space [120]. Most research on trajectory clustering focuses on relational clustering. For example, Dianfeng Qiao et al. [120] have used spatial neighbor analysis to cluster trajectories related to natural phenomena such as hurricanes and ocean currents. S. Sharmila and B. A. Sabarish analyzed and compared various distance measures in the

context of clustering spatial trajectory data [121].

Spatio-temporal or geometric-temporal trajectory refers to both the spatial and temporal information of an object or system, as it moves and changes over time in a given environment. Spatial-temporal data is increasingly becoming universal as a result of the widespread use of GPS, GSM, and RFID. In the recent paper on spatio-temporal trajectory clustering [122], authors propose an extension of the DBSCAN (Density Based Spatial Clustering of Applications with Noise) algorithm for spatio-temporal sub-trajectories clustering and the concepts of entropy and silhouette index to validate the clusters. We can also mention this paper on hierarchical clustering of trajectories for the exploration of spatio-temporal periodic patterns [123], where a new algorithm (using spatio-temporal semantic information such as direction, speed and time) is proposed and compared with three clustering methods: Kernel function, Grid-based and Traclus.

3.2.2 Diverse formulations for trajectory Clustering

Trajectory clustering methods are broadly categorized into non-parametric (feature-based or raw-data-based) and parametric (model-based) approaches [124, 80]. Non-parametric approaches, which make no assumptions about data distribution, assign trajectories to clusters based on proximity or object data, while parametric approaches use probability models to infer cluster membership. Feature-based methods extract key features for clustering, while raw-data-based approaches work directly with the data, preserving its longitudinal structure. Model-based methods, on the other hand, rely on statistical models for clustering. Non-parametric methods, particularly raw-data-based approaches, are effective for complex trajectory data, allowing relational clustering algorithms to directly handle longitudinal patterns without assuming underlying distributions. Figure 3.2 illustrates the three major approaches: (a) feature-based, (b) raw-data-based, and (c) model-based.

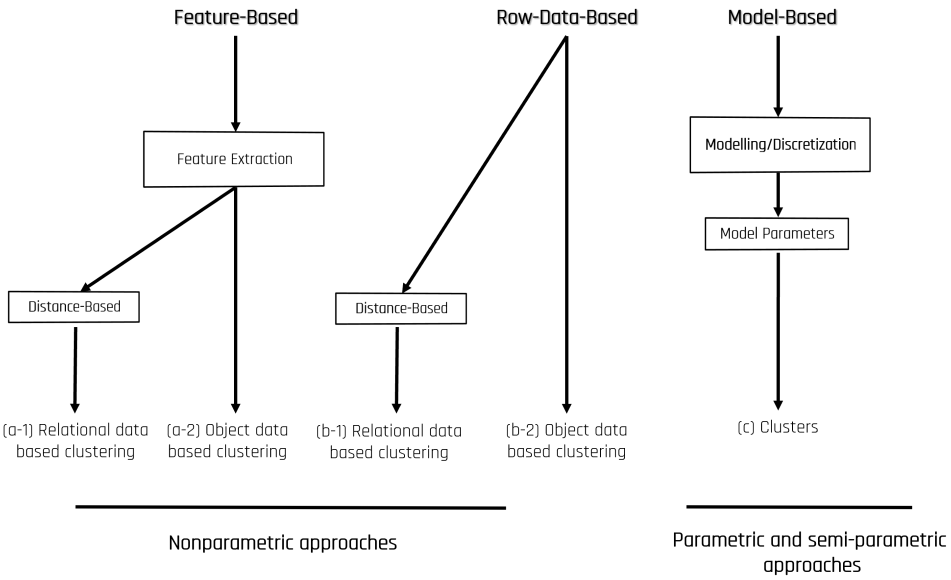


Figure 3.2: Three trajectory clustering approaches: (a) feature-based , (b) raw-data-based, (c) model-based.

Overview of trajectory clustering methods shows that some methods are based on object data, whereas most methods are based on proximity data (relational clustering), i.e. a matrix of dissimilarities or distances between objects. Naturally, algorithms in the latter category can also be applied to object data after computing a distance matrix. Relational clustering approaches thus offer greater adaptability and allow the use of non-Euclidean distance metrics. This flexibility facilitates the efficient clustering of complex data, such as temporal, spatial and sequential data, which present hidden structures or do not lend a Euclidean distance anytime. Indeed, such data, which cannot be represented in a Euclidean space, are not compatible with object-based algorithms that are based on the properties of Euclidean spaces, particularly for the computing of barycenters. Used an optimal distance function adapted to each type trajectory data or to each dimension in data considerably improves clustering performance [125].

In Figure 3.3, we illustrate the categorization of the main relational clustering algorithms from the two broad families of traditional and popular clustering approaches, i.e. partitioning approaches and hierarchical approaches [79], frequently used for trajectory clustering [117]. Hierarchical methods produce a dendrogram, a nested structure of successive groups of input data. Hierarchical methods can be agglomerative, such as Agglomerative Nesting (AGNES) and Sequential Agglomerative Hierarchical Non-overlapping (SAHN), or divisive, such as Divisive Analysis (DIANA) [41]. In contrast, partitioning methods generally aim to provide a unique partition of the data into a fixed number of clusters, often by locally optimizing an objective function. Partitioning methods can be hard partitioning, such as Partitioning Around Medoids (PAM) [41], or soft partitioning, which includes fuzzy algorithms, such as Fuzzy Clustering (FANNY) [41], Relational Fuzzy c-means (RFCM) [57], Non-Euclidean Relational Fuzzy c-means (NERF) [59], and Fuzzy c-medoids algorithm (FCMdd) [64]. Soft partitioning methods also include evidential clustering algorithms, such as Evidential Clustering (EVCLUS) [66], Relational Evidential c-means (RECM) [63] and Evidential c-medoids [62]. We can also include in this list soft partition methods based on possibilistic and rough theories, such as Relational Possibilistic C-Means (RPCM) [89] and Rough Membership C-Means (RMCM) [89]. However, these methods are not studied in this review.

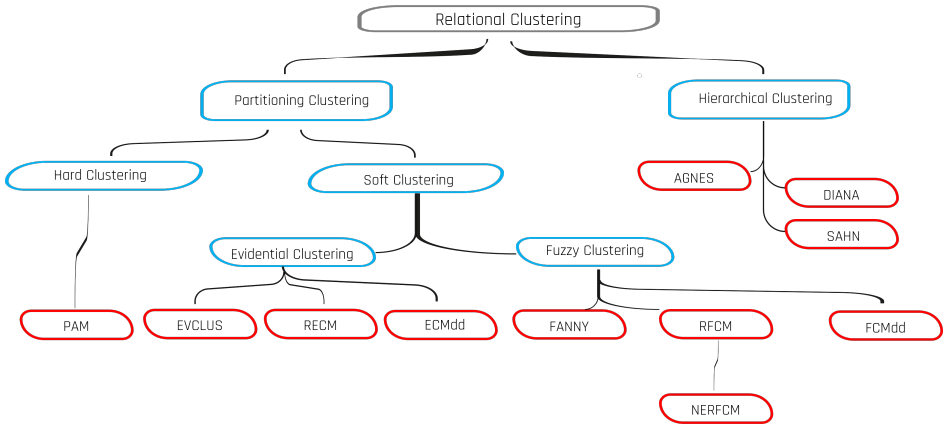


Figure 3.3: An overview of relational clustering algorithms.

3.2.3 Trajectory similarity measures

Computing the similarity between pairs of trajectories is a fundamental operation in trajectory analytics that is necessary for distance-based clustering of trajectories (view in Section 3.2.2). The similarity is used to determine the distance between two trajectories. Similarity can be defined as the cost of transforming one trajectory into another based on a specific similarity function. In the literature, there are many papers on comparative study of trajectory similarity measures that allow us to observe thus the advantages and disadvantages of these measures.

Prior research has reviewed four basic measures, including Euclidean distance (ED), Dynamic Time Warping (DTW), Longest Common Subsequence (LCS), and Fréchet distance [126]. More recent work presents an empirical comparative evaluation of popular trajectory similarity functions, observing the advantages and disadvantages of these measures [127, 115]. Comparisons are based on several criteria, including metric and non-metric properties. A distance function is metric if it respects uniqueness, symmetry, non-negativity and triangle inequality. Another distinction is between discrete and continuous measures, depending on the format of the trajectories. Finally, criteria such as point offset or length variation, as well as robustness to noise, outliers and missing data, are also taken into account. These works also highlight the comparison in terms of computational cost, memory usage, accuracy, and the amount of data to be processed beforehand to determine their suitability for static or flow extraction applications. In paper [128], the authors begin with a thorough conceptual and theoretical comparison. This comparison highlights the similarities and differences between the measures in terms of the different characteristics above.

We propose a classification of the most important and commonly used similarity measures based on the type and format of the trajectories (see Figure 3.4). For distance measures within the same category, theoretically, they have the same targeted information type (continuous or categorical) and the same requirement of trajectory data type (sequential, temporal only, or spatio-temporal trajectories).

It is worth noting that this classification is not unique, and many other aspects can be used to connect and distinguish those distance measures. For example, ED [129], DTW [130], STLC (spatio-temporal linear combine) [11], Fréchet distance [131], spatio-temporal Euclidean distance (STED)[129], and spatial locality in-between polylines (SLIP) are all derived by calculating the Euclidean distance between certain parts of the given two trajectories. While edit distance on real sequence (EDR) [13], edit distance with real penalty (ERP)[132], edit distance with projections (EDwP) [133], LCS, and spatio-temporal longest common subsequence (STLCSS) [134] actually reflect the proportion of the two trajectories that are close to each other based on a distance function. From the perspective of the use of distance functions, they can be grouped into the L_{p-norm} family (ED, optimal matching (OM), DTW, STLC, Fréchet distance, STED and SLIP) and edit distance family (EDR, ERP, EDwP, LCS and STLCSS).

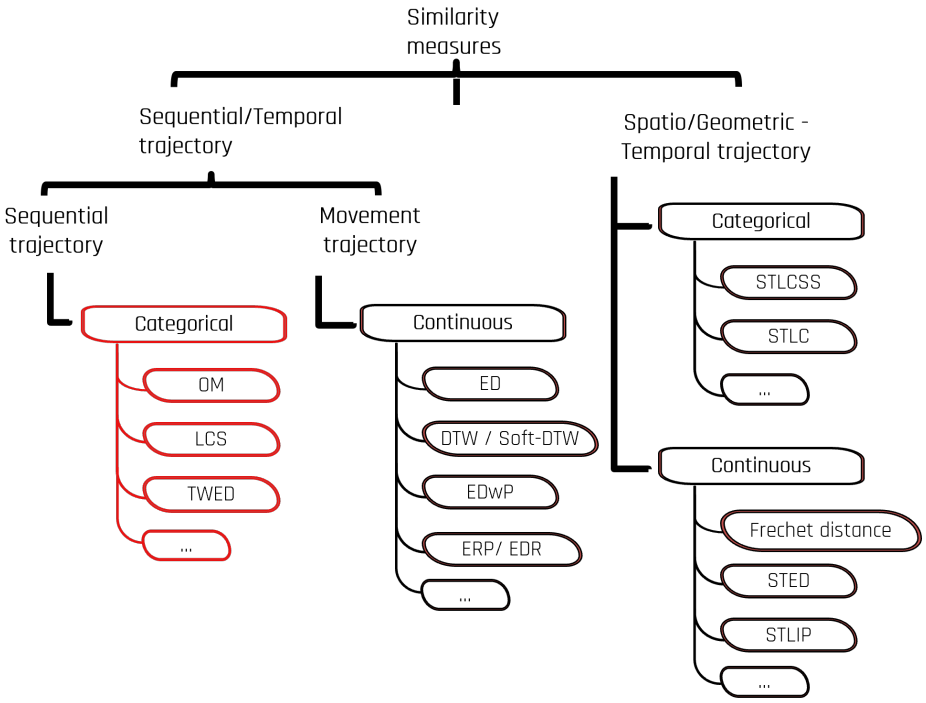


Figure 3.4: Categorization of trajectory distance measures.

3.3 Review of sequential trajectory clustering

3.3.1 Related sequential trajectory data

Sequential trajectories or categorical sequences are widely used in various fields to describe the evolutionary state of the process under study. In the social sciences, sequences are defined through longitudinal categorical or discrete data. In computer science, sequential trajectories are defined as categorical time series. Longitudinal and time series data are confusing when it comes to analyze trajectory data. There are no strict and formal definitions on which a large number of data analysts agree. In the literature, we have noted that there are few papers on the topic. And in those few papers that mention the topic, the definition is only briefly [135]. Based on our review, we propose the following definitions illustrated in Figure 3.5. Time series data refer to observations (values) of a single object collected over an extended time period. Cross-sectional data, in contrast, consists of observations (values) of multiple objects at a single time point. Longitudinal data (or panel data), involving multiple objects observed at multiple time points. Indeed, longitudinal data are often referred to as cross-sectional time series, because they incorporate temporal information based on multiple objects.

Longitudinal data, like time series, can be structured as a three-dimensional tensor (or data cube) defined by items N, variables M, and time points T, where each cell $X_{jt}^{(i)}$ represents a measurement for objet i at time t on feature j [136]. Both longitudinal and time series data can be balanced or unbalanced ($T^{(i)} = T$), spaced equally or unequally in time, and impacted by missing data, with $T^{(i)} \leq T$. Specific properties of longitudinal data allow for a lag, or the temporal difference $\left|T_t^{(i)} - T_l^{(i)}\right|$ between two objects. Longitudinal data can be

formatted as long or wide. Wide format accommodates balanced data and places all subject i data in one row, while long format uses one row per observation. When $M > 1$, we refer to multidimensional (in contrast to unidimensional) longitudinal data or multivariate (versus univariate) time series. In this case, the trajectories studied are multidimensional sequential trajectories. However, in this section, we focus first on one-dimensional sequential trajectories for $M = 1$.

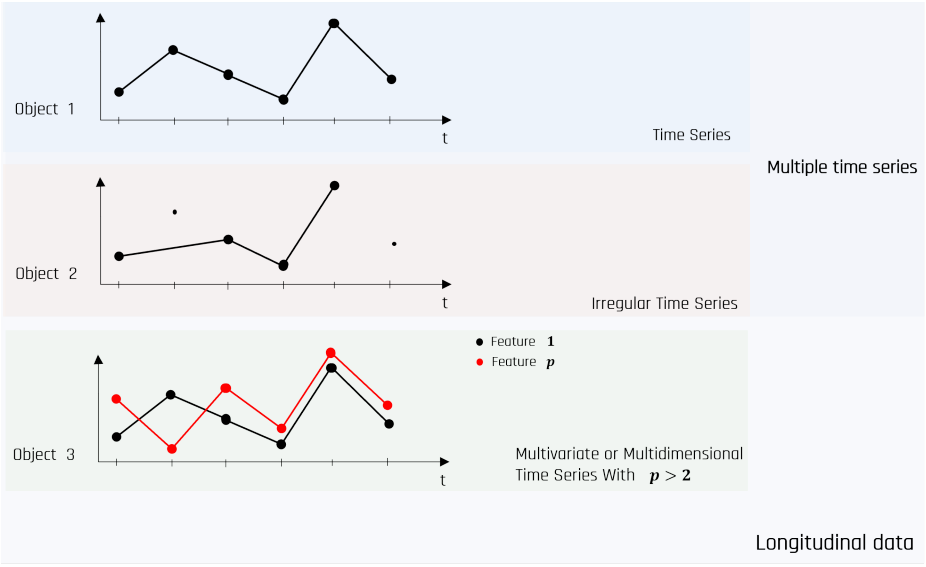


Figure 3.5: Illustration of categorical or discrete time series and longitudinal data.

3.3.2 Related works on sequential trajectory clustering

Model-based clustering approachs was early work for sequential trajectory clustering. Commonly used techniques include latent class modeling approaches, i.e. growth mixture modeling (GMM), group-based trajectory modeling (GBTM), latent class analysis (LCA) and latent transition analysis (LTA) [33]. Indeed, latent class modeling approaches have long been used in the social sciences [16] and, increasingly, in healthcare. For example, Deville-Stoetzel, N and al. [137], used latent class analysis to identify patient profiles patterns based on their experience of access and continuity at team-based primary healthcare clinics in Canada. Other model-based clustering approach including Markov models (MM), have been studied in the literature [138]. For Markov model-based clustering approaches, the model type is often specified a priori, such as hidden Markov models (HMMs), variable-length Markov chains (VLMCs), double-chain Markov models (DCMMs) or mixture transition distribution (MTD) models. The model structure (e.g., the number of hidden states in an HMM) can be determined by model selection techniques and parameters estimated using maximum likelihood algorithms.

Feature-based clustering approaches are often explored first for trajectory clustering, because they simplify the complexity related to the temporal dimension. Feature extraction transforms data into a cross-sectional tabular format. Although most extraction methods are generic, the specific features extracted generally depend on the application [124].

For sequential trajectories, extracted features can include entropy measures such as Gini dispersion, Shannon, Chebycheff, and other measures such as, uncertainty coefficient, Pearson measure, Sakoda measure, Φ^2-measure, see [139] for a detailed review of candidate features. Feature-based clustering approaches use both existing object-based clustering algorithms and existing relational clustering algorithms.

State-of-the-art of Row-data-based clustering of sequential trajectory shows that the most popular works are often extensions of traditional clustering algorithms to consider the temporal dimension (object-based clustering), or relational clustering algorithms with adapted similarity measures[32, 33]. Dias and Cortinhal introduced the sequential K-means (SKM) [140], an extensions of k-means for categorical time series, which iteratively assigns each object to the nearest cluster centroid represented by a transition probability matrix based on the symmetric Kullback–Leibler distance. Although numerous works on the adaptation of traditional clustering algorithms to various types of trajectories characterized by continuous data can be found in the literature, for example longitudinal k-means [141], sequential trajectories remain under-explored and need more attention.

Table 3.1 summarizes the strengths and limitations of the model-based, feature-based, and row-data-based clustering approaches, highlighting the key advantages and challenges of each method.

Approach	Strengths	Limitations
Model-based	• Directly models the underlying dynamics. • Good at handling noise and uncertainty using probabilities.	• Assumptions about the model can be too strict. • Sensitive to model misspecification. • Can be complex to estimate in some cases.
Feature-based	• Extracts useful features (statistical summaries, metrics) to simplify comparison. • Reduced dimensionality helps with analysis and interpretation.	• May lose fine dynamic or structural details. • Clustering quality depends heavily on the chosen features. • Requires a solid feature engineering phase.
Row data-based (Including distance-based and object-based)	• Uses raw data, preserving all inherent information. • Flexible in choosing similarity measures (distance, alignment, etc.).	• Computation-intensive, especially for large datasets or long series. • Sensitive to noise and outliers. • Needs careful selection of the distance measure to properly reflect similarity.

Table 3.1: Strengths and limitations of sequential trajectory clustering approaches.

In this review, we focuse on Row data-based clustering of sequential trajectory using relational or proximity data (distance-based methods) and we propose a comparative framework of multidimensional sequential trajectories clustering. In a distance-based cluster approach, trajectories are clustered based on their pairwise similarity, as measured

by a user-specified dissimilarity metric, i.e., distance measure. This approach comprises two main steps. The first step consists of a sequential analysis, including the extraction and processing of sequences and the calculation of proximity data between sequences (relational data). This enables fast experimentation with different measures of similarity suitable to the application at hand. The main goal of sequence analysis is the knowledge discovery from event or state sequences describing trajectories. The second step consists in applying a clustering algorithm based on these relational data. The advantage of a distance-based approach is that domain knowledge can be taken into account in specifying the distance measure to capture the relevant properties of the trajectories. Another advantage is that the pairwise distances between trajectories yields a hierarchy which provides additional information on the sequential trajectory.

In the next subsections, we describe the ontology of processing, similarity and clustering of sequential trajectories. These steps involve sequential trajectory data processing (Section 3.3.2.1), the definition of similarity functions (Section 3.3.2.2), and the application of relational clustering methods (Section 3.3.2.3). The overall framework summarizing these stages is illustrated in Figure 3.6.

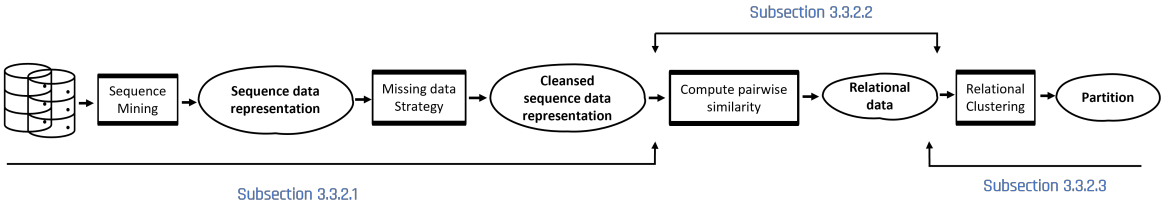


Figure 3.6: General framework of raw-data distance-based sequential trajectory clustering.

3.3.2.1 Sequential trajectory data processing

Sequence mining and/or sequence construction from categorical or discrete longitudinal or time series data is the first step in the sequential trajectory analysis. But in many instances, sequences are not directly present or visible in the data, and have to be reconstructed. Chongming Gao and al. [142], consider that extracting knowledge from trajectory data is the most important step, and consists in generating a good representation of the trajectory. Sequence construction is a laborious task, due to the nature of trajectory data. Moreover, the literature contains few references to help identify the main formats and types of structure of sequential data, [143] representing one of the few exceptions. The following ontology can be used to define and describe the main formats and representations of sequences. Indeed, the main difference between sequences is that they contain states or events [32]. Both are referred to as status at a point in time. It is also important to determine whether status alignment is based on an internal temporal reference (where temporality is explicitly defined) or on an external reference (where temporality is missing). Based on the characteristics of trajectory source data, we can highlight the following commonly used sequential representations [144, 145]:

- *The states sequence (STS) format:* STS is an intuitive and common representation where successive states (statuses) of an individual are listed in consecutive columns. Each

column typically corresponds to a predetermined time unit, but it can also handle sequences without time references.

- *The state permanence sequence (SPS) format:* Proposed by Aassve et al. in 2007 [146], in SPS format, each distinct state in the sequence is paired with its duration. In one variant, state/duration pairs are enclosed in parentheses.
- *The distinct successive state (DSS) format:* DSS provides more compact representations by listing only one of several identical successive states.
- *The vertical time-stamped event (TSE) format:* TSE lists the events experienced by an individual along with the time of occurrence, allowing sequences of events to be constructed easily from this format. Events are listed from the first significant change of state, instead of starting with the initial state.

Consider the treatment trajectory of patient i with a serious chronic pathology, where their care pathway is tracked monthly for one year from diagnosis. The variable j records the patient's care status for each month of follow-up. Different representations of their treatment sequence $S_j^{(i)}$ are shown in Figure 3.7. Each state in the sequence is chosen from a finite domain or alphabet $\Sigma = \{D, C, T, S\}$, which defines the set of all possible care statuses that a patient can experience during follow-up. $|\Sigma|$ is the number of states in the alphabet, i.e. the cardinality.

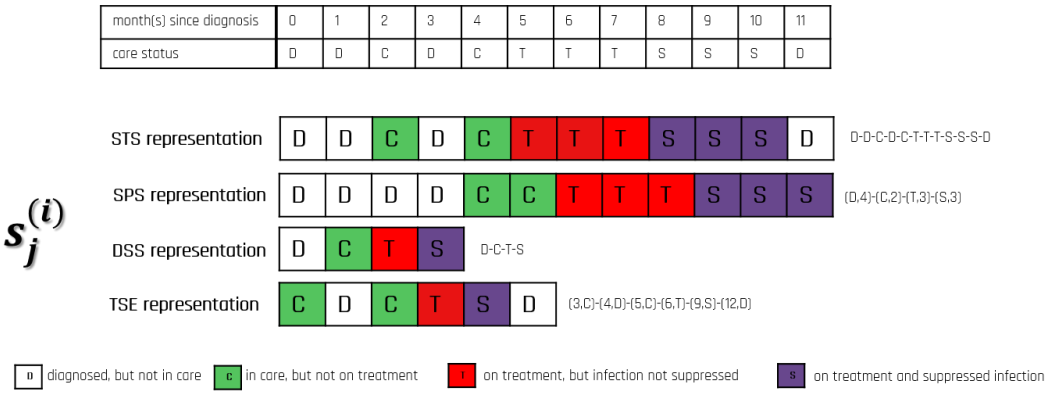


Figure 3.7: Examples of Sequence data representations.

In sequential analysis, it has been shown that the methods used to handle missing data have a significant impact on the results. For sequential clustering, the choice of imputation method directly affects the dissimilarity measure and can even determine which measure should be used. The three most common strategies for addressing missing data in sequences are: treating missing data as an additional state in the alphabet, imputing it, or simply deleting any missing data. The first strategy is generally applied to non-random missingness, where its presence is meaningful, as opposed to missing completely at random or missing at random [147]. Gap coding treats missing values as distinct states, creating a separate alphabet for them. Imputation can be simple (mean, median, mode) or multiple. Multiple imputation, based on predictive methods such as random forest or multinomial regression models, considers the longitudinal nature of the data, allowing covariates, including time-dependent ones, to be included in the model.

3.3.2.2 Sequential trajectory similarity functions

A measure of sequential dissimilarity quantitatively assesses the degree of dissimilarity between two sequences, providing insights into their resemblance. Sequential trajectory dissimilarity measures can be classified into three categories, [32]: i) Distribution-based measures; ii) Attribute-based measures counting common attributes across sequences; and iii) Edit distances, which assess the cost of operations required to transform one sequence into another. In this section, we introduce these three families of similarities, highlighting their objectives and properties, including state distribution (*Distribution*), state duration (*Duration*) and state order (*Sequencing*) in sequences. We can also include the number of distinct states in the sequence (*Experience or Alphabet*) and the time of appearance of states within the sequence (*Timing*) [148].

Distances between probability distributions. The first approach distributions-based used to measure the dissimilarity between sequences is based on Chi-square (Chi) [149], i.e. Chi distance between state distributions, and focuses on the longitudinal state distribution within each sequence. Indeed, each sequence is represented by a distribution vector. For example, in the example sequence in Figure 3.7, the state distribution is $(4/12, 2/12, 3/12, 3/12)$. Given these distribution vectors, the dissimilarity between two sequences is evaluated via Euclidean or Chi-square distance. The squared Chi-square measure weights the squared differences for each state by the inverse of the overall proportion of time spent for the state, gives more importance to a rare state than to a frequent state. Let $p_{k|s_1}$ be the proportion of time spent in state k in sequence s_1, and p_k the overall proportion of time spent in state k, the Chi distance reads as follows:

$$d_{Chi}(s_1, s_2) = \sum_{k=1}^{|\Sigma|} \frac{(p_{k|s_1} - p_{k|s_2})^2}{p_k}. \tag{3.1}$$

Chi-square is sensitive to the duration parameter and robust for sequencing and timing. Several improvements to address these limitations have been developed [149]. Other related measures based on the Chi-square measure have also been proposed, including distance based on conditional distributions of subsequent states. This was proposed by [150] and defined as the sum of position-dependent distances computed at successive positions.

Distances based on counts of common attributes or attribute-based. These are dissimilarity measures based on the common subsequences which LCS measure. They are adapted to handle transitions between state while they take into account the order in the sequence. They are also able to handle sequences of different lengths. If $A(s_1, s_2)$ is the number of common attributes, between sequence s_1 and s_2 then the sequences are close for a large value of $A(s_1, s_2)$ and dissimilarity measure is as follows:

$$d(s_1, s_2) = A(s_1, s_1) + A(s_2, s_2) - 2A(s_1, s_2). \tag{3.2}$$

The dissimilarity is maximal when $A(s_1, s_2) = 0$, i.e., when the two sequences have no common attribute. We can mention two attribute-based dissimilarities introduced by C. Elzinga [151] in Sequence similarity: the longest common prefix (LCP) and the longest common subsequence (LCS). LCP use the number of successive common positions starting from the beginning of the sequences, that can be noted $A_p(s_1, s_2)$. A derivative

of LCP is the reversed longest common prefix or longest common suffix (RLCP). It looks for the common elements from the end rather than from the beginning of the sequences. For LCS, it use the number of common consecutive states in the subsequence $A_s(s_1, s_2)$. Thus in the same way, [152], proposed Simple Hamming distance which measuring the dissimilarity between two sequences by the number of positions with non-matching states. It applies only to pairs of sequences of the same length and is very sensitive to timing mismatches. We get the Hamming distance with Equation 3.2 by using $A_h(s_1, s_2)/2$ as proximity measure, where $A_h(s_1, s_2)$ is the number of no-matching positions.

Edit distances or optimal matching (OM). It was first introduced in sequence analysis by [153]. OM transforms the sequence s_1 into the sequence s_2 using three possible operations: insertion, deletion, and substitution. Dissimilarity is computed as the cost associated to the smallest number of operations. Let $A_i = A_1, ..., A_\ell$ be a set sequence of actions (insertion, deletion and substitution) of length ℓ to transform the sequence s_1 to s_2. And $A_i(E^k, E^{k'})$ is the action for aligning E^k, the k^{th} state of s_1 against $E^{k'}$, the k'^{th} state of s_2. We can use several possible action sequences to match the two sequences. Considering this, let $A_i^j = A_1^j, ..., A_{\ell_j}^j$, be the set of all possible transformations and $\gamma(A_i^j) \in \mathbb{R}_0^+$ the cost of each action or elementary transformation. The OM dissimilarity is then :

$$d_{OM}(s_1, s_2) = \min_j \sum_{i=1}^{\ell_j} \gamma(A_i^j). \tag{3.3}$$

The method seems simple and relatively intuitive, but the choice of the costs is a delicate operation. This topic is subject to lively debates in the literature [154] mostly because of the difficulties to establish an explicit and sound theoretical frame. Generally insertion and delete actions have the same cost and are called indel action. Substitution costs can be theoretical or empirical, for the empirical case, the costs are determined from the estimated transition rates, a detailed knowledge is provided in [154]. Among the main variants of OM, we can mention Generalised Hamming distance and Levenshtein II. The generalised Hamming distance does not allow indels while Levenshtein II distance discard substitutions and measure the distance by counting the number of indels necessary to transform one sequence into another. Dynamic Hamming distance (DHD), proposed by [155], focuses on OM without indels, such as generalized Hamming, and proposed that substitution costs should depend on the position t in the sequence. Localised OM, proposed by [156], aims to make indel costs dependent on the two adjacent states. The motivation is that inserting or deleting a state similar to its neighbors would only change the length of the spell in that state, without affecting the sequencing. We can also cite OM sensitive to spell length by [157] and OM between sequences of transitions by [158]. Other edit distances have been proposed for specific sequences. For example, the Time Warp Edit Distance (TWED) is a distance measure for matching discrete time series with temporal elasticity [130].

3.3.2.3 Relational clustering of sequential trajectory

Sequential analysis allows the construction of trajectories from categorical longitudinal data or categorical time series (sequential processing) and the measurement of similarities between these trajectories (sequential similarity), as shown in Figure 3.8. Relational clustering allows to group objects from the distance matrix between trajectories obtained

with the dissimilarity measure. In the literature, sequential trajectory clustering are often based on algorithms like Ascending Hierarchical Classification (CAH) with usually Ward aggregation method or Partitioning Around Medoids (PAM) [32], and are both relational clustering techniques using medoids. In this work, we include two soft relational partitioning algorithms, fuzzy and evidential clustering, presented in Figure 3.3. We used medoid-based versions of soft clustering algorithms such as relational fuzzy c-medoids (FCMdd) [64] and relational evidential c-medoids (ECMdd) [62]. Medoid-based clustering offers the advantage of not requiring the distance or dissimilarity matrices between trajectories to be Euclidean, as is the case with mean-based clustering approaches or c-mean relational algorithms. Indeed, this will enable us to explore and use several optimal distance functions for each trajectory dimension.

3.4 Review of multidimensional sequential trajectory clustering

Figure 3.8 illustrates the clustering process of sequential trajectories, from raw longitudinal data to the final partitioning. The framework integrates both unidimensional and multidimensional sequential trajectory clustering (MST). It follows a structured pipeline, starting with data preprocessing, followed by similarity computation, and finally relational clustering and cluster validation.

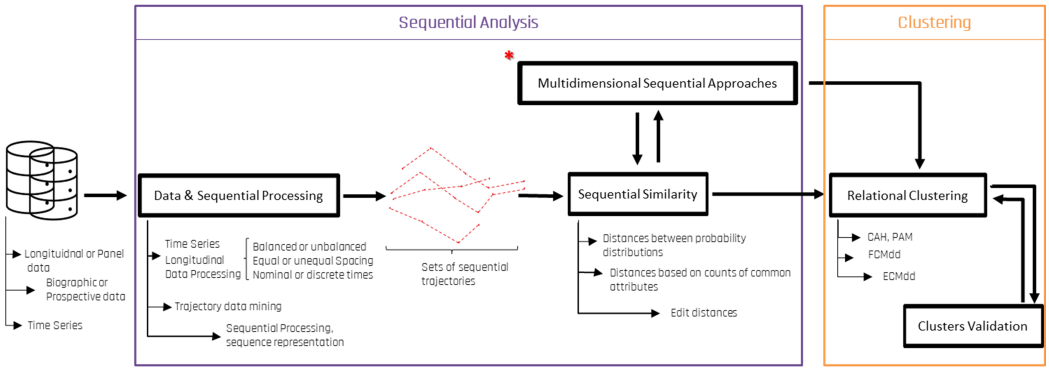


Figure 3.8: Illustration of the clustering process of sequential trajectories.

3.4.1 Definitions and concepts of multidimensional sequential trajectory

Multidimensional sequential trajectories (MST) refer to the simultaneous evolution of multiple events or states across several categorical or discrete longitudinal datasets or time series for a given set of objects. In fact, an MST includes multiple categorical sequences. Let χ be a three-tensor multichannel sequence data constituted of a set of N individuals in $\mathbb{R}^{N\times M\times T}$. The trajectory $\tau^{(i)}$ of an individual i can be defined as the set of sequences S_j (see Figure 3.9), with $j = 1, .., M$. The sequence S_j of the variable j is a list of states or events $E_t^{(k)}$, ordered by $t = 1, ..., T^{(i)}$ chosen from a finite alphabet Σ (sequence domain), with $k = 1, ..., \Sigma$. An alphabet is the set of all possible states that can appear in the sequences.

A sequence S_j can be represented as a succession of pairs (E_t, T_t), with E_t representing a state and T_t a date of state measurement.

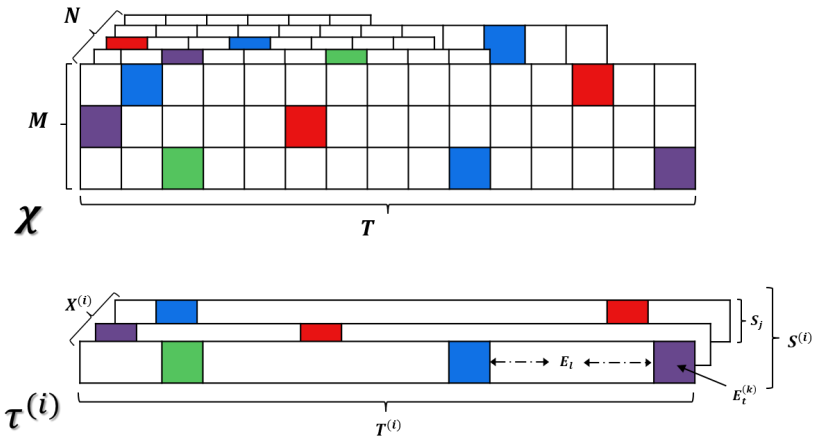


Figure 3.9: Illustration of a sequence extraction ($S^{(i)}$) and a trajectory or pathway construction ($\tau^{(i)}$) from multidimensional longitudinal data χ.

3.4.2 Review of row-data distance-based clustering for MST

In multidimensional sequence analysis (processing multiple sequences and/or computing similarities between them), it is often complex to simplify successive sequence elements into a unique and limited set of states. Methodological adaptations is then necessary to handle the diversity and complexity of the status [112]. The concept of measuring similarity in sequences has traditionally focused on unidimensional data [159]. However, the extension to multidimensional sequences has been explored, and adapted measures, such as LCS or DTW are used to compare multidimensional sequential trajectories. Clustering methods for multidimensional sequential trajectory found in the literature can be grouped into five main categories.

Combination of states (Cstat). Cstat consists of creating a new state variable that combines the simple states composing each dimension [112]. For example, in the case of the care trajectories defined in Section 3.3.2.1, one can notice that the care status variable is a result of combinations of states. Possible non-combined states would include: diagnosed/not diagnosed, on treatment/not on treatment; suppressed infection/infection not suppressed, in care/not in care. But in certain cases, this can quickly lead to a large alphabet, i.e. a very large set of states. Thus, in the case of four dimensions each containing three simple states, the combined variable would potentially have $3 \times 3 \times 3 \times 3 = 81$ states. Such an extension of the alphabet may be impractical with optimal matching when it comes to setting substitution costs specifically adapted to each pair of states.

Combination of costs (Ccost). Ccost method, most common and often called multiple sequence analysis (MSA) or multichannel sequence analysis (MCSA), is based on combining the costs of substitution matching actions between different dimensions. A possible combination is to sum or average the costs defined for each dimension [112], or to apply weighted sum [7]. In [160], MCSA was used to identify six life trajectory clusters of 304

students transitioning to adulthood based on Education, Career, Partnership, and Parenthood data. Substitution costs were defined separately for each variable and averaged for overall analysis. In practice, taking into account multiple dimension within an OM framework, as defined by equation 3.3, is relatively simple and only requires adapting the substitution costs and the indel costs, so that they reflect the relationship between equivalent dimension. The multichannel version of these terms can be expressed as follows:

$$\Gamma^{sub}(E^k, E^{k'}) = \frac{\sum_{c=1}^{N^c} \gamma^{sub^c}(E^k, E^{k'})}{N^c}; \quad \Gamma^{ind}(E^k, E^{k'}) = \frac{\sum_{c=1}^{N^c} \gamma^{ind^c}(E^k, E^{k'})}{N^c}. \tag{3.4}$$

with N^c the number of channel or dimension, and $(E^k, E^{k'}) \in \Sigma^c$ the alphabet of each dimension studied. γ^{sub^c} and γ^{Ind^c} are respectively the specific substitution and indel cost matrices associated with each dimension studied.

Combination of distance (Cdist). Cdist method consists of computing a distance matrix for each sequence dimension separately, then combining them into a single distance matrix by a linear combination [15]. In [161], Philippe Blanchard used Cdist method to study the socio-political careers of activists in the main French association devoted to anti-Aids (AIDES) based on questionnaire data. OM method was used to calculate similarity matrices for each sequence dimension.These matrices are aggregated by a sum function into a single matrix used in relational clustering.

Combination of clusters (Cclust). Cclust method uses sequence clustering created separately for each sequence dimension and then deriving the joint final cluster from clustering constructed for each dimension [15]. This approach is not widely adopted because of its limites to generate a large number of clusters, which quickly become sparse as the number of dimensions increases.

Globally Interdependent Multiple Sequence Analysis (GIMSA) [14]. GIMSA method computes a distance matrix for each dimension, then uses multidimensional scaling (MDS) followed by multiple factor analysis techniques like symmetrical or canonical Partial Least Squares (PLS). First, Optimal Matching (OM) or another dissimilarity measure calculates distances within each dimension. Next, MDS reduces the dimensionality of each dissimilarity matrix, mapping objects while preserving pairwise distances. Given a multiple dissimilarity matrix **D**, MDS finds the largest eigenvalues of each matrix and, symmetric PLS combines them to maximize covariance, based for their linear relationships and structural strength.

3.4.3 Characteristics and taxonomy

Cstat, Ccost, Cdist, Cclust, and GIMSA methods can be systematically compared and classified according to the criteria multidimensionality, parsimony and interdependence [32] in Table 3.2. Multidimensionality refers to the explicit and flexible contribution of each sequential dimension to the final clustering results. Cstat method masks dimensions by combining multiple sequences into a single one. For example, it is not possible to have specific parameters for each dimension, or to evaluate the impact of each dimension on the results. Parsimony refers to the question of whether a method can produce a limited number of trajectory clusters. Cclust method leads to a high number of clusters. Interdependence

refers to the consideration of relationships between dimensions. In Cdist, the relationship between multiple sequences is hidden. Methods Cstat and Ccost, focus on the dependency between dimensions transversely, at each point of the sequence. In Clust and GIMSA methods, the focus is on the dependency between dimensions via global sequences. We can also categorize methods according to the type of approach. This is the step at which multidimensionality is taken into account in the diagram in figure 3.8. Distance matrices are computed independently for each dimension in Cdist, Cclust and GIMSA, and different dissimilarity measures can be used. The dissimilarity matrix for each sequence can be computed based on the same distance function. Different distance functions can be used for each sequence dimension [32]. For the Ccost method, it is relatively related to the OM distance family.

Table 3.2: Taxonomy of multidimensional sequence analysis strategies.

Methods	Multidimensional	Parsimony	Interdependence	Type
1 - Combination of states (Cstat)	**No**	Yes	local	Processing
2 - Combination of costs (Ccost)	Yes	Yes	local	Similarity
3 - Combination of distance (Cdist)	Yes	Yes	**No**	Clustering
4 - Combination of clusters (Cclust)	Yes	**No**	global	**No**
5 - Globally Interdependent Multiple Sequence Analysis (GIMSA)	Yes	Yes	global	Similarity

3.5 Experimental comparison of multidimensional methods

3.5.1 Experimental setting

To compare the performance of the five MST clustering methods, both synthetic and real datasets described by categorical, ordinal, and discrete sequences are considered. Several dissimilarity matrices were computed in line with the recommendations of the five approaches (Section 3.4) to account for multichannel sequences. Among the most commonly used methods for treating missing data in sequences, described in Section 3.3.2.1, multiple imputation via the random forest model or adding a new status for missing values were explored. Motivated to the literature [7], we use the following dissimilarity measures. For synthetic data, we use optimal matching with substitution costs based on transition rates. For real data, the optimal distance for each dimension is selected from among OM, HAM, TWED, LCS, CHI2, and EUCLID. All matrices were normalized based on overall dispersion, i.e. in each matrix, dissimilarities $d(e_k, e_{k'})$ are normalized as $d(e_k, e_{k'})/T$, where $T = \sum_{k=1}^{n} d(e_k, g)$, and g is the overall prototype of the matrix, computed as $l = argmin_{1 \leq h \leq h} \sum_{k=1}^{n} d(e_k, e_h)$.

Relational clustering algorithms, as described in Section 3.3.2.3, are used to partition the dissimilarity matrices generated by each approach, according to 3.8 workflow. The relational hard clustering algorithms CAH and PAM is applied to these proximity data to obtain a set

of groups $\mathscr{G} = (G_1, \ldots, G_K)$. FCMdd is also applied to these matrices to initially produce a fuzzy partition into K fuzzy clusters. Subsequently, a hard partition $\mathscr{G} = (G_1, \ldots, G_K)$ is derived from this fuzzy partition by defining the hard cluster $G_k(k = 1, ..., K)$ as $G_k = \{\mathbf{x}_i : u_{ik} \geq u_{im} \quad \forall m \in \{1, \ldots, K\}\}$. ECMdd is similarly applied to these similarity matrices to initially establish a credal partition into 2^K credal clusters. Then, a hard partition $\mathscr{G} = (G_1, \ldots, G_{2^K})$ is derived from this credal partition by a pignistic transformation [62]. The quantity u_{ik} represents the membership degree of object $\mathbf{x}_i(i = 1, \ldots, n)$ in fuzzy cluster $k(k = 1, \ldots, K)$. The quantity m_{ik} indicates the membership belief of object $\mathbf{x_i}(i = 1, \ldots, n)$ in credal cluster $k(k = 1, \ldots, 2^K)$. Each relational clustering algorithm was run (until the convergence to a stationary value of the adequacy criterion) 50 times and the best result was selected according to the adequacy criterion (minimum of the objective function).

To compare the clustering results, an external criteria, such as the corrected Rand index (CR) [162], and internal criteria, like the Average Silhouette Width (ASW) and Hubert's C (HC) index [163], are considered. Clustering validation criteria include CR $[-1, 1]$ and ASW $[-1, 1]$, both maximized for better clustering, and HC $[0, 1]$, minimized for closer alignment with the ideal partition. The experiments were conducted on a high-performance computing (HPC) server. The server is equipped with two AMD EPYC 7452 32-Core processors, offering a total of 64 physical cores with hyper-threading, and 512 GB of RAM. These powerful hardware specifications enabled efficient and accurate execution of the experiments, ensuring reliable results. The programs were implemented using R version 3.6.3.

3.5.2 Synthetic categorical longitudinal data

We simulated data for two multidimensional sequential trajectories characterized by longitudinal categorical and/or ordinal data sets. First one dataset includes two independent variables, a three-modality categorical variable and a four-modality ordinal variable, while the second one contains four variables comprising two correlated ordinal variables with three and four modalities and two independent categorical variables with three and four modalities. Each data set contains 150 objects split into four groups of different sizes, distributed according to distinct probability distributions. More details on the model simulations are available in Section B.1 of Appendix B.

We build the sequences of each trajectory, containing 2 and 4 sequences respectively. Dissimilarity between pairs of sequences is calculated using the OM measure for both datasets. Depending on MST, one or more dissimilarity matrices are calculated for each trajectory. For the Cstat, Ccost and GIMSA methods, a single dissimilarity matrix is obtained. Cdist and Cclust, in contrast, generate one dissimilarity matrix per sequence. The CAH, PAM, FCMdd and ECMdd clustering algorithms were applied to the dissimilarity matrices to obtain a four-cluster partition, whether hard, fuzzy or credal. The hard cluster partitions (obtained from the fuzzy or credal partition transformation, or obtained directly from CAH and PAM) were compared with the class partition known a prior. Table 3.3 shows the performance of the CAH, PAM, FCMdd and ECMdd clustering algorithms for each of the Cstat, Ccost, Cdist, Cclust and GIMSA multidimensional sequential approaches. Performance was evaluated using the CR index.

Table 3.3: Performance of the multidimensional sequential clustering approaches on the synthetic datasets: The Rand index.

Methods	Synthetic dataset 1				Synthetic dataset 2			
	CAH	PAM	FCMdd	ECMdd	CAH	PAM	FCMdd	ECMdd
Cstat	0.583	0.605	0.554	0.628	0.582	0.588	0.572	0.607
Ccost	0.528	0.612	0.282	0.645	0.576	0.596	0.496	0.612
Cdist	0.593	0.674	0.500	0.602	0.590	0.606	0.471	0.663
Cclust	0.575	0.614	0.502	0.637	0.529	0.612	0.501	0.649
GIMSA	0.594	0.594	0.567	0.502	0.529	0.608	0.507	0.643

Table 3.3 shows that Cdist method performed the best on both synthetic datasets, achieving the highest values of the Rand index. This means that this method successfully clustered multidimensional data in a manner consistent with the real class structure. Among the algorithms used by the Cdist method, PAM performed best on synthetic dataset 1, with a value of 0.674, while ECMdd performed best on synthetic dataset 2, with a value of 0.663. These algorithms were able to find the optimal cluster medoids, taking into account the different dissimilarities between the variables. The other methods and algorithms exhibited lower or similar performance, depending on the variance between variables and between classes.

3.5.3 The biofam dataset

The biofam dataset, created using information from the Swiss Household Panel's 2002 retrospective biographical survey by [164], contains individuals aged 30 or over at the time of the survey. The dataset examines trajectories of family life between the ages of 15 and 30, focusing on aspects such as residence status, marital status, and parenthood. This random sample of 2,000 individuals provides 16 years of follow-up per individual, resulting in 32,000 observations for each of the three dimensions.

Muller et al. [164] used this dataset for life course classification with optimal matching (OM) and identified five optimal clusters. In this study, we aimed to obtain a five-class clustering using two different configurations for dissimilarity calculation. The first configuration applied standard OM across all dimensions, while the second used LCS for residence status, CHI2 for parenthood, and OM for marital status. This choice is motivated by the characteristics of the variables and similarity measures. The CAH, PAM, FCMdd, and ECMdd clustering algorithms were applied to the dissimilarity matrices to generate hard, fuzzy, and credal partitions into five clusters, which were compared through internal performance criteria, ASW and HC. Table 3.4 and Table 3.5 show the performance of the CAH, PAM, FCMdd and ECMdd clustering algorithms for each of the Cstat, Ccost, Cdist, Cclust and GIMSA multidimensional sequential approaches. Performance was evaluated using the ASW and HC index.
Tables 3.4 and 3.5 compare the performance of multidimensional sequential clustering methods on the biofam dataset. The Table 3.4 is based on OM dissimilarities for all dimensions, while the Table 3.5 uses OM, LCS and CIH2 dissimilarities for each of the three dimensions.

Table 3.4: Performance of the multidimensional sequential clustering approaches on the biofam dataset with Standard OM dissimilarity function.

Methods	CAH		PAM		FCMdd		ECMdd	
	ASW	HC	ASW	HC	ASW	HC	ASW	HC
Cstat	0.344	0.139	0.433	0.089	0.318	0.165	0.433	0.089
Ccost	0.367	0.093	0.394	0.087	0.330	0.101	0.356	0.092
Cdist	0.369	0.017	0.404	0.025	0.376	0.097	0.336	0.037
Cclust	0.363	0.091	0.399	0.083	0.337	0.096	0.403	0.079
GIMSA	0.339	0.097	0.362	0.156	0.260	0.252	0.357	0.154

Table 3.5: Performance of the multidimensional sequential clustering approaches on the biofam dataset with Standard OM, LCS and CHI2 dissimilarities functions.

Methods	CAH		PAM		FCMdd		ECMdd	
	ASW	HC	ASW	HC	ASW	HC	ASW	HC
Cdist	0.474	0.042	0.508	0.048	0.515	0.055	0.687	0.097
Cclust	0.405	0.059	0.454	0.041	0.410	0.051	0.322	0.074
GIMSA	0.325	0.102	0.362	0.156	0.260	0.252	0.209	0.150

The PAM and ECMdd algorithms display the highest performance in terms of silhouettes (ASW) and stability (HC) of the final partitions. For the Cstat method, PAM and ECMdd show similar ASW and HC values, at 0.433 and 0.089 respectively. In contrast, for the Cdist method and the ECMdd algorithm, these values are 0.687 and 0.097 respectively. Considering the ASW and HC performance criteria separately, the Cstat and Cdist methods perform best in the 3.4 table, while in the 3.5 table, the Cdist method stands out.

3.5.4 American youth (AY) dataset

The National Longitudinal Survey of Youth (1979 – 2012) is a project tracking the transition to adulthood trajectory of young Americans on various life aspects from 1979 to 2012 [165] with an annual survey. The sample includes 12,685 participants and 243,071 observations, with an average follow-up of over 19 years per individual. In this work, we identified life trajectory typologies for these young people by considering multiple dimensions, including social, economic, health, education, and employment aspects. Twelve categorical sequences, each with two to eight status, were considered. We retained data for 10,532 individuals with at least 12 years of follow-up to ensure homogeneous sequence durations, facilitating multi-channel method comparisons. Sequences for poverty and employability status had 13% and 16% missing data, respectively. In one configuration, missing data were treated as possible values, adding an NA status to the alphabet. In another, a random forest model was used to impute missing data. Standard OM (with constant substitution and insertion costs) was applied to all dimensions for similarity measures. Based on [166], life trajectories were grouped into three classes, and CAH, PAM, FCMdd, and ECMdd clustering algorithms were applied to create hard, fuzzy, and credal partitions into three clusters. The hard partitions

were compared using internal performance criteria, ASW and HC. Tables 3.6 and 3.7 present the performance of the CAH, PAM, FCMdd and ECMdd clustering algorithms for each of the Cstat, Ccost, Cdist, Cclust and GIMSA multidimensional sequential approaches using extended alphabet and multiple imputation to treat missing data. Performance was evaluated using the ASW and HC indices.

Table 3.6: Performance of the multidimensional sequential clustering approaches on the American youth dataset with Standard OM dissimilarity function with the extended alphabet.

Methods	CAH		PAM		FCMdd		ECMdd	
	ASW	HC	ASW	HC	ASW	HC	ASW	HC
Cstat	0.119	0.306	0.083	0.250	0.158	0.171	-0.042	0.644
Ccost	0.274	0.117	0.234	0.217	0.297	0.084	0.111	0.236
Cdist	0.396	0.105	0.342	0.207	0.399	0.087	0.212	0.225
Cclust	0.502	0.108	0.523	0.097	0.530	0.092	0.223	0.272
GIMSA	0.110	0.338	0.124	0.318	0.188	0.280	0.078	0.252

Table 3.7: Performance of the multidimensional sequential clustering approaches on the American youth dataset with Standard OM dissimilarity function with multiple imputation.

Methods	CAH		PAM		FCMdd		ECMdd	
	ASW	HC	ASW	HC	ASW	HC	ASW	HC
Cstat	0.133	0.291	0.037	0.300	0.178	0.170	-0.170	0.745
Ccost	0.277	0.186	0.279	0.161	0.298	0.065	0.094	0.248
Cdist	0.385	0.089	0.391	0.154	0.402	0.069	0.288	0.071
Cclust	0.480	0.096	0.495	0.095	0.497	0.089	0.199	0.226
GIMSA	0.102	0.349	0.118	0.318	0.195	0.264	0.070	0.263

Tables 3.6 and 3.7 present the performance of sequential multidimensional clustering methods on AY dataset using the OM dissimilarity function with respectively an extended alphabet and multiple imputation for the treatment of missing data. The Cclust method seems to show the best results in terms of ASW index values in both tables (0.530 and 0.497) followed by the Cdist approach. The FCMdd clustering algorithm performs best for the two approaches Cclust and Cdist in both tables, in opposition to the ECMdd algorithm which performs lowest compared to all approaches in both tables. We can also observe that although the Cclust method shows superior performance when missing data is processed with an alphabet extension (in Tables 3.6), and in Tables 3.7, methods Cdist and Cclust show improved performance, i.e. when missing data is treated by multiple imputation.

3.5.5 Primary Biliary Cirrhosis (PBC) dataset

Available on UCI Machine Learning[1], these data are from 312 patients with PBC. The initial protocol included visits at 6 months, 1 year, and with visits due to worsening

[1] https://archive.ics.uci.edu/ml/index.php

health. We search for typologies of treatment trajectories by simultaneously considering the maximum amount of categorical information to extract treatment sequences for each patient. We applied dissimilarity measures adapted to each sequence refers to [5]. Matching subsequences function was used for binary variable sequences to handle missing values and alignment issues. CHI2 was applied to sequences of categorized age and gender, considering only state frequencies. OMslen was used for other variables to account for sequence length variability. We considered survival status (alive, transplanted, deceased) as the a prior class partition to compare multidimensional sequential methods [167]. Performance was evaluated using the CR index. The CAH, PAM, FCMdd, and ECMdd algorithms were applied to dissimilarity matrices, generating hard, fuzzy, and credal partitions into three clusters. The hard partitions were compared with the known a priori class partition. Table 3.8 shows the performance of the CAH, PAM, FCMdd and ECMdd clustering algorithms for each of the Cstat, Ccost, Cdist, Cclust and GIMSA multidimensional sequential approaches. Performance was evaluated using the CR index.

Table 3.8: Performance of the multidimensional sequential clustering approaches on the Primary Biliary Cirrhosis dataset with DHD, CHI2, and OMslen dissimilarity functions: The Rand index.

Methods	CAH	PAM	FCMdd	ECMdd
Cdist	0.428	0.425	0.474	0.537
Cclust	0.477	0.475	0.503	0.532
GIMSA	0.424	0.517	0.463	0.508

Table 3.8 presents the performance of multidimensional sequential clustering methods on PBC dataset, using the dissimilarity functions DHD, CHI2 and OMslen. The Rand Index measures the similarity between the clusters obtained by the clustering algorithms. We observe that for this dataset, the Cclust method obtains the slightly highest values on two of the four clustering algorithms (CAH, FCMdd). However, the Cdist method shows a slightly high performance for the ECMdd clustering algorithm and values approaching Cclust for the CAH and PAM algorithms, although it is slightly lower for FCMdd. Overall, the Cclust method seems to perform best for these data, closely followed by Cdist.

3.6 Results and discussion

3.6.1 Comparative analysis and discussion

In this section, we propose a comparative analysis of methods Cstat, Ccost, Cdist, Cclust, and GIMSA, state-of-the-art of row-data-base relational clustering approaches for MST. We used several MST characterized by real and simulated datasets, composed of categorical/ordinal/discrete time series and longitudinal data, with different dimensions and time lengths, see Table 3.9. The dissimilarity matrices were computed using either a single dissimilarity function for all trajectory dimensions, or different dissimilarity functions, has a strong impact clustering results. In this context, Studer and Ritschard [5] have formulated guidelines, aimed at orienting the appropriate selection of the measure. Sébastien Massoni and his colleagues [168] have suggested the use of an optimal convex combination of different

dissimilarity measures to determine the most appropriate one. We use critical difference plots based on classical Friedman and Nemenyi non-parametric statistical tests [169], to analyze the differences in performance (RI, ASW and HC) of the single-view relational clustering algorithms (CAH, PAM, FCMdd, ECMdd) for each MST clustering method (Cstat, Ccost, Cdist, Cclust, GIMSA).

Table 3.9: Characteristics of differences between datasets and run-times by method, #Σ corresponds to the maximum alphabet in all dimensions.

Datasets	Nrow	Ncol	#Σ	K	Times (mins)					Measures
					Cstat	Ccost	Cdist	Cclust	GIMSA	
Synthetic 1	150	2	4	4	0.008	0.010	0.011	0.014	0.012	OM
Synthetic 2	150	4	4	4	0.011	0.012	0.022	0.031	0.023	OM
Biofam: Cf. 1	2000	3	3	5	0.033	0.370	0.040	0.451	0.022	OM
Biofam: Cf. 2	2000	3	3	5			0.064	0.747	0.022	OM, LCS, CHI2
AY	10532	12	9	3	0.435	0.321	0.058	0.676	0.454	OM
AY imputation	10532	12	9	3	0.371	0.478	0.020	0.786	0.879	OM
PBC	312	8	5	3			0.064	0.121	0.075	NMS, OMslen, CHI2

The RI-based critical difference diagram presented in Figure 3.10, allows us to observe that there is no significant difference between methods, however the Cclust is the best of all followed closely by Cdist. The ASW-based critical difference diagram is shown in Figure 3.11. The figure shows that Cdist is the best approach of all, followed by Cclust, but the two approaches are not significantly different. The Ccost, Cstat and GIMSA approaches are clearly less accurate. We observe that Cdist and Cclust are both significantly different from the approaches Ccost, Cstat and GIMSA. In the HC-based critical difference diagram shown in figure 3.12, we can see that Cdist is the most stable, followed by Cclust. GIMSA and Cstat remain the least stable. We can conclude that, in the majority of cases on experimental data, Cdist method performs best and more robustly, followed by Cclust. Methods Ccost, Cstat and GIMSA are ranked respectively after these.

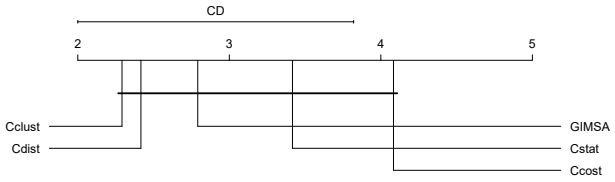


Figure 3.10: Critical difference diagram between Cstat, Ccost, Cdist, Cclust and GIMSA based on Rand Index.

However, a critical review can also be made of the differences between these five methods. Indeed, these approaches differ in their theoretical background and are selected according to the specific goals of MST analysis, each offering a unique perspective for understanding multiple sequential data. For example, the differentiation criteria presented in Section 3.4.3, i.e. multidimensionality, parsimony, type and local and global interdependence,can or cannot be appropriate depending on the research questions or data. Another main

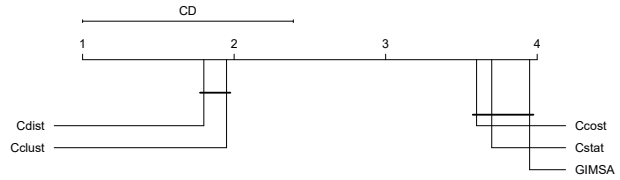


Figure 3.11: Critical difference diagram between Cstat, Ccost, Cdist, Cclust and GIMSA based on Average Silhouette Width.

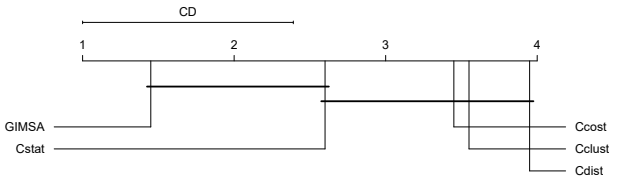


Figure 3.12: Critical difference diagram between Cstat, Ccost, Cdist, Cclust and GIMSA based on Hubert's C.

distinction can be the assumption of independence of the dimensions assumed by Ccost and Cdist methods. They assume implicitly that the sequential measurements (status) were done independently in each dimension (sequences) [15]. We can wish to use Ccost and Cdist even if dimension independence is not effective, for example because the costs and distances obtained are simpler to interpret. In such cases, it is crucial to be conscious of the constraints imposed by the procedure on costs for Ccost and on dissimilarities for Cdist.

A comparison of the temporal complexities of the approaches (Table 3.9) shows that the Cdist method has generally lower computation times than the others. In detail, we observe that the Cstat method is generally the fastest for small datasets, and its runtime increases with dataset size. Ccost method has variable run times, and Cclust method generally has the highest run times of all methods, particularly for large datasets. Run times for the GIMSA method are generally comparable to those of Cstat and Ccost. The missing data processing method (state extension or multiple imputation) has an impact on time complexity and differs from methods. Multiple imputation appears to reduce the time complexity of the Cstat and Cdist methods, but increases the runtime of the Ccost, Cclust and GIMSA methods. Moreover, although the main focus of this work was on the comparison between these five approaches, the procedure we have presented can serve as a guide for performing multidimensional sequential trajectory clustering. The work also contributes to the comparison of hard (CAH and PAM) and soft (FCMdd and ECMdd) clustering algorithms in the context of sequential trajectory analysis. We can thus see on Figure 3.13 of the critical difference diagram, that the PAM algorithm is the best, followed by ECMdd, and CAH is the least performing of all. However, although we can classify them, these differences are not statistically significant for PAM, ECMdd and FCMdd.

Our work has some limitations. Although we have exploited simulated and real datasets that are very different from each other, and studied different scenarios of these data, we cannot claim to have taken into account all possible situations of MST. In addition, choices related to the clustering algorithms and dissimilarity measures used could have been different, which could have led to alternative conclusions. For dissimilarity measures, although a large

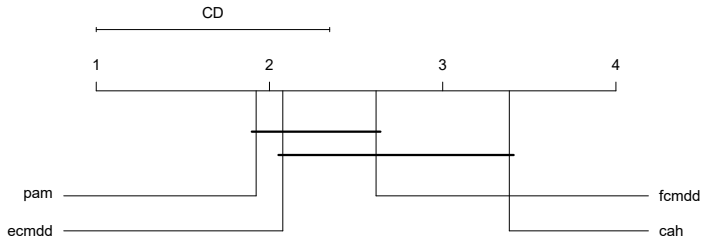


Figure 3.13: Critical difference diagram between clustering algorithms CAH, PAM, FCMdd and ECMdd based on Rand Index.

variety of measures exist, we concentrated on those that are sensitive to the differences characteristic of each dimension of our data.

3.6.2 Exploring multi-view relational methods for MST clustering

Cdist is the most powerful and robust method for MST clustering according to Section 3.6.1, even though its main limitation is the ignorance of dimension correlation. Our approach aims to extend Cdist into a collaborative dimensional framework with relevance weights for each dimension, which we named Codist (distance collaborative). In contrast to Cdist, Codist matches all columns of the classification Table 3.2, because it provides a global interdependence of the dimensions (sequences) and, more importantly, allows direct interpretability of the clustering results, which no method in the literature offers. This proposition provides an application framework, offering a new approach to clustering multi-dimensional sequences. To compare the performance of Codist against methods in the literature (Section 3.4), we used eDOL dataset and two clustering algorithms based on multi-relational partitioning hard and fuzzy designed to learn a relevance weight for each dissimilarity matrix, i.e. multi-relational dynamic clustering algorithm (MRDCA) [125], and the multi-relational fuzzy c-medoids (MRFCMdd) [61]. The comparisons presented in Table 3.10 focus on variants where matrix relevance weights are calculated locally for each cluster, with optimization constraints based on a sum function. A detailed analysis of the other variants is available in Section 3.6.2.2. For more details on relational multi-view partitioning algorithms, see Section B.2 of Appendix B. The same setting contexts of the single-view relational algorithms (defined in section 5.6.2.1) have been used for the application framework of the multi-view relational algorithms.

The data come from the eDOL project, presented in Section 2.1.1 of chapter 2. We are interested on barometers composed of eight attributes measured weekly, as detailed in Table 2.1, allowing for the evaluation of pain intensity and monitoring: pain, fatigue, mood, stress, sleep, body comfort, sports and non-sports activities. These dataset, as described in Section 2.1.3.2, are complex and subjective, and can be treated either as categorical time series (CTS), discrete time series (DTS), or continuous time series (TSC). We worked on data from 636 patients, extracted from eDOL in June 2023, totaling 14,090 self-reported care events, with an average follow-up of 5 months and a total duration of approximately 19 months. We used dissimilarity measures adapted to discrete time series, the time warp edit distance (TWED) for all eight dimensions. The TWED distance, noted $\delta_{\lambda,\nu}$, measures the minimum cost of operations needed to transform one discrete sequence into another. Its computation

has a complexity of $O(n^2)$. It depends on two parameters: $\lambda > 0$, which penalizes time gaps, and $\nu \in [0, 1]$, which controls its elasticity. The PAM, FCMdd, and ECMdd single-view relational clustering algorithms were applied to the dissimilarity matrices obtained from Cstat, Ccost, Cdist, Cclust and GIMSA. For the new Codist method, corresponding multi-view relational clustering algorithms, include MRDCA, and MRFCMdd were applied to the dissimilarity matrices.

Table 3.10: Performance of the multidimensional sequential clustering approaches on the eDOL dataset with TWED dissimilarity function.

Methods	PAM/MRDCA		FCMdd/MRFCMdd		ECMdd	
	ASW	HC	ASW	HC	ASW	HC
Cstat	0.477	**0.033**	0.372	0.298	0.449	0.251
Ccost	0.560	0.154	0.462	0.417	0.345	0.214
Cdist	0.348	0.066	0.505	**0.155**	0.568	0.155
Cclust	0.160	0.399	0.471	0.302	0.432	0.189
GIMSA	0.200	0.497	0.341	0.417	0.504	0.284
Codist	**0.568**	0.130	**0.585**	0.201	-	-

Table 3.10 presents the performance of the clustering algorithms PAM, FCMdd, and ECMdd, along with the hard and fuzzy multi-view counterparts, i.e. MRDCA, and MRFCMdd, across the Cstat, Ccost, Cdist, Cclust, GIMSA, and Codist methods. Performance was evaluated using the ASW and HC indices. We shows that Codist combined with fuzzy clustering (MRFCMdd) offers the best performance, achieving the high compactness (ASW=0.585), making it the most efficient approach, but less stable (HC = 0.201). Cstat with hard clustering (PAM) has good stability. Cdist balances compactness and stability, particularly with fuzzy clustering.

We conducted an exploratory evaluation of Codist, comparing in detail the different variants of the MRFCMdd and MRDCA algorithms. Indeed, relevance weights are calculated at each iteration of the algorithm and can be estimated locally for each cluster, or globally for all clusters. Readers are referred to Section B.2 of Appendix B for further details.

3.6.2.1 Experimental setting

On the eDOL datasets, using the TWED dissimilarity function, the proximity matrices obtained on each sequential dimension were normalized according to their global dispersion. This involves each dissimilarity $d_{ii'} = \delta_{\lambda,\nu}\left(S_j^{(i)}, S_j^{(i')}\right)$ in a dissimilarity matrix $j = 1, \ldots, 8$, was normalized according to the formula $2 \times d_{ii'}/(m + d_{ii'})$, where m represents the maximum possible dissimilarity of the matrix j. The default parameters $\nu = 0.5$ and $\lambda = 0.5$ are used. The algorithms were run 10 times, with the number of clusters fixed to 3. Next, we select the ASW values corresponding to the lowest cost of the objective function of the algorithm. The fuzzy partitions are transformed into hard partitions using the maximum

rule, as follows: $G_k = \{\mathbf{x_i} : u_{ik} \geq u_{im} \quad \forall m \in \{1, \ldots, K\}\}$. The default value $m = 1.5$ of the fuzziness parameter is used.

3.6.2.2 Results

Table 3.11 shows that the MFCMdd-RWG-S and MFCMdd-RWL-S algorithms have the lowest objective function costs (0.098). MFCMdd-RWL-S offers the best partitioning performance, with an ASW of 0.539. MRDCA algorithms have higher costs and lower silhouette indices, indicating lower performance.

Table 3.11: Performance of clustering algorithms in terms of ASW.

Algorithmes	J_{Cost}	ASW
MFCMdd-RWG-P	6.330	0.520
MFCMdd-RWG-S	**0.098**	**0.528**
MFCMdd-RWL-P	6.310	0.521
MFCMdd-RWL-S	**0.098**	**0.539**
MRDCA-RWG	8.280	0.481
MRDCA-RWL	8.240	0.509

A qualitative comparison of fuzzy partitions using contingency matrix or matching matrix between partition MFCMdd-RWL-S and the others, illustrated in Figure 3.14, shows that classification errors are relatively low, with similar results in 96% of cases. Figure 3.15 shows a high degree of membership of patients in clusters 1 and 2 to their respective clusters, indicated by a median close to 1. In contrast, cluster 3 has more imprecise objects, given the degree of fuzziness of several patients close to $1/3$.

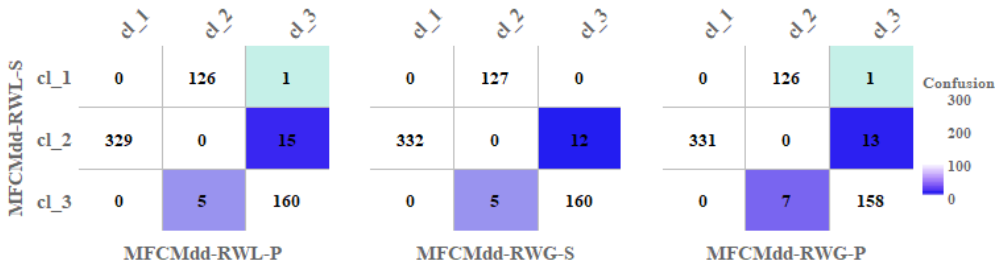


Figure 3.14: Comparison of fuzzy partitions using matching matrix between partitions.

Table 3.12 shows the relevance weight values for each dimension or sequence (pain, stress, fatigue, sleep, mood, body comfort, non-sports activity, sports activity) in each cluster according to the best partition given by MFCMdd-RWL-S. The values represent the relative importance of each dimension in the characterization of each cluster. Cluster 1 is characterized by high values for the dimensions of Pain, Fatigue, Sleep and Body Comfort, suggesting profiles associated with chronic pain with fatigue, sleep disturbance and altered body comfort. Cluster 2 shows higher values for the dimensions of stress, mood, non-sports activity and sports activity, indicating trajectories where stress, emotional aspects and physical activity play a more prevalent role. Cluster 3, on the other hand, appears to be

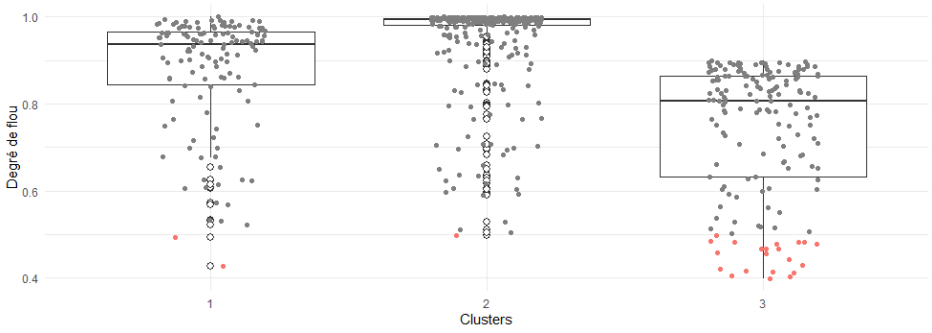


Figure 3.15: Distribution of fuzziness.

more generalist than the others, with no specific characteristics in terms of symptoms or behaviours.

Table 3.12: Relevance weight values for each trajectory dimension in each cluster.

	Cluster 1	Cluster 2	Cluster 3
Pain	**0.147**	0.136	0.142
Stress	0.135	**0.137**	0.131
Fatigue	**0.148**	0.135	0.146
Sleep	**0.143**	0.125	0.135
Mood	0.129	**0.133**	0.130
Body Comfort	**0.120**	0.095	0.115
Non-sports Activity	0.086	**0.124**	0.109
Sports Activity	0.093	**0.114**	0.091

3.7 Conclusions

In this work, we propose a review of trajectory clustering methods. In this context, we provide diverse formulations of definitions and methods of trajectory clustering in a first part. In a second part, we discuss sequential trajectory clustering methods, focusing on a comparative analysis of multidimensional methods. Four types of trajectory are thus highlighted, including sequential, temporal, spatial and spatio-temporal trajectories. Common clustering methods are organized into three main categories, depending upon whether they work directly with the raw-data, indirectly with features extracted from the raw-data, or indirectly with models built from the raw-data. The most recent studies on clustering multidimensional sequential trajectories (MST) based directly on raw data have been grouped into five main methods: additive (Ccost or Cdist), combinative (Cstat or Cclust) or factorial analysis (GIMSA).

In the light of critical and experimental analysis of the real and simulated MST datasets considered, the Cdist method is the best performing and most robust, followed by the Cclust method. However, we have demonstrated that Cdist assumes dimension independence and hides the relationship between dimensions, and that Cclust is not parsimonious in terms of

the final number of clusters. It is important to be aware of these limitations when choosing these methods. We have also proposed to explore the Codist method, an extension of Cdist, which is more efficient and addresses the limitations of the Cdist method. Codist also has the advantage of learning dimensional relevance weights, allowing clustering results to be explained. This work also contributes to the comparison of hard (CAH and PAM) and soft (FCMdd and ECMdd) clustering algorithms in the context of sequential trajectory analysis. When data are complex and subjective, soft algorithms, particularly ECMdd, perform well and offer a finer representation of uncertainty.

Although the initial results illustrate the performance of the Codist method, these promising results open up new opportunities for application. In sum, although there is a possibility for improvement in methods presented in this chapter, the main opportunity for future work on MST clustering could be to work on new evidential clustering algorithms to effectively represent uncertainty and imprecision and to enable collaboration between multi-dimensions or multi-sequences using object-based or relational-based data.

In the next chapters, i.e. chapter 4, chapter 5 and chapter 6, we explore new Frameworks and propose new algorithms for evidential clustering of MST or, more generally, for multidimensional trajectory data. Chapter 4 focuses on the exploration of feature-based evidential clustering, while Chapters 5 and 6 present algorithms for distance-based and object-based evidential clustering respectively.

Chapter 4

Feature-Based Evidential C-Means Clustering

Abstract – In this chapter, we explore the feature-based clustering of trajectory data related to chronic pain from eDOL. By employing a soft clustering approach, we aim to uncover distinct care trajectories. Our method involves extracting and selecting key features from the data using advanced techniques like Tsfresh and unsupervised feature selection. We then apply Evidential C-Means algorithms to identify care trajectory typologies, taking into account uncertainty based on subjectivity and the nature of the data. The results reveal two clear clusters associated with discomfort and well-being, along with an uncertain cluster representing intermediate characteristics. Through descriptive analysis, statistical tests, and multinomial regression, we identify patient profiles within these trajectories, enhancing our understanding of chronic pain management.

Contents

4.1 Introduction

Chronic pain affects millions of patients in France, approximately 30% of the general adult (18 years and older) population [170]. This high prevalence of chronic pain has a significant impact on individuals suffering from it, as well as on society as a whole. Patients with chronic pain face a deterioration in their daily functioning, both physically and socially, compromising their quality of life. Improvements in the treatment and management of chronic pain can increase patients' quality of life and reduce societal costs. However, as highlighted in the study [28], currently available treatments are often outdated, have limited effectiveness, and can lead to significant adverse effects. Furthermore, due to the complexity of care pathways for patients with chronic pain, the outcomes in terms of health improvement are often unsatisfactory. Evaluating state of the art machine learning approaches in chronic pain research has shown that further research is needed on machine learning approaches for personalizing treatments, rehabilitation, and self-management of chronic pain [171]. Thus, identifying typologies of care pathways and better characterizing patient profiles would allow healthcare professionals to improve care outcomes and better support patients. This could help identify predictive factors for treatment success and guide physicians in the initial choice of treatment and patient follow-up.

The eDOL project, initiated in 2017 by the ANALGESIA Institute, as outlined in Section 2.1.1 of chapter 2, has brought innovation to chronic pain management through an mHealth application that accompanies patients in their daily pain management, leading to better treatment. This project aims to enhance care pathways by identifying clusters and characterizing patient profiles more effectively. It facilitates collecting of follow-up data to evaluate pain intensity and its longitudinal impact, fostering new research directions.

Prior work has already been conducted in the field of time series trajectory clustering, as extensively reviewed in reference [124, 172, 173]. According to these works, there are three main approaches. Firstly, feature-based approaches, which are our focus in this work. These approaches aim to extract significant features from time series for clustering purposes. Secondly, raw-data-based approaches, which use the time series data directly without prior feature extraction. Finally, model-based approaches, which employ statistical or probabilistic models for clustering time series. Our decision to opt for feature-based approaches is based on several considerations. Our time series data exhibits strong irregularities, including unequal spacing between data points and varying lengths characterized by differing follow-up duration (unbalanced time series). These irregularities are compounded by the complex and uncertain nature of our chronological data. Uncertainty due both the uncertain medical nature of the data and the way in which they are collected, mainly by patients themselves, who provide subjective assessments of their pain, stress, fatigue, comfort, etc., in the course of their treatment. Thus, all these factors pose challenges when performing clustering directly on raw data [174]. By adopting feature-based approaches, we address these challenges and enhance the clustering process.

This article presents a soft clustering application approach that utilizes feature extraction and selection from sequential data to enhance interpretability and optimize clustering performance. Indeed, the extraction of features from time series has enabled us to deal effectively with the complex and noisy nature of the data [124]. Feature selection, as opposed

to projection, maintained interpretability, reduced dimensionality without significant loss of information, and improved clustering performance [175]. Evidential Clustering, a credal variant of Fuzzy C-Means, was used to modelling uncertainty and imprecision in the data [1]. The first step involves extracting features from the available time series and selecting the most important attributes. For this purpose, we use a time series feature extractor (Tsfresh) and unsupervised feature selection methods, including unsupervised Random Forest, Laplacian Score, and unsupervised Spectral Feature Selection. The second step involves using the evidential c-means (ECM) clustering algorithm on the most important attributes selected. The interpretability of the determined partitions through shape values, descriptive analysis, statistical tests, and repeated measures multinomial regression on clinical and demographic data allowed us to determine the profile of patients in the identified trajectories.

This chapter is organized as follows. We begin by introducing our methodological approach in Section 4.2, which includes feature extraction, unsupervised feature selection, and the Evidential Clustering Method. In Section 4.3, we apply our methodology to the dataset, providing a detailed overview of the data and showcasing the results of feature extraction, selection, and clustering. Finally, Section 4.4 provides a thorough discussion of our findings, their implications, and potential future research directions.

4.2 Research framework

In this work, our objective is to identify and describe typologies of chronic pain without having prior knowledge. However, due to the complexity, uncertainty, noise, and temporal dependence of our data, as presented in Section 4.3.1, traditional automatic classification methods become inadequate. Therefore, on one hand, we resort to feature extraction and selection techniques that contribute to improving both the accuracy and interpretability of most learning algorithms by focusing on the most informative features while discarding irrelevant or redundant ones [176]. These techniques allow us to identify and extract the most relevant and informative features from the raw data, thereby reducing noise and redundancy. Varying length multivariate time series can be characterized by multiple summary statistics; however, some statistics may contain useless or redundant information, and some features may be coupled [177]. Then, by focusing on the most discriminating characteristics extracted by selection, our model can achieve greater precision in grouping care trajectories. Additionally, the selected features provide us with a better understanding of the model and underlying relationships present in the data, thus enhancing result interpretability (see Figure 4.1).
Furthermore, in order to account for the uncertainty related to the possibility of each data point belonging partially to multiple clusters with different degrees of imprecision and uncertainty, we opted for a soft clustering approach [178]. Among the three types of clustering methods, namely (1) density-based approaches, (2) hierarchical approaches, and (3) prototype-based approaches, we chose the prototype-based ones, based on the k-means algorithm, for several reasons. In addition to benefiting from the advantages offered by soft clustering variants of prototype-based approaches, such as considering the uncertainty and complexity of the data, the key advantage of prototype-based methods is the concept of a prototype (or centroid), which allows for a more meaningful representation of clusters. Moreover, they exhibit reduced computational complexity, making them efficient even

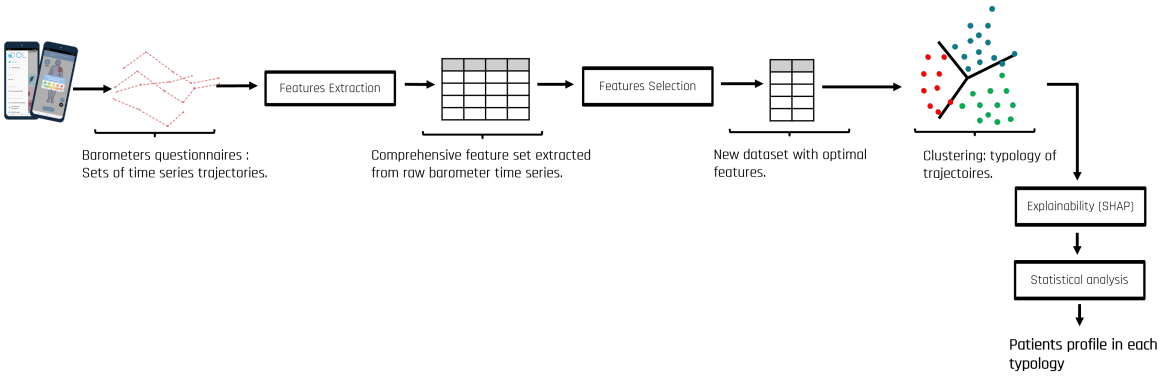


Figure 4.1: Formulations of feature-based for trajectory Clustering.

for large amounts of data compared to hierarchical approaches. Unlike density-based approaches, which tend to merge data into a single group, prototype-based methods can effectively separate groups, even when they are strongly intertwined. However, unlike the K-means method (or Hard C-means (HCM)) that categorically assigns an individual's membership to a specific class, its soft variants (C-means algorithms) propose a smoother approach. They minimize an objective function to determine the degree of uncertainty of each data point regarding its membership in a specific class. Fuzzy C-means (FCM) [179] and evidential C-means (ECM) [1] are two variants of the C-means clustering algorithm that allow for soft partitioning of the data. FCM utilizes fuzzy set theory to assign partial degrees of membership to each data point for each cluster, while ECM uses belief function theory to determine belief masses for each data point for each cluster. In previous works on soft clustering applied to time series, fuzzy clustering is one of the most commonly used partitioning tools [180, 181]. D'Urso et al. [182, 183, 184, 185] propose robust techniques based on quantiles or DTW for feature-based time series clustering using fuzzy c-means and its relational variant fuzzy c-medoid. The same authors also contribute to model-based fuzzy clustering of time series with their approach based on B-splines [186]. A number of studies comparing HCM and FCM on different types of data have been conducted in the literature [187, 188, 189, 190]. While some research concluded on the performance of HCM, others also specified that FCM was the better performer. For time series data, these same results are highlighted, regardless of the approach used, i.e. feature-based, raw data or model-based. In conclusion, as mentioned in their comparative review of image segmentation between HCM and FCM [191], Pugazhenthi A. and al. underline that the optimal clustering method depends on the qualitative requirements of the clustering process, the applications and the unique characteristics of the data used. FCM can process data with imprecise boundaries compared to HCM. However, the approach proposed by the FCM algorithm does not provide satisfactory results if the studied data contains noisy or outlier data [1]. ECM allows for modeling both uncertainty and imprecision, something that FCM cannot achieve as it is limited to modeling imprecision alone. We therefore chose to use ECM, not only for its strengths mentioned before, but also because its separability performance compared to k-means on our data has been demonstrated [d].

Thus, our approach is based on three steps, and its entire operation is illustrated in Figure 4.2: (1) Transformation and extraction of time series features, (2) Attribute selection through a comparison of three unsupervised selection algorithms, and (3) Partitioning, evaluation,

and analysis of clusters.

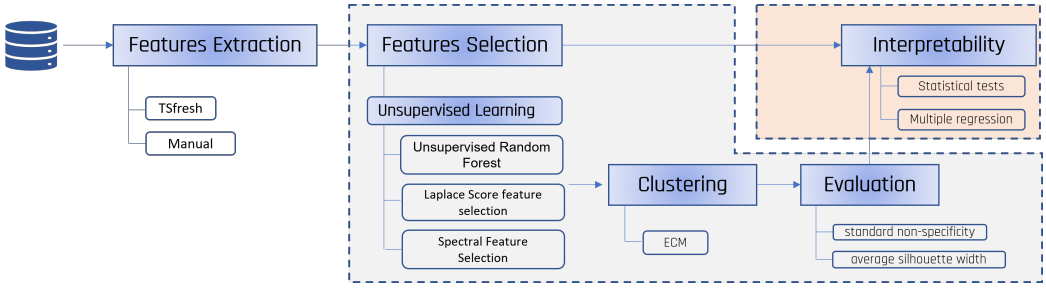


Figure 4.2: Flowchart of the global workflow.

4.2.1 Transformation and feature extraction

Clustering based on raw time series data involves working with data that is highly noisy and irregular, limiting the interpretability of the results. These include random variations, measurement errors, outliers, and heterogeneous series lengths. Such disturbances can introduce inconsistencies and errors in clustering analysis as they can distort the underlying relationships and structures present in the data. Thus, feature extraction from time series plays a crucial role in improving the quality and usability of the data, as well as the interpretability of machine learning results. This extraction helps capture the inherent structure of the series, reduce uncertainty, manage complexity, and mitigate the effects of noise. Several feature-based clustering methods have been proposed to address these concerns [192, 193, 194]. While most feature extraction methods are generic in nature, [124] explain that the extracted features should generally depend on the application domain. In other words, a set of features that works well for one application may not be relevant for another. [195] developed **tsfresh** (Time Series Feature Extraction on the basis of Scalable Hypothesis tests), a Python package, which is widely used in the literature and automatically extracts over 100 features from a time series. These features can include basic characteristics of the time series (e.g., maximum value, mean, or number of peaks) and complex features (e.g., symmetry or time shift statistics). Furthermore, these extracted features reflect the shape and inherent nature of the data model of the series (e.g., distribution or autocorrelation). Some typical features are illustrated in Figure 4.3.

Thus, a large number of features, which capture information about the behavior, dispersion, homogeneity, kurtosis, asymmetry, and trend of the raw time series of the barometers, were computed using the **tsfresh** package. The features considered in this study are derived, on the one hand, from research on feature extraction from temporal data for medicine [196, 197, 177]. Guo, C. and al. have evaluated several clinical and biostatistical studies on time series summary statistics as features for clinical prediction tasks [177]. On the other hand, some features are derived from well-known statistical techniques such as standard deviation and mean, and the remaining features were recommended in our study to capture the discrete nature of our time series.

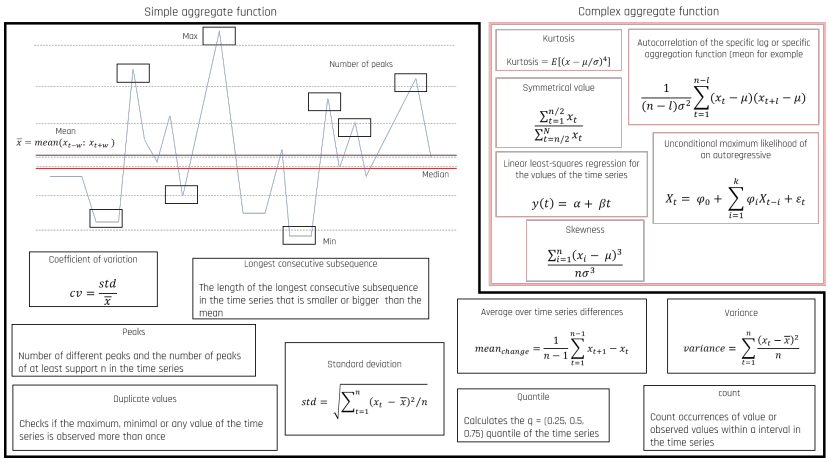


Figure 4.3: The name, design principle, and computing method of some features in Tsfresh

4.2.2 Unsupervised Feature Selection

Intuitively, not all features have the same importance. Some features may be redundant, some may be irrelevant, and some may even distort the clustering results. Reducing the dimensionality of such a representation improves interpretability and enhances the performance of the clustering procedure. The idea is to construct a clustering approach that is interpretable, meaning that it is suitable for an expert in the field to perform subsequent analysis. In order to select relatively more concise and useful features, we have proposed and compared three unsupervised feature selection algorithms. Unsupervised feature selection is a complex research area that has received much attention in recent years [198, 199, 200]. This is mainly due to its ability to identify and select relevant features without requiring class label information. Although many feature selection methods have been proposed over the decades, most of them were specifically developed for supervised classification tasks [201]. Unsupervised feature selection also differs from feature extraction, which uses transformation techniques to reduce the dimensionality of features. In the context of unsupervised selection, it becomes challenging to determine which features to retain in the absence of class labels, as there are no clear criteria to guide the selection. However, in addition to being unbiased and performing well when labels or prior knowledge are lacking, unsupervised selection also has the advantage of reducing the risk of overfitting the data [202]. Unlike supervised selection methods that may struggle with new data classes, unsupervised selection offers greater adaptability.

In the literature, unsupervised feature selection methods can be classified into three main approaches: filtering methods, wrapper methods, typically based on a specific clustering algorithm, and hybrid methods that seek to combine the advantages of filtering and wrapper approaches [201]. We focus on filtering methods and wrapper methods, which are distinguished by their speed and scalability. Filtering and wrapper methods offer distinct approaches to unsupervised feature selection. Univariate filtering evaluates features individually, while multivariate filtering considers feature relationships, adept at handling redundancy. Iterative wrapper methods address feature selection as an estimation problem, avoiding combinatorial search unlike sequential methods. In general, there is no best approach or unsupervised selection method for all types of data and domains; each

approach has its own advantages and disadvantages. Nevertheless, from our literature study, we can highlight some methods that have proven effective for variable selection in the healthcare domain, namely Random Forests, Laplacian, and Spectral algorithms [203, 204]. In this paper, we compare the Laplacian Score-based univariate filtering algorithm, the Unsupervised Spectral Feature Selection-based multivariate filtering algorithm, and the unsupervised Random Forest feature selection-based iterative wrapper algorithm.

Unsupervised Random Forest feature selection (URF) [205, 206]. Random Forest has been extended to handle unlabeled data and enable unsupervised learning. In this approach, the original data is considered as class 1, and a synthetic second class of the same size is created and labeled as class 2. To construct this synthetic class, random samples are drawn from the univariate distributions of the original data. Thus, class 2 has a distribution of independent random variables, each having the same univariate distribution as the corresponding variable in the original data. This destroys the dependency structure present in the original data. Using this artificial two-class configuration, Random Forest is applied for learning. Formulating the problem in this way has several advantages, such as outlier detection, missing value imputation, etc. However, the most significant gain in our context is the ability to estimate the importance of each variable.

Laplacian Score feature selection (LS) [207]. It is a linear unsupervised feature selection method. For each feature, the algorithm computes a Laplacian score that indicates its ability to preserve locality. The approach is based on the observation that two data points are likely to be related to the same subject if they are close to each other. In many learning problems, the local structure of the data space is more meaningful than the global structure. To model this local geometric structure, Laplacian Score constructs a graph of nearest neighbors and searches for features that respect this structure. The algorithm relies primarily on Laplacian Eigenmaps and Locality Preserving Projection methods.

Unsupervised Spectral Feature Selection (US) [208]. The algorithm utilizes Laplacian decomposition, also known as spectral decomposition or eigendecomposition of the Laplacian matrix, to find the most relevant features in data sets. Indeed, from a neighborhood graph, a similarity matrix measuring the relationships between pairs of objects is constructed. This matrix, unlike the Laplacian approach, captures the global structure of the data. Graph theory demonstrates that information about the structure of a graph can be obtained by analyzing its spectrum. Spectral feature selection focuses on how to choose features based on the graph structures.

In summary, unsupervised feature selection methods generate a score for each feature in order to rank them. Some methods calculate feature importance using the coefficients of a classification, sometimes using pseudo-labels (as in the case of unsupervised Random Forest). Other methods perform selection by preserving the original structure of the data based on a similarity matrix between objects. This matrix is constructed using all the features, and from it, a form of mapping is performed using spectral information (as in the case of unsupervised Spectral) or Laplacian information (as in the case of Laplacian score). These methods differ in their approaches, the feature importance measures they use, their ability to reduce the dimensionality of the data, and the complexity of the algorithm they involve.

4.2.2.1 Common process for unsupervised filtering

All unsupervised filtering feature selection algorithms are based on studying the similarity between data sets. In unsupervised filtering algorithms, the integration and projection steps play an important role in reducing the dimensionality of the input data (see Figure 4.4). Integration involves combining the input features to create new representations of the data, while projection reduces the dimension by projecting the data into a lower-dimensional space. The integration step is typically performed by applying a mathematical function (Laplacian, Spectral) that preserves some of the underlying structure of the data. The projection step involves mapping the integrated features to the new feature space. This can be done by applying additional linear mathematical transformations or techniques to the integrated features.

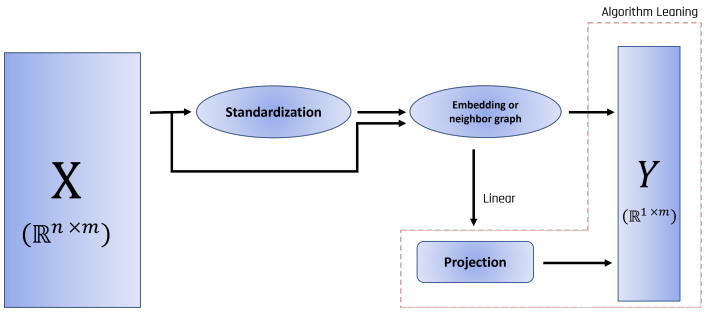


Figure 4.4: Common workflow of unsupervised features selection

Let's consider a dataset $\mathbf{X}$ consisting of n objects, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, where $\mathbf{x}_i \in \mathbb{R}^m$. Let $F_1, F_2, \ldots, F_m$ be the m features, and $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_m$ be the corresponding feature vectors. The similarity between pairs of objects, which does not require class label information, is captured by the scalar s. This similarity value is constructed using the Radial Basis Function (RBF) kernel. The RBF kernel measures the similarity between two instances $\mathbf{x}_i$ and $\mathbf{x}_j$ based on their Euclidean distance and is mathematically defined as follows:

$$s_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right). \tag{4.1}$$

Where $||\mathbf{x}_i - \mathbf{x}_j||$ represents the Euclidean distance between objects $\mathbf{x}_i$ and $\mathbf{x}j$, and σ is a parameter that controls the influence of distance on similarity. In the context of Laplacian and Spectral approaches, $\mathbf{S} = [s_{ij}]$ is used to capture the relationships and structures in the data. These algorithms utilize s_{ij} to perform data integration operations, focusing on preserving the similarity between objects. They use a utility function $U(.)$ to evaluate the extent to which the set of features preserves the data's similarity structure. Solving this function is often done through a greedy selection of the best features that maximize their individual utility $U(F)$.

$$\max_S U(\mathbf{S}) = \max_S (\mathbf{S}) \sum_{\mathbf{f} \in S} U(\mathbf{f}) = \max_S (\mathbf{S}) \sum_{\mathbf{f} \in S} \tilde{\mathbf{f}}' \tilde{\mathbf{S}} \tilde{\mathbf{f}}. \tag{4.2}$$

Which $U(S)$ is utility of feature set similarity S, $U(\mathbf{f})$ utility of feature F, $\tilde{\mathbf{f}}'$ and $\tilde{\mathbf{S}}$ are transformation of respectively feature vector $\mathbf{f}$ and feature similarity matrix S. Thus, the

two algorithms differ in how the vector $\mathbf{f}$ and the similarity matrix $\mathbf{S}$ are transformed into $\tilde{\mathbf{f}}$, $\tilde{\mathbf{f}}'$, and $\tilde{\mathbf{S}}$.
The Laplacian Score selection method begins by constructing an undirected nearest neighbor graph $\mathbb{G}$ from the source data $\mathbf{X}$. $\mathbb{G}$ is constructed using the nearest neighbors approach. However, it's important to note that the nearest neighbors algorithm typically relies on Euclidean distance for distance measurements. The weight matrix $\mathbf{S}$ of the graph $\mathbb{G}$, which captures the local structure of the data space, is defined as s_{ij} if nodes i and j are connected, and 0 otherwise. This weight matrix represents the connections or similarities between edge graph nodes. Next, we construct the diagonal matrix $\mathbf{D}_{ii} = \sum_j \mathbf{S}_{ij} \quad \forall i$, and the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{S}$. Here, $\mathbf{1} = [1, \ldots, 1]^T$ represents a column vector, the Laplacian Score of feature F_i is calculated using the following formula [207]:

$$LS_{score}(F_i) = \frac{\tilde{\mathbf{f}}_i^{'} \mathbf{L} \tilde{\mathbf{f}}_i}{\mathbf{f}_i^{'} \mathbf{D} \mathbf{f}_i}, \quad where \quad \tilde{\mathbf{f}}_i = \mathbf{f}_i - \frac{\mathbf{f}_i^{'} D\mathbf{1}}{\mathbf{1}^{'} D\mathbf{1}}. \tag{4.3}$$

The Spectral selection method, unlike the Laplacian score, utilizes an eigendecomposition of the normalized Laplacian matrix in its selection process and aims to preserve a global structure of the data. To begin, the similarity matrix $\mathbf{S} = [s_{ij}]$ is computed on the source data $\mathbf{X}$. Next, the undirected graph $\mathbb{G}$ is constructed from the obtained similarity matrix $\mathbf{S}$. The Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is obtained by subtracting the similarity matrix $\mathbf{S}$ from the degree matrix $\mathbf{D}_{ii} = \sum_j \mathbf{S}_{ij} \quad \forall i$, which represents the degree of each node in the graph. The degree matrix is obtained by calculating each diagonal element $\mathbf{D}_{ii}$ as the sum of the weights of the edges connected to node i in the similarity graph $\mathbb{G}$. Thus, $\mathbf{L}$ captures the connectivity information of the data, but it is based on similarity rather than direct neighborhood relationship, and the normalized Laplacian $\mathbf{L}$ is defined as follows $\mathcal{L} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}}$. Now, we can obtain the eigenvectors and eigenvalues of the Laplacian matrix. For $\mathcal{L}$, we compute its spectral decomposition (λ_j, ξ_j) where ξ_j is its eigenvector and λ_j is the corresponding eigenvalue, with $0 \leqslant j \leqslant n-1$, and $\lambda_0 \leq \lambda_1, \ldots, \leq \lambda_{n-1}$ satisfying the equation $\mathcal{L} \times \xi_j = \xi_j \times \lambda_j$. This is known as the eigendecomposition, and according to Equation 4.2, the spectral score of feature F_i is defined as follows [208]:

$$SP_{score}(F_i) = \frac{\tilde{\mathbf{f}}_i^{'} \mathcal{L} \tilde{\mathbf{f}}_i}{1 - \tilde{\mathbf{f}}_i^{'} \xi_0}, \quad where \quad \tilde{\mathbf{f}}_i = \frac{D^{\frac{1}{2}} \mathbf{f}_i}{\| D^{\frac{1}{2}} \mathbf{f}_i \|}, \tag{4.4}$$

and $\lambda_0 = 0$, $\xi_0 = \mathbf{D}^{\frac{1}{2}} \mathbf{1}$, and (λ_0, ξ_0) is usually called the trivial eigenpair of the graph.

It can be noted that in Laplacian Feature Selection, the graph is constructed directly from the source data, whereas in Spectral Feature Selection, the graph is constructed from the similarity matrix obtained from the source data using a kernel function. Additionally, in Laplacian Feature Selection, the Laplacian matrix is computed from the weight matrix, whereas in Spectral Feature Selection, it is computed from the similarity matrix. This reflects the difference in graph construction and similarity measures used in the two approaches. Furthermore, although both algorithms use the matrix $\mathbf{D}$, the way it is computed differs. In Laplacian Score, it is computed by diagonalizing the similarity matrix $\mathbf{S}$, while in Spectral Feature Selection, it is obtained directly from the graph by computing the node degrees.

Algorithm 4 Spectral Selection

Input : $\mathbf{X}$, σ
Output: $scores$ - the ranked feature list

Construct $\mathbf{S}$, the similarity matrix from $\mathbf{X}$
Construct undirected graph $\mathbb{G}$ from $\mathbf{S}$
Build $\mathbf{D}$ and $\mathcal{L}$ from $\mathbb{G}$
for each feature vector $\mathbf{f}_i$ of F_i **do**
  $\tilde{\mathbf{f}}_i \leftarrow \frac{D^{1/2}\mathbf{f}_i}{\|D^{1/2}\mathbf{f}_i\|}$ $scores(i) \leftarrow SP_{score}(F_i)$
end
Ranking $scores$ in ascending order
return $scores$

Algorithm 5 Laplacian Score

Input : $\mathbf{X}$, σ
Output: $scores$ - the ranked feature list

Construct nearest neighbor graph $\mathbb{G}$ from $\mathbf{X}$
Construct $\mathbf{S}$, the weight matrices from $\mathbb{G}$
Build $\mathbf{D}$ and $\mathbf{L}$ from $\mathbf{S}$
for each feature vector $\mathbf{f}_i$ of F_i **do**
  $\tilde{\mathbf{f}}_i \leftarrow \mathbf{f}_i - \frac{\mathbf{f}_i' D\mathbf{1}}{\mathbf{1}' D\mathbf{1}}\mathbf{1}$ $scores(i) \leftarrow LS_{score}(F_i)$
end
Ranking $scores$ in ascending order
return $scores$

4.2.2.2 Data normalization

The extracted data undergo a number of transformations (centering and/or scaling normalization) before being used by the selection algorithms. In the literature, a consensus among researchers emphasizes the significance of normalization, as it grants equal importance to all input attributes in various machine learning approaches, thereby facilitating rapid convergence. Moreover, our study underscores the importance of considering data fluctuations stemming from the application of multiple aggregation functions for transformation. However, it is crucial to note that the Random Forest algorithm operates distinctively as it is not a distance-based model but rather a tree-based one. In Random Forest, each node performs splits based on sorted lists, necessitating absolute values for branching. The algorithm's predictive capabilities rely on data partitioning rather than direct feature value comparison, rendering normalization unnecessary. To ensure a standardized evaluation process and foster effective comparisons, we conducted a comparative analysis of selection performance using both normalized and non-normalized data across all algorithms. We consider both centering and scaling normalization. Mathematically, let D be the dataset, σ be the standard deviation of D, and μ be its mean. Then, the normalized D is given by the following equation:

$$\text{Centered and Scaled:} \quad \hat{x} = \frac{x - \mu}{\sigma} \quad \forall x \in D \tag{4.5}$$

4.2.2.3 Performance evaluation metrics for selection

The performance of a selection is measured based on the quality of the obtained clustering using the selected features. Specifically, the data corresponding to the selected variables are first normalized. Then, the best partition and the corresponding optimal number of clusters are calculated on this normalized data using a Grid-search approach. The clustering algorithm used for evaluation is ECM, and the quality criteria are the normalized nonspecificity (Section 4.2.3) of the partitions for determining the number of clusters and the Average Silhouette Width (ASW) for assessing the best separability.

4.2.2.4 Evaluation and comparison of selection algorithms

As observed in Figure 4.2, the proposed selection approach involves multiple iterations, with comparisons within each algorithm and a final comparison among the three algorithms to ensure robustness. In practice, each selection algorithm was evaluated using the elbow method with different selection options. For each option, we first performed a Grid-search to find the optimal number of clusters C, ranging from 2 to 10, that minimizes the criterion of normalized nonspecificity [209]. Then, we calculated the Average Silhouette Width corresponding to the optimal number of clusters for each selection. The best selection that maximizes the ASW is retained, as a high ASW score indicates a better situation, where points within the same cluster are tightly grouped and well separated from other clusters. The worst value is -1, and values close to 0 indicate overlapping clusters. Thus, this criterion allows for comparing the clustering quality produced by different partitioning methods with varying numbers of clusters [210]. Moreover, it does not rely on specific statistical assumptions about the data. Finally, the best selection from each algorithm is retained, and a final evaluation comparing the best performances of the three algorithms helps determine the optimal selection. Figure 4.5 summarizes the complete feature selection approach based on data from feature extraction.

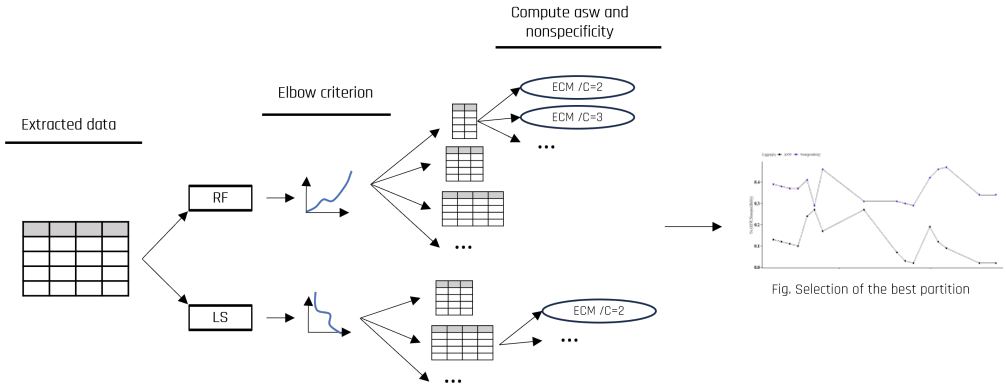


Figure 4.5: Summary of the complete feature selection approach based on data from feature extraction.

4.2.3 Evidential c-means (ECM) Clustering

The ECM algorithm [1], presented in Section 2.3.5.4, is an extension of k-means based on the theory of belief functions (Section 2.3.5.1). This approach handles ambiguous, uncertain, or atypical objects by assigning belief masses to subsets of clusters. The theoretical framework and the general functioning of ECM, including the objective function in Equation 2.47, and iterative steps in algorithm 2, are detailed in the Background chapter (chapter 2).

The performance of the ECM algorithm is evaluated using the *standardized non-specificity*, a internal metric that quantifies the imprecision in the clustering results. This measure, introduced in [209], is defined as:

$$N^*(c) \triangleq \frac{1}{nlog_2(c)} \sum_{i=1}^{n} N(m_i) \quad \text{with} \quad N(m) \triangleq \sum_{A \in 2^\Omega / \emptyset} m(A) log_2 |A| + m(\emptyset) log_2 |\Omega| \quad (4.6)$$

Algorithm 6 Feature Selection approach algorithm

Input : Dataset **X**, Feature Selection algorithms $(FS) = \{$**URF, LS, US**$\}$
Output: Optimal parameter for Feature Selection

foreach FS algorithm **do**
  Run Feature Selection algorithm FS with default parameters using **X**;
  Manually identify different selection levels based on algorithm's importance order (elbow method);
  foreach Selection Options (opt) **do**
    Extract data $\mathbf{X}_{opt}$ from **X**,
    for $c = 2$ **to** 10 **do**
      Set ECM hyperparameters $\alpha = 0.1$, $\beta = 1.1$, and $\delta^2 = 80$, cf. [d];
      Run ECM using c, $\mathbf{X}_{opt}$, and compute Non-specificity (N), cf. 4.6;
    end
  end
  Select c^* corresponding to the minimum of N;
  for each parameter of FS **do**
    Run ECM using c^* and compute Average Silhouette Width (asw);
  end
  Select asw^* corresponding to the maximum of asw;
end
Select best parameter corresponding to the maximum asw^*;
return Optimal Parameter for Feature Selection;

The standardized non-specificity takes values in $[0, 1]$ and must be minimized. A lower non-specificity reflects a better specificity of belief masses assigned to clusters. This metric is particularly useful for evaluating ECM in scenarios where imprecision plays a significant role, such as multidimensional and uncertain data.

4.2.4 Cluster analysis

We evaluate the interpretability of clusters by examining the number of selected features at each iteration. The use of a limited number of features promotes easier human understanding and a more concise representation, which is a key characteristic of interpretable clusters. Furthermore, once different groups of trajectories are identified, it is essential to categorize each cluster by calculating comparative statistics and profiling patients in each cluster using statistical analysis techniques such as descriptive statistics, statistical tests, and logistic regressions. Categorization involves quantifying the overall magnitude of differences with the proportion of explained variance and comparing means conditionally to the clusters of active variables. Indeed, a variable is considered more active if its proportion of explained variance is high. Additionally, if its conditional mean is large within a given cluster, it plays an important role in characterizing the membership of observations in that cluster. Descriptive analysis, on the other hand, will assess the significance of other variables in the study (personal questionnaire and self-questionnaire) relative to the cluster's variable of

interest. Mean and standard deviation Mean(sd), are presented for quantitative attributes, and numbers and proportions for qualitative attributes n(%). For results from statistical tests performed, Student tests for quantitative attributes and Chi-squared test for qualitative attributes are used. In order to account for repeated measures of certain explanatory variables, as described in Section 4.3.1, we employ appropriate paired statistical tests. We used the paired Student's t-test for quantitative variables and the Fisher's exact test for categorical variables. P-Value were adjusted using Bonferroni correction to control family-wise error rate.

4.3 Application and results

Statistical analysis is performed using R software v3.6.3 and Python software v3.10.4. The Type I error rate α is set to 0.05 for two-tailed statistical tests. No adjustment of the α threshold is made. Missing values corresponding to periods of non-observation are not imputed and are treated as such. Patients with less than one month of follow-up (less than 5 recordings) are excluded from the analysis (Figure 4.6). For each patient, the date of study inclusion is defined as their index date.

4.3.1 Data presentation

The data used in this study were collected via the eDOL platform, a digital health tool described in detail in Section 2.1.2 of the chapter 2. This platform combines a smartphone application for patients and a web interface for physicians, enabling comprehensive data collection for clinical and therapeutic monitoring.

As outlined in Section 2.1.3.1, the eDOL platform captures three main types of data: socio-demographic and lifestyle information, clinical evaluations, and barometric data. For this study, the focus is primarily on the barometric data, which are collected weekly through self-assessments on attributes such as sleep, mood, fatigue, stress, body comfort, and pain. These attributes are summarized in Table 2.1. The barometric data present unique challenges for analysis, as detailed in Section 2.1.3.2. These include irregularity in the time series due to inconsistent patient participation, the discrete nature of the measurements on a scale of 0 to 10, and the inherent subjectivity of the data, which reflects patients' perceptions in real-life conditions. These factors introduce uncertainty and complexity that must be accounted for during feature extraction and clustering.

The feasibility clinical study conducted in 2019 and published in 2022 [28] showed a continuous adherence and utilization rate of the eDOL application at 61.9%. However, we did not have sufficient patient follow-up data during the manual version of eDOL (version 1), which lasted 1 year between 2019 and 2020. Indeed, the sensitivity analysis of the switch to version 2 (chatbot-assisted) shows that the utilization rate increased from 21.09% (version 1) to 62.01% (version 2, the date at which we extracted the data used in this study). To date, 1590 patients have been included in the study, with an adherence rate of 62.01% (986 patients). Approximately 38% of patients (604 patients) were excluded due to the absence of barometric and personal data. Among the 986 patients who have enrolled with completed data, from 664 patients have been analyzed. Because, we excluded 322 patients from the

study due to insufficient adherence to the barometer completion. These patients had less than one month of follow-up, which resulted in less than 5 times of barometer questionnaire completion. In fact, we considered a minimum follow-up period of one month, equivalent to 5 completions of the questionnaire. This decision was based on the observation that patients need a period of one month to adapt to using the application. The Figure 4.6 summarizes the patient selection process for the study. A total of 14,090 fillings have been considered, with an average of 22 fillings per patient, corresponding to an average follow-up period of 5 months, and a total follow-up duration of 73 weeks (approximately 19 months). The patients were over 18 years old, with an average age of 48 years, and predominantly employed with a `A levels + 2` education level.

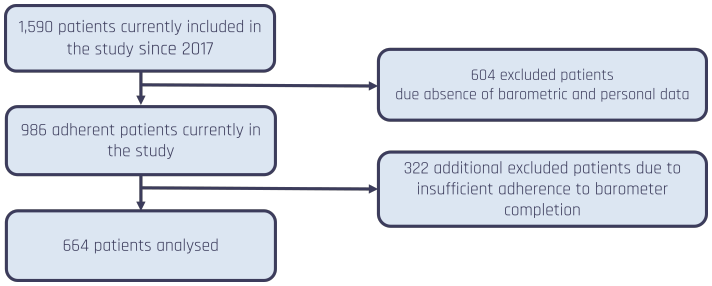


Figure 4.6: Patient Flow-Chart.

We examined the characteristics (trend and dispersion variables) of the time series for the barometers using box plots (cf. Figure 4.7). On average, the scores for pain and fatigue are relatively high, while the scores for stress and sports activity are relatively low. We also note the low variability of the fillings series for all barometers, except for sports and non-sports activities, which are relatively more dispersed. These characteristics describe the time series of the barometers and provide new insights into the evolution and variation of the fillings series, which can be extracted and considered instead of raw data [195, 211, 212].

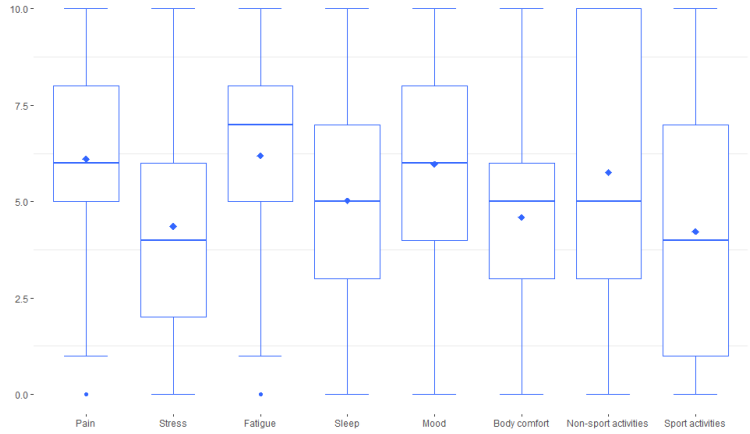


Figure 4.7: Boxplot of barometers – visualization of trend and dispersion of all fillings series for the eight barometers – ♦ corresponds to the mean.

According to Figure 4.8, the density curves indicate that the features Pain, Stress, Fatigue, Sleep, Mood, and Body Comfort have density curves that resemble normal distributions. On the other hand, the characteristics Non-sport Activities and Sport Activities have density

curves that deviate from normality. The Non-sport Activities feature exhibits a right-skewed distribution, indicating a higher frequency of higher values compared to lower values. The Sport Activities feature shows a bimodal distribution, suggesting the presence of two distinct groups within the data. Overall, the normal-like distributions of certain features indicate a relatively balanced spread of data, while the deviations from normality in others highlight the presence of outliers or potential data clusters that require further investigation. Furthermore, the barometers are not correlated with each other. However, linear causal relationships indicate that the scores for fatigue, stress, and pain evolve in the same direction, while sleep, body comfort, and mood also show a positive trend. Finally, The width, shape, and position of the density curves further confirm the dispersion and variability observed in Figure 4.7 above.

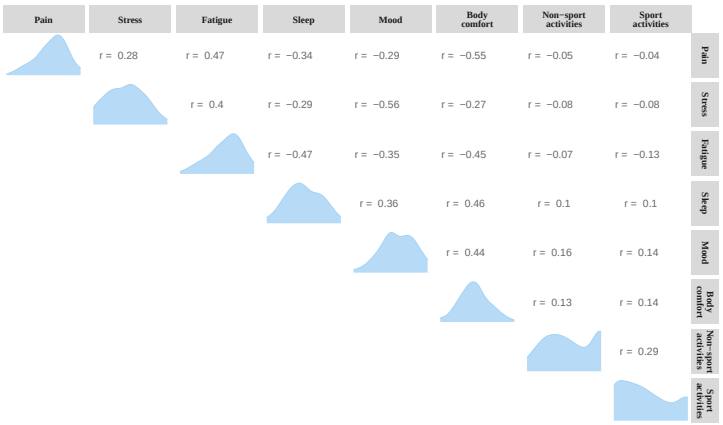


Figure 4.8: Assessing Normality and Correlation in Variable Distributions of barometers.

4.3.2 Applying feature extraction and selection

We initially extracted features based on manual observations through visualizations of time series during the data exploration stage. These manually extracted features included maximum, minimum, mean, and median values. Subsequently, additional features were automatically extracted using tsfresh. As a result, a large number of features capturing information about behavior, dispersion, homogeneity, kurtosis, asymmetry, and trend of the raw barometer series were extracted. In total, 23 features were computed from the raw data of each of the 8 barometers. The resulting table had a size of 664 patients and 184 variables. Features with a high proportion of missing values, i.e., greater than 80%, were removed, resulting in the elimination of 6 variables. The elimination threshold of 80% was based on manual observations and evaluations of repeated experiments. The remaining variables with missing values, 5 in total, were imputed using a semi-parametric multiple imputation approach (Predictive Mean Matching - PMM). After feature extraction and handling of missing data, we were left with a final dataset consisting of 178 variables. From this stage, we can now proceed to selecting the most relevant variables for our analysis.

The unsupervised feature selection algorithm, Spectral Feature Selection, assigns equal importance to all variables, while the Laplacian and Random Forest algorithms offer different levels of selection based on their importance. This characteristic of Spectral Feature

Selection is due to its approach of constructing the Laplacian matrix used for calculating variable importance scores. Specifically, when variables are strongly correlated, as in our case after feature extraction, it becomes more challenging to select the most relevant variables because the spectral approach produces a Uniform Laplacian [208]. In this case, during the decomposition of eigenvectors, the obtained importance scores are uniform or equal for all variables. Additionally, the Spectral algorithm solely relies on similarity measures in its approach, without considering the individual contributions of variables, i.e., differences between the values of the data sets. This leads to an equal importance assigned to variables, which conceals the differences between them. This characteristic of Spectral Feature Selection does not align with our objective, which is to eliminate the least important variables to improve the quality of our clustering and simplify for explanatory modeling. Therefore, we focused on selection based on the Laplacian and Random Forest algorithms.

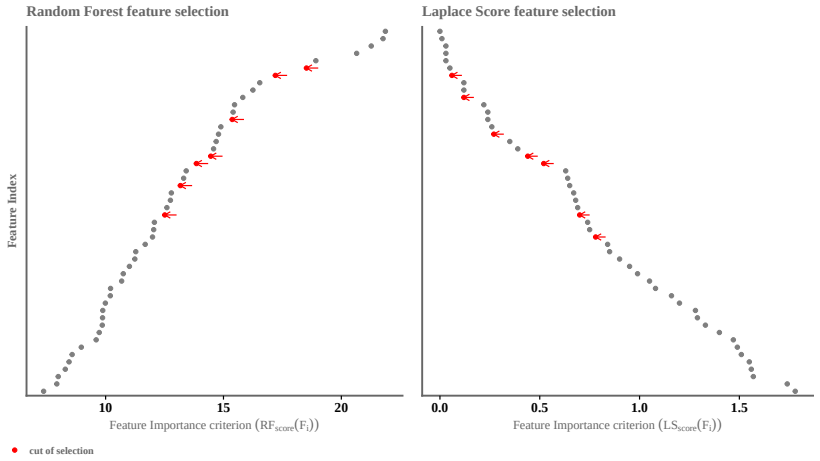


Figure 4.9: Visualization of the top 50 most important attributes and selection options .

According to the importance curves (Figure 4.9), the LS and URF algorithms offer several selection options based on minimizing and maximizing importance criteria, respectively. In total, we studied 15 selection options. For the Unsupervised Random Forest Feature Selection algorithm, we identified options 6, 7, 13, 18, 19, 22, 26 (first variables) based on the order of importance established by the algorithm. For Laplacian Score Feature Selection, the selections concern 7, 10, 15, 18, 19, 26, 29 (first variables). As for the Unsupervised Spectral Feature Selection algorithm, which assigns equal importance to all variables, we used all 184 available variables. For each selection option, we performed a grid search to determine the optimal number of clusters C (ref. to Section 4.2.2.4). This search was conducted both for normalized data using the standardized approach and for raw data without normalization. Interestingly, all selection options, whether combined with normalization or not, showed an optimal number of clusters C equal to 2 or 3, considering minimal Non-specificity, as can be seen in Figure 4.10. In all selection options with a large number of variables, the non-specificity value was high, regardless of the value of C. In all cases, the Nonspecificity value was maximized as C increased. Next, we evaluated the separability quality of the partitions for each selection using the optimal number of clusters (C optimal). The optimal selection for each algorithm, which maximizes separability (ASW) and minimizes compactness (Non-specificity), is retained (Figure 4.11). The final best selection is obtained with the URF selection algorithm, which retained 7 variables primarily related to the trend of the source

data. This selection exhibits partitioning quality with an ASW of 0.27 and a Non-specificity of 0.29, and it is based on non-normalized data (Table 4.1).

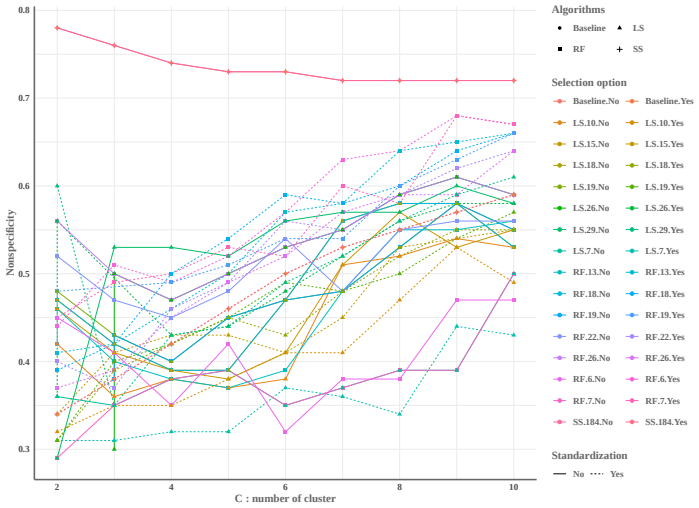


Figure 4.10: Grid search optimal number of clusters; Baseline corresponds to all features.

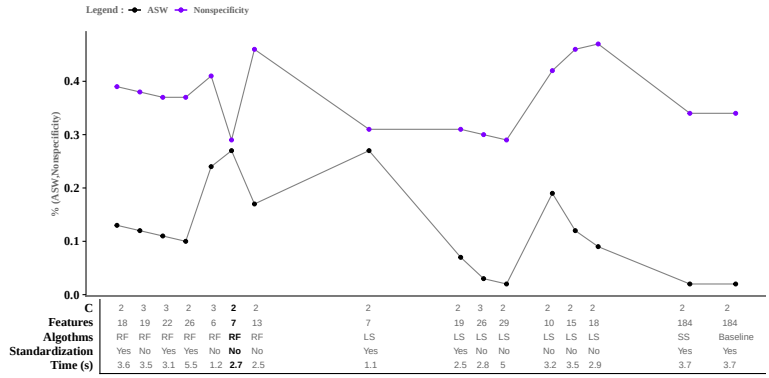


Figure 4.11: Selection of the best partition.

4.3.3 Applying evidential clustering and analysis of results

The parameters of the ECM clustering algorithm were optimized during the final classification. The optimal values of the hyperparameters for a better partition were $\alpha = 0.1, \beta = 1.1$, and $\delta^2 = 80$. Initialization by k-means was also chosen, and a type of simple focal sets was used to obtain atypical (empty set) and uncertain clusters. Thus, with the optimal number of clusters set to 2, ECM identifies a total of 2^2 partitions. The credal partition is converted into a hard partition using the maximum mass function for each patient. Indeed, a hard partition $\mathscr{G} = (G_1, \ldots, G_{2^K})$ is derived from credal partition by defining the hard cluster $G_k (k = 1, ..., 2^K)$ as $G_k = \left\{ \mathbf{x_i} : m_{ik} \geq m_{il} \quad , \forall l \in \left\{1, \ldots, 2^K\right\} \right\}$. The quantity m_{ik} indicates the membership belief of object $\mathbf{x_i} (i = 1, \ldots, n)$ in credal cluster $k (k = 1, \ldots, 2^K)$. This allows us to obtain three hard partitions, i.e. two singleton clusters (cl_1 and cl_2) and an uncertain subset (cl_1, cl_2) which we have labeled $cl\_uncertain$, as shown in Figure 4.12. The respective sizes of these clusters are 216, 157, and 291 elements.

Table 4.1: Final Comparison: Optimal Selections of each selection algorithm. Standardization indicates if performance (high asw and low nonspecificity (NS)) is achieved when data is normalized or not.

Algorithms	Nbr of features	Standardization	C	NS	ASW	Time
Laplacian Score	7	Yes	2	0.31	0.27	1.1
Random Forest	**7**	**No**	**2**	**0.29**	**0.27**	**2.7**
Spectral Score	184	Yes	2	0.34	0.02	3.7
Baseline	184	Yes	2	0.34	0.02	3.7

The average diameters of the clusters are 27.84, 29.12, and 28.92. The average distances within the clusters are 9.12, 9.96, and 8.61, while the median distances are 8.65, 9.27, and 8.14. The Dunn index value is 0.04 for each cluster, suggesting limited separation between the clusters. This means that the clusters are relatively close to each other. Finally, the entropy is 1.07 for each cluster, indicating some uniformity in the distribution of elements within each cluster.

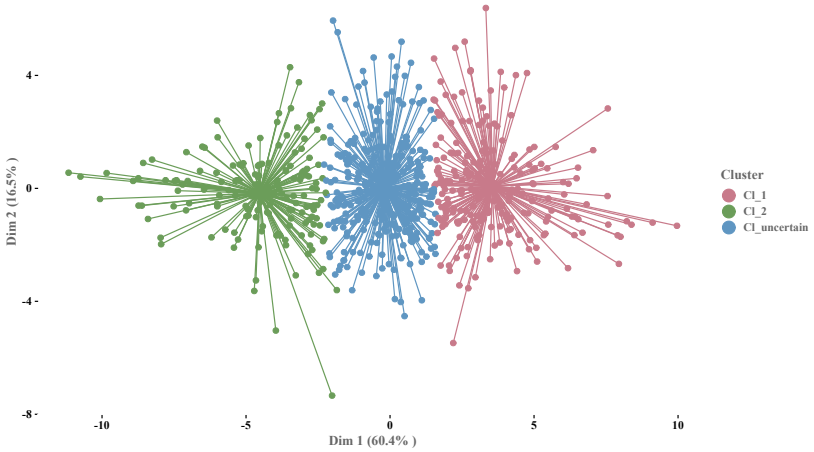


Figure 4.12: Visualization of the crisp credal partition clusters on the normalized PCA factorial axes of the selected features

Advanced analysis of the clustering model parameters reveals the significant impact of the uncertain cluster on the quality of the identified typologies. By removing the uncertain elements, several improvements are observed. The silhouette of the final partition significantly improves (ASW=0.55), undicating increased separability between the `cl_1` and `cl_2` clusters. Additionally, clustering stability indicators such as the determination coefficient (R2sq with uncertain set = 0.52, R2sq without incertain set = 0.75) and the Calinski-Harabasz Fisher index (CHsq with incertain set = 365.95, CHsq without uncertain set = 689.68) confirm that eliminating uncertain points enhances dissimilarity between clusters and increases the variability explained by clustering. Analysis of variance and density also support this trend. In conclusion, excluding uncertain points improves the quality of the final partition in terms of cluster separability and explained variability. These results highlight the crucial importance of considering data uncertainty to obtain clusters that reflect reality.

4.3.3.1 Evidential clustering compared with various clustering approaches

To compare the performance of ECM with various partitioning algorithms (HCM and FCM) and a hierarchical clustering algorithm (CAH) on our complex data, we utilize the best combination of feature selection obtained in Section 4.3.2.

The hard clustering algorithms CAH and HCM are applied to these selected data to obtain a set of groups $\mathscr{G} = (G_1, \ldots, G_K)$. FCM will also be applied to these selected data to initially produce a fuzzy partition into K fuzzy clusters. Subsequently, a hard partition $\mathscr{G} = (G_1, \ldots, G_K)$ is derived from this fuzzy partition by defining the hard cluster $G_k (k = 1, ..., K)$ such $G_k = \{\mathbf{x_i} : u_{ik} \geq u_{im} \quad \forall m \in \{1, \ldots, K\}\}$. The quantity u_{ik} represents the membership degree of object $x_i (i = 1, \ldots, n)$ in fuzzy cluster $k (k = 1, \ldots, K)$. For CAH, we used Ward's $(ward.D2)$ dissimilarity measure for the link between clusters. Thus, based on the hard ECM partition and to define a hard partition comparison framework for the algorithms overall, we posit $k = 3$. Each clustering algorithm was run (until the convergence to a stationary value of the adequacy criterion) 100 times and the best result was selected according to the adequacy criterion. To compare the clustering results provided by these approaches, an external criteria, such as the Adjusted Rand Index (ARI), and internal criteria, like the Average Silhouette Width (ASW), are considered.

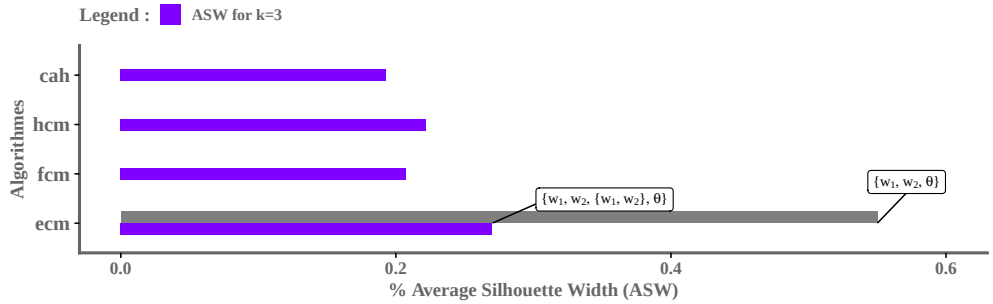


Figure 4.13: Evidential clustering compared with various clustering approaches. With the atypical cluster $\emptyset = \{\}$.

The Figure 4.13 compares the clustering silhouette quality of ECM with other clustering algorithm variants using the ASW criterion. Overall, ECM appears to perform best, followed by HCM, FCM and CAH. The gray bar corresponds to the ASW value when the uncertain partition is not considered, as discussed in the second paragraph of section 4.3.3. ARI allows us to compare the assignments between different clusters determined by the algorithms. It considers pairs of patients and measures the degree to which these pairs are grouped in the same way in two partitions obtained from two different algorithms. Thus, we observe in the Figure 4.14, that in most cases, the algorithms (FCM and CAH) do not agree with ECM regarding the classification of patients in the uncertain cluster (cluster 3). Indeed, FCM and CAH classify a significant portion of patients from the uncertain cluster into cluster 2. Additionally, all three algorithms also classified a number of patients from ECM's cluster 1 in their cluster 3.
The low performance of CAH on our data is certainly due to the complexity of the data and, in particular, the high presence of outliers and noisy values. However, HCM and FCM, although robust to noisy data, do not have the ability to exclude outliers in the process of

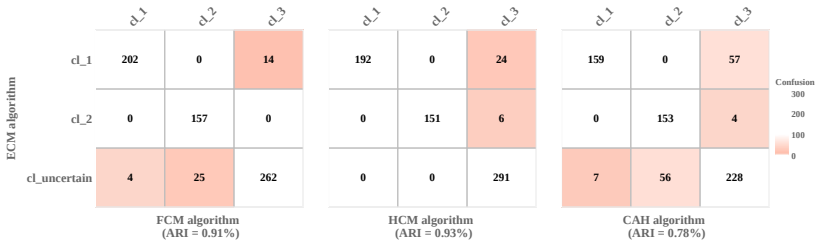


Figure 4.14: Comparing based of degree of similarity between partition furnished by ECM with various partitioning approaches.

selecting prototypes. We also remarked that when the fuzziness parameter (m) of FCM is less than 1.3 (its value considered in the final result (Figure 4.13), its performance is very close to that of HCM. For values of m greater than 1.5 (its default value in the algorithm), performance drops, leading to complete fuzziness, meaning that all data points have equal membership to all clusters.

4.3.3.2 Categorization of clustering results

The categorization of clusters is based on comparing the means of active variables across clusters while considering the overall interpretability of each variable, see Figure 4.15. The idea is to compare the means of the variables that give the best partition, selected in the section 4.3.2, and conditional on the clusters obtained. By analyzing the total explained variability, it appears that the definition of groups is mainly influenced by aspects such as sleep, fatigue, and bodily comfort. These variables contribute significantly, accounting for approximately 65% of the variability. Stress and mood, on the other hand, contribute slightly (between 30% and 40%) to this clustering. Comparing the means of active variables during clustering shows that `cl_2` is strongly determined by variables such as `Sleep_mean`, `Sleep_median`, `Bodily.comfort_mean`, and `Mood_mean`. On the other hand, cluster `cl_1` is more explained by variables like `Fatigue_mean`, `Stress_mean`, and `Pain_mean`. `cl_2` groups patients suffering from chronic pain related to Fatigue and Stress. In contrast, `cl_1` groups patients suffering from chronic pain related to Sleep, Bodily comfort, and Mood. Patients belonging to the uncertain cluster are those who exhibit common characteristics of both groups.

The analysis of variability explained by the variables performed above is a traditional approach used to try to understand the decisions made by the clustering algorithm in its grouping process, i.e. to explain the contribution of the variables and categorize the clusters. Various explainable artificial intelligence (XAI) methods, such as SHapley Additive exPlanations (SHAP) [213], have recently been proposed to estimate feature contributions agnostically to learning processes. The use of SHAP in the explicability of totally unsupervised models such as clustering is post-hoc (central approaches). Indeed, we train a supervised model on the cluster assignment given by the clustering algorithm with regard to our input data. We considered the supervised RF algorithm for exploring the explicability of the ECM algorithm. Note that it was possible to choose other supervised learning algorithms for this task, such as XGBoost [214], or to choose the best performing one by comparing them. However, our choice was motivated by the fact that unsupervised RF performed best for the feature selection step on our data. We trained the RF to predict

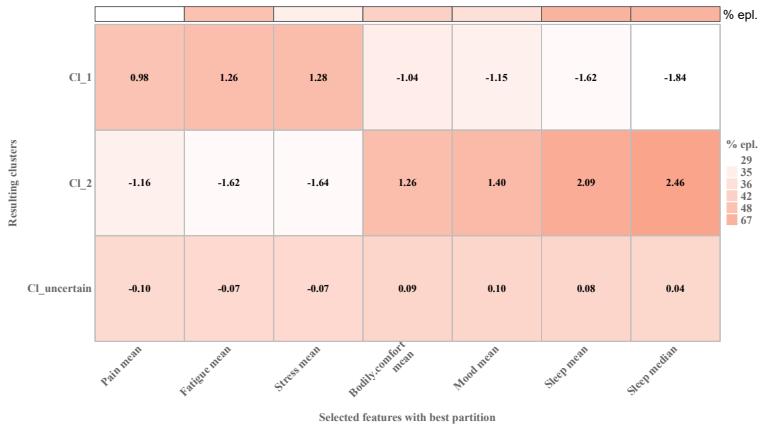


Figure 4.15: Categorization of clustering results: Comparison of means of active variables by group and proportion of variance explained. %epl: percentage of variability explained by the clustering of each feature .

the cluster numbers assigned by ECM. The optimized parameters (number of trees to grow (225) and maximum depth (2)) achieved an ARI of 98%. This model was used with SHAP to explain the clustering results obtained by ECM. In Figure 4.16, we can observe the SHAP values and the average SHAP values obtained for each feature.

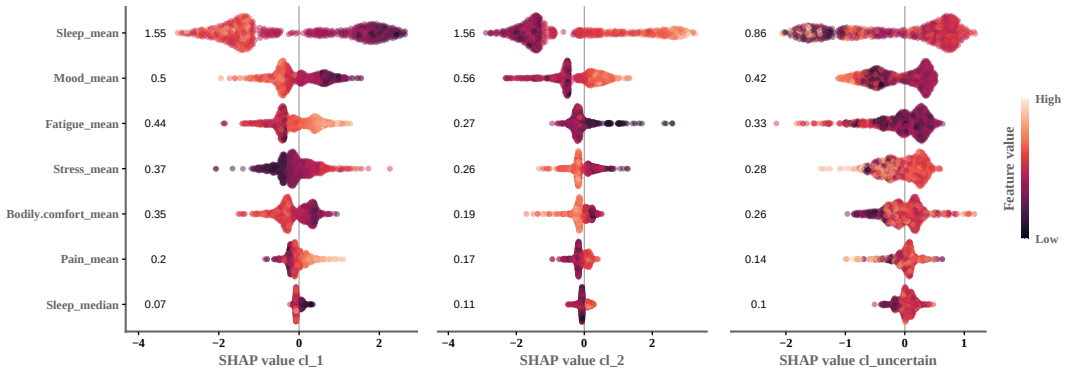


Figure 4.16: Density scatter plot showing SHapley Additive exPlanation (SHAP) values for each feature, reflecting how much impact each feature has on a predictive random forest model output trained on ECM clustering assignment.

The averages of the SHAP values confirm the observations obtained in the Figure 4.15. The variables `Sleep_mean`, `Sleep_median` and `Mood_mean` contributed more to `cl_2` while the variables `Fatigue_mean`, `Stress_mean` and `Pain_mean` contributed more to `cl_1`. However, we can observe that `Bodily.comfort_mean` behaved differently from previous interpretations. SHAP offers more interpretability; for example, we can observe the weights of each individual in the clustering locally in each cluster and in each variable. While patients in cluster 1 tend to have high mean sleep over time, those in cluster 2 have low mean sleep. We also observe that the SHAP values in the 3rd graph of the figure (uncertain cluster versus (cluster 1 and 2)) are distributed evenly on either side of the vertical line (v). This means either a single cluster crossed by v, or two clusters on either side.

4.3.3.3 Descriptive analysis of socio-demographic factors associated with clusters

The interpretability of clusters through descriptive analysis and statistical tests of factors associated with membership in each cluster allowed us to identify the profile of these patients. Our initial examination focused on the clearly identified clusters (clusters 1 and 2) and socio-demographic variables (Table 4.2). Statistical analysis of socio-demographic variables among patients with chronic pain from `cl_2` and `cl_1` clusters revealed several significant results ($p < 0.05$). Regarding age, a significantly different distribution was observed between the two clusters, with a higher proportion of patients over 50 years old in the discomfort cluster (26%) compared to the well-being cluster (12%). Furthermore, the frequency of pain showed a significant association, with a larger proportion of patients reporting continuous pain throughout the day in the discomfort cluster (26%) compared to the well-being cluster (11%). As for smoking, a significant difference was found between the two clusters, with patients in the discomfort cluster being less likely to be smokers (18%) compared to the well-being cluster (31%). Similarly, alcohol consumption differed significantly between clusters, with a higher proportion of patients in the discomfort cluster reporting alcohol consumption (69%) compared to the well-being cluster (58%). Variables such as nocturnal awakening, sleep disturbance, and frequency of episodes were also significantly associated with the discomfort and well-being clusters. However, variables such as education level, number of children and family situation did not show significant differences between the two clusters ($p > 0.05$). In summary, the statistical analysis of socio-demographic variables highlights significant differences between the `cl_1` and `cl_2` groups. These differences mainly concern age, smoking, alcohol consumption, pain frequency, and crisis frequency.

Table 4.2: Descriptive analysis of socio-demographic features associated with clusters .

Variables	Cl_2, $N = 157$[1]	Cl_uncertains, $N = 291$[1]	Cl_1, $N = 216$[1]	p-value[2]
Age				<0.001
18 to 30 years excluded	3 (1.9%)	5 (1.7%)	12 (5.6%)	
30 to 50 years included	113 (72%)	237 (81%)	179 (83%)	
Over 50 years	40 (26%)	49 (17%)	25 (12%)	
Number of Children				<0.11
1 to 2 children	69 (44%)	143 (49%)	81 (38%)	
No children	74 (47%)	118 (41%)	108 (50%)	
More than 2 children	14 (8.9%)	30 (10%)	27 (12%)	
Education Level				<0.12
Advanced	32 (20%)	64 (22%)	32 (15%)	
High School Diploma	34 (22%)	54 (19%)	54 (25%)	
Intermediate	57 (36%)	86 (30%)	67 (31%)	
Secondary	34 (22%)	87 (30%)	63 (29%)	
Smoker				<0.020
No	129 (82%)	219 (75%)	150 (69%)	
Yes	28 (18%)	72 (25%)	66 (31%)	
Blood Alcohol Level				<0.043

No	48 (31%)	95 (33%)	90 (42%)	
Yes	109 (69%)	196 (67%)	126 (58%)	
Pain Frequency				<0.001
Throughout day	41 (26%)	58 (20%)	24 (11%)	
Frequent	19 (12%)	29 (10.0%)	11 (5.1%)	
Night and day	48 (31%)	131 (45%)	134 (62%)	
Rare	3 (1.9%)	1 (0.3%)	1 (0.5%)	
Very frequent	46 (29%)	72 (25%)	46 (21%)	
Pain Crisis Frequency				<0.001
Exceptional	13 (8.3%)	3 (1.0%)	1 (0.5%)	
Frequent	52 (33%)	136 (47%)	99 (46%)	
Rare	60 (38%)	74 (25%)	37 (17%)	
Very frequent	32 (20%)	78 (27%)	79 (37%)	
Night Awakening				<0.001
At least once per night	23 (15%)	65 (22%)	44 (20%)	
Never	33 (21%)	17 (5.8%)	8 (3.7%)	
Several times per night	25 (16%)	88 (30%)	104 (48%)	
Sometimes	76 (48%)	121 (42%)	60 (28%)	
Prevents Sleep				<0.001
Never	33 (21%)	25 (8.6%)	6 (2.8%)	
Sometimes	79 (50%)	137 (47%)	81 (38%)	
Often	34 (22%)	102 (35%)	71 (33%)	
Every night	11 (7.0%)	27 (9.3%)	58 (27%)	
Family Situation				<0.4
Single	45 (29%)	84 (29%)	79 (37%)	
Relationship	109 (69%)	199 (68%)	133 (62%)	
Widow / Widower	3 (1.9%)	8 (2.7%)	4 (1.9%)	

[1]n(%),
[2]Fisher's Exact Test for Count Data with simulated p-value; Pearson's Chi-squared test
[2]Student's t-test

4.3.3.4 Descriptive analysis of clinical factors associated with clusters

The descriptive analysis of clinical factors associated with clusters revealed several significant results ($p < 0.05$), as shown in Table 4.3. These results highlight significant differences between the `cl_2` and `cl_1` groups across various parameters. Firstly, variables related to pain-related perceptions and beliefs showed significant differences between the two clusters. Cluster `cl_2` had a higher proportion of patients with high levels of catastrophizing (21%) compared to cluster `cl_1` (48%). Additionally, kinesiophobia (fear of movement) was more prevalent in cluster `cl_1` (73%) than in cluster `cl_2` (54%). Significant differences were also observed regarding depression, anxiety, and cognitive disorders. Cluster `cl_1` exhibited higher rates of depression, anxiety, and cognitive disorders compared to cluster `cl_2`. In terms of impact on daily life, patients in cluster `cl_1` reported higher levels of

impact on work, physical appearance, and life satisfaction compared to patients in cluster `cl_2`. Furthermore, sleep disturbances were more prevalent among patients in cluster `cl_1` compared to those in cluster `cl_2`. In summary, the analysis of clinical factors reveals significant distinctions between the `cl_2` and `cl_1` groups, with important clinical implications. These differences involve pain-related perceptions and beliefs, personality traits, the presence of depression, anxiety, and cognitive disorders, impact on daily life, and the prevalence of certain health problems.

Table 4.3: Descriptive analysis of clinical factors associated with clusters

Variables	Cl_2, $N = 308$[1]	Cl_uncertain, $N = 514$[1]	Cl_1, $N = 352$[1]	p-value[2]
Catastrophism				<0.001
Proven Catastrophizing	66 (21%)	159 (31%)	168 (48%)	
Unproven Catastrophizing	242 (79%)	355 (69%)	183 (52%)	
Kinesiophobia				<0.001
Confirmed kinesiophobia	167 (54%)	331 (64%)	260 (73%)	
Unproven kinesiophobia	140 (46%)	185 (36%)	98 (27%)	
Alexithymia				<0.001
Proven alexithymia	60 (38%)	146 (50%)	137 (64%)	
Unknown alexithymia	97 (62%)	144 (50%)	78 (36%)	
Optimist	12.55 (2.27)	11.95 (2.04)	11.52 (2.03)	<0.001
Belief in justice	18.4 (4.4)	17.3 (4.1)	16.4 (4.1)	<0.001
Belief in injustice	13.6 (5.4)	15.6 (4.8)	17.2 (4.6)	<0.001
Depression				<0.001
Proven depression	66 (22%)	215 (42%)	206 (58%)	
No-proven depression	240 (78%)	302 (58%)	151 (42%)	
Anxiety				<0.001
Proven anxiety	24 (7.8%)	123 (24%)	153 (43%)	
No-proven anxiety	282 (92%)	394 (76%)	204 (57%)	
Cognitive problems				<0.001
No-probable disorders	156 (52%)	177 (35%)	61 (17%)	
Probable disorders	146 (48%)	329 (65%)	289 (83%)	
Sleep problems	316 (58%)	364 (41%)	127 (21%)	<0.001
Impact on work	11.7 (5.5)	13.2 (5.4)	13.8 (5.1)	<0.001
Impact on physique	6 (9)	10 (12)	8 (12)	<0.001
Life satisfaction	22 (6)	19 (7)	16 (6)	<0.001

[1]Mean(sd) or n(%),
[2]Fisher's Exact Test for count data with simulated p-value; Pearson's Chi-squared test
[2]Pairwise Student's t-tests

4.3.3.5 Identification of patient profiles

The phase of statistical tests was used to select variables that significantly explain membership in a cluster. However, this analysis does not take into account multidimensional interaction. To address this, we used multinomial logistic regression, which isolated the

effects of each explanatory variable, identifying residual effects on cluster membership. Furthermore, for variables with multiple occurrences, we will use mixed multinomial logistic models. The use of this more advanced approach allows us to obtain a more comprehensive understanding of the marginal effects and complex interactions of factors that influence membership in different clusters, see Figure 4.17.

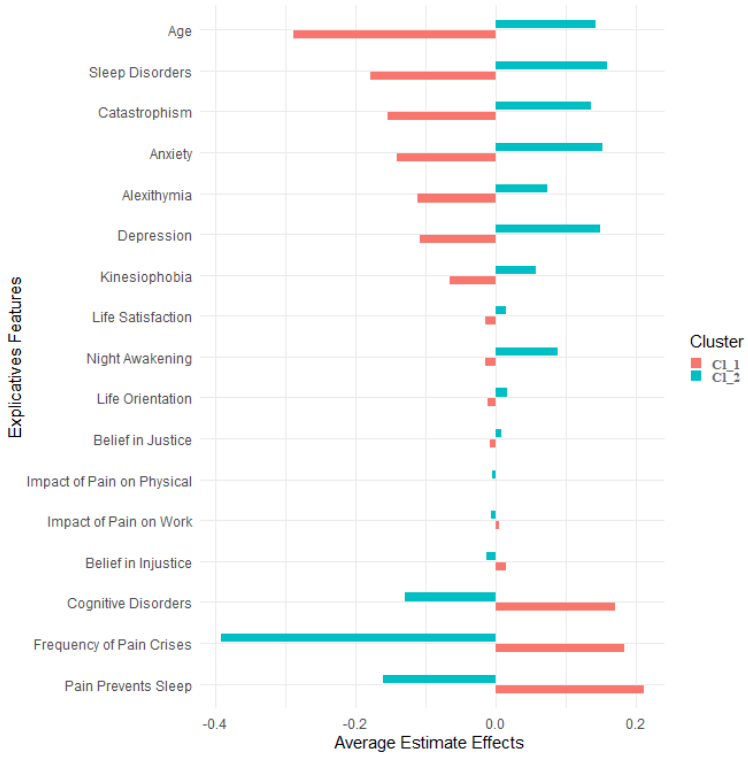


Figure 4.17: Estimation of the average marginal effects of significant explanatory variables in a multinomial context. Marginal effects measure the impact that an instantaneous unit change in one variable has on the outcome variable (clustering variable) while all other variables are held constant.

The results of multinomial regressions explaining the clusters reveal diverse profiles among the identified cluster patients. These profiles cover socio-demographic aspects, pain perceptions by patients, their psychological states, comorbidities, and quality of life. In cluster 2, generally older patients who do not wake up at night are observed. In terms of the psychological profile, these patients do not show signs of depression, anxiety, sleep disorders, catastrophism, or kinesiophobia. Regarding quality of life, they appear to have overall good health, although autonomy and mobility issues are present. For cluster 1, patients are characterized by a higher frequency of pain crises, preventing them from sleeping every night. They perceive their pain as perpetual. Patients in this cluster also likely exhibit cognitive disorders. In terms of quality of life, these are patients who express high satisfaction in their life. These results highlight the heterogeneity of profiles among patients with chronic pain, emphasizing the importance of a personalized approach to their management.

4.4 Discussion

We conducted multiple experiments by varying feature representations, unsupervised feature selection techniques, and clustering algorithms. Within the clustering approach proposed in this study, we observed differences among clustering algorithms and applied feature selection techniques. Among these experiments, concise representations were achieved through feature selection using the Random Forest (RF) algorithm for hand-crafted features and tsfresh extraction features. When combined with ECM clustering using the Euclidean distance metric, this approach exhibited superior performance, achieving an ASW score of 0.27. The main application of our results would be to enable an assessment of the trajectories of chronic pain and the care instituted in parallel. Analysis of these trajectories, combined with detailed patient characterization, could lead to the creation of predictive tools for the effectiveness of care and/or the evolution of the pathology. These tools would provide personalized medicine for each patient profile, assisting clinicians in managing their patients.

However there are several possibilities for improving this work. First, the linearity of our approach, due to the use of the linear clustering algorithm ECM, can be seen as a limitation. However, we also believe that a nonlinear algorithm will not answer the doctors' questions about explicability. Furthermore, preliminary study on a nonlinear exploration using the Kernel k-means algorithm [215] shows not satisfying results are not very satisfying. Indeed, we can observed that separability over several nonlinear kernels into three clusters yields silhouettes (asw) inferior to those of ECM (asw=0.27) on the selected data and baseline data (Figure 4.18). However, the exploration was conducted with the default configuration of kernels hyperparameters, and the simple change of initialization of kernel parameters completely changes the cluster population, but with silhouettes close to the default parameters. This raises an important question about the repeatability of the results [216]. Other nonlinear algorithms based on neural networks, such as Deep Embedded Clustering (DEC) [56], or other nonlinear soft clustering kernels like the fuzzy c-mean kernel [217], could be explored. We have not verified the ASW of these algorithms, but we doubt that the results are more interesting on this type of data, especially given the problems of reproducibility and repeatability, a linear approach might be more stable.

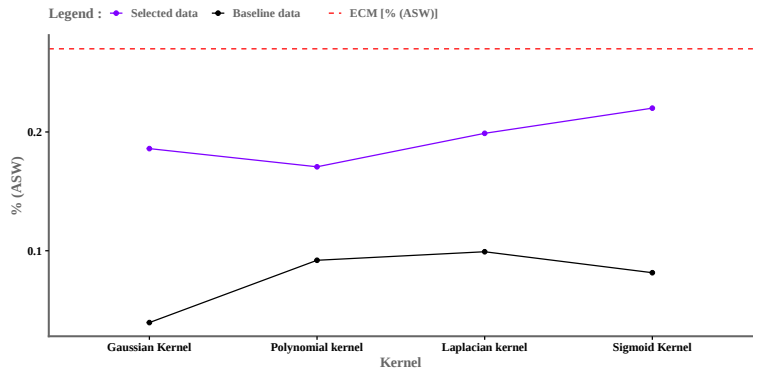


Figure 4.18: Kernel k-means clustering based on the kernel type .

Also, like all partitioning clustering approaches, ECM presents the limitation associated with defining the number of clusters (c). However, the optimal c with clustering algorithms

like Mean Shift reinforces our results. Mean Shift clustering conducts classification of a set of data points by using steepest ascent to local maxima in a kernel density estimate [218]. Based on a linear approach for nearest neighbor searches and a Gaussian or RBF (Radial Basis Function) kernel applied to our data from the selection step, we observe on the Figure 4.19, that the optimal number of clusters is between 2 and 4, with silhouette accuracy between 17% and 39% respectively, with a very large neighbor search radius ($r = 3.5$). The parameter bandwidth r is very important in the Mean Shift algorithm and must be chosen in the same way as the number of clusters in partitioning algorithms. We therefore tested several values of r, knowing that the most commonly used default value is 1, this confirms the noisy aspect of our data.

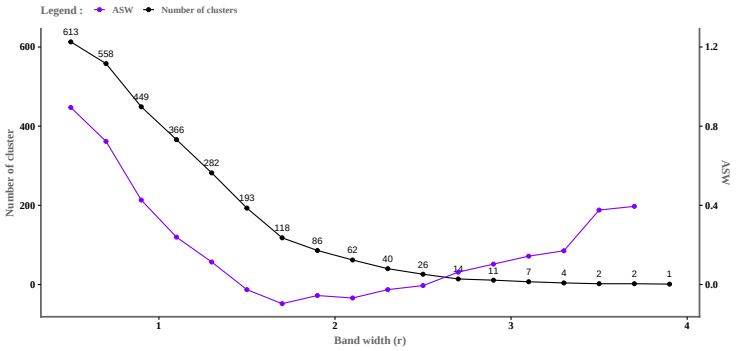


Figure 4.19: Number of clusters depending on bandwidth.

It is also important to note that the comparison between ECM and the other variants of clustering algorithms in Section 4.3.3.1 has limitations in generalizability. Because, as we specified in the section 4.2, according to Pugazhenthi A. and al., the optimal clustering method depends on the qualitative requirements of the clustering process, the applications and the unique characteristics of the data used.

From a practical perspective, incorporating new explanatory variables (e.g., medical hospitalization, drug, etc.) could contribute to a more detailed characterization of patient profiles within each cluster. From a general standpoint, although the approach applied in this paper, which is based on feature extraction and selection, yields results with easy interpretability, we believe that clusters separability performance could be enhanced. Furthermore, feature extraction may have limitations when the extracted features do not capture all the information present in the raw data, particularly when data is complex, subjective and uncertain. Indeed, an evidential clustering approach that directly uses raw data (longitudinal or time series data) might offer improved clustering performance for this type of data while accounting for the uncertainties inherent in these data. This entails selecting appropriate similarity functions and employing an evidential relational clustering approach. The algorithms studied show a low capacity to efficiently separate our data, despite the careful search for the optimal number of clusters. Thus, in future work, we plan to explore mixed clustering, i.e. applying hierarchical clustering with the ECM partition as prior knowledge (for example, semi-supervised CAH [219]). Indeed, the ECM partition is considered as a preliminary (pre-classified) partition. The resulting dendrogram will mainly allow a more refined and parsimonious suggestion of potentially identifiable sub-partitions. The result will not be an immediate increase in data separability performance. However,

it will allow us to manage outliers and noisy data, while identifying potential sub-clusters within the clusters generated by the ECM. By taking advantage of the robustness of ECM and hierarchical clustering, we aim to improve our in-depth understanding of clustering results and manage uncertain individuals more effectively. In the context of our future perspectives as well, we wish to emphasize the importance of fairness in clustering approaches. It would be interesting for researchers working on the ECM algorithm to examine whether the model exhibits bias towards certain specific observations based on certain variables. This could contribute to a better understanding of the performance of the ECM algorithm and its ongoing improvement.

4.5 Conclusion

We have proposed an approach to identify typologies of care trajectories for patients suffering from chronic pain. This involves clustering longitudinal or time series data related to chronic pain, and more generally, clustering repeated-measurement data. This approach is based on feature extraction from raw-data, unsupervised feature selection of the most significant characteristics, and the use of an evidential clustering technique that takes into account uncertainties in the data. This approach, based on feature extraction from the time series, effectively dealt with the complex and noisy nature of the data. Feature selection, as opposed to projection, maintained interpretability, reduced dimension without significant loss of information, and improved clustering performance. It also made it possible to deal with the complex and irregular nature of the data, as well as the varying lengths of the sequences, thus ensuring a good balance between interpretability and performance of the results. The advantage of evidential clustering over fuzzy clustering and k-means clustering lies in its ability to model both uncertainty and imprecision in the data, while fuzzy clustering can only model imprecision, offering a better representation of nuances and variations in complex, raw data.

The results suggest that patients whose pain is related to fatigue and stress can be grouped together (cluster 1), while patients with pain related to sleep, physical comfort, and mood form another group (cluster 2). These groups exhibit distinct profiles covering socio-demographic aspects, patients' pain perceptions, psychological states, comorbidities, and quality of life. In cluster 2, older patients are found who do not experience sleep disturbances or mental health issues but face mobility challenges. In contrast, cluster 1 is characterized by a higher frequency of pain crises, a perception of perpetual pain, and possible indications of cognitive issues. These findings highlight the importance of considering the identified groups or clusters to guide a personalized and targeted approach in the management of patients with chronic pain, with the aim of improving their well-being and quality of life.

In the next chapter (chapter 5), we aim to improve silhouette accuracy by exploring clustering methods based on theory of belief function and using a similarity measure tailored to raw data, such as Soft-DTW or editing measures like Optimal Matching. We introduce a new relational clustering method, called Multi-View Evidential C-Medoid clustering with adaptive weightings (MECMdd).

Chapter 5

Distance-Based Multi-View Evidential C-Medoids Clustering

Abstract – In this chapter, we address the challenges of evidential clustering of trajectory data using the row-data distance-based approach. Indeed, in real-world clustering applications, proximity data (or relational data) are often the easiest to collect as they can handle objects characterized by complex and multiple types of data, by selecting the appropriate distance function. Relational clustering, which aims to identify clusters of similar objects based on their mutual relationships, is essential for dealing with such data. Nevertheless, most existing relational clustering methods cannot efficiently handle multi-view datasets while representing uncertainty and imprecision when confronted with objects in overlapping clusters. We introduce a new relational clustering method, called Multi-View Evidential C-Medoid clustering with adaptive weightings (MECMdd), to address this limitation. Based on the theory of belief functions, our approach characterizes partial knowledge in cluster assignment. The approach integrates view weight assignments, estimated locally for each cluster with a weight sum constraint (MECMdd-RWL-S) or a weight product constraint (MECMdd-RWL-P), and globally with a weight sum constraint (MECMdd-RWG-S) or a weight product constraint (MECMdd-RWG-P), within a collaborative learning framework. We evaluated our proposed algorithms through multiple experiments on real-world datasets, comparing their performance and advantages against related and state-of-the-art methods. Additionally, we applied them to a real-world case study on chronic pain care pathways, highlighting their effectiveness and interpretability.

Contents

5.1 Introduction

Clustering is an essential data analysis technique focused on uncovering underlying structures within a collection of objects. In general, these objects, characterized by matrices, sequences, time series, images, text, etc., can be clustered using one of three common representations: vector data, symbolic data or relational data. For relational data, we have a matrix of dissimilarity data, denoted as $\boldsymbol{\mathcal{R}} = [\tau_{ij}]$, where τ_{ij} is the pairwise dissimilarity (often a distance) between objects i and j. Relational clustering uses this dissimilarity matrix $\boldsymbol{\mathcal{R}}$ to partition data into clusters, uncovering underlying latent structure. Relational clustering is particularly useful when computing the distance between objects is complex, when the distance measure cannot be expressed in a simple mathematical way, or when clusters of similar objects cannot be efficiently represented by a single prototype, such as a centroid for example.

The literature presents numerous techniques for relational clustering. Among the efficient methods, partition-based approaches such as the k-means algorithm and its relational equivalent, Partitioning Around Medoids (PAM) [41], are extensively utilized. These approaches are favored for both computational simplicity and ability to adapt to large data sets. K-means and its relational extensions, identifies clusters using a centroid, respectively a medoid, and generates a hard partition, in which each object is distinctly assigned to a single cluster. However, in many real-world applications, objects from various groups are not well-separated, making hard clustering less accurate. To address this limitation and better capture clusters ambiguity, soft clustering methods have been developed, which enable the expression of uncertainty and imprecision in the resulting partition.

Representing the core paradigm of soft clustering, Fuzzy c-means (FCM) [69] represents uncertainty by allowing an object to belong to different groups with different degrees of fuzziness. Different relational variants of the FCM algorithm have been proposed [220, 221, 222], with fuzzy c-medoid (FCMdd) [64] being the most popular. Although fuzzy clustering approaches have shown promising results, they are not robust to noise and outliers, and they do not explicitly describe the imprecision of cluster assignment results. In fact, both uncertainty and imprecision in clustering often need to be expressed. For example, as Figure 5.1 shows, objects located between different clusters are indistinguishable and can be considered imprecise and/or uncertain. An evidential clustering variant of FCM, named evidential c-means (ECM), has been proposed to address this issue [1]. ECM, which is based on the belief functions theory [91, 96], creates a credal partition that improves hard and fuzzy partitions [223]. This enhancement allows for more precise expression of partial knowledge regarding cluster membership of various objects. Robust and efficient relational variants of ECM, such as centroid-based RECM [63] and EVCLUS [66], have been developed. Additionally, medoid-based variants like ECMdd [62, 107] and median-based variants such as MECM [108] have been introduced.

In order to address the limitations of relational fuzzy clustering, relational evidential clustering effectively express in a rich manner the imprecision and uncertainties regarding the class membership of the objects, and allow a direct representation of outliers. Unfortunately, these clustering methods are based on the assumption of single-view data. In several real-world applications, data have multiple representations or sources, which

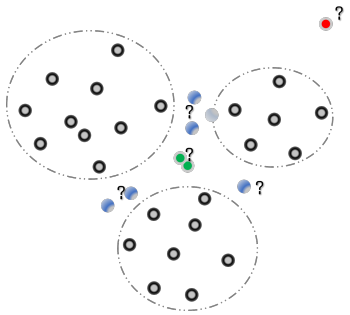


Figure 5.1: Illustration of imprecision and uncertainty in clustering.

usually have complementary information and can be used to improve the accuracy of the clustering task [224]. Integrating these views allows for more robust and accurate clustering results, instead of considering only a single data view. For multi-view data, the traditional approach involves clustering based on a global matrix, created by a uniform combination of partial similarities. However, this method can be inefficient, as different similarity matrices do not necessarily contribute equally to defining the categories of similar objects [60]. It then becomes necessary to learn cluster-dependent relevance weights for each similarity matrix in order to achieve meaningful clustering across all similarity matrices. In the field of belief-based clustering, this issue has recently been explored for vector data by simultaneously performing clustering and feature discrimination [225, 226]. However, this approach has not yet been investigated for relational data.

In this contribution, we propose MECMdd, a new medoid-based multi-view relational evidential clustering framework. Initially, we developed MECMdd-RWL-S and MECMdd-RWG-S, which are algorithms incorporating relevance weights for each relational view, i.e. each dissimilarity matrix, estimated locally and globally, respectively. These algorithms leverage a sum-based matching function for the dissimilarity matrix weights, constraining that the sum of the weights equals 1. Despite MECMdd-RWL-S and MECMdd-RWG-S being able to characterize uncertainty and imprecision in cluster assignment, they are sensitive to small variations in the weights of the dissimilarity matrices and, thus, to the variability of clustering data. Indeed, the sum constraint tends to favor a relatively balanced distribution of weights, which may limit the algorithm's ability to optimally adjust the weights for certain complex data. To address these limitations, we extended MECMdd by introducing a product-based matching function for the weights, leading to the development of MECMdd-RWL-P and MECMdd-RWG-P. These extensions provide a more flexible and adaptive framework for weight adjustment in multi-view clustering.

The rest of the chapter is organized as follows. Sections 5.2 and 5.3 present, respectively, previous work on relational multi-view clustering and the general context of evidential clustering. Sections 5.4 and 5.5 include a detailed presentation of the proposed MECMdd clustering algorithm, along with a discussion of the complexity of the proposed algorithms compared to other multi-view clustering methods. The performance and effectiveness of the new algorithms are analyzed using synthetic and real datasets in Section 5.6, which also includes an analysis of parameter sensitivity, and an application to eDOL dataset. Finally, Section 5.7 exposes a conclusion and perspectives about the work.

5.2 Related works on relational multi-view clustering

Big data is more and more represented through multiple views or modalities. To fully explore and exploit multi-view data, a number of multi-view clustering methods have been developed [227]. Concerning relational clustering of multi-view data, several approaches have been explored in the literature.
In [125], Francisco de A.T. and al. propose partitioning hard clustering algorithms based on multiple dissimilarity matrices (MRDCA). MRDCA is an extension of the dynamic hard clustering algorithm for relational data [228]. These algorithms cluster objects by simultaneously considering their relational descriptions from p multiple dissimilarity matrices and determine relevance weights for each matrix. The MRDCA has two variants, MRDCA-RWL and MRDCA-RWG, which can compute relevance weights for each dissimilarity matrix, respectively, locally for each cluster, or globally for the whole partition, such that each matrix has different influence on the clustering process. However, these algorithms consider that the relationship between objects and clusters is hard, i.e. an object cannot belong to more than one cluster at the same time. In real data, imperfect information is omnipresent. These input data may incorporate uncertainty due to the lack of information, conflicting evidence, ambiguity, measurement errors. In addition to uncertainty, imprecision, vagueness, and inconsistency can be associated with data [229]. This, and the inherent variability of the data linked to aleatoric uncertainty makes hard partition too rigid. Consequently, soft relational clustering approaches based on fuzzy clustering have been proposed.

In [60], Frigui and al proposed the Clustering and Aggregating Relational Data (CARD) algorithm that uses multiple dissimilarity matrices to represent the relational descriptions. Frigui and others propose the extension $CARD_R$ [57] based on the relational fuzzy c-means (RFCM), and Non-Euclidean Relational Fuzzy c-means (NERF) algorithms [59], as well as the $CARD_F$ [41], a variant based on the Fuzzy Clustering (FANNY) algorithm, to group relational data represented in the form of multiple dissimilarity matrices. CARD aggregates the matrices and learns a relevance weight for each matrix of each cluster. Francisco de A.T. et al. [61] introduced multiple relational fuzzy c-medoids (MRFCMdd), an extension of the FCMdd algorithm [64]. This algorithm uses several dissimilarity matrices to partition objects, with the aim of obtaining a consensus partition. MRFCMdd provides a partition and a prototype for each fuzzy cluster, and learns a relevance weight for each dissimilarity matrix. MRFCMdd is also designed to estimate weights in two ways: locally, i.e. a relevance weight for each matrix in each cluster (MRFCMdd-RWL), and globally, i.e. a global clustering weight for each matrix (MRDCA-RWG). Fuzzy clustering [230], by generating a fuzzy partition, offers a way to handle uncertainty by allowing objects to belong to multiple clusters to varying degrees. While fuzzy clustering has shown some success, it does not fully capture partial information. Yet, in real-world applications, uncertainty and imprecision are common in cluster structures, making it challenging to distinguish cluster information for certain objects. To address this challenge, new approaches have been proposed in the framework of possibilistic theory [84], rough set [231], and belief function theory (or evidence theory) [1]. The belief function theory is particularly well known for its rich modeling of partial information. The evidential c-means (ECM) clustering [1], which is the evidential counterpart of fuzzy clustering, is effective in handling uncertain and imprecise cluster structures and producing meaningful clustering results.

Few multi-view clustering derived from ECM have been proposed [225, 226]. Liu and al. [225] propose a new adaptive weighted multi-view evidential clustering (WMVEC) method. They introduce view weights to capture the different contributions of each view in clustering and employ entropy regularization to regulate the distribution of view weights. In [226], they further develop WMVEC method and present a weighted multi-view evidential clustering with feature preference (WMVEC-FP) to eliminate the interference of redundant features, thereby improving the clustering performance. Unfortunately, WMVEC and WMVEC-FP are built upon the assumption of access to multi-view vector data. In this contributions, we propose a new adaptive weighted multi-view evidential relational medoid-based clustering (MECMdd) method. In the next Section 5.3, we introduce the evidential clustering framework.

5.3 Background

The background necessary for the development of the MECMdd method is introduced in Section 2.3.5.2 of chapter 2. Indeed, relational evidential clustering methods, such as ECMdd [107], extend the principles of evidential c-means (ECM) to relational data. Unlike object-based methods, ECMdd operates on pairwise dissimilarity matrices, allowing it to process more complex data structures. As discussed in Section 2.3.5.5, ECMdd optimizes a credal partition by minimizing its objective function, as defined in Equation 2.52, which balances penalties for imprecision and the handling of outliers through the belief masses assigned to the empty set. The iterative optimization process is described in algorithm 3, which alternates between updating the credal partition and the medoids prototypes.

While the theoretical foundation of ECMdd is detailed in the chapter 2, additional examples of credal partitions further illustrate its functionality. Table 5.1 provides an example of a credal partition for four objects classified into two clusters. This example highlights the algorithm's ability to represent diverse scenarios: certain assignments (e.g., $m_1(\{\omega_1\}) = 1$), Bayesian distributions (e.g., m_2), imprecise beliefs (e.g., m_3), and outliers (e.g., $m_4(\emptyset) = 1$).

Table 5.1: Example of a credal partition.

A	$m_1(A)$	$m_2(A)$	$m_3(A)$	$m_4(A)$
$\emptyset$	0	0	0	1
$\{\omega_1\}$	1	0.9	0	0
$\{\omega_2\}$	0	0.1	0	0
Ω	0	0	1	0

Let $v_k^{\Omega} \in E$ such that $A_j = \{\omega_k\}$ ($|A_j| = 1$) represent the medoid of a specific cluster, and $v_j^{2^{\Omega}} \in E$ such that $A_j \subseteq \{\omega_1, \ldots, \omega_c\}$, $|A_j| > 1$, represent an imprecise cluster. When $|A_j| = 1$, the dissimilarity τ_{ij} is computed as $\tau(e_i, v_k^{\Omega})$. Otherwise, for $|A_j| > 1$, the dissimilarity is defined as shown in Equation 2.56.

5.4 MECMdd with constraint based on sum for weights learning

5.4.1 MECMdd with relevance weight for each dissimilarity matrix estimated locally

This algorithm is designed to provide a credal partition and a prototype for each credal cluster, while also learning a relevance weight for each dissimilarity matrix with constraint based on sum for weights learning. These weights are estimated locally, changing at each iteration of the algorithm, and vary from one cluster to another.

5.4.1.1 Formulation

Let $E = \{e_1, \ldots, e_n\}$ be the set of n objects and p dissimilarity matrices defined by $\boldsymbol{\mathcal{R}}_l = (\tau_l(e_i, e_j))$ for $l = 1, \ldots, p$. The relational evidential c-medoids clustering algorithm with relevance weights for each dissimilarity matrix estimated locally (referred to as MECMdd-RWL-S) aims to find the partition $\mathbf{M}$, the corresponding prototypes $\mathbf{V}$, and compute the vector of relevance weights (one for each cluster) $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k, \ldots, \boldsymbol{\lambda}_c)$, such that $\boldsymbol{\lambda}_k$ is a vector of size p. The objective function of MECMdd-RWL-S is:

$$J_{MECMdd-RWL-S}(\boldsymbol{M}, \boldsymbol{V}, \boldsymbol{\Lambda}) = \sum_{i=1}^{n} \sum_{A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^{\beta} \sum_{l=1}^{p} (\lambda_{jl})^{s} \tau_{ijl} + \sum_{l=1}^{p} \delta_l^2 \sum_{i=1}^{n} m_{i\emptyset}^{\beta}. \quad (5.1)$$

Where $\sum_{l=1}^{p} (\lambda_{jl})^{s} \tau_{ijl}$ represents the global relation between an object $e_i \in E$ and the subset prototype $v_j^{2^\Omega} \in E$, parameterized by $s > 1$. The weight λ_{jl} is the view weight of l^{th} view in the subset A_j, $\tau_{ijl} = \tau_l(e_i, A_j) = \tau_l(e_i, v_j^{2^\Omega})$ is the local dissimilarity between $e_i \in A_j$ and the subset prototype $v_j^{2^\Omega} \in E$ in dissimilarity matrix $\boldsymbol{\mathcal{R}}_l$, and is obtained using Equation (2.56). The parameter α, and β are the same as in ECMdd. The parameter δ works as a threshold, assumed at a fixed distance from any individuals, at which objects can be considered outliers, and controls the rate of individuals considered outliers. Motivated by the noise clustering (NC) approach, presented as robust to atypical individuals [100], and by the adaptive weighted multi-view evidential clustering proposal [225], the threshold δ based on several dissimilarity matrices is calculated as follows $\sum_{l=1}^{p} \delta_l^2 \sum_{i=1}^{n} m_{i\emptyset}^{\beta}$. Indeed, as Denœux et al. [1] underline, the parameter δ, assumed to be at a fixed distance from all individuals, allows robustness to atypical individuals, by controlling the proportion of individuals considered as outliers. We formulate the hypothesis that this distance is variable for each dissimilarity matrix.
$J_{MRFCMdd-RWL-S}$ is optimized under the following constraints:

$$\sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij} + m_{i\emptyset} = 1 \qquad \forall i = \{1, \ldots, n\}, \quad (5.2)$$

$$m_{ij} \geq 0 \qquad \forall i = \{1, \ldots, n\}, \quad \forall A_j \subseteq \Omega, \quad (5.3)$$

$$\sum_{l=1}^{p} \lambda_{jl} = 1 \qquad \forall A_j \subseteq \Omega, \quad (5.4)$$

$$\lambda_{jl} \geq 0 \qquad \forall l = \{1, \ldots, p\}, \quad \forall A_j \subseteq \Omega. \quad (5.5)$$

5.4.1.2 Optimization

Similarly to ECM and its variants, we provide a Gauss-Seidel method to minimize (5.1), constrained with (5.2)-(5.3) and (5.4)-(5.5). More specifically, we employ the Lagrange optimization method to alternately update one variable while leaving others fixed. The details of each update step are provided in Theorems 5.4.1, 5.4.2, and 5.4.3 for the variables $\mathbf{M}$, $\mathbf{\Lambda}$, and $\mathbf{V}$, respectively.

Theorem 5.4.1 (Updating $\mathbf{M}$, with $\mathbf{V}$ and $\mathbf{\Lambda}$ fixed). *The credal partition, denoted* $\mathbf{M}$*, which minimizes the clustering objective function* $\mathcal{J}$*-MECMdd-RWL-S w.r.t constraints (5.2) and (5.3) is computed using the following expression:*

$$m_{ij} = \frac{\left(|A_j|^{\alpha} \sum_{l=1}^{p} (\lambda_{jl})^s \tau_{ijl}\right)^{-\frac{1}{\beta-1}}}{\sum_{A_k \neq \emptyset} \left(|A_k|^{\alpha} \sum_{l=1}^{p} (\lambda_{kl})^s \tau_{ikl}\right)^{-\frac{1}{\beta-1}} + \left(\sum_{l=1}^{p} \delta_l^2\right)^{-\frac{1}{\beta-1}}}, \tag{5.6}$$

$$m_{i\emptyset} = 1 - \sum_{A_j \neq \emptyset} m_{ij}. \tag{5.7}$$

Theorem 5.4.2 (Updating $\mathbf{\Lambda}$, with $\mathbf{M}$ and $\mathbf{V}$ fixed). *The vectors of relevance weights* $\boldsymbol{\lambda}_k = (\lambda_{k1}, \ldots, \lambda_{kp})$*, which minimizes the clustering objective function* $\mathcal{J}$*-MECMdd-RWL-S, with* $s > 1$*, and constraints (5.4) and (5.5), is obtained according to:*

$$\lambda_{jl} = \frac{\left(\sum_{i=1}^{n} s|A_j|^{\alpha} m_{ij}^{\beta} \tau_{ijl}\right)^{-\frac{1}{s-1}}}{\sum_{h=1}^{p} \left(\sum_{i=1}^{n} s|A_j|^{\alpha} m_{ij}^{\beta} \tau_{ijh}\right)^{-\frac{1}{s-1}}}. \tag{5.8}$$

Theorem 5.4.3 (Updating $\mathbf{V}$ while fixing $\mathbf{M}$ and $\mathbf{\Lambda}$). *The prototype of the singleton cluster* $e_{h^\star} = v_k^{\Omega}$ *of subset* $A_j = \{\omega_k\}$ *with* $e_{h^\star} \in E$*, which minimizes the clustering objective function* $\mathcal{J}$*-MECMdd-RWL-S, is first computed according to (5.9):*

$$h^\star = \underset{1 \leq h \leq n}{\operatorname{argmin}} \sum_{i=1}^{n} m_{ij}^{\beta} \sum_{l=1}^{p} (\lambda_{jl})^s \tau_l(e_i, e_h). \tag{5.9}$$

Next, the prototype of the imprecise cluster $e_{h^\diamond} = v_j^{2^\Omega}$ *of subset* $A_j \subseteq \{\omega_1, \ldots, \omega_k\}$ *with* $e_{h^\star} \in E$ *is determined according to equations (5.10) and (5.11) using (5.9):*

$$h^\diamond = \underset{1 \leq h \leq n}{\operatorname{argmin}}\ \sigma_{ij}^2 + \eta \frac{1}{|A_j|} \sum_{\omega_k \in A_j} \sum_{l=1}^{p} \tau_l(e_i, v_k^{\Omega}), \tag{5.10}$$

and

$$\sigma_{ij}^2 = \frac{1}{|A_j|} \sum_{\omega_k \in A_j} \left[\sum_{l=1}^{p} \tau_l(e_i, v_k^{\Omega}) - \frac{1}{|A_j|} \sum_{\omega_k \in A_j} \sum_{l=1}^{p} \tau_l(e_i, v_k^{\Omega})\right]^2. \tag{5.11}$$

Proof. The prototypes minimizing the clustering objective function J-MECMdd-RWL-S is computed in two steps, depending on whether the subset A_j represents a singleton cluster or an imprecise cluster [107, 108, 232].

For singleton clusters $A_j = \{\omega_k\}$ (i.e., $|A_j| = 1$), the prototype $e_{h^\star} = v_k^\Omega$ is computed using Equation (5.9). The proof of this proposition is simple. This is justified by the principle that the medoid of a singleton cluster should minimize the weighted sum of dissimilarities between the objects in the dataset and the candidate prototypes. The term $\sum_{i=1}^{n} m_{ij}^{\beta} \sum_{l=1}^{p} (\lambda_{jl})^s \tau_l(e_i, e_h)$ reflects both the membership degree m_{ij} and the importance of feature subsets λ_{jl}. The minimization ensures that the chosen prototype $e_{h^\star}$ best represents the cluster under the given weighting scheme.

For imprecise clusters $A_j \subseteq \{\omega_1, \ldots, \omega_k\}$ (i.e., when $|A_j| > 1$), the prototype $e_{h^\diamond} = v_k^{2^\Omega}$ is determined through Equations (5.10) and (5.11). This proposition is motivated by two conditions: (1) the prototypes of the singleton clusters $\omega_k \in A_j$ should have small dissimilarities to the objects e_i in A_j, measuring the uncertainty within the imprecise cluster, and (2) the prototype should be close to all involved singleton prototypes $\omega_k \in A_j$ included in A_j to ensure representativeness. Based on these conditions, the variance of th dissimilarities of object e_i to the medoids of all the included specific classes of A_j could be taken into account to express the degree of uncertainty. The smaller the variance is, the higher uncertainty we have for object e_i. Meanwhile, this is to distinguish the outliers, which may have equal dissimilarities to the prototypes of some specific classes, but obviously not a good choice for representing the associated imprecise classes. The above assumptions allow us to define a medoid of $|A_j| > 1$ as follows:

$$h^\diamond = \underset{i:e_i \in E}{\operatorname{argmin}} \left\{ f\big(\{\tau(e_i, v_k^\Omega); \omega_k \in A_j\}\big) + \eta \frac{1}{|A_j|} \sum_{\omega_k \in A_j} \tau(e_i, v_k^\Omega) \right\}, \tag{5.12}$$

where ω_k is an element of A_j, v_k^Ω is its corresponding prototype, and f denotes the variance function among the dissimilarity values. In this contribution, we use the variance function σ_{ij}^2, given by Equation (5.11) to describe the variance of the dissimilarities between object e_i and the medoids of the involved specific classes in A_j. But, according to [107, 108], others variance function f could be use.

5.4.2 MECMdd with relevance weight of each dissimilarity matrix estimated globally

The variant of MECMdd-RWL-S with global relevance weight, called MECMdd-RWG-S, has also been proposed to provide a credal partition and prototype for each cluster, while learning relevance weights for all clusters in each view.

5.4.2.1 Formulation

MECMdd-RWG-S aims to find a partition $\mathbf{M}$, the corresponding prototypes $\mathbf{V}$, and a single relevance weight vector $\boldsymbol{\lambda}$ of size p. The objective function of MECMdd-RWG-S can be represented as follows:

$$J_{MECMdd-RWG-S}(\mathbf{M},\mathbf{V},\boldsymbol{\lambda}) = \sum_{i=1}^{n}\sum_{A_j\neq\emptyset}|A_j|^{\alpha}m_{ij}^{\beta}\sum_{l=1}^{p}(\lambda_l)^s\,\tau_{ijl} + \sum_{l=1}^{p}\delta_l^2\sum_{i=1}^{n}m_{i\emptyset}^{\beta}. \quad (5.13)$$

Where $\sum_{l=1}^{p}(\lambda_l)^s\,\tau_{ijl}$ represents the global correspondence between an object $e_i \in E$ and the cluster prototype $v_k^{\Omega} \in E$, parameterized by $s > 1$ and by the weight vector $\boldsymbol{\lambda} = (\lambda_1,\ldots,\lambda_p)$. $\tau_{ijl} = \tau_l(e_i, A_j) = \tau_l(e_i, v_j^{2^{\Omega}})$ is the local dissimilarity between $e_i \in E$ and the subset prototype $v_j^{2^{\Omega}} \in E$ in dissimilarity matrix $\boldsymbol{\mathcal{R}}_l (l = 1,\ldots,p)$, and is obtained from (2.56).

$J_{MRFCMdd-RWG-S}$ is optimized with the constraints (5.2)-(5.3) and (5.14)-(5.15):

$$\sum_{l=1}^{p}\lambda_l = 1, \quad (5.14)$$

$$\lambda_l \geq 0, \qquad \forall l = \{1,\ldots,p\}. \quad (5.15)$$

5.4.2.2 Optimization

We use the same efficient iterative approach in Section 5.4.1, to minimize (5.13). These are the Lagrange optimization technique to update variables one after the other, alternately. The details of each update step are provided in Theorems 5.4.4, 5.4.5, and 5.4.6 for the variables **M**, $\boldsymbol{\lambda}$, and **V**, respectively.

Theorem 5.4.4 (Updating **M** while fixing **V** and $\boldsymbol{\lambda}$). *The calculation of the credal partition* **M**, *which minimizes the clustering objective function J-MECMdd-RWG-S w.r.t. constraints (5.2) and (5.3) is derived from the following expression:*

$$m_{ij} = \frac{\left(|A_j|^{\alpha}\sum_{l=1}^{p}(\lambda_l)^s\tau_{ijl}\right)^{-\frac{1}{\beta-1}}}{\sum_{A_k\neq\emptyset}\left(|A_k|^{\alpha}\sum_{l=1}^{p}(\lambda_l)^s\tau_{ikl}\right)^{-\frac{1}{\beta-1}} + \left(\sum_{l=1}^{p}\delta_l^2\right)^{-\frac{1}{\beta-1}}}, \quad (5.16)$$

$$m_{i\emptyset} = 1 - \sum_{A_j\neq\emptyset} m_{ij}. \quad (5.17)$$

Theorem 5.4.5 (Updating $\boldsymbol{\lambda}$, while fixing **M** and **V**). *Considering the constraint function based on the sum of the relevance weights (5.14) and positivity (5.15) in the optimization scheme of (5.13), the updating term of the weight vector* $\boldsymbol{\lambda} = (\lambda_1,\ldots,\lambda_p)$ *is obtained by:*

$$\lambda_l = \frac{\left(\sum_{i=1}^{n}\sum_{A_j\neq\emptyset} s\,|A_j|^{\alpha}m_{ij}^{\beta}\tau_{ijl}\right)^{-\frac{1}{s-1}}}{\sum_{h=1}^{p}\left(\sum_{i=1}^{n}\sum_{A_j\neq\emptyset} s\,|A_j|^{\alpha}m_{ij}^{\beta}\tau_{ijh}\right)^{-\frac{1}{s-1}}}. \quad (5.18)$$

Theorem 5.4.6 (Updating **V** while fixing **M** and $\boldsymbol{\lambda}$). *With* **M** *and* $\boldsymbol{\lambda}$ *fixed, the prototype of the singleton cluster* $e_{h^\star} = v_k^{\Omega}$ *of* $A_j = \{\omega_k\}$ *with* $e_{h^\star} \in E$*, which minimizes* $\mathcal{J}$*-MECMdd-RWG-S, is first computed according to (5.19). Following this, the prototype of the imprecise cluster* $e_{h^\diamond}$*, denoted as* $v_k^{2^\Omega}$*, associated with* $A_j \subseteq \{\omega_1, \ldots, \omega_k\}$*, where* $e_{h^\diamond} \in E$*, is determined based on equations (5.10) and (5.11):*

$$h^\star = \underset{1 \leq h \leq n}{\operatorname{argmin}} \sum_{i=1}^{n} m_{ij}^{\beta} \sum_{l=1}^{p} (\lambda_l)^s \tau_l(e_i, e_h). \tag{5.19}$$

Proof. The detailed proofs process can be found in proof 5.4.1.2.

5.4.3 Computation algorithms and complexities

The algorithms MECMdd-RWL-S and MECMdd-RWG-S converges after a fixed number of iterations. Updating the credal partition **M** and the local relevance weight vector $\boldsymbol{\Lambda}$ using Lagrangian multipliers does not increase the complexity, neither does the medoid-searching scheme for prototypes of singleton clusters. We provide a summary of the proposed in Algorithm 7 for MECMdd-RWL-S and in Algorithm 8 for MECMdd-RWG-S.

Algorithm 7 MECMdd-RWL-S

Input: p dissimilarity matrices.
Parameters: number of clusters c, and hyperparameters α, $\beta > 1$, $\delta_l > 0$, $\eta > 0$,$\gamma \in [0, 1]$
Initialization: $\mathbf{V}^0$ and $\boldsymbol{\Lambda}^0$
while no convergence of the solution **do**
  (1) Compute the prototypes of meta-clusters using (5.10) and (5.11);
  (2) Compute the credal partition matrix **M** using (5.6) and (5.7);
  (3) Compute the best relevance weight vector $\boldsymbol{\Lambda}$ according to (5.8);
  (4) Compute the new prototype set **V** using (5.9)
end
Output: The optimal **M**, **V** and $\boldsymbol{\Lambda}$

Algorithm 8 MECMdd-RWG-S

Input: p dissimilarity matrices.
Parameters: refer to parameters of Algorithm 7
Initialization: $\mathbf{V}^0$ and λ^0
while no convergence of the solution **do**
  (1) Compute the prototypes of meta-clusters using (5.10) and (5.11);
  (2) Compute the credal partition matrix **M** using (5.16) and (5.17);
  (3) Compute the best relevance weight vector $\boldsymbol{\lambda}$ according to (5.18);
  (4) Compute the new prototype set **V** using (5.19)
end
Output: The optimal **M**, **V** and $\boldsymbol{\lambda}$

The algorithmic time complexity of ECMdd shows that one iteration requires $O(n2^c + cn^2)$ [107], where n is the dataset size and c is the number of clusters. The update process for mass membership **M** in MECMdd-RWL-S is similar to ECMdd with a single view. For a

$n \times n$ dissimilarity matrix, ECMdd has a complexity of $O(n2^c)$, while MECMdd-RWL-S, with p views, has a complexity of $O(pn2^c)$. Similarly, updating prototypes and calculating dissimilarities has a complexity of $O(cn^2)$ for ECMdd, and $O(pcn^2)$ for MECMdd-RWL-S. Computing the best relevance weight vectors involves calculating 2^c denominators and the numerator once, repeated for the p relevance weight vectors, resulting in a complexity of $O(p2^c)$. Thus, the total time complexity of MECMdd-RWL-S and MECMdd-RWG-S is $O(t(pn2^c + pcn^2))$, for t iterations.

5.5 MECMdd with constraint based on product for weights learning

5.5.1 MECMdd with relevance weight for each dissimilarity matrix estimated locally

This algorithm is designed to provide a credal partition and a prototype for each credal cluster, while also learning a relevance weight for each dissimilarity matrix with constraint based on product for weights learning. These weights are estimated locally, changing at each iteration of the algorithm, and vary from one cluster to another.

5.5.1.1 Formulation

Let $E = \{e_1, \ldots, e_n\}$ be the set of n objects and $\boldsymbol{\mathcal{R}}_l = (\tau_l(e_i, e_j)), \quad \forall l \in \{1, \ldots, p\}$ be p dissimilarity matrices. The relational evidential c-medoids clustering algorithm with relevance weights for each dissimilarity matrix estimated locally based on a weight product constraint for optimization (referred to as MECMdd-RWL-P) aims to find the partition $\mathbf{M}$, the corresponding prototypes $\mathbf{V}$, and compute the vector of relevance weights (one for each cluster) $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k)$, such that $\boldsymbol{\lambda}_k$ is a vector of size p. It aims to locally optimize an adequacy criterion, whose objective function, that measures the fit between the clusters and their prototypes. The objective function of MECMdd-RWL is similar to MECMdd-RWL-S, and is represented as follows:

$$J_{MECMdd-RWL-P}(\mathbf{M}, \mathbf{V}, \boldsymbol{\Lambda}) = \sum_{i=1}^{n} \sum_{A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^{\beta} \sum_{l=1}^{p} (\lambda_{jl})^s \tau_{ijl} + \sum_{l=1}^{p} \delta_l^2 \sum_{i=1}^{n} m_{i\emptyset}^{\beta}. \quad (5.20)$$

The difference with MECMdd-RWL-S lies in the hyperparameter s fixed to 1 and the constraints used to optimize the objective function $J_{MRFCMdd-RWL-P}$ are:

$$\sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij} + m_{i\emptyset} = 1 \qquad \forall i = \{1, \ldots, n\}, \quad (5.21)$$

$$m_{ij} \geq 0 \qquad \forall i = \{1, \ldots, n\}, \quad \forall A_j \subseteq \Omega, \quad (5.22)$$

$$\prod_{l=1}^{p} \lambda_{jl} = 1 \qquad \forall A_j \subseteq \Omega, \quad (5.23)$$

$$\lambda_{jl} > 0 \qquad \forall l = \{1, \ldots, p\}, \quad \forall A_j \subseteq \Omega. \quad (5.24)$$

5.5.1.2 Optimization

Similar to MECMdd-RWL-S, this algorithm begins with an initial partition and alternates between three steps until convergence. We use the same efficient iterative approach, in Section 5.4.1, to minimize (5.20), subject to the constraints (5.21)-(5.22) and (5.23)-(5.24). Specifically, we apply the Lagrange optimization method to alternately update one variable at a time while keeping the others fixed. The details of each update step are provided in Theorems 5.5.1, 5.5.2, and 5.5.3 for the variables $\mathbf{M}$, $\mathbf{\Lambda}$, and $\mathbf{V}$, respectively.

Theorem 5.5.1 (Updating $\mathbf{M}$, with $\mathbf{V}$ and $\mathbf{\Lambda}$ fixed). *The credal partition $\mathbf{M}$, which minimizes the clustering objective function J-MECMdd-RWL-P w.r.t constraints (5.21) and (5.22) is computed using the following expression:*

$$m_{ij} = \frac{\left(|A_j|^{\alpha} \sum_{l=1}^{p} (\lambda_{jl})^s \tau_{ijl}\right)^{-\frac{1}{\beta-1}}}{\sum_{A_k \neq \emptyset} \left(|A_k|^{\alpha} \sum_{l=1}^{p} (\lambda_{kl})^s \tau_{ikl}\right)^{-\frac{1}{\beta-1}} + \left(\sum_{l=1}^{p} \delta_l^2\right)^{-\frac{1}{\beta-1}}}, \tag{5.25}$$

$$m_{i\emptyset} = 1 - \sum_{A_j \neq \emptyset} m_{ij}. \tag{5.26}$$

Proof. To prove the Theorem 5.5.1, we employ Lagrange multipliers, denoted as $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_n\}$, to tackle the constrained minimization problem of the J-MECMdd-RWL-P objective function concerning $\mathbf{M}$. The Lagrangian function associated to (5.20), (5.21), and (5.22) is:

$$\mathcal{L}(\boldsymbol{M}, \boldsymbol{\xi}) = J_{MECMdd-RWL-P}(\boldsymbol{M}, \boldsymbol{V}, \boldsymbol{\Lambda}) - \sum_{i=1}^{n} \xi_i \left(\sum_{\substack{A_j \subseteq \Omega, \\ A_j \neq \emptyset}} m_{ij} + m_{i\emptyset} - 1 \right). \tag{5.27}$$

By differentiating the Lagrangian with respect to the m_{ij}, $m_{i\emptyset}$ and ξ_i, we obtain the derivatives (5.28), (5.29) and (5.30).

$$\frac{\partial \mathcal{L}(\mathbf{M}, \xi_i)}{\partial m_{ij}} = \beta m_{ij}^{\beta-1} |A_j|^{\alpha} \sum_{l=1}^{p} (\lambda_{jl})^s \tau_{ijl} - \xi_i, \tag{5.28}$$

$$\frac{\partial \mathcal{L}(\mathbf{M}, \xi_i)}{\partial m_{i\emptyset}} = \beta m_{i\emptyset}^{\beta-1} \sum_{l=1}^{p} \delta_l^2 - \xi_i, \tag{5.29}$$

$$\frac{\partial \mathcal{L}(\mathbf{M}, \xi_i)}{\partial \xi_i} = \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij} + m_{i\emptyset} - 1. \tag{5.30}$$

Annulling the derivatives (5.28) and (5.29) gives:

$$m_{ij} = \left(\frac{\xi_i}{\beta}\right)^{\frac{1}{\beta-1}} \left(\frac{1}{|A_j|^{\alpha} \sum_{l=1}^{p} (\lambda_{jl})^s \tau_{ijl}}\right)^{\frac{1}{\beta-1}}, \tag{5.31}$$

$$m_{i\emptyset} = \left(\frac{\xi_i}{\beta}\right)^{\frac{1}{\beta-1}} \left(\frac{1}{\sum_{l=1}^{p} \delta_l^2}\right)^{\frac{1}{\beta-1}}. \tag{5.32}$$

Using (5.31) and (5.32) in the annulation of (5.30), we can obtain the updating (5.33):

$$\left(\frac{\xi_i}{\beta}\right)^{-\frac{1}{\beta-1}} = \sum_{A_k \neq \emptyset} \left(\frac{1}{|A_k|^{\alpha} \sum_{l=1}^{p} (\lambda_{kl})^{s} \tau_{ikl}}\right)^{\frac{1}{\beta-1}} + \left(\frac{1}{\sum_{l=1}^{p} \delta_l^2}\right)^{\frac{1}{\beta-1}}. \tag{5.33}$$

Introducing (5.33) in (5.31) gives the update formulas (5.25) - (5.26).

Theorem 5.5.2 (Updating $\boldsymbol{\Lambda}$, with $\mathbf{M}$ and $\mathbf{V}$ fixed)**.** *Considering the constraints (5.23) and (5.24), the vectors of relevance weights $\boldsymbol{\lambda}_k = (\lambda_{k1}, \ldots, \lambda_{kp})$, which minimizes the clustering objective function (5.20), with $s = 1$, are computed according to:*

$$\lambda_{jl} = \frac{\prod_{h=1}^{p} \left(\sum_{i=1}^{n} |A_j|^{\alpha} m_{ij}^{\beta} \tau_{ijh}\right)^{\frac{1}{p}}}{\sum_{i=1}^{n} |A_j|^{\alpha} m_{ij}^{\beta} \tau_{ijl}}. \tag{5.34}$$

Proof. In order to prove Theorem 5.5.2, we employ the Lagrange multiplier, denoted as $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_{2^c})$, to solve the constrained minimization problem with respect to $\boldsymbol{\Lambda}$. The Lagrangian function associated to (5.20), (5.23), and (5.24) is written as follows :

$$\mathcal{L}(\boldsymbol{\Lambda}, \boldsymbol{\xi}) = J_{MECMdd-RWL-P}(\mathbf{M}, \mathbf{V},) - \sum_{A_j} \xi_j \left(\prod_{l=1}^{p} \lambda_{jl} - 1\right). \tag{5.35}$$

We differentiate the Lagrangian $\mathcal{L}(\boldsymbol{\Lambda}, \boldsymbol{\xi})$ with respect to the variables $\boldsymbol{\Lambda}$ and $\boldsymbol{\xi}$:

$$\frac{\partial \mathcal{L}(\boldsymbol{\Lambda}, \xi_j)}{\partial \lambda_{jl}} = \sum_{i=1}^{n} |A_j|^{\alpha} m_{ij}^{\beta} \tau_{ijl} - \xi_j \prod_{h \in \{1,\ldots p\}, h \neq l} \lambda_{jh}, \tag{5.36}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\Lambda}, \xi_j)}{\partial \xi_j} = \prod_{l=1}^{p} \lambda_{jl} - 1. \tag{5.37}$$

Then, we can isolate the expression of ξ_j by multiplying (5.36) with λ_{jl} and by reducing the equation using (5.23) to obtain equation (5.38):

$$\lambda_{jh} = \xi_j \frac{1}{\sum_{i=1}^{n} |A_j|^{\alpha} m_{ij}^{\beta} \tau_{ijl}}. \tag{5.38}$$

Next substituting (5.38) into the annulation of the derivative (5.36) allows to obtain a straightforward formula for the Lagrange multipliers:

$$\xi_j = \prod_{h=1}^{p} \left(\sum_{i=1}^{n} |A_j|^{\alpha} m_{ij}^{\beta} \tau_{ijh}\right)^{\frac{1}{p}}. \tag{5.39}$$

Combining (5.38) and (5.39) finally results in (5.34).

Theorem 5.5.3 (Updating **V** while fixing **M** and **Λ**). *The prototype of the singleton cluster* $e_{h^\star} = v_k^\Omega \quad \forall k = \{1, \ldots, c\}$ *of subset* $A_j = \{\omega_k\}$ *with* $e_{h^\star} \in E$*, which minimizes the clustering objective function* $\mathcal{J}$*-MECMdd-RWL-P, is computed according to (5.40):*

$$h^\star = \underset{1 \leq h \leq n}{argmin} \sum_{i=1}^{n} m_{ij}^{\beta} \sum_{l=1}^{p} (\lambda_{jl})^s \tau_l(e_i, e_h) \tag{5.40}$$

The prototype of the imprecise cluster $e_{h^\diamond} = v_j^{2^\Omega}$ *of subset* $A_j \subseteq \{\omega_1, \ldots, \omega_k\}$ *with* $e_{h^\diamond} \in E$ *is determined according to equations (5.41) and (5.42):*

$$h^\diamond = \underset{1 \leq h \leq n}{argmin}\ \sigma_{ij}^2 + \eta \frac{1}{|A_j|} \sum_{\omega_k \in A_j} \sum_{l=1}^{p} \tau_l(e_i, v_k^\Omega), \tag{5.41}$$

$$\sigma_{ij}^2 = \frac{1}{|A_j|} \sum_{\omega_k \in A_j} \left[\sum_{l=1}^{p} \tau_l(e_i, v_k^\Omega) - \frac{1}{|A_j|} \sum_{\omega_k \in A_j} \sum_{l=1}^{p} \tau_l(e_i, v_k^\Omega) \right]^2 \tag{5.42}$$

The parameter η, which is typically set to 1, helps to identify outliers from potential medoids when determining prototypes of meta-clusters and has minimal effect on the final partitioning results.

Proof. The proof of Theorem 5.5.3 is straightforward according to the proving details of the proof 5.4.1.2.

5.5.1.3 Computational algorithm and complexity

For ease of implementation, we provide a summary of the proposed method in Algorithm 9.

Algorithm 9 MECMdd-RWL-P

Input: p dissimilarity matrices.
Parameters: number of clusters c, and hyperparameters $\alpha, \beta > 1$, $\delta_l > 0$, $\eta > 0$,$\gamma \in [0, 1]$
Initialization: $\mathbf{V}^0$ and $\mathbf{\Lambda}^0$
while no convergence of the solution **do**
  (1) Compute the prototypes of meta-clusters utilizing (5.41) and (5.42);
  (2) Compute the credal partition matrix **M** using (5.25) and (5.26);
  (3) Compute the best relevance weight vector **Λ** according to (5.34);
  (4) Compute the new prototype set **V** using (5.40);
end
Output: The optimal **M**, **V** and **Λ**

In [107], the algorithmic complexity of ECMdd was analyzed. It was shown that one iteration of the optimization procedure necessitates $O(n2^c + cn^2)$ operations where n denotes the dataset size and c is the number of clusters. The update process of mass membership **M** of our proposition algorithm, MECMdd-RWL-P is similar to ECMdd with single view. For a given $n \times n$ dissimilarity matrix, in this step ECMdd have $O(n2^c)$ complexity and MECMdd-RWL-P introduces an additional sum on p views, then its complexity at this step is $O(pn2^c)$. With the same approach the complexity for updating the prototypes and calculating the

dissimilarity between objects and classes is $O(cn^2)$ for ECMdd, thus $O(pcn^2)$ for MECMdd-RWL-P. The step of computation of the best relevance weight vectors needs the computation of 2^c denominators, the computation of the numerator just once, and to repeat that for the p vector of relevance weights, so complexity of $O(p2^c)$. Therefore, the total time complexity is $O(pn2^c + pcn^2 + p2^c)$. In summary, if the clustering process needs t iterations to converge, the total time complexity of MECMdd-RWL-P is $O(t(pn2^c + pcn^2))$.

5.5.2 MECMdd with relevance weight of each dissimilarity matrix estimated globally

The MECMdd-RWL-P algorithm can suffer numerical instabilities when it produces single clusters or clusters containing objects with zero dissimilarity between them, i.e. $\sum_{i=1}^{n} m_{ij}^{\beta}\tau_{ijl} \rightarrow 0$. Indeed, this term is at the denominator of the weight update terms, Equation (5.34), and if it becomes too small, the value of the fraction tends to infinity. We propose a variant, with global relevance weight, called MECMdd-RWG-P, to provide a credal partition and prototype for each cluster, while learning relevance weights for all clusters in each view via a weight product constraint.

5.5.2.1 Formulation

Let $E = \{e_1, \ldots, e_n\}$ be the set of n objects and $\boldsymbol{\mathcal{R}}_l = (\boldsymbol{\tau}_l(e_i, e_j)), \quad \forall l = \{1, \ldots, p\}$ be p dissimilarity matrices. MECMdd-RWG-P aims to find a partition $\mathbf{M}$, the corresponding prototypes $\mathbf{V}$, and a single relevance weight vector $\boldsymbol{\lambda}$ of size p. The objective function of MECMdd-RWG-P is represented as follows:

$$J_{MECMdd-RWG-P}(\mathbf{M}, \mathbf{V}, \boldsymbol{\lambda}) = \sum_{i=1}^{n} \sum_{A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^{\beta} \sum_{l=1}^{p} (\lambda_l)^s \tau_{ijl} + \sum_{l=1}^{p} \delta_l^2 \sum_{i=1}^{n} m_{i\emptyset}^{\beta}, \quad (5.43)$$

with $s = 1$. The objective function is optimized under (5.21), (5.22) and the following additional constraints:

$$\prod_{l=1}^{p} \lambda_l = 1, \quad (5.44)$$

$$\lambda_l > 0, \qquad \forall l = \{1, \ldots, p\}. \quad (5.45)$$

5.5.2.2 Optimization

In this subsection, we use the iterative approach, similar to Section 5.4.1, to minimize (5.43). To achieve this, we utilize the method of Lagrangian optimization, allowing for the alternate updating of variables while keeping the others fixed. The details of each update step are provided in Theorems 5.5.4, 5.5.5, and 5.5.6 for the variables $\mathbf{M}$, $\boldsymbol{\lambda}$, and $\mathbf{V}$, respectively.

Theorem 5.5.4 (Updating $\mathbf{M}$ while fixing $\mathbf{V}$ and $\boldsymbol{\lambda}$). *The calculation of the credal partition, represented as* $\mathbf{M}$*, which minimizes the clustering objective function* $\mathcal{J}$*-MECMdd-RWG-P w.r.t. constraints* (5.21) *and* (5.22) *is derived from the following expression:*

$$m_{ij} = \frac{\left(|A_j|^{\alpha} \sum_{l=1}^{p} (\lambda_l)^s \tau_{ijl}\right)^{-\frac{1}{\beta-1}}}{\sum_{A_k \neq \emptyset} \left(|A_k|^{\alpha} \sum_{l=1}^{p} (\lambda_l)^s \tau_{ikl}\right)^{-\frac{1}{\beta-1}} + \left(\sum_{l=1}^{p} \delta_l^2\right)^{-\frac{1}{\beta-1}}}, \tag{5.46}$$

$$m_{i\emptyset} = 1 - \sum_{A_j \neq \emptyset} m_{ij}. \tag{5.47}$$

Proof. The proof proceeds in the same way as that presented in Theorem 5.5.1. We utilize the Lagrangian multipliers, denoted as $\boldsymbol{\xi}$, to address the constrained minimization problem concerning $\mathbf{M}$. Lagrangian function of (5.43) can be written as follows:

$$\mathcal{L}(\mathbf{M}, \boldsymbol{\xi}) = J_{MECMdd-RWG-P}(\mathbf{M}, \mathbf{V}, \boldsymbol{\lambda}) - \sum_{i=1}^{n} \xi_i \left(\sum_{\substack{A_j \subseteq \Omega, \\ A_j \neq \emptyset}} m_{ij} + m_{i\emptyset} - 1 \right). \tag{5.48}$$

By applying differentiation to the Lagrangian with respect to the variables m_{ij}, $m_{i\emptyset}$, and ξ_i, and subsequently setting the resulting derivatives to zero, we obtain the same equations (5.28), (5.29) and (5.30), whose resolutions allow us to obtain the optimal solutions of m_{ij} at (5.46) and $m_{i\emptyset}$ at (5.47).

Theorem 5.5.5 (Updating $\boldsymbol{\lambda}$, while fixing $\mathbf{M}$ and $\mathbf{V}$)**.** *Considering the constraint function based on the product of the positive relevance weights (5.44), and (5.45) in the optimization scheme of (5.43), the updating term of the weight vectors $\boldsymbol{\lambda}$ is obtained by:*

$$\lambda_l = \frac{\prod_{h=1}^{p} \left(\sum_{i=1}^{n} \sum_{A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^{\beta} \tau_{ijh} \right)^{\frac{1}{p}}}{\sum_{i=1}^{n} \sum_{A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^{\beta} \tau_{ijl}}. \tag{5.49}$$

Proof. To prove Theorem 5.5.5, the Lagrange multiplier $\boldsymbol{\xi}$, is applied to tackle the constrained minimization issue related to $\boldsymbol{\lambda}$. Thus, the objective function (5.43) is modified to include ξ_j, under the constraints (5.44) and (5.45), enabling to obtain the following Lagrangian function:

$$\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\xi}) = J_{MECMdd-RWG-P}(\mathbf{M}, \mathbf{V}, \boldsymbol{\lambda}) - \xi \left(\prod_{l=1}^{p} \lambda_l - 1 \right). \tag{5.50}$$

The next of the proof proceeds in the same way as the Theorem 5.5.2. By taking the derivatives of the Lagrangian with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\xi}$ variables and setting the derivatives to zero, we obtain:

$$\frac{\partial L(\boldsymbol{\Lambda}, \xi)}{\partial \lambda_l} = \sum_{i=1}^{n} \sum_{A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^{\beta} \tau_{ijl} - \xi \prod_{h \in \{1,\ldots p\}, h \neq l} \lambda_l = 0 \tag{5.51}$$

$$\frac{\partial L(\boldsymbol{\Lambda}, \xi)}{\partial \xi} = \prod_{l=1}^{p} \lambda_l - 1 = 0, \tag{5.52}$$

Multiplying by λ_l equation (5.51) and reducing the formula using (5.52) results in (5.53):

$$\lambda_l = \frac{\xi}{\sum_{i=1}^{n} \sum_{A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^{\beta} \tau_{ijl}}. \tag{5.53}$$

Substituting (5.53) into (5.51) gives then:

$$\xi = \prod_{h=1}^{p} \left(\sum_{i=1}^{n} \sum_{A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^{\beta} \tau_{ijh} \right)^{\frac{1}{p}}. \tag{5.54}$$

Finally, combining (5.53) and (5.54) gives (5.49).

Theorem 5.5.6 (Updating **V** while fixing **M** and $\boldsymbol{\lambda}$). *With the credal partition matrix* **M** *and dissimilarity matrix weight vector* $\boldsymbol{\lambda}$ *fixed, the prototype of the singleton cluster* $e_{h^\star} = v_k^{\Omega}$ *of* $A_j = \{\omega_k\}$ *with* $e_{h^\star} \in E$*, which minimizes the clustering objective function of MECMdd-RWG-P, is first computed according to (5.55).*

$$h^\star = \underset{1 \leq h \leq n}{\operatorname{argmin}} \sum_{i=1}^{n} m_{ij}^{\beta} \sum_{l=1}^{p} (\lambda_l)^s \tau_l(e_i, e_h). \tag{5.55}$$

Following this, the prototype of the imprecise cluster $e_{h^\diamond}$*, denoted as* $v_k^{2^\Omega}$*, associated with* $A_j \subseteq \{\omega_1, \ldots, \omega_k\}$*, where* $e_{h^\diamond} \in E$*, is determined based on equations (5.41) and (5.42).*

Proof. The proof of Theorem 5.5.6 is straightforward according to the proving details of the proof 5.4.1.2.

5.5.2.3 Computational algorithm and complexity

For ease of implementation, we provide a summary of the proposed method in Algorithm 10. Following a similar reasoning as in Section 5.5.1.3, we can conclude that the total time complexity of MECMdd-RWG-P is also of the order of $O(t(pn2^c + pcn^2))$ for t iterations.

Algorithm 10 MECMdd-RWG-P

Input: p dissimilarity matrices.
Parameters: refer to parameters of Algorithm 9
Initialization: $\mathbf{V}^0$ and $\boldsymbol{\lambda}^0$
while no convergence of the solution **do**
  (1) Compute the prototypes of meta-clusters utilizing (5.41) and (5.42);
  (2) Compute the credal partition matrix **M** using (5.46) and (5.47);
  (3) Compute the best relevance weight vector $\boldsymbol{\lambda}$ according to (5.49);
  (4) Compute the new prototype set **V** using (5.55);
end
Output: The optimal **M**, **V** and $\boldsymbol{\lambda}$

Remark. *It is interesting to note that, numerical instability, such as overflow or division by zero, can always arise when calculating the relevance weight of each dissimilarity matrix. This typically happens when the algorithm generates credal clusters such that* $\sum_{i=1}^{n} \sum_{A_j \neq \emptyset} m_{ij}^{\beta} d_{ijl} \to 0$. *However, the likelihood of these types of numerical inconsistencies occurring is greater when relevance weights are estimated locally. Additionally, the exponential growth in the number of clusters significantly increases computational complexity, making it challenging to interpret the assignment of objects to high-cardinality meta-clusters. To address this, constraints can be imposed to limit the number of focal elements in the meta-cluster. For instance, similar in ECM [1], we can restrict the meta-clusters focal set to contain most 2 elements.*

5.6 Empirical results

In this section, experiments were conducted on fifteen real-world multi-view and single-view datasets, as well as on a synthetic dataset and a real case study, to evaluate the performance and advantages of the proposed algorithms. First, in Section 5.6.1, we provide a detailed evaluation of the performance of the proposed algorithms MECMdd-RWL-S, MECMdd-RWG-S, MECMdd-RWL-P, and MECMdd-RWG-P, comparing them using validity measures and tools adapted to credal partitioning, along with a sensitivity analysis of the parameters of the proposed methods. Then, in Section 5.6.2, we assess the performance of our algorithms against related and state-of-the-art algorithms, including a detailed description of the datasets, the comparison process, and statistical significance analysis. Finally, in Section 5.6.1.4, we present a real-world case study of clustering chronic pain care pathways based on subjective and complex temporal data using the MECMdd-RWL-S and MECMdd-RWL-P algorithms.

Remark. *All experimental data and R code are available in a public GitHub repository*[1]*. And the algorithms also deployed in the python library* `evclust`[2].

5.6.1 Comparison of proposed algorithms on Iris dataset

In this experiment, we evaluate the performance based on credal partitions and weight of views of the MECMdd-RWL-S, MECMdd-RWL-P, MECMdd-RWG-S and MECMdd-RWG-P algorithms using the Iris dataset to highlight its strength. The Iris dataset comprises 150 objects distributed across 3 clusters under the edit framework $\Omega = \{\omega_1, \omega_2, \omega_3\}$, where each object is described by four features. We segmented the features relating to sepals (length and width) into one view, and those describing petals (length and width) into another view of the data. The relational data derived from each view was obtained using a Euclidean metric. The four algorithms were executed with the same parameters: $\alpha = 2$, $\beta = 1.5$, $\delta = 10$, $\gamma = 0.5$, and $\eta = 1$. The number of clusters was set to $c = 3$, and we considered a 'full' set of focal elements to compute all 2^c subsets of Ω. Each clustering algorithm was run 10 times until convergence to a stationary value of the objective function, and the best partition, corresponding to the minimum value of this function, was selected for detailed comparative analysis.

[1] The repository can be accessed at https://github.com/armelsoubeiga/MECMdd.
[2] evclust https://pypi.org/project/evclust.

5.6.1.1 Validity metrics

We considered two defined entropy for belief functions, namely Credal Rand Indices (CRI) [233] and Nonspecificity (NS) [209]. Based on the mass of belief, these metrics are adapted in the multi-view credal partition context to assess the quality, performance, and behavior of the partitions generated by the proposed algorithms. Inspired by [224], we also adapted the silhouette index, a metric commonly used to evaluate clustering performance, to the framework of evidential multi-view relational clustering with relevance weights for each view. The definition of this metric is provided below, applied in the case where view weights are estimated locally. However, for global weights algorithms, the metric is modified by using the globally estimated relevance weight λ_l instead of λ_{jl}.

Credal rand indices Let $\mathbf{M}$ and $\mathbf{M}'$ be two partitions to be compared and $\mathbf{R} = (m_{ij})_{1\leq i\leq j\leq n}$ and $\mathbf{R}' = (m'_{ij})_{1\leq i\leq j\leq n}$ be respectively their relational representation [84]. $\mathbf{R}$ is defined as a tuple, in which m_{ij} is a so-called pairwise mass function, defined by (5.56) and (5.57), where s_{ij} denotes the proposition that objects i and j belong to the same cluster, and $\neg s_{ij}$ denotes the negation of s_{ij} on the frame $\Theta_{ij} = \{s_{ij}, \neg s_{ij}\}$.

$$m_{ij}(\emptyset) = m_i(\emptyset) + m_j(\emptyset) - m_i(\emptyset)m_j(\emptyset), \tag{5.56}$$

$$m_{ij}(\{s_{ij}\}) = \sum_{k=1}^{c} m_i(\{\omega_k\})\, m_j(\{\omega_k\}). \tag{5.57}$$

The credal rand indices (CRI) is defined as follows :

$$CRI(\mathbf{M}, \mathbf{M}') = 1 - \frac{2\sum_{i<j}\varphi(m_{ij}, m'_{ij})}{n(n-1)}, \tag{5.58}$$

where φ is a distance between mass functions and n the number of objects. In this contribution, we used the CRI consistency index, i.e., where φ is the degree of conflict. However in [233], the authors proposed two others generalizations of the CRI where φ corresponds to Jousselme's distance and the L1 distance between belief functions. The CRI varies from 0 to 1, with 1 corresponding to the optimal value, and an evidential partition $\mathbf{M}$ is said preferable to another one $\mathbf{M}'$ when $\mathbf{M}$ is more consistent with the true classes and more precise [84], in other words, when $CRI(\mathbf{M}) > CRI(\mathbf{M}')$. $\mathbf{M}'$ represented here the credal partition of the true partition, i.e the vector of labels for the real classes encoding a hard partition has been transformed into a credal partition [233].

Nonspecificity The average nonspecificity (NS) of a credal partition, as defined in [233], and corresponds to the mean of the non-specificity of each m_i:

$$NS(c) \triangleq \frac{1}{n}\sum_{i=1}^{n} N(m_i) \quad \text{with} \quad N(m) \triangleq \frac{1}{log_2(c)} \sum_{A\in 2^{\Omega}/\emptyset} m(A) log_2 |A| + m(\emptyset) log_2 |\Omega| . \tag{5.59}$$

The non-specificity $1 \geq NS(c) \geq 0$ must be minimized, as lower values of this index indicate more certain partitions.

Silhouette index Kaufman and Rousseeuw [41] introduced the silhouette index to evaluate the compactness and separation of clusters for clustering algorithms generating a hard partition. The silhouette value, ranging in $[-1, +1]$, quantifies an object's fit within its cluster versus its separation from other clusters. Higher values indicate a better fit within the cluster. The average silhouette weight (ASW) of the individual silhouettes $s(e_i)$ of each object e_i is generally used for global interpretation:

$$ASW = \frac{1}{n} \sum_{e_i \in E} s(e_i), \tag{5.60}$$

where

$$s(e_i) = \frac{b(e_i) - a(e_i)}{max(b(e_i), a(e_i))}, \tag{5.61}$$

with $a(e_i)$ representing the average dissimilarity between e_i and all other points of the cluster to which e_i belongs (if e_i is the only observation in its cluster, $s(e_i) = 0$ without further calculations). For all other clusters A_k, the distance $d(e_i, A_k)$ is equal to the average dissimilarity of e_i to all observations of A_k such that $e_i \notin A_k$. Based on [224], we can define in a multi-view clustering framework with relevance weights, a(.) and b(.) as :

$$a(e_i) = \frac{\sum_{e \in A_j} \sum_{l=1}^{p} \lambda_{jl} d_j(e_i, e)}{|A_j|}, \tag{5.62}$$

$$d(e_i, A_k) = \frac{\sum_{e \in A_k} \sum_{j=1}^{p} \lambda_{kl} d_j(e_i, e)}{|A_k|}, \tag{5.63}$$

$$b(e_i) = \min_{A_k \neq A_j} d(e_i, A_k). \tag{5.64}$$

The credal partition of each object is converted into a hard credal partition using the Equation (2.46), and all views are assigned the same relevance weight for $\{\emptyset\}$.

5.6.1.2 Comparison of credal partitions

The comparison of the partitions below allows us to evaluate the quality and performance of the four proposed variants of multi-view evidential relational clustering. Figure 5.2 (a) assesses the quality and coherence of the credal partitions obtained with respect to the true partition of Iris flowers according to their species. It can be observed that the four credal partitions exhibit similar levels of coherence. The credal partition generated by MECMdd-RWG-P is the best constructed, with a Credal Rand Index (CRI) of 84%, indicating a good match with the real partition of the flowers, whereas MECMdd-RWG-S is the least performant, with a CRI of 77%. Figure 5.2 (b) shows that the credal partitions generated by MECMdd-RWL-P and MECMdd-RWG-P are the most specific, with Non-specificity (NS) values of 0.046 and 0.048, respectively, meaning that they assign less belief mass to meta-clusters, promoting well-defined clusters. However, according to the silhouette of the obtained partitions (Figure 5.2 (c)), the MECMdd-RWG-S and MECMdd-RWL-S algorithms

are the most performant compared to the others in terms of Average Silhouette Width (ASW), which indicates a better separation of clusters. We can conclude that although the MECMdd-RWL-P and MECMdd-RWG-P algorithms, where weights are optimized under a product constraint, exhibit higher CRI and NS values, demonstrating that they generate more coherent and specific partitions, the MECMdd-RWL-S and MECMdd-RWG-S algorithms, optimized under a sum constraint, achieve higher ASW values, indicating better cluster separation quality. This highlights that validity criteria are not limited to mere performance but also reflect the significance of the generated credal partitions. Higher CRI and NS values suggest increased coherence with the ground truth and reduced uncertainty, while higher ASW values indicate good compactness and separation of clusters, even though this may imply lower specificity in the credal partitions.

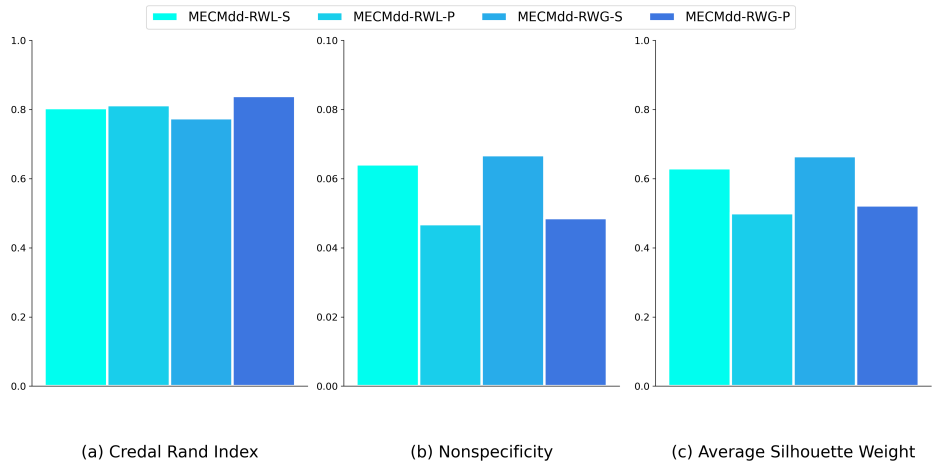


Figure 5.2: Performance and quality of clustering on Iris data, measured using the CRI, NS and ASW validity criteria. The performance of Figures (a) and (c) are to be maximized, while that of Figure (b) is to be minimized.

For advanced qualitative and visual analysis, the hard credal partition is then computed using the maximum of belief on the credal partition, based Equation (2.46). We can observe from Figure 5.3 that all four proposed algorithms identified imprecise subsets, i.e. objects belonging to the meta-clusters $\{\omega_1, \omega_2\}$ and $\{\omega_1, \omega_3\}$, while the MECMdd-RWG-P algorithm additionally identifies the meta-clusters $\{\omega_2, \omega_3\}$ and Ω subset. MECMdd-RWL-S and MECMdd-RWG-S algorithms classified 5 and 3 objects in the meta-clusters, respectively, while MECMdd-RWL-P and MECMdd-RWG-P classified 2 and 3 objects, respectively, as imprecise. This explains the higher non-specificity (Figure 5.2) of the first algorithms, as they are more uncertain about object assignment. The comparison with the true class of the objects shows that the proposed algorithms are mostly in agreement, classifying the objects into the expected clusters, except for MECMdd-RWL-P and MECMdd-RWG-S, which most disagrees respectively on 35 and 30 objects.

5.6.1.3 Comparison of weight of views

Figure 5.4 shows the relevance weights for the sepal and petal views of the Iris dataset. The values represent the relative importance of each view in characterizing the clustering, either globally (Figure 5.4 (a) and (c)) or in characterizing each cluster locally (Figure 5.4 (b) and

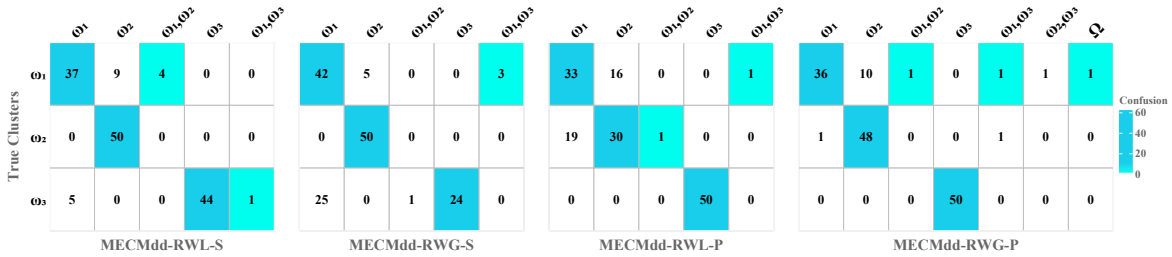
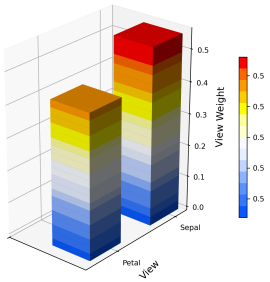
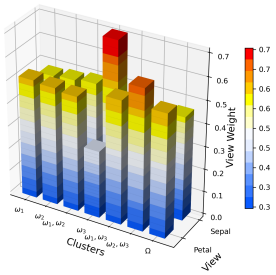


Figure 5.3: Qualitative comparison between obtained credal partitions and real clusters of iris flowers using confusion matrices.

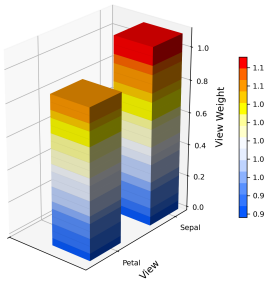
(d)). The sepal view is the most relevant in the overall clustering process in both weight learning approaches, whether constrained by sum or product. However, a local observation shows that the petal view is more informative for the majority of clusters, while the sepals describe the membership of the flowers only in cluster $\{\omega_3\}$ for MECMdd-RWL-S and in clusters $\{\omega_1\}$ and $\{\omega_2, \omega_3\}$ for MECMdd-RWL-P.



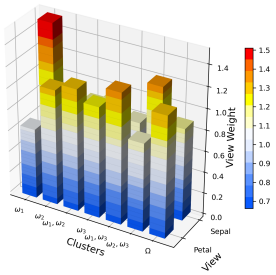
(a) MECMdd-RWG-S



(b) MECMdd-RWL-S



(c) MECMdd-RWG-P



(d) MECMdd-RWL-P

Figure 5.4: Illustration and comparison of the weights of the sepal and petal views of the Iris data from the proposed algorithms. (a) and (c) illustrate the global weights and (b) and (d) illustrate the local weights in each cluster.

Figure 5.5 illustrates the dispersion of the best partition of the weights for each algorithm, selected in Section 5.6.1. The algorithms with a product constraint learn to generate a wider dispersion of relevance weights, which promotes a more pronounced differentiation between views, particularly for MECMdd-RWL-P. This may indicate better local specialization in

the clustering process. In contrast, the algorithms with a sum constraint promote a more balanced distribution of weights, enabling a more homogeneous approach in terms of view relevance. The local constraints (RWL) tend to show more flexibility and adapt more precisely to local characteristics, while the global constraints (RWG) favor consistency across the entire dataset.

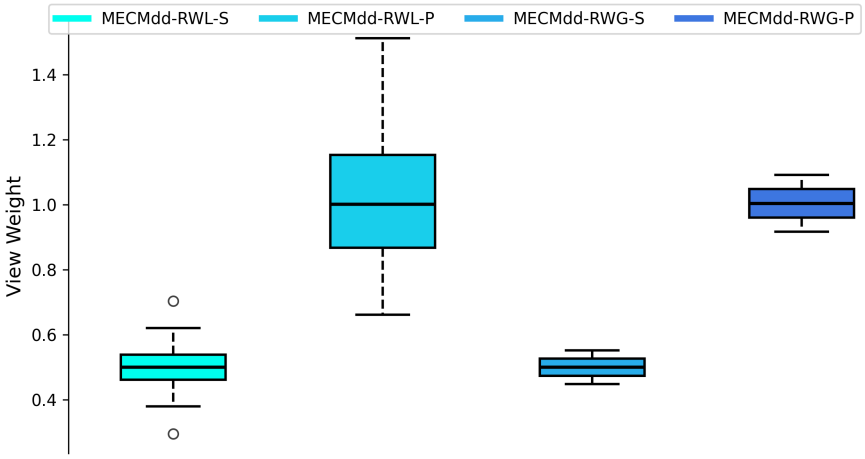
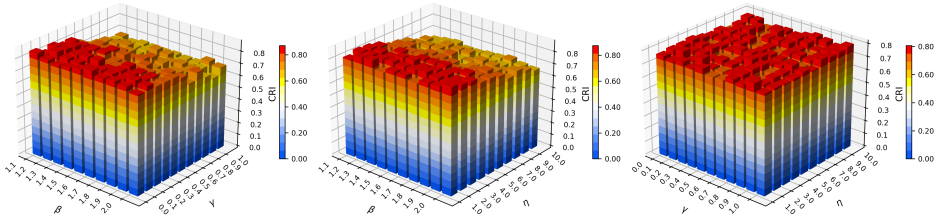


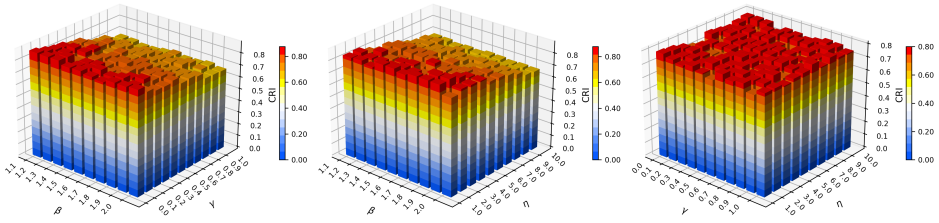
Figure 5.5: Distribution of view weights estimated by the MECMdd variants under sum and product constraints.

5.6.1.4 Parameters and Convergence analysis

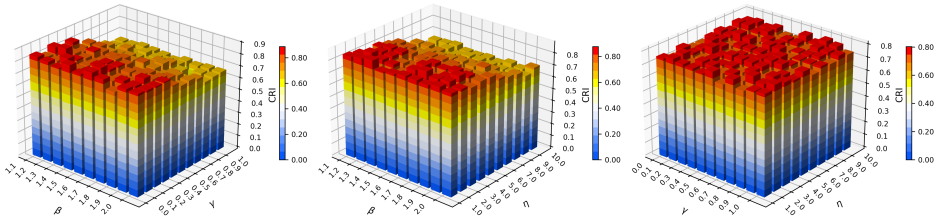
In this section, we first focus on the sensitivity of the results of the MECMdd-RWL-(S and P) and MECMdd-RWG-(S and P) algorithms to the parameter β, and to the additional parameters γ and η compared to ECM. We measured the CRI for several combinations of β, γ, and η values. The other parameters declared in Section 5.6.1 were kept constant, i.e. $\alpha = 2$ and $\delta = 10$. The results are presented in Figure 5.6. It can be observed that the values of $\beta \times \gamma$ and $\beta \times \eta$ have a significant impact on the performance of the algorithms. Indeed, in the first two plots of Figures 5.6 (a), (b), (c), and (d), we observe that the performance of the credal partitions (high CRI) is maximized for specific values of γ and η. Overall in these first experiments, the partitions seem most effective when $\gamma \leq 0.5$ and $\eta \leq 6$. The selection of appropriate values for γ and η can lead to an improvement in the algorithm's performance. These initial results do not show an obvious trend in the choice of the combination $\gamma \times \eta$, and at this stage, this choice must be made iteratively.
We also explored the convergence of the best partitions obtained (Section 5.6.1) on the Iris dataset for the MECMdd-RWL-S, MECMdd-RWG-S, MECMdd-RWL-P, and MECMdd-RWG-P algorithms. As shown in Figure 5.7, initially, as the number of iterations increases, the value of the objective function decreases significantly. However, in the final stages, this decrease slows down until convergence, which generally occurs after about ten iterations. This shows that each algorithm reaches the minimum value at every iteration. Each curve represents how the value of the objective function changes as the number of iterations increases.
Finally, in this section, we also observed the behavior of the choice of δ on the clustering performance of each developed MECMdd variant. As a reminder, δ controls the distance



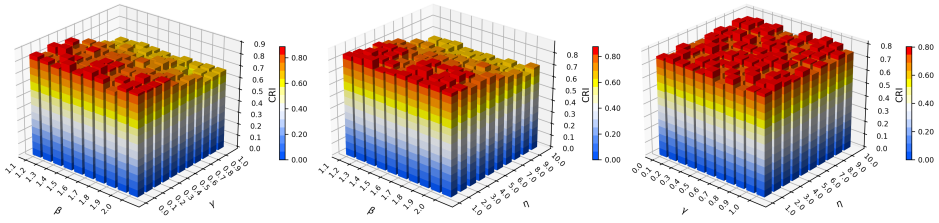
(a) MECMdd-RWL-S



(b) MECMdd-RWL-P



(c) MECMdd-RWG-S



(d) MECMdd-RWG-P

Figure 5.6: Sensitivity of parameters β, γ and η on the performance of credal partitions based on CRI of MECMdd variants. The bars represent CRI values according to different parameter combinations.

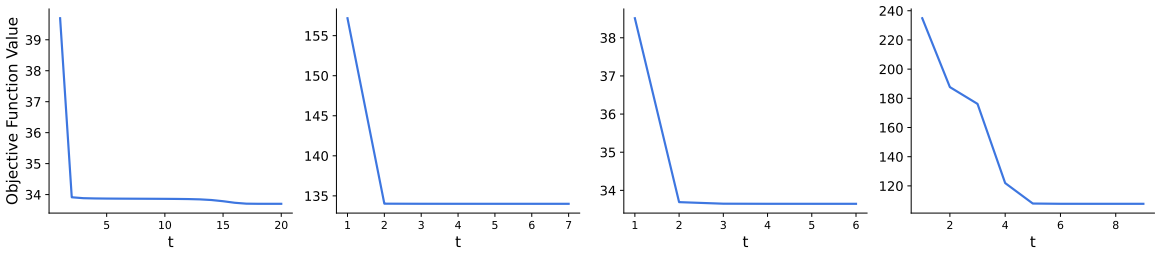


Figure 5.7: Object function value convergence, x-axes represent the iterations and y-axes the function values.

between objects and the empty focal set. The proposed algorithms allow setting a different δ for each view, based on the idea that views have different dispersions as they represent different sources of information describing the same objects. For instance, sepals and petals are different pieces of information, but describe the same iris flowers. To this end, we used the parameters set in Section 5.6.1 and varied δ. We considered two scenarios: one where δ is set by default to the same value for all views, and another where δ is assigned a different value for each view based on its dispersion. Each algorithm was run 10 times for both configurations of δ, and table 5.2 shows different results for the CRI and mean ASW validation criteria. Choosing a specific δ value for each view significantly improves the performance of the algorithm compared to using a default value for all views.

Table 5.2: The sensitivity of the choice of the δ value on the clustering performance based on the CRI and ASW of each variants of MECMdd algorithm. The Quantile_{95} corresponds to the value greater than 95% of the distances between the objects.

Choice of δ	Value of δ		RWL-S		RWG-S		RWL-P		RWG-P	
	Petal	Sepal	CRI	ASW	CRI	ASW	CRI	ASW	CRI	ASW
Default	3	3	0.788	0.535	0.780	0.505	0.436	0.529	0.505	0.515
Quantile_{95}	5	2	**0.821**	**0.591**	**0.847**	**0.543**	**0.804**	**0.562**	**0.803**	**0.553**

5.6.2 Comparison with other algorithms on real datasets

We start by outlining the experimental setting used to compare our algorithms with other methods on real datasets in Section 5.6.2.1. Then, in Section 5.6.2.2, we present the results obtained on these datasets in terms of clustering performance. Finally, in Section 5.6.2.3, we perform a detailed statistical analysis of the results to assess the superiority or effectiveness of the proposed algorithms compared to other methods.

5.6.2.1 Experimental setting

We performed experiments on fifteen publicly available real datasets. Data includes simple vector data, multi-view data, categorical data and multivariate time series data. The

state-of-the-art methods used to compare the performance of our methods consist of two classical single-view relational clustering methods, namely FCMdd[64] and ECMdd [62], a vector single-view ECM[1], and three representative multi-view relational and vector clustering methods, which include MRDCA[125], MRFCMdd[61], and WMVEC[225].

For simple vector data, we considered four benchmark datasets selected from the UCI machine learning repository [234], including Thyroid gland, Ecoli, Abalone, and Wine. The main details of each dataset are summarized in Table 5.3, where n corresponds to the number of instances, p to the number of features, and c represents the number of predefined clusters.

Table 5.3: Summary of the simple vector data.

Datasets	**n**	**p**	**c**
Thyroid gland (TG)	215	5	3
Ecoli (Ecoli)	336	7	8
Abalone (ABL)	4177	8	3
Wine (Wine)	178	13	3

For clustering algorithms that operate on single-view relational data, we computed a single dissimilarity matrix by using the Euclidean distance across all attributes together. Inversely, multi-view relational algorithms required the construction of multiple dissimilarity matrices, each derived from a different attribute with Euclidean distance computed individually per attribute. Multi-view vector algorithms and single-view vector algorithms, on the other hand, applied directly to the raw data, treating each attribute either as a distinct view or as a unified table combining all attributes, respectively.

The Table 5.4 shows basic information about the multi-view datasets used. We used four publicly available multi-view datasets, including Multiple features [235], Prokaryotic Phyla [225], WebKB [225], and MSRC [225]. The Prokaryotic Phyla dataset includes 551 prokaryotic species, characterized by three multi-view data such as text and two genomics representations. MSRC is an image dataset and contains 210 samples of 7 types (bicycle, tree, car, airplane, building, cow, face). There are five views in MSRC, and 30 images in each category. WebKB is a dataset consisting of web pages and hyperlinks for 4 classes from computer science departments, described by three views. Multiple features dataset contains 2000 handwritten digits from Dutch utility maps, each digit represented by six feature view sets, we utilized three sets: Fourier coefficients, profile correlations, and Karhunen-Love coefficients as in [235]. For each view in a multi-view dataset, one dissimilarity matrix was computed considering all attributes in the view, according to the Euclidean distance. Single-view clustering algorithms were applied to the Euclidean matrix obtained by considering all attributes across all views.
To expand the applicability of our proposed multi-view relational clustering techniques and to further validate their effectiveness, we also employed real-world datasets of categorical and categorical/continuous time-series types. Table 5.5 outlines the features of three categorical datasets of varied sizes from the UCI Machine Learning repository: Car Evaluation, Breast Cancer, and Zoo. In these experiments, we transformed the categorical data into numerical

Table 5.4: The details of the used multi-view datasets

Datasets	n	c	p	Views name	Features
Multiple features (MF)	2000	10	3	Fourier coefficients	76
				Profile correlations	216
				Karhunen-Love coefficients	64
Prokaryotic Phyla (PP)	551	4	3	Gabor	393
				Wavelet moments	3
				CENTRIST	438
WebKB (WKB)	203	4	3	Text on web pages	1703
				Anchor text in hyperlinks	203
				Text in the tittle	203
MSRC (MSRC)	210	7	5	Local binary patterns	256
				Histogram of oriented gradients	100
				Color moments	48
				Scale-invariant feature transform	200
				Global image scene transform	512

format via one-hot encoding, enabling us to apply numerical vector methods like ECM and WMVEC [41]. In the ECM approach, the algorithm utilizes all newly derived one-hot encoded attributes collectively as a single view. Inversely, with the mean-based multi-view method WMVEC, each set of encoded attributes from a specific variable is treated as an independent view. The treatment of categorical data for relational clustering can be performed using appropriate dissimilarity measures. Various measures have been proposed in the literature (see [236] for a comprehensive review). We thus used two commonly employed dissimilarity measures: Hamming distance [152] for individual binary categorical variables and Gower distance [237] for individual multiple categories variables; when the dataset included both binary and multiple categories variables, we used the Gower distance. Using these dissimilarity metrics, we then applied medoid-based single-view clustering methods (FCMdd and ECMdd) to the resulting dissimilarity matrices derived from all attributes combined. For the multi-view medoid-based clustering algorithms (MRDC, MRFCMdd and MECMdd), we computed separate dissimilarity matrices for each categorical variable, by treating each as an individual view in the multi-view framework.

Table 5.5: Datasets information of the used categorical datasets.

Datasets	n	p	c
Car Evaluation (CAR)	1728	6	4
Breast Cancer (BC)	286	9	2
Zoo (ZOO)	101	16	7

In Table 5.6, we show the characteristics of the four multivariate time series datasets selected. The multivariate continuous time series datasets are available online at the UCR time series [238]. The Biofam dataset is multivariate categorical time series from the data of the retrospective biographical survey carried out by the Swiss Household Panel [145]. As shown

in this Table, datasets with different number of time series, different length of time series, and different number of classes are selected. In this experiments, we use Dynamic Time Warping (DTW) [239] distance for clustering continuous time series data. DTW is the most well known technique for evaluating dissimilarity of time series with respect to their shape information. Unlike the Euclidean distance, DTW can compare time series of variable size and is robust to shifts or dilatation across the time dimension. The dissimilarity between pairs of categorical time series is calculated using the Optimal Matching (OM) [118] measure. Each uni-variate time series is considered as a view. The single-view medoid-based clustering algorithms were applied to the dissimilarity matrix obtained by a linear sum combination, considering all dissimilarities of all views.

Table 5.6: Multivariate time series selected for experimental studies.

Datasets	**n**	**p**	**c**	**Length**	**Type of features**
UWaveGestureLibrary (UGL)	120	3	8	315	Continuous
StandWalkJump (SWJ)	12	4	3	2500	Continuous
BasicMotions (BM)	40	6	4	100	Continuous
Biofam dataset (BF)	2000	3	2	30	Categorical

All dissimilarity matrices were normalized according to their overall dispersion [240] to have the same dynamic range. This means that each dissimilarity $d(e_k, e_{k'})$ in a given dissimilarity matrix has been normalized as $d(e_k, e_{k'})/T$, where $T = \sum_{k=1}^{n} d(e_k, g)$ is the overall dispersion and $g = e_{h^\star} \in E = \{e_1, \ldots, e_n\}$ is the overall prototype, which is computed according to $h^\star = argmin_{1 \leq h \leq n} \sum_{k=1}^{n} d(e_k, e_h)$.

The relational and vector hard clustering algorithms used in the comparison were respectively applied to the similarity matrices or vector data to obtain a set of groups $\mathscr{G} = (G_1, \ldots, G_c)$. From all fuzzy partitions, we generate hard partitions with the maximum principle rule i.e. by defining the hard cluster $G_k(k = 1, ..., c)$ as $G_k = \{e_i : u_{ik} \geq u_{im} \quad \forall m \in \{1, \ldots, c\}\}$, where the quantity u_{ik} represents the membership degree of object $e_i(\forall i = \{1, \ldots, n\})$ in fuzzy cluster $k(\forall k = \{1, \ldots, c\})$. For the proposed algorithms and credal partitions used in the comparison, we can employ pignistic transformation [241, 1] BetP in (2.44) to turn the credal partition into a fuzzy and hard partition. With such conversion, the mass of belief of the meta-cluster is evenly distributed to the relevant singleton clusters. Each clustering algorithm was run (until the convergence to a stationary value of the adequacy criterion) two times and the best result was selected according to the adequacy criterion and the parameters presented in Table 5.7.

5.6.2.2 Results on real datasets

To evaluate the performance of the different clustering algorithms and compare them, we assume the ground truth is known. The hard partition generated by each algorithm for each dataset was then compared with the predefined partition. The parameters of different methods are presented in Table 5.7, and the comparison criteria used were the Adjusted Rand Index (ARI) and the Precision (P) [162]. Higher values of these two indices indicate better clustering results. The higher value of each index is kept according to the number of iterations (2 times) and hyper-parameters of each algorithm (Table 5.7). Based on the

analysis of evidential clustering parameters in [1, 62, 226], α have little impact on the results and is considered fixed at 2 for all experiments. In the rest of the work, we have renamed the following datasets UWaveGestureLibrary (UGL), StandWalkJump (SWJ) and Multiple features (MF).

Table 5.7: The parameters setting of different algorithms. $Q_{0.95}$ is the quantile function at 95%, p number of views.

Algorithms	**Parameters**
ECM	$\alpha = 1; \beta \in \{1.1, 1.2, \ldots, 2.0\}; \delta = 10$
ECMdd	$\alpha = 1; \beta \in \{1.1, 1.2, \ldots, 2.0\}; \delta \in \{10, Q_{0.95}(\tau_{ij:i\neq j})\}; \eta \in \{1, 2\}; \gamma \in \{0, 1\}$
MECMdd	$\alpha = 1; \beta \in \{1.1, 1.2, \ldots, 2.0\}; \delta^p \in \{10, Q_{0.95}(\tau_{ij:i\neq j})\}; \eta \in \{1, 2\}; \gamma \in \{0, 1\}; s = 2$
WMVEC	$\alpha = 1; \beta \in \{1.1, 1.2, \ldots, 2.0\}; \lambda \in \{e^0, e^5, e^{10}\}$
FCMdd MRFCMdd	$m \in \{1.1, 1.5, 2, 2.5, 3.0\}$; $s = 2$
MRDC	*none*

Tables 5.8 and 5.9 show the performance of the clustering algorithms on each dataset, respectively according to ARI and Precision (P) criteria. We can observe from the results that, compared with clustering methods based on a single view (i.e. ECM, ECMdd and FCMdd), algorithms adapted to multi-view data, such as MECMdd, WMVEC, MRFCMdd and MRDC, generally show superior performance on multi-view datasets (MF, PP, WKB and MSRC). This underlines the inadequacy of using single-view clustering methods to cluster multi-view data. We can also observe that, in general, relational algorithms perform better than vector algorithms, the reason being that relational methods use an adapted distance measure, unlike the Euclidean distance used with vector algorithms. In particular, while medoid-based relational clustering approaches can cluster row data time series using adapted measures, vector-based algorithms are not applicable. In general, variants of algorithms that locally estimate the weights of the views (RWL) yield better results. We conclude from these experiments that the proposed algorithms, i.e. MECMdd-RWL-S, MECMdd-RWG-S, MECMdd-RWL-P and MECMdd-RWG-P, present comparable or superior performances to the reference algorithms used in this comparative experiment. In addition, they provide the advantage of a more robust representation of uncertainty in uncertain data contexts, and can be transformed into hard and fuzzy partitions. In Section 5.6.2.3, we examine in detail the statistical significance of these performances.

5.6.2.3 Results statistical analysis

In this section, we applied non-parametric Friedman and Nemenyi tests to analyze the performance of the MECMdd variants in terms of Adjusted Rand Index (ARI) and Precision (P). The critical difference (CD) diagrams based on these tests are presented in Figures 5.8 and 5.9, which allow us to visualize the relative performance of the algorithms. The two critical

Table 5.8: Clustering performance of hard, fuzzy, and credal partitions algorithms with different relational datasets based ARI. Values in bold indicate the best results per dataset. The "-" indicates inappropriate methods for a given dataset.

Algorithm	ABL	Ecoli	Wine	TG	PP	MF	CAR	BC	ZOO	BF	UGL	SWJ	BM	WKB	MSRC
ecm	0.59	0.80	0.72	0.53	0.48	0.90	0.51	0.55	0.90	-	-	-	-	0.74	0.80
wmvec	0.66	0.70	0.72	0.63	0.62	0.92	0.56	0.54	0.67	-	-	-	-	0.53	0.88
fcmdd	0.61	0.78	0.72	0.55	0.58	0.61	0.52	0.50	0.91	0.51	0.87	0.56	0.92	0.56	0.81
mrfcmdd-rwl	0.62	0.36	0.64	0.57	0.59	0.87	0.54	0.58	0.82	0.45	0.88	0.55	0.88	0.59	0.83
mrfcmdd-rwg	0.62	0.79	0.66	0.55	0.60	0.86	0.49	0.56	0.63	0.42	0.87	0.59	0.89	0.64	0.88
mrdca-rwl	0.62	0.80	0.85	0.52	0.62	0.92	0.66	0.56	0.90	0.50	0.89	0.55	0.90	0.64	0.89
mrdca-rwg	0.62	0.76	0.71	0.57	0.60	0.92	**0.67**	**0.59**	0.89	0.50	0.84	0.50	0.91	0.62	0.84
ecmdd	0.62	0.72	0.73	0.61	0.50	0.92	0.53	0.51	0.85	0.41	0.86	0.65	0.61	0.80	0.81
mecmdd-rwl-s	0.62	0.82	0.76	**0.95**	**0.65**	0.92	0.56	0.57	0.93	0.44	**0.89**	**0.65**	0.86	0.61	0.86
mecmdd-rwg-s	0.63	0.72	0.75	0.64	0.59	**0.93**	0.51	0.57	0.91	0.43	0.87	0.59	0.90	0.82	0.86
mecmdd-rwl-p	**0.68**	0.34	0.86	0.81	0.61	0.91	0.51	0.50	**0.98**	**0.53**	0.86	0.64	**0.98**	**0.85**	**0.93**
mecmdd-rwg-p	0.63	**0.89**	**0.89**	0.70	0.60	0.91	0.51	0.50	0.72	0.51	0.86	0.59	0.83	0.84	0.88

Table 5.9: Clustering performance of hard, fuzzy, and credal partitions algorithms with different relational datasets using Precision. Values in bold indicate the best results per dataset. The "-" indicates inappropriate methods for a given dataset.

Algorithm	ABL	Ecoli	Wine	TG	PP	MF	CAR	BC	ZOO	BF	UGL	SWJ	BM	WKB	MSRC
ecm	0.41	0.77	0.59	0.59	0.45	0.47	0.61	0.63	0.92	-	-	-	-	0.69	0.34
wmvec	**0.45**	0.68	0.59	0.69	**0.53**	0.58	0.70	0.58	0.33	-	-	-	-	**0.70**	0.51
fcmdd	0.42	0.65	0.59	0.69	0.46	0.30	0.62	0.58	0.89	0.52	0.46	0.21	0.80	0.41	0.34
mrfcmdd-rwl	0.44	0.29	0.47	0.65	0.47	0.39	0.62	0.55	0.77	0.55	0.50	0.29	0.75	0.46	0.39
mrfcmdd-rwg	0.44	0.71	0.50	0.61	0.49	0.33	0.56	0.55	0.80	0.33	0.47	0.27	0.81	0.54	0.41
mrdca-rwl	0.43	0.73	0.78	0.60	**0.53**	**0.64**	0.70	0.61	0.88	0.42	0.51	0.23	0.76	0.51	0.42
mrdca-rwg	0.43	0.60	0.55	0.66	0.48	0.63	0.70	0.63	0.75	0.51	0.39	0.22	0.82	0.52	0.43
ecmdd	0.44	0.64	0.60	0.68	0.44	0.41	0.63	0.58	0.76	0.58	0.44	**0.37**	0.57	0.59	0.36
mecmdd-rwl-s	0.44	0.77	0.63	**0.93**	0.50	0.35	0.66	**0.64**	0.92	**0.58**	**0.52**	**0.37**	0.70	0.51	0.49
mecmdd-rwg-s	0.44	0.73	0.63	0.70	0.48	0.57	**0.71**	0.58	0.92	0.44	0.46	0.32	0.78	0.48	0.48
mecmdd-rwl-p	0.44	0.28	0.79	0.87	0.50	0.56	0.61	0.58	**0.94**	0.51	0.42	0.33	**0.94**	0.59	**0.60**
mecmdd-rwg-p	0.44	**0.84**	**0.84**	0.69	0.48	0.55	0.60	0.58	0.39	0.51	0.43	0.33	0.84	0.65	0.55

difference diagrams highlight distinct rankings based on the Adjusted Rand Index (ARI) and Precision. In the ARI-based diagram, MECMdd-RWL-S and MECMdd-RWL-P stand out as the top-performing algorithms, followed by MRDCA-RWL. This indicates that algorithms calculating relevance weights locally (RWL) achieve the best performance in terms of cluster consistency. MECMdd-RWG-S and MECMdd-RWG-P, which compute weights globally, are ranked next, demonstrating good performance but slightly behind their local counterparts. In the Precision-based diagram, a similar pattern is observed with MECMdd-RWL-S and MECMdd-RWL-P leading the rankings, followed by MECMdd-RWG-S and MECMdd-RWG-P. This suggests that RWL algorithms not only provide more consistent partitions but also exhibit better accuracy in assigning objects to clusters. The global weight algorithms (RWG) also perform well but show a slight decrease in comparison to the local variants.

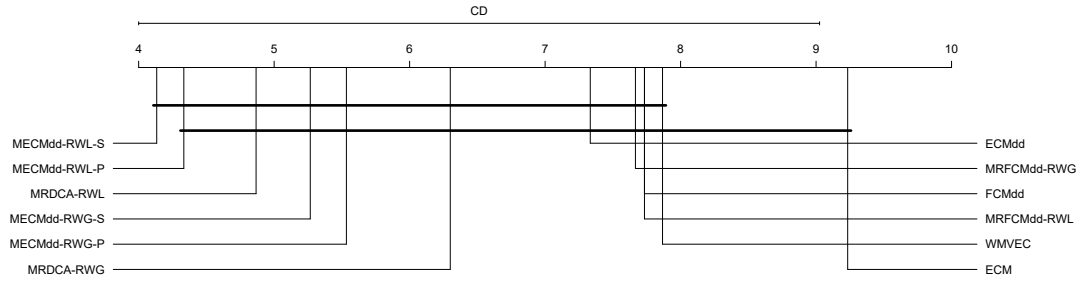


Figure 5.8: Friedman critical difference test diagram between MECMdd variants based on Ajuste Rand Index (ARI) considering all data on all data.

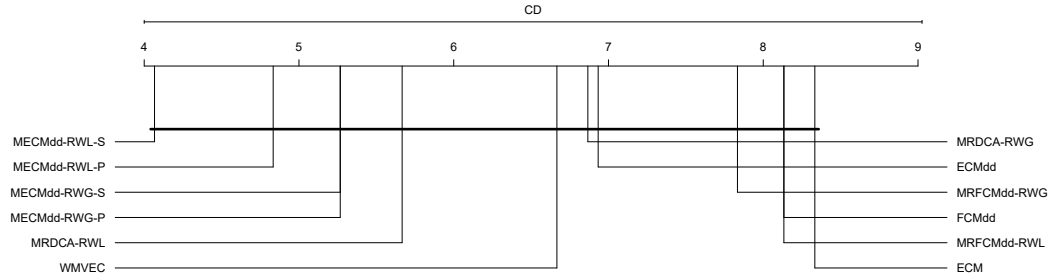


Figure 5.9: Friedman critical difference test diagram between MECMdd variants based on Precision (P) considering all data on all data.

5.6.3 Case study: application to real-life care pathways

In relational clustering analysis, the dissimilarity matrices can be generated using different sets of variables (or different view) and a fixed dissimilarity function. Alternatively, a fixed set of variables and different dissimilarity functions could have been utilized, or using different sets of variables and dissimilarity functions. Indeed, the use of an optimal distance function adapted to each dimension or view considerably improves clustering performance. In this real-life case study, we illustrate this advantage of relational clustering approaches.

The objective of this case study is to identify the typical care pathways of patients suffering from chronic pain, with explainability regarding their membership in a given typology.

The data come from the eDOL project, presented in Section 2.1.1 of chapter 2. We are interested on barometers composed of eight attributes measured weekly, as detailed in Table 2.1, allowing for the evaluation of pain intensity and monitoring: pain, fatigue, mood, stress, sleep, body comfort, sports and non-sports activities. These dataset, as described in Section 2.1.3.2, are complex and subjective, and can be treated either as categorical time series (CTS), discrete time series (DTS), or continuous/numerical time series (CNTS). We worked on data from 636 patients, extracted from eDOL in June 2023, totaling 14,090 self-reported care events, with an average follow-up of 5 months and a total duration of approximately 19 months.

We executed the MECMdd-RWL-S and MECMdd-RWL-P algorithms 5 times, using pairwise trajectory similarity measures as Optimal Matching (OM) [118], Time Warp Edit Distance (TWED), and Dynamic Time Warping (DTW) [239], when the data are respectively considered as CTS, DTS, and CNTS data. We performed both algorithms with the same parameters, based on the work of [i] (contribution of chapter 4): $\alpha = 2$, $\beta = 1.5$, $\delta = 10$, with the number of clusters fixed at $c = 2$, [i] identified, via a careful search involving several iterations, that 2 clusters corresponded to an optimal solution for evidential clustering. We fixed $\gamma = 0.5$, and $\eta = 1$, since it gives efficient performances on the data sets. The ASW and NS validity metrics were measured for each obtained partition. The experimental workflow is summarized in Figure 5.10. Indeed, each variable is considered separately as a view and transformed into either a categorical, discrete or continuous data sequence. The similarity metric (OM, TWED, or DTW) is then calculated between pairs of objects to give a distance matrix. For each metric, we obtain 8 relational matrices, which are then used in the clustering process with matrix weighting in each algorithm.

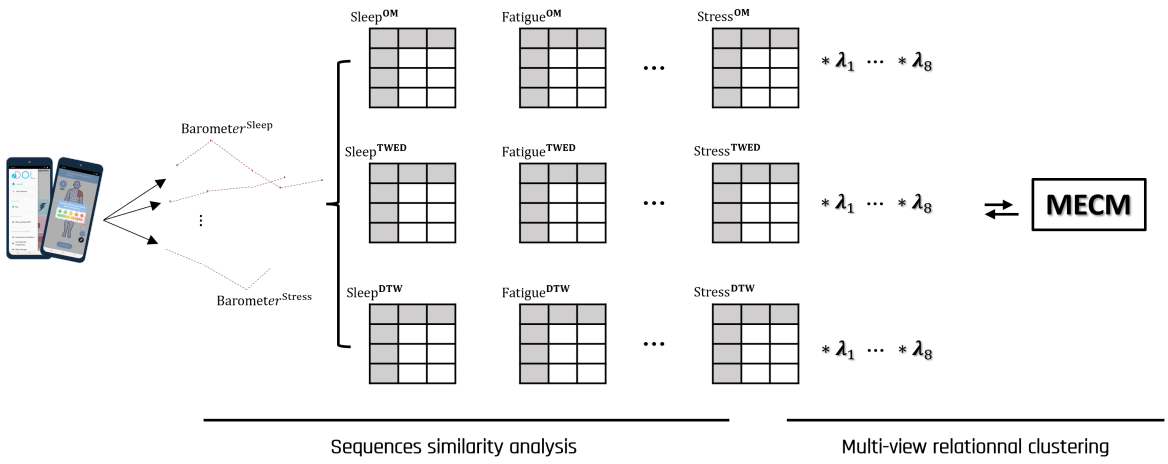


Figure 5.10: Illustration of experimental workflow. ⇆ means the iteration process for calculating optimum weights.

Table 5.10 shows the clustering results of the MECMdd-RWL-S and MECMdd-RWL-P algorithms for the three similarity measures OM, TWED, and DTW. For each algorithm and each measure, the result with the lowest objective function value was considered. We observe that the OM similarity measure gives the best results for both algorithms, far ahead of the other two measures, which yield similar results. This clearly shows that treating these data as CTS is most recommended, as similarity measures used for categorical sequences

like OM seem to better capture the unique characteristics of these data. We also note that the MECMdd-RWL-P algorithm generates partitions where uncertainty is minimized, with more precise and specific cluster assignments (as evidenced by a lower NS value), while MECMdd-RWL-S creates well-defined and well-separated clusters (as evidenced by a higher ASW value). In both algorithms, we obtain two specific clusters (singletons), which we call Cl_1 and Cl_2, and an imprecise cluster $\{Cl\_1, Cl\_2\}$, intermediate between the two clusters, which we call $Cl\_uncertain$.

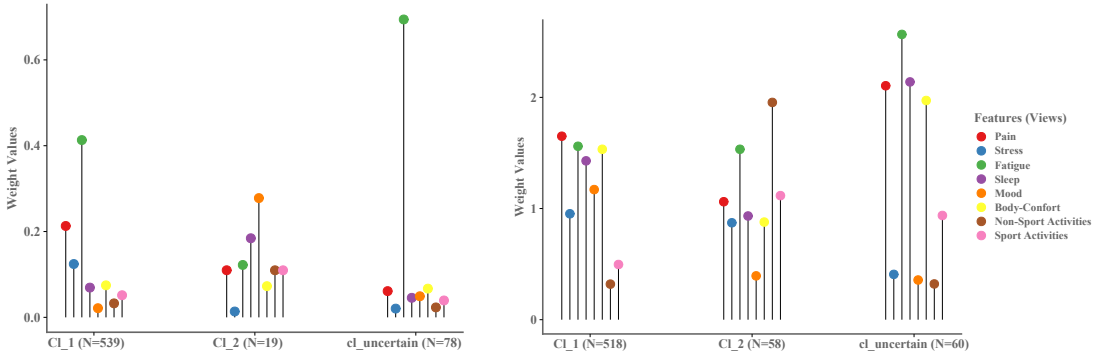
Table 5.10: Clustering results of the used eDOL dataset based ASW and NS performance metrics and OM, TWED and DTW similarity metrics.

Similarity Metric	MECMdd-RWL-S		MECMdd-RWL-P	
	ASW	NS	ASW	NS
OM	**0.730**	**0.122**	**0.679**	**0.091**
TWED	0.520	0.238	0.516	0.231
DTW	0.616	0.259	0.513	0.252

Analysis of variable relevance weights (Figure 5.11) allows for post-hoc categorization or explainability of clusters. Learning the relevance weights of views (features) is a major advantage of the algorithms we proposed, both in terms of performance and explainability of results. In a classical approach, this explicative task is performed with other tools like SHapley Additive exPlanations (SHAP) [213] after the clustering process. The comparison of relevance weights shows that for the MECMdd-RWL-S algorithm (Figure 5.11 (a)), the Cl_1 cluster is strongly determined by variables such as **pain**, **stress**, **fatigue**, and **sports and non-sports activities**. In contrast, the Cl_2 cluster is more explained by variables such as **sleep** and **mood**. Cluster Cl_2 groups patients with chronic pain related to sleep and mood, while Cl_1 groups patients with chronic pain related to stress and fatigue. Patients belonging to the uncertain cluster exhibit characteristics common to both groups. These results obtained with MECMdd-RWL-S are similar to results obtained in [213]. The comparison of relevance weights for the MECMdd-RWL-P algorithm, however, is not comparable to this study. Indeed, this algorithm gives a high weight to all features in the Cl_1 cluster, with the exception of features relating to sporting or non-sporting activities, which are determinant in the Cl_2 cluster.

5.7 Conclusion

In this contribution, we propose medoid-based relational multi-view evidential clustering algorithm that are able to partition objects taking into account simultaneously their relational descriptions given by multiple dissimilarity matrices. Referred to as MECMdd, it based on the framework of belief functions. MECMdd is able to address the uncertainty and imprecision of multi-view relational data clustering, and provides a credible partition, which extends fuzzy, possibilistic and rough partitions. It can automatically learn the importance of each views by estimated a weight locally for each cluster with a weight sum constraint (MECMdd-RWL-S) or locally with a weight product constraint (MECMdd-RWL-P) and



(a) MECMdd-RWL-S (b) MECMdd-RWL-P

Figure 5.11: Visualization of the feature weights on eDOL dataset.

globally with a weight sum constraint (MECMdd-RWG-S) or globally with a weight product constraint (MECMdd-RWG-P) in a collaborative learning framework. We define two objective functions and develop corresponding iterative optimization algorithms to obtain the optimal solutions, respectively. Experimental results obtained on several real datasets and a case study demonstrate the potential and superiority of our proposed algorithm.

In the next contribution in chapter 6, we introduced a new algorithm, called Soft-ECM, based on belief function theory, but able to group data not defined in a Eucludian space using barycenters as ECM, i.e. based on row data directly.

Chapter 6

Row Data-Based Soft-Evidential C-Means Clustering

Abstract – In this chapter, we address the challenges of evidential clustering of trajectory data using the row-data object-based approach. We reformulate the Evidential C-Means, problem to accommodate non-Euclidean data such as mixed data (numerical and categorical) and non-tabular data like trajectory data, for example time series, which evidential traditional algorithms cannot handle due to their reliance on Euclidean space properties. To overcome these limitations, we propose a novel algorithm, Soft-ECM, capable of positioning centroids of imprecise clusters consistently using only a semi-metric. Our experiments demonstrate that Soft-ECM achieves results comparable to conventional fuzzy and evidential clustering methods on numerical data, while also proving effective for mixed data. Furthermore, we highlight its advantages in integrating semi-metrics such as DTW for time series data, showcasing its versatility and potential in diverse clustering applications. Furthermore, we applied Soft-ECM to the real-life case study of chronic pain trajectory clustering, demonstrating its effectiveness.

Contents

6.1 Introduction

In numerous real-world applications, only partial information is available for objects, and in these cases hard clustering can result in poor accuracy. To address this problem and capture the degree of ambiguity regarding the class membership of the objects, soft clustering variants have been proposed, including Fuzzy C-Means[69] and Evidential C-Means[1]. These variants allow to describe the uncertainty and/or imprecision in the partition. Fuzzy C-Means (FCM) [69] is based on fuzzy set theory, while Evidential C-Means (ECM) clustering is based on belief function theory [1]. ECM allows objects to belong not only to singleton clusters of the solution set Ω, but also to any subset of Ω (i.e. *meta-clusters*) with different belief masses. This additional flexibility over FCM provides a deeper insight into the data and improves robustness against outliers.

However, ECM was originally designed for tabular data and it handles only quantitative attributes, i.e. data in a vector space. Indeed, ECM is based on Euclidean distance: the problem formulation assumes the capability to compute barycenters as a mean position of cluster elements. However, this is not always feasible. In order to adapt ECM to different types of data, variants have been proposed. Among these variants, ECMdd [62] and RECM [63] use relational similarity measures, while CECM [103] introduces clustering constraints to better guide data partitioning and employs a specific Mahalanobis distance for each cluster. ECM+ [106] improves the center definition when Mahalanobis distances are used. Other variants, such as CatECM [105], are designed for categorical data. Credal C-Means (CCM) [102] and Belief C-Means (BCM) [101] modify how objects are assigned to meta-clusters. These approaches reconsider the computation of cluster centers by considering distances between objects and clusters, avoiding the arithmetic mean approximation, which improves accuracy for highly uncertain data. Furthermore, some data types are not tabular, such as time series for which hard clustering methods have already been adapted [242]. To the best of our knowledge, no evidential clustering methods have yet been proposed to handle such data.

In this contribution, we propose Soft-ECM, a new algorithm that reformulates the ECM problem as a constrained multi-class clustering, where singleton clusters and meta-clusters (imprecise clusters) are individually optimized. In contrast to the classical ECM approach, meta-clusters are here treated like singleton clusters, allowing greater flexibility in dealing with uncertainties. A new constraint on meta-cluster barycenters ensures that they are positioned consistently with those of singleton clusters. In this way, the Soft-ECM algorithm is more flexible, and allows the use of non-Euclidean dissimilarity measures. This makes it possible to deal with complex data, such as mixed data (tabular data with both quantitative and qualitative features) or even non-tabular data such as time series.

6.2 Soft Evidential C-Means (Soft-ECM)

We propose to reformulate the ECM problem[1], presented in Section 2.3.5.2 of chapter 2, in a more generic way, so that the evidential clustering problem can be solved with dissimilarity

[1]We can observe that ECM uses the properties of Euclidean distance to define both the function to be optimized (eq. 2.47) and the meta-cluster centroids (eq. 2.51).

measures other than Euclidean distance. In ECM case, the centroids of meta-cluster defined as in Equation 2.51 is correct when the data are in a Euclidean space. Intuitively, our approach relaxes the definition of meta-cluster centroid as the explicit isobarycenter of singletons' centroids. In Figure 6.1, the meta-cluster centroid $\boldsymbol{v}_{\{A,B\}}$ is constrained by both being as close as the barycenter of its examples (green dashed lines) and being as close as barycenter of the singletons clusters centroids (red dashed lines).

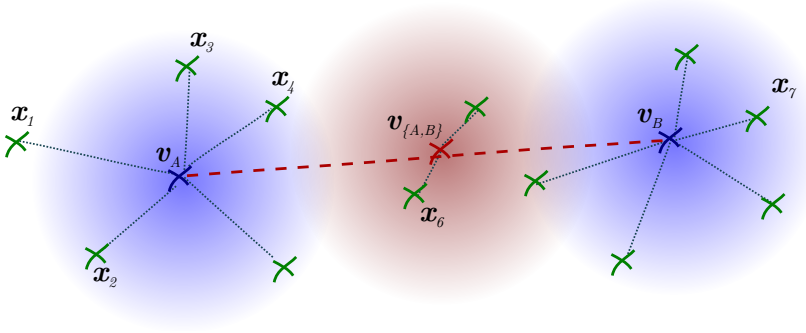


Figure 6.1: Illustration of a 2-cluster evidential clustering. Examples are represented in green. Points in blue $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$ indicate cluster centers, A and B. The red point $\boldsymbol{v}_{\{A,B\}}$ is the center of meta-cluster $\{A, B\}$. This point is both the barycenter of examples belonging in the meta-cluster $\{A, B\}$; and the barycenter of $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$. The fine dotted lines indicate the distances considered in computing the new clustering inertia.

Let $\mathbf{X} = (\mathbf{x}_i)$ denotes a set of objects in $\mathbf{X}$ (e.g., time series) and d denotes a semi-metric between these objects.[2] In a classical way [242], we can then define the barycenter (medoid) of a set of objects as an element of $\mathbb{X}$ that minimizes inertia. More formally, the barycenter (medoid), $\mathbf{v}_E$, of the subset $E \subseteq \boldsymbol{X}$ is defined as follows:

$$\mathbf{v}_E = \underset{\boldsymbol{x}\in\mathbb{X}}{\operatorname{argmin}} \sum_{\boldsymbol{e}\in E} d(\boldsymbol{e}, \boldsymbol{x}). \tag{6.1}$$

Then, the Soft-ECM problem is formalized as the following constrained minimization problem:

$$\begin{aligned} \min_{\boldsymbol{M},\boldsymbol{\mathcal{V}}} J^{\alpha,\beta,\delta}(\boldsymbol{M},\boldsymbol{\mathcal{V}}) &= \sum_{i=1}^{n} \sum_{A\subseteq\Omega\setminus\emptyset} |A|^{\alpha} m_i(A)^{\beta} d(\mathbf{x}_i, \mathbf{v}_A) + \sum_{i=1}^{n} \delta^2 m_i(\emptyset)^{\beta} \\ \text{subject to } & \sum_{A\subseteq\Omega} m_i(A) = 1,\ \forall i \in \{1, n\}, \\ & \mathbf{v}_A = \underset{\mathbf{x}\in\mathbb{X}}{\operatorname{argmin}} \sum_{\omega_k\in A} d(\mathbf{v}_{\{\omega_k\}}, \mathbf{x}),\ \forall A \subseteq \Omega. \end{aligned} \tag{6.2}$$

In this formulation, $\boldsymbol{M} = (m_i\,(A))$ represents the mass functions on a focal element $A \subseteq \Omega$, while $\boldsymbol{\mathcal{V}} = \{\mathbf{v}_A\}$ corresponds to the barycenters of all singleton clusters and meta-clusters. In contrast to ECM formulation (eq. 2.47), which focused on the optimization of cluster positions only and defined the positions of meta-clusters by the equation 2.51, here we

[2] a semi-metric is a symmetric, positive and defined dissimilarity measure metric, but it does not necessarily hold the triangular inequality.

optimize both. The second constraint enforces the centroid of each meta-cluster A to be the barycenter of the cluster centroids ($\boldsymbol{v}_{\omega_k}$ where $\omega_k \in A$). We find again the expression of the equation that defines the barycenter (eq. 6.1).

Theorem 6.2.1. *In the particular case where $\mathbf{X} = \mathbb{R}^p$ and d is a Euclidean distance, the problems of the equations 2.47 and 6.2 are equivalents.*

Proof. To prove theorem 6.2.1, we start with the definition of the barycenter notion given in Equation 6.1. Indeed, in the case where d is the Euclidean distance, we establish the equivalence between the centroid $\mathbf{v}_A$ of singleton clusters for $|A_j| = 1$ and meta-clusters for $|A_j| > 1$, defined from this Equation $\mathbf{v}_A = \text{argmin}_{\mathbf{x}\in\mathbf{X}} \sum_{\omega_k \in A} d(\mathbf{v}_{\{\omega_k\}}, \mathbf{x})$ (minimization of inertia). $\mathbf{v}_A(A \subseteq \Omega)$ is a set of centroids for the set of clusters (singleton and meta-clusters)

We define the mapping $\bar{\cdot} : \mathbb{R}^{p\times c} \mapsto \mathbb{R}^{p\times 2^c}$, which constructs a set of centroids for the set of clusters from the singleton clusters centers. In inverse, $\underline{\cdot} : \mathbb{R}^{p\times 2^c} \mapsto \mathbb{R}^{p\times c}$, selects the centroids of singleton clusters from the set of centroids. Then, for any $\boldsymbol{M}$ and $\boldsymbol{\mathcal{V}}$, we have:

$$\begin{aligned} J_{ECM}(\boldsymbol{M}, \boldsymbol{\mathcal{V}}) &= J^{\alpha,\beta,\delta}\left(\boldsymbol{M}, \bar{\boldsymbol{\mathcal{V}}}\right), \\ J^{\alpha,\beta,\delta}\left(\boldsymbol{M}, \boldsymbol{\mathcal{V}}\right) &= J_{ECM}\left(\boldsymbol{M}, \underline{\boldsymbol{\mathcal{V}}}\right). \end{aligned} \tag{6.3}$$

This expresses that if the parameters of J_{ECM} consider only the centroids of singleton clusters, the objective function includes the centroids of both singleton clusters and meta-clusters (when computed using the Euclidean distance). We can then clearly see that $\underline{\bar{\mathcal{V}}} = \mathcal{V}$. With these definitions, we can now prove the theorem by proceeding by contradiction. Suppose $\boldsymbol{M}^*$ and $\boldsymbol{\mathcal{V}}^*$ are the optimal solution of $J^{\alpha,\beta,\delta}(\boldsymbol{M}, \boldsymbol{\mathcal{V}})$ in Equation 6.2, and there exists a better solution than $(\boldsymbol{M}^*, \underline{\boldsymbol{\mathcal{V}}^*})$ to minimize J_{ECM}. We denote $(\boldsymbol{M}', \boldsymbol{\mathcal{V}}')$ this solution such that :

$$J_{ECM}\left(\boldsymbol{M}', \boldsymbol{\mathcal{V}}'\right) < J_{ECM}\left(\boldsymbol{M}^*, \underline{\boldsymbol{\mathcal{V}}^*}\right). \tag{6.4}$$

Using the equalities in (6.3), we have:

$$J^{\alpha,\beta,\delta}\left(\boldsymbol{M}', \bar{\boldsymbol{\mathcal{V}}}'\right) < J^{\alpha,\beta,\delta}\left(\boldsymbol{M}^*, \underline{\bar{\boldsymbol{\mathcal{V}}}^*}\right), \tag{6.5}$$

and thus:

$$J^{\alpha,\beta,\delta}\left(\boldsymbol{M}', \bar{\boldsymbol{\mathcal{V}}}'\right) < J^{\alpha,\beta,\delta}\left(\boldsymbol{M}^*, \boldsymbol{\mathcal{V}}^*\right). \tag{6.6}$$

Moreover, $\left(\boldsymbol{M}', \bar{\boldsymbol{\mathcal{V}}}'\right)$ satisfies the constraints of Equation 6.2. The mass constraints are respected as $(\boldsymbol{M}', \boldsymbol{\mathcal{V}}')$ was a solution of Equation 2.47, which includes these constraints. Additionally, by construction of $\bar{\mathcal{V}}'$, the added elements are barycenters of pure clusters, minimizing inertia under Euclidean distances. Thus, $\left(\boldsymbol{M}', \bar{\boldsymbol{\mathcal{V}}}'\right)$ is a better solution than the optimal solution for Soft-ECM, which is a contradiction. Therefore, no solution better than $(\boldsymbol{M}^*, \underline{\boldsymbol{\mathcal{V}}^*})$ exists for J_{ECM}.

6.3 Optimization scheme for Soft-ECM

The Soft-ECM problem is a bi-level optimization problem [243], i.e. an optimization problem in which parameters are themselves defined by another optimization problem. We propose a heuristic that relaxes the hard constraint of identifying the meta-clusters barycenters into an additional numerical constraint in the global minimization function. The relaxed minimization problem of Soft-ECM then becomes:

$$\begin{aligned}\min_{\boldsymbol{M},\boldsymbol{\mathcal{V}}} & \sum_{i=1}^{n}\sum_{A\subseteq\Omega\setminus\emptyset}|A|^{\alpha}m_i(A)^{\beta}d(\mathbf{x}_i,\mathbf{v}_A)+\sum_{i=1}^{n}\delta^2 m_i(\emptyset)^{\beta}\\ & +\lambda\sum_{A\subseteq\Omega\setminus\emptyset}\sum_{\omega_k\in A}d(\mathbf{v}_{\{\omega_k\}},\mathbf{v}_A),\\ \text{subject to } & \sum_{A\subseteq\Omega}m_i(A)=1,\quad \forall i\in\{1,\dots,n\}.\end{aligned} \tag{6.7}$$

While the first and second terms are similar to those of ECM, the third term imposes the additional constraint of consistency between the barycenters of the $\{\omega_k\}$ clusters and those of meta-clusters. The new hyperparameter $\lambda \geq 0$ adjusts the importance given to this constraint. The higher λ, the more the solution respects the inter-clusters positioning constraint. We can use an alternating optimization scheme to solve the relaxed Soft-ECM optimization problem (see Algorithm 11), with the following two steps alternating (until convergence or a fixed number of iterations):

1) Optimization of $\boldsymbol{M}$, with $\boldsymbol{\mathcal{V}}$ fixed. The credal partition $\boldsymbol{M}$, which minimizes the objective function, is computed using the expression below:

$$\begin{aligned}m_i(A) &= \frac{|A|^{-\frac{\alpha}{\beta-1}}d\left(\mathbf{x}_i,\mathbf{v}_A\right)^{-\frac{1}{\beta-1}}}{\sum_{B\neq\emptyset}|B|^{-\frac{\alpha}{\beta-1}}d\left(\mathbf{x}_i,\mathbf{v}_B\right)^{-\frac{1}{\beta-1}}+\delta^{-\frac{2}{\beta-1}}},\\ m_i(\emptyset) &= 1-\sum_{A\neq\emptyset}m_i(A).\end{aligned} \tag{6.8}$$

This solution was proposed for ECM [1] and is still valid for Soft-ECM. Indeed, the term added to the objective function does not depend on $\boldsymbol{M}$, so the optimization with respect to $\boldsymbol{M}$ of the problem in eq. 6.7 is the same as the problem in eq. 2.47.

2) Optimization of $\boldsymbol{\mathcal{V}}$, with M fixed. The optimization of eq. 6.7 when $\boldsymbol{M}$ is fixed is equivalent to the following unconstrained optimization problem:

$$\underset{\mathcal{V}}{\operatorname{argmin}}\sum_{A\subseteq\Omega\setminus\emptyset}\left(\sum_{i=1}^{n}|A|^{\alpha}m_i(A)^{\beta}d(\mathbf{x}_i,\mathbf{v}_A)+\lambda\sum_{\omega_k\in A}d(\mathbf{v}_{\{\omega_k\}},\mathbf{v}_A)\right). \tag{6.9}$$

We adopt a different approach from ECM and use a numerical method (gradient descent) to find an approximate solution. The use of optimization tools based on *automatic differentiation* allows to solve this problem efficiently. We note that the optimization of this function requires the differentiability of d.

Soft-ECM converges after a finite number of iterations as $J^{\alpha,\beta,\delta,\lambda}$ decreases at each step and is a positive function. Indeed, updating the credal partition $\mathbf{M}$ with Lagrange multipliers does not increase the objective function $J^{\alpha,\beta,\delta,\lambda}$ [1]. Regarding the optimization with respect to $\boldsymbol{\mathcal{V}}$, the gradient descent ensure to not increase the objective function.

Algorithm 11 Soft-ECM Algorithm

Input: $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$: n objects in $\mathbb{R}^{n \times p}$.
Parameters:
c: number of clusters
α: weighting for cardinality
$\beta > 1$: weighting for belief
$\delta_l > 0$: threshold for outliers
$\epsilon, \xi > 0$: convergence thresholds
ρ: learning rate

Initialization:
$\mathbf{v}^0$: Initialize c prototypes randomly
$\mathbf{v}_A^0$: Compute the mean of $\mathbf{v}^0$ for $|A| > 1$

$t \leftarrow 0$
repeat
  (1) Compute the credal partition matrix $\mathbf{M}$
  (2) Update the prototype vector:
  $\theta \leftarrow 0$
  repeat
    $\mathcal{V}^{\theta+1} = \mathcal{V}^{\theta} - \rho \nabla_{\mathcal{V}} J^{\alpha,\beta,\delta,\lambda}(\mathbf{M}, \mathcal{V}^{\theta})$
    $\theta \leftarrow \theta + 1$
  until $\|\mathcal{V}^{\theta+1} - \mathcal{V}^{\theta}\| < \xi$;
  $t \leftarrow t + 1$
until $\|\mathbf{M}^t - \mathbf{M}^{t-1}\| < \epsilon$;

Output: Optimal solutions $\mathbf{M}$ and $\mathcal{V}$.

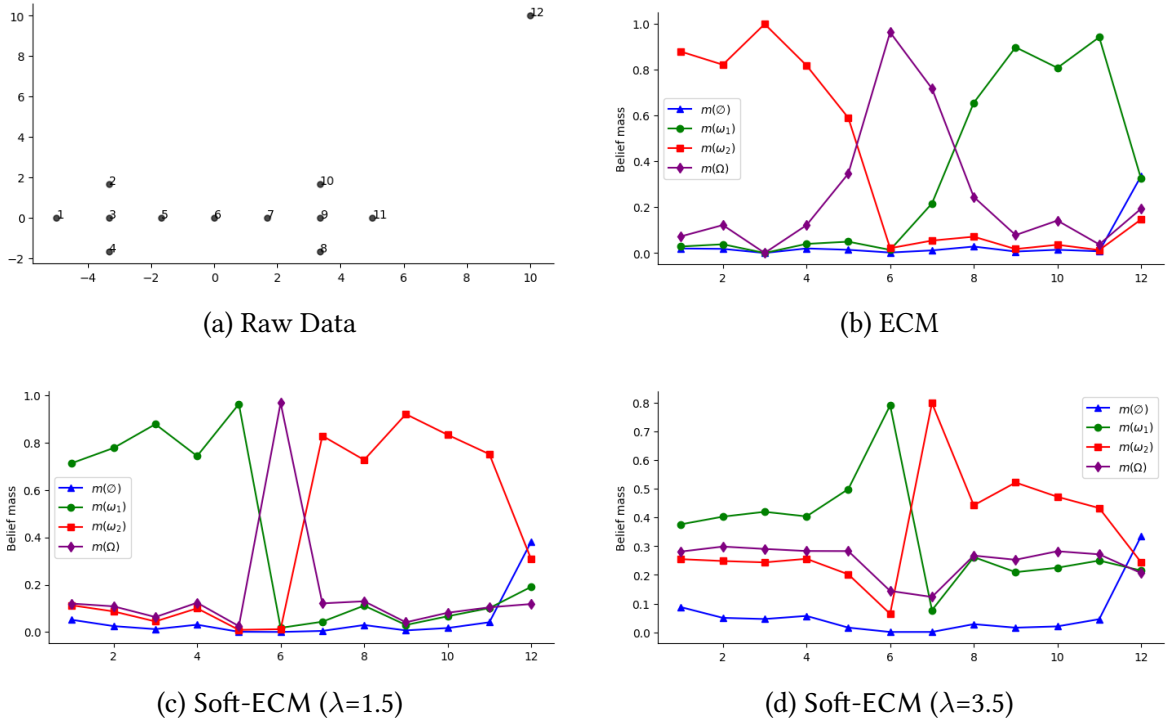
6.4 Experiments

In this section, we present the results of several experiments. First, we demonstrate the ability of Soft-ECM to perform clustering comparable to ECM on a classic toy problem. Next, we compare various fuzzy clustering methods with our approach using real-world datasets of different types, including tabular data (numerical and categorical) and time series. Finally, we showcase the clustering of synthetic time series to illustrate how our approach addresses the challenges of fuzzy clustering on complex data with a semi-metric.

6.4.1 Experiments with synthetic data

We compare the behavior of Soft-ECM and ECM on a classic dataset, *Diamond* [1]. It consists of twelve objects (see Figure 6.2a), where objects 1 to 11 being normal, and object 12 is an

outlier. Soft-ECM and ECM were run with the same parameters: $\alpha = \frac{1}{6}$, $\beta = 2$, $\delta = 11$ and, for Soft-ECM, $\lambda = 1.5$. The data were partitioned into $c = 2$ clusters. Figures 6.2b and 6.2c depict the 2^c cluster masses (focal elements) – $m(\omega_1)$, $m(\omega_2)$, $m(\Omega)$, and $m(\emptyset)$ – for ECM and Soft-ECM, respectively.



(a) Raw Data

(b) ECM

(c) Soft-ECM (λ=1.5)

(d) Soft-ECM (λ=3.5)

Figure 6.2: Illustration of the Diamon dataset clustering [1]. (a) illustration of raw data. (b), (c) and (d) mass distributions for each example for ECM and Soft-ECM. A high mass indicates cluster or meta-cluster membership.

Objects 1 to 5 and 8 to 11 are assigned to two distinct clusters by ECM and Soft-ECM, while object 12 is identified as an outlier. For objects 6 and 7, ECM assigns most of the mass to the ignorance (meta-cluster) Ω, as they are located near the centroid of Ω, which is defined as the average of the centroids of ω_1 and ω_2 (see Figure 6.2b). In contrast, Soft-ECM optimizes this centroid more effectively, with most belief mass assigned to ω_2 for object 7 (cf. Figure 6.2c). When we increase λ, until 3.5 with this dataset, it further constrains inter-cluster positioning. As shown in Figure 6.2d, object 6 was then classified into the ω_1, consistent with the results obtained by the authors in (BCM) [101] and aligned the raw data.

6.4.2 Experiments and comparisons with real data

6.4.2.1 Experimental setting

We evaluate here the performance of the Soft-ECM algorithm against ECM and its state-of-the-art variants, using nine publicly available real-world datasets.[3]. The datasets are of different types (numerical, categorical and time series). The characteristics of the datasets are summarized in Table 6.1.

[3] https://archive.ics.uci.edu

Data	n	p	c	Length	Metric(s)	Inputs
Abalone	4177	8	3	-	Euclidean	Numerical
Ecoli	336	7	8	-	Euclidean	Numerical
Glass	214	9	6	-	Euclidean	Numerical
Breast Cancer (BC)	286	9	2	-	Hamming	Categorical
Lung	32	56	3	-	Hamming	Categorical
Soybean	47	35	4	-	Hamming	Categorical
ERing	30	4	6	65	Euclidean, Soft-DTW	Time series
BasicMotions (BM)	40	6	4	100	Euclidean, Soft-DTW	Time series
AtrialFibrillation (AF)	15	2	3	640	Euclidean, Soft-DTW	Time series

Table 6.1: Experimental dataset characteristics, n: number of objects, p: number of dimensions and c: number of clusters.

The algorithm Soft-ECM was implemented in Python and based on the PyTorch [244] framework for solving eq. 6.9.[4] The benchmark algorithms used to evaluate the performance of our algorithm include ECM for numerical data, Evidential C-Means for categorical data (CatECM [105]), and Evidential C-Medoids (ECMdd [62]) for the relational format of data. Additionally, we compare our results with a hard clustering approach, specifically K-means and its variants K-modes and TimeSeriesKMeans (TSKmeans [245]).

Each algorithm was executed 10 times for each of the following parameter settings: $\alpha = 2.0$, $\delta = 10$, $\beta \in \{1.1, 1.2, \ldots, 2.0\}$, $\lambda \in \{1.0, 2.0, \ldots, 10.0\}$. Each run was was stopped upon convergence to a stationary value with $\epsilon = 10^{-3}$. The number of subsets for the meta-clusters was limited to 2 for results interpretability and computational reasons. For each dataset, the number of clusters corresponds to the prior number of clusters. For credal partitions, the pignistic [1] transformation was applied to convert them into hard partitions. Euclidean, Soft-DTW and Hamming metrics were used for clustering numerical, temporal and categorical data respectively (see Table 6.1).

6.4.2.2 Analysis of λ parameter

In this section, we examine the sensitivity of Soft-ECM results to the new parameter λ on each dataset. We use the average normalized specificity as the internal validity index of a credal partition, as defined by (6.10), following the approach proposed in [1], where $0 \leq N^{\star}(c) \leq 1$. We computed $N^{\star}$ for several combinations of λ and β values for each real-world dataset.

$$N^{\star}(c) = \frac{1}{n \log_2(c)} \sum_{i=1}^{n} \left[\sum_{A \in 2^{\Omega} \setminus \emptyset} m_i(A) \log_2 |A| \right]. \tag{6.10}$$

Figure 6.3 illustrates the performance of Soft-ECM. We can observe that the values of λ and β have a significant impact. Selecting appropriate values for λ on each dataset improves the performance of Soft-ECM. These first results show no evident trend in the evolution of RI and, at this stage, we recommend a tuning of the λ parameter for each new dataset.

[4] Code available: https://gitlab.inria.fr/tguyet/ecm-ts.

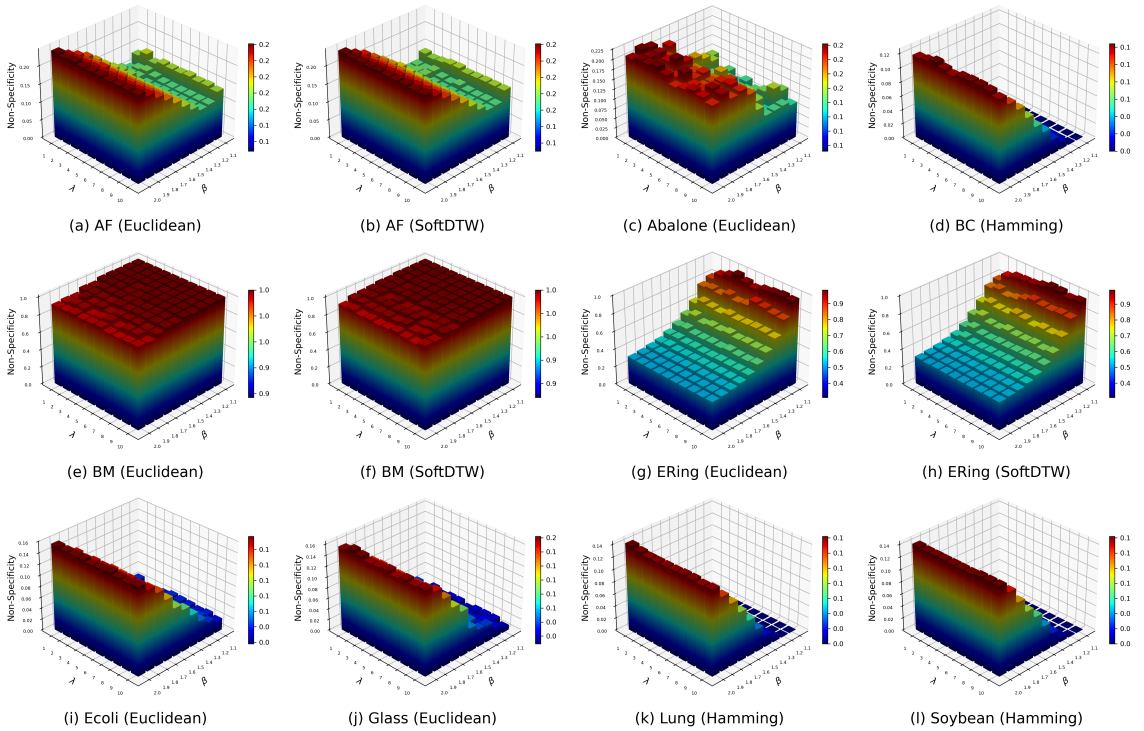


Figure 6.3: Parameter sensitivity β and λ on the performance of the Soft-ECM (N^*) algorithm.

6.4.2.3 Comparison with other algorithms on real-world datasets

Two standard performance measures are computed on the partitions: accuracy and Rand Index (RI) [162]. For each dataset, the parameters β and λ that minimize the non-specificity ($N^\star$) are selected for the credal partitions. Next, we compute the mean and standard deviation (σ) of the performance measures across the 10 executions. Table 6.2 shows the performances obtained for each algorithm on each dataset. The accuracy is calculated using a prior known class of data. Indeed, for each algorithm and each dataset, we calculated the mean and standard deviation statistics in mean(σ) format in Table 6.2, for the 10 executions (according to the parameters defined in Section 6.4.2.1 and those selected in Section 6.4.2.2).

The results show that Soft-ECM performs well across different data types, both in terms of RI and accuracy. For numerical data, Soft-ECM achieves competitive results, often close to or better than *ECM* and *K-Means*, with an average RI of 56.6% and accuracy of 51.6%. However, it shows slightly higher variability in performance, according to σ. On categorical data, Soft-ECM outperforms *K-Modes* and *CatECM* in many cases, though its performance varies across iterations, with higher σ values. For time series datasets, where both Euclidean and Soft-DTW metrics are used, Soft-ECM demonstrates competitive performance, often surpassing *TSKmeans*, especially with Soft-DTW metrics on Accuracy. Overall, Soft-ECM performs well in about 83% of cases across different data types, and proves its versatility across multiple data types and metrics.

In conclusion, this experimentation confirms that Soft-ECM achieves comparable or superior performance to the benchmark algorithms used in this comparative experimentation. This is beyond the result of the Proposition 6.2.1, demonstrating the usefulness of a flexible problem

	ECM	ECMdd	CatECM	Soft-ECM	K-Means	K-Modes	TSKmeans
				Rand Index			
Abalone	**0.580 (0.00)**	0.578 (0.023)	-	0.566 (0.031)	0.554 (0.016)	-	-
Ecoli	0.886 (0.004)	0.826 (0.052)	-	0.807 (0.060)	**0.892 (0.017)**	-	-
Glass	0.637 (0.046)	0.630 (0.057)	-	**0.665 (0.005)**	0.641 (0.027)	-	-
BC	-	0.581 (0.000)	0.581 (0.000)	**0.585 (0.008)**	-	0.581 (0.000)	-
Lung	-	0.523 (0.061)	0.550 (0.058)	0.482 (0.122)	-	**0.597 (0.000)**	-
Soybean	-	0.790 (0.105)	0.905 (0.061)	**0.905 (0.018)**	-	0.905 (0.120)	-
ERing (Eucl.)	-	0.846 (0.037)	-	0.784 (0.040)	-	-	**0.914 (0.024)**
BM (Eucl.)	-	0.477 (0.103)	-	**0.573 (0.086)**	-	-	0.426 (0.051)
AF (Eucl.)	-	0.418 (0.044)	-	0.496 (0.071)	-	-	**0.402 (0.048)**
ERing (SoftDTW)	-	**0.904 (0.026)**	-	0.760 (0.017)	-	-	0.862 (0.030)
BM (SoftDTW)	-	0.564 (0.048)	-	**0.586 (0.094)**	-	-	0.411 (0.044)
AF (SoftDTW)	-	0.468 (0.078)	-	**0.525 (0.093)**	-	-	0.399 (0.014)
				Accuracy			
Abalone	0.513 (0.000)	0.498 (0.013)	-	**0.516 (0.017)**	0.488 (0.026)	-	-
Ecoli	0.816 (0.002)	0.744 (0.048)	-	0.714 (0.067)	**0.821 (0.028)**	-	-
Glass	0.550 (0.040)	0.539 (0.054)	-	**0.585 (0.002)**	0.569 (0.021)	-	-
BC	-	0.703 (0.000)	0.703 (0.000)	**0.707 (0.009)**	-	0.703 (0.000)	-
Lung	-	0.497 (0.031)	0.494 (0.053)	0.469 (0.071)	-	**0.531 (0.000)**	-
Soybean	-	0.704 (0.103)	0.872 (0.095)	0.698 (0.122)	-	**1.000 (0.000)**	-
ERing (Eucl.)	-	0.667 (0.075)	-	0.557 (0.075)	-	-	**0.810 (0.052)**
BM (Eucl.)	-	**0.474 (0.021)**	-	0.410 (0.050)	-	-	0.400 (0.024)
AF (Eucl.)	-	0.474 (0.021)	-	**0.513 (0.077)**	-	-	0.460 (0.038)
ERing (SoftDTW)	-	0.700 (0.067)	-	0.547 (0.032)	-	-	**0.797 (0.060)**
BM (SoftDTW)	-	0.468 (0.059)	-	**0.472 (0.069)**	-	-	0.395 (0.026)
AF (SoftDTW)	-	0.493 (0.056)	-	**0.503 (0.078)**	-	-	0.467 (0.000)

Table 6.2: Clustering accuracy (*Rand Index* and *Accuracy*) of algorithms on real datasets. Bold numbers indicate best results by dataset. - refers to inappropriate algorithms for a given dataset.

formulation. In addition, it offers the advantage of a better fuzzy representation in uncertain data contexts.

6.4.3 Experiments with time series data

Regarding time series datasets, a Euclidean metric and a semi-metric, the Dynamic Time Warping (DTW), are experimented. DTW is particularly useful for datasets where there is temporal flexibility between instances of the same class (including time series of different lengths). Since DTW is only a semi-metric, new approaches have been developed to adapt clustering algorithms to DTW [242] but to our knowledge, there is no method for performing credal partitioning of time series using DTW. In this section, we present the results of two experiments obtained on artificial time series analysis. They illustrate the relevance and ability of our algorithm to perform fuzzy partitioning of time series using DTW, or more specifically Soft-DTW [246], which is its differentiable version.

In a first experiment, we use classic synthetic time series, named *Cylinder-Bell-Funnel*, illustrated in figure 6.5. The dataset includes 50 items of each class. These items are grouped according to index: from 1 to 50, from 51 to 100 and from 101 to 150, to facilitate interpretation of the mass plots. To illustrate the flexibility of Soft-ECM, we applied it with

two dissimilarity measures: the L_2 metric and the Soft-DTW semi-metric. The results are shown in figure 6.4.
The mass plots illustrate the expected partitioning for this dataset with $c = 3$. When we use the L_2 metric, the algorithm fails to identify the correct clusters. Indeed, the examples are assigned equal masses for the three clusters, whatever the true behavior of the time series. In contrast, with Soft-DTW, each expected cluster is mainly assigned to a singleton cluster, representing a better clustering. This experiment demonstrates that our algorithm effectively addresses the problem of fuzzy partitioning of time series, exploiting the advantages of Soft-DTW.

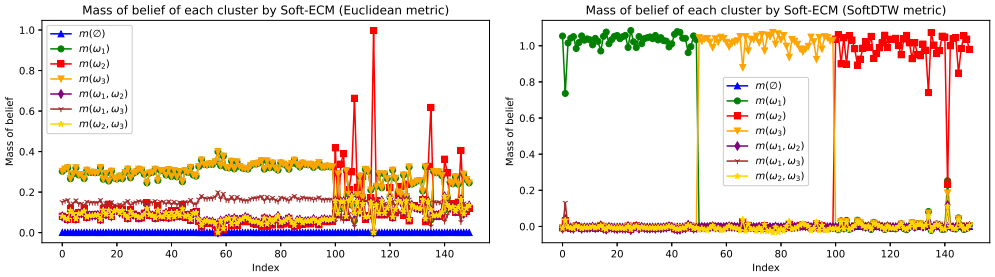


Figure 6.4: On top, plot of mass obtained using L_2 distance. On bottom, mass graph obtained by Soft-DTW.

In a second experiment, we aim to demonstrate the advantages of fuzzy clustering for these time series. The previous task can be easily addressed using the DBA algorithm, for example, but would not consider situations where time series could be fuzzily assigned to several classes because they mix two behaviors. In this experiment, we therefore used class *Bell*, class *Funnel* and a new class constructed by combining a series *Bell+Funnel* (M-shaped time series).

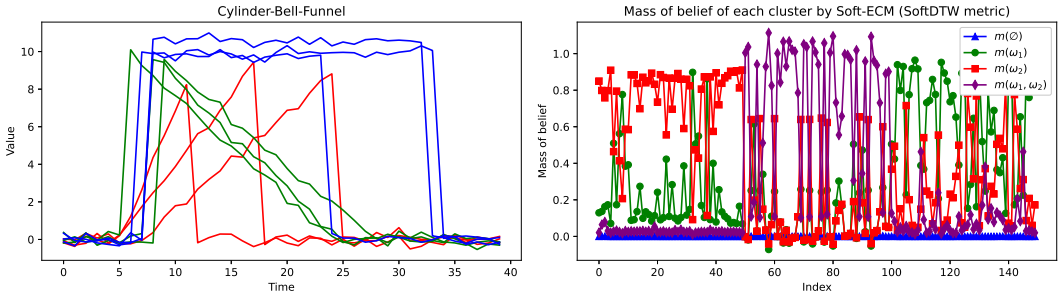


Figure 6.5: Illustration of the three-cluster raw-data on top: *Cylinder* (3 series in blue), *Bell* (3 series in green) and *Funnel* (3 series in red). On bottom, mass plot obtained by clustering with Soft-DTW for 2 classes (*Bell* and *Funnel*) and mixed examples (*Bell+Funnel*).

Figure 6.5 illustrates the results obtained by Soft-ECM clustering with $c = 2$ and a Soft-DTW. We can see that the algorithm identifies two classes (here, the *Bell* and *Bell+Funnel* classes), and the third class is identified as a mixture of the first two. For other initializations of the algorithm, the two clusters that are identified are sometimes *Bell* and *Funnel*. For this synthetic data, both results are acceptable, and of particular interest is to show the ability of the model to identify examples as mixtures of the other two. This demonstrates that the Soft-ECM algorithm is able to identify meaningful time series meta-clusters.

6.4.4 Case study: application to real-life care pathways

The objective of this case study is to identify typical care pathways for patients with chronic pain. The data are from the eDOL project, presented in Section 2.1.1 of chapter 2, which collects clinical, socio-demographic and barometric information from patients and their physicians concerning pain status. We are interested on barometers composed of eight attributes measured weekly, as detailed in Table 2.1, allowing for the evaluation of pain intensity and monitoring: pain, fatigue, mood, stress, sleep, body comfort, sports and non-sports activities. These dataset, as described in Section 2.1.3.2, are complex and subjective, and can be treated either as categorical time series (CTS), discrete time series (DTS), or continuous/numerical time series (CNTS). We worked on data from 636 patients, extracted from eDOL in June 2023, totaling 14,090 self-reported care events, with an average follow-up of 5 months and a total duration of approximately 19 months.

We execute Soft-ECM with soft-DTW as the similarity metric between pairs of trajectories, and in this case, the eDOL data have been interpreted as CNTS, since soft-DTW considers the time series as continuous. We carried out 10 trials, for each initialization value of the regularization parameter $\lambda \in \{.5, 1.0, 1.5, 2.0, \ldots, 5.0\}$. To make the results comparable to the different approaches and algorithms used in other contributions for clustering eDOL data, we performed Soft-ECM with the same parameters: $\alpha = 2$, $\beta = 1.5$, $\delta = 10$, and $c = 2$. Since the real classes are not known for the data sets, we used the non-specificity ($N^\star$) and the average silhoute index (ASW) to evaluate the performance of each result. Considering this criterias, the result giving the best partition corresponds to the highest ASW and lowest $N^\star$.
Figure 6.6 presents the average with error bars (std: standard deviation) of $N^\star$ and ASW on the 10 trials of each values of λ. We observed that, for the eDOL data set the proposed algorithm Soft-ECM performs, but has high variance. Best performance is obtained when $\lambda = 1.5$, and we selected the partition with the minimal objective function value for this value of λ with $ASW = 0.34$ and $N^\star = 0.37$.

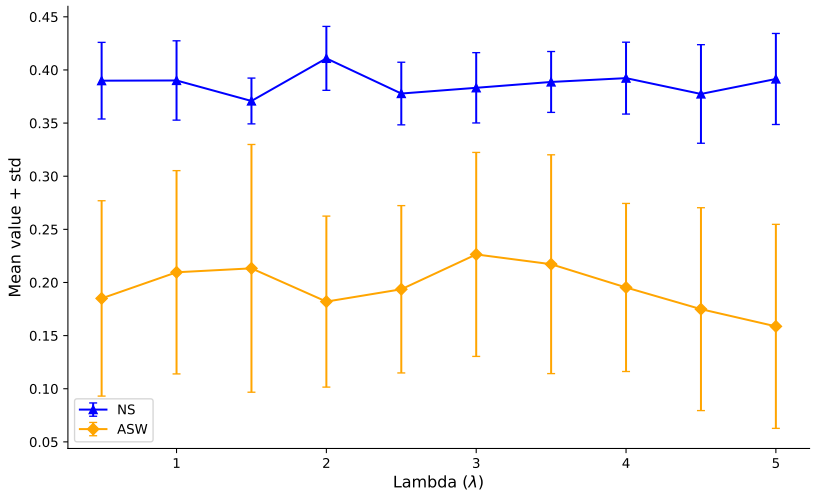


Figure 6.6: Average and std performance for ASW and N^* varying with λ of the Soft-ECM algorithm.

Hard Partition is derived through the selected best partiton **M** (membership belief) using Equation 2.46. We obtain two specific clusters (singletons), which we call Cl_1 and Cl_2, and an imprecise cluster $\{Cl\_1, Cl\_2\}$, intermediate between the two clusters, which we call Cl_1_2. The number of patients in each cluster is $Cl\_1 = 99$, $Cl\_2 = 95$ and $\{Cl\_1, Cl\_2\} = 442$. Figure 6.7 shows the trajectories of the centroids in each cluster, which can be analyzed as the average trajectories for each typology. For each centroid, we observe the curves of the 8 dimensions (sequences) monitored in eDOL. Unfortunately, Soft-ECM does not offer direct interpretability of patients' cluster membership across dimensions but other advanced tools, such as XAI explanations for multivariate time series, can also be used [247, 248].

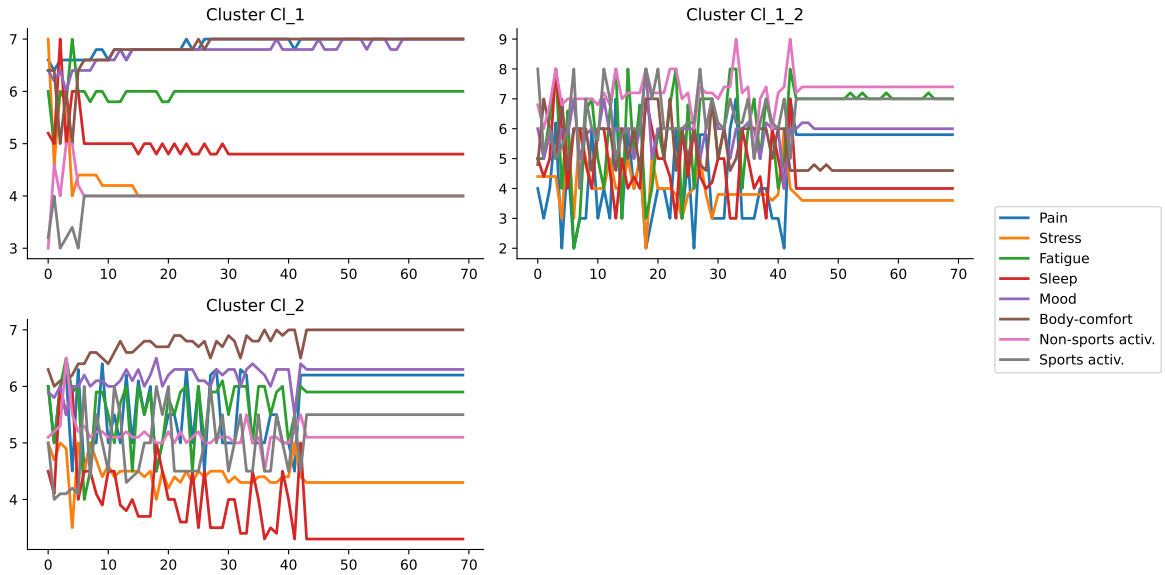


Figure 6.7: Visualization of the shape (trajectory) of the centroids of each cluster of the best partition obtained with Soft-ECM algorithm.

However, the analysis of the average total duration of each value ($[0-10]$) in each dimension allows to obtain the explained average of each dimension to the cluster membership of the objects. Combined with the analysis of Figure 6.8, which shows the transversal frequency distributions of the values at each time point, we can conclude that cluster 1 groups patients whose chronic pain is related to sleep and mood, while the chronic pain of patients in cluster 2 is determined by stress and fatigue. Indeed, over the first 18 weeks of follow-up, we observe that the trajectory of patients in cluster 1 is characterized by a moderate level of sleep and mood, which improves from the 12^{th} week onwards. For cluster 2, although stress levels improve, patients' state of fatigue declines from a high to a moderate level. These results are generally comparable to the observations obtained in the previous chapters (chapter 4 and chapter 5).

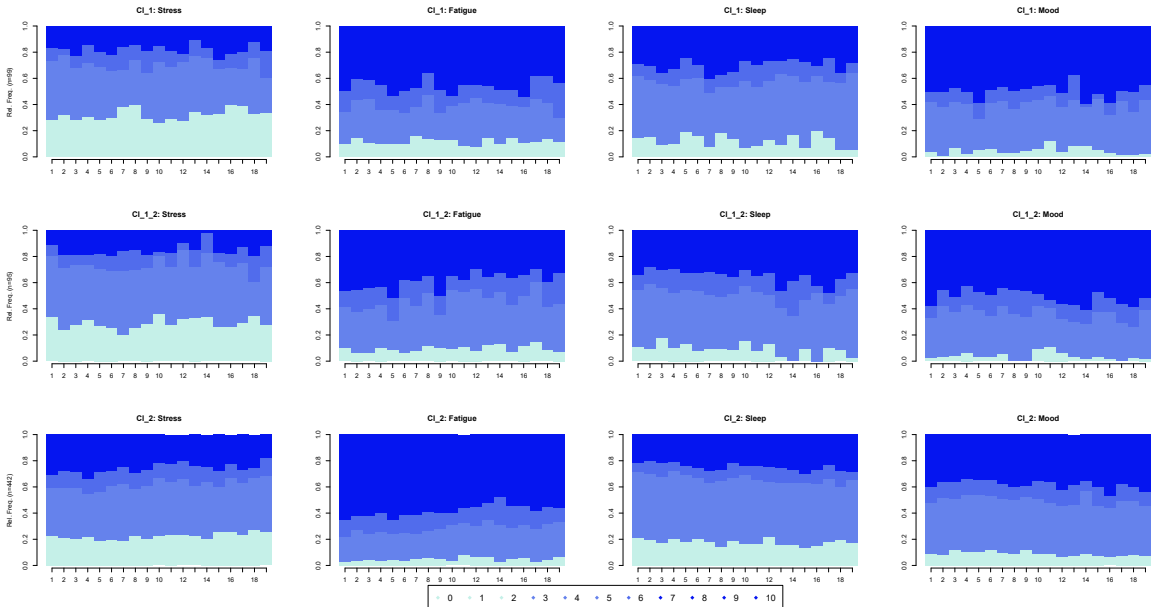


Figure 6.8: Sequence frequency graph illustrating the most frequent measured values, each represented by a transversal stacked bar.

6.5 Conclusion

In this contribution, we have proposed a generalization of the Evidential C-Means algorithm, to Soft Evidential C-Means, named Soft-ECM, to provide a evidential clustering solution adapted to complex data and not necessarily defined in a Euclidean space. Experimental results show that Soft-ECM is able to produce comparable or even superior performance to benchmark algorithms.

Furthermore, we have shown the capability of Soft-ECM to effectively achieve a fuzzy clustering with new types of data and compared a semi-metric, such as time series with a Dynamic Time Warping. In addition, a major advantage for uncertain data analysis is its ability to offer a more effective representation of fuzziness. Although this extension introduces new hyperparameters and is based on a numerical solution scheme, the flexibility provided by Soft-ECM in the use of a semi-metric and the improved computation of meta-cluster barycenters highlight its relevance for a wide scope of problems.

Chapter 7

evclust: Python library for evidential clustering

Abstract – A recent developing trend in clustering is the advancement of algorithms that not only identify clusters within data, but also express and capture the uncertainty of cluster membership. Evidential clustering addresses this by using the Dempster-Shafer theory of belief functions, a framework designed to manage and represent uncertainty. This approach results in a credal partition, a structured set of mass functions that quantify the uncertain assignment of each object to potential groups. The Python framework evclust, presented in this chapter, offers a suite of efficient evidence clustering algorithms as well as tools for visualizing, evaluating and analyzing credal partitions.

Contents

7.1 Introduction

Clustering is a fundamental task in data analysis, grouping objects based on similarity. Traditional methods like k-means [41] are efficient but assign each object to a single cluster, which can be limiting in applications where uncertainty or overlap between clusters must be accounted. Soft clustering methods allow partial memberships to multiple clusters, addressing this limitation. Fuzzy clustering, for example, computes degrees of membership for each object in all clusters, as implemented in Python libraries such as `scikit-fuzzy` [249]. Possibilistic clustering [250], on the other hand, relaxes the sum constraint on membership degrees, allowing better handling of outliers. Rough clustering approaches [251] define lower and upper approximations for clusters, providing a nuanced view of cluster membership. Building on these approaches, evidential clustering is a more general framework that quantifies uncertainty using the Dempster-Shafer theory of belief functions [99]. Instead of assigning simple degrees of membership, it uses mass functions that distribute belief over subsets of clusters. This richer representation allows evidential clustering to capture complex uncertainty patterns, such as partial ignorance, where an object may belong to multiple clusters without a clear preference [99].

While R has tools like evclust [252] for evidential clustering, Python lacked a comprehensive library for this purpose. To fill this gap, we introduce evclust, a Python library that implements state-of-the-art evidential clustering algorithms. evclust provides a flexible and user-friendly interface for analyzing credal partitions, along with tools for visualization and evaluation. Based on the Python ecosystem, it integrates seamlessly with popular libraries such as `numpy`, `pandas`, `matplotlib`, `scipy`, and `scikit-learn`. See Table 7.1 for software metadata.

This chapter is organized as follows. Section **??** explains the core concepts behind evidential clustering. Section 7.2 focuses on the evclust software, outlining its structure and main features, including clustering algorithms, utility tools, and performance metrics. Section 7.3 demonstrates how evclust can be used in practice. Finally, Section 7.4 discusses experimental results and wraps up with the conclusion.

Table 7.1: Metadata for the evclust package.

Metadata	**Description**
Current code version	v0.2
Permanent link	`https://github.com/armelsoubeiga/evclust`
Legal code license	MIT License
Code versioning system	git
Software languages,	Python
Compilation requirements	`Python` > 3.8, `scipy` > 1.10 `scikit-learn` > 1.3.0, `numpy`, `pandas`
Operating systems	Cross-platform (Windows, macOS, Linux)
Documentation	Available on `https://evclust.readthedocs.io`
Support email	`armel.soubeiga@uca.fr`

7.2 Description and architecture of evclust software

7.2.1 Evidential Clustering Algorithms in evclust

Table 7.2 presents an overview of the evidential clustering algorithms implemented in evclust. Some of these algorithms require attribute data, whereas others can handle proximity data, i.e., a matrix of dissimilarities, or distances between objects. Each clustering algorithm integrated into evclust includes an example section in its documentation, which provides a practical illustration of how to use the algorithm. In evclust, for example, ECM is implemented in function $ecm()$.

```
ecm(x, c, g0 = NULL, type = 'full', pairs = NULL,
    Omega = TRUE, ntrials = 1, alpha = 1, beta = 2,
    delta = 10, epsi = 0.001, disp = TRUE)
```

This function requires at least two key inputs: the data matrix $\mathbf{X}$, which has dimensions $n \times p$ (n is the number of samples, and p is the number of attributes), and c, the number of clusters.

The $g0$ argument is optional and allows to provide an initial matrix of cluster prototypes. The $type$ parameter defines the focal sets: $'full'$ (All subsets of Ω), $'simple'$ (Only the empty set, singletons, and Ω), and $'pairs'$ (The empty set, singletons, Ω, and either all pair subsets) or only the specific pairs provided through the $pairs$ argument. If $Omega = False$, the set Ω is excluded from the focal sets.

Other parameters control aspects of the algorithm, like the number of trials ($ntrials$), the weighting coefficients ($alpha$, $beta$, $delta$), and the stopping criterion ($epsi$). One can display the progression of the algorithm by setting $disp = True$. For more details, use `help(ecm)`.

The other functions in Table 7.2 follow the same parameter-setting scheme, with some modifications depending on the algorithm, including or excluding specific parameters.

Table 7.2: Overview of clustering algorithms in evclust.

Method	Inputs	Complexity	Function
ECM – Evidential c-Means [1]	Attribute data	$\mathcal{O}(n2^c)$	`ecm()`
RECM – Relational Evidential c-Means [63]	Proximity data	$\mathcal{O}(nc^2 + cn^2)$	`recm()`
k-EVCLUS – k Evidential Clustering[67, 253]	Proximity data	$\mathcal{O}(n^2c^2)$	`kevclus()`
CatECM – Categorical Evidential c-Means[105]	Categorical Attribute data	$\mathcal{O}(n2^c)$	`catecm()`
EGMM – Evidential Gaussian Mixture Model[254]	Attribute data	$\mathcal{O}(n2^c)$	`egmm()`
BPEC – Belief Peak Evidential Clustering[68]	Attribute data	$\mathcal{O}(n^2 + n2^c)$	`bpec()`
ECMdd – Evidential c-Medoids [107]	Proximity data	$\mathcal{O}(cn^2 + n2^c)$	`ecmdd()`
MECM – Median Evidential c-Means[108]	Proximity data	$\mathcal{O}(n2^c)$	`mecm()`
WMVEC – Weighted Multi-View Evidential Clustering[226]	Multi-view Attribute data	$\mathcal{O}(nw2^c)$	`wmvec()`
WMVEC-FP – WMVEC with feature preference[226]	Multi-view Attribute data	$\mathcal{O}(nwp2^c)$	`wmvec_fp()`
MECMdd-RWG – Multi-veiw Evidential C-Medoids [g],[j]	Multi-view Proximity data	$\mathcal{O}(pn2^c + pcn^2)$	`mecmdd_rwg()`
MECMdd-RWL – Multi-veiw Evidential C-Medoids [g],[j]	Multi-view Proximity data	$\mathcal{O}(pn2^c + pcn^2)$	`mecmdd_rwl()`
CCM – Credal C-Means[102]	Attribute data	$\mathcal{O}(n2^c)$	`ccm()`

7.2.2 Utils Functions in evclust

The evclust Python library provides a set of powerful utility functions, see Table 7.3, designed to facilitate the exploration and analysis of credal partitions produced by evidential clustering algorithms. These functions allow users to summarize, transform, and visualize credal partitions, making them highly versatile for clustering tasks in various domains.

Credal partitions, which are more general than traditional clustering outputs, can be converted into classical structures such as hard, fuzzy, or rough partitions [98]. This transformation is achieved using functions like `extractMass()`, which computes key outputs such as the plausibility and belief distributions, lower and upper approximations of clusters, and other derived metrics. Visualization is another critical aspect supported by evclust. The library includes functions such as `ev_plot()` for representing clusters and approximations in a two-dimensional space, `ev_pcaplot()` for PCA-based cluster visualization, and `ev_tsplot()` for analyzing clustering results in time-series data. These tools provide an intuitive understanding of the results and help to interpret complex clustering structures. In addition to these core functionalities, evclust offers tools for creating and managing focal sets (`makeF()`), summarizing the properties of credal partitions (`ev_summary()`), and detecting potential outliers. These utilities not only enhance the usability of evidential clustering but also enable a deeper exploration of uncertainty and ambiguity in cluster memberships, making evclust a valuable resource for data scientists and researchers. Table 7.3 summarizes the main utility functions available in evclust.

The evclust library provides access to several real-world example datasets through its `datasets` module. These datasets, such as the Decathlon dataset from `load_decathlon()`, the well-known Iris dataset `load_iris()`, Protein data `load_protein()`, the fourclass dataset `load_fourclass()`, and the multi-view ProPt dataset `load_prop()`, are available for users to explore and apply clustering algorithms.

Table 7.3: Utility functions in evclust.

Function	Description
`makeF()`	Creation of a matrix of focal sets.
`ev_summary()`	Extracts basic information to summarize a credal partition.
`ev_plot()`	Generates plots of a credal partition.
`ev_pcaplot()`	Plots PCA results with cluster colors of a credal partition.
`extractMass()`	Computes different outputs from a credal partition.

7.2.3 Metrics Functions in evclust

The evclust library includes a set of metrics functions that facilitate the evaluation and comparison of credal partitions. These functions address key aspects such as uncertainty quantification, relational representations, and the comparison of clustering results, offering

versatile tools for evidential clustering analysis.

One of the core metrics implemented in evclust is the measure of *nonspecificity()*, which quantifies the imprecision of a credal partition. Introduced by Klir and Wierman [209], Non-Specificity accounts for the uncertainty inherent in mass functions, distinguishing between imprecision and conflict. This metric is particularly useful for analyzing the degree of uncertainty in credal partitions and has been employed to determine the number of clusters in evidential clustering [1]. To facilitate the comparison of credal partitions, evclust implements the `credalRI()` function, which computes the Rand Index (RI) adapted for evidential clustering. This function compares two credal partitions by first transforming them into their relational representations and then evaluating the level of agreement between them. The relational representation of a credal partition, computed using `pairwise_mass()`, is expressed as a set of three relational matrices: M_e, M_1, and M_0, corresponding to the masses assigned to the empty set, singleton sets, and their complements, respectively. The `credalRI()` function then uses these representations to calculate the RI while allowing the user to specify a weighting scheme via its `type` argument. Overall, these metrics functions provide robust tools for evaluating evidential clustering results, enabling users to assess the level of uncertainty, extract insights from relational structures, and compare multiple credal partitions effectively.

Table 7.4: Metrics functions in evclust.

Function	Description
`nonspecificity()`	Computes the non-specificity of a credal partition.
`credalRI()`	Computes the Rand Index to compare two credal partitions.
`pairwise_mass()`	Computes relational representations for credal partition.

7.3 Illustrations

In this section, we demonstrate the use of the main functions in the evclust library through the analysis of some datasets.

7.3.1 Example with Iris dataset

This section demonstrates the use of evclust for evidential clustering with the Iris dataset. The Iris dataset includes measurements of sepal length, sepal width, petal length, and petal width for 50 flowers from each of three iris species. Assuming uncertainty in species assignments and the possibility of overlapping species across clusters, we employ the `ecm` algorithm to analyze the data and account for these uncertainties.

```
# Import requirements
import evclust
from evclust.ecm import ecm
from evclust.datasets import load_iris
```

```
# Import test data
df = load_iris()
df = df.drop(['species'], axis = 1) # del label column
```

The ecm algorithm can be applied to these data by running function ecm():

```
# ECM with c=3
clus = ecm(x=df, c=3, beta = 2,  alpha=1, delta=10)
```

We can summary the output of the ecm model, to see Focal sets or Number of outliers

```
from evclust.utils import ev_summary
ev_summary(clus)
```

```
------ Credal partition ------
3 classes,
150 objects
Generated by ecm
Focal sets:
[[0. 0. 0.]
    [1. 0. 0.]
    [0. 1. 0.]
    [1. 1. 0.]
    [0. 0. 1.]
    [1. 0. 1.]
    [0. 1. 1.]
    [1. 1. 1.]]
Value of the criterion = 38.82
Nonspecificity = 0.22
Prototypes:
[[7.06131634 3.03675091 6.05972886 2.1474559 ]
    [4.96375502 3.3462016  1.49213248 0.24695422]
    [6.01335287 2.76720722 4.77762377 1.64225065]]
Number of outliers = 0.00
```

The output of the ev_summary function provides an overview of the credal partition obtained from the ecm clustering algorithm. In this case, three clusters were identified across 150 objects. The focal sets represent the subsets of clusters considered during the evidential clustering process, including single clusters and combinations of clusters. The prototypes indicate the centroids of the identified clusters in the feature space, showing the average characteristics of each cluster. The clustering criterion achieved a value of Value of the criterion = 38.82, indicating the fit of the model to the data, while the Nonspecificity = 0.22 reflects the degree of imprecision in the partitioning. No outliers were detected in this analysis, suggesting a robust clustering outcome.
Figure 7.1 shows a simple plot on the left and a plot on the axes of a PCA on the right.

```
from evclust.utils import  ev_plot, ev_pcaplot

ev_plot(x=clus, X=df)
ev_pcaplot(data= df, x=clus, normalize=False,
    splite=False, cex=8, cex_protos=5)
```

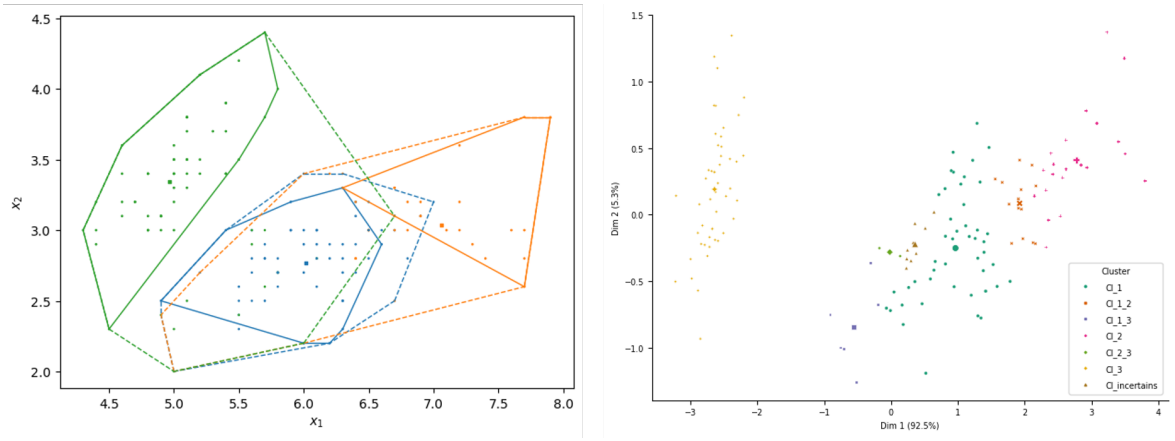


Figure 7.1: Illustration of visualization of credal partition using `ev_plot` and `ev_pcaplot` functions.

7.3.2 Example with multi-view dataset

We illustrate the example of multi-view evidential clustering with the WMVEC algorithm using the `wmvec` function from evclust. The function is configured to learn $c = 4$ clusters and for `type="simple"` means that we are only interested in the focal set equal to 2.

```
#Import of modules
from evclust.wmvec import wmvec
from evclust.datasets import load_prop

# Multi-view data import
df = load_prop()

# wmvec clustering fit
clus = wmvec(X=df, c=4, alpha=2, delta=5, maxit=20,
        epsi=1e-3, beta=1.1, lmbda=403, type='simple',
        disp=True)
```

The Prop dataset contains three attribute views with different dimensions. When imported using the `load_prop()` function, it returns a list containing these three views, each represented as a DataFrame. The importance weight of each view can be accessed through `clus['param']['R']`. This allows users to analyze the different attribute views and their corresponding weights in clustering tasks.

```
clus['param']['R']
```

```
[0.31174353, 0.34414724, 0.34410923]
```

7.4 Conclusion and Perspectives

Evidential clustering offers a novel framework for clustering, where uncertainty about cluster memberships is modeled using Dempster-Shafer mass functions. This approach provides a richer representation of imprecision and ambiguity compared to traditional

clustering methods, making it especially suitable for complex and uncertain datasets.

In this contribution, we introduced evclust, a Python library that implements a comprehensive set of evidential clustering algorithms. These algorithms are designed to address a wide range of clustering challenges, including handling non metric dissimilarity data, discovering clusters with complex shapes, and performing model-base clustering. Additionally, evclust provides tools for visualizing, evaluating, and exploiting credal partitions. The open-source nature of the library and its implementation in Python ensure broad accessibility and seamless integration into various workflows, benefiting from the widespread use of Python in the data science community. evclust caters to a diverse audience, including researchers, data analysts, students, programmers, and decision-makers, with applications extending beyond clustering to data analysis and decision-making under uncertainty.

We plan to expand evclust in several directions. Our road-map includes adding new algorithms to address emerging challenges in data science, such as Belief Shift Clustering (BSC), Dynamic evidential c-means clustering (DECM), Deep Evidential Clustering (DEC), Decision tree-based evidential clustering (DTEC), Transfer learning-based evidential c-means clustering (TECM), etc. We also aim to enhance the library's usability through simplified documentation and tutorial notebooks, making it more accessible for educational and practical purposes. Encouraging community contributions via platforms like GitHub will ensure continuous improvement and adaptation to the evolving needs of users. We aim to improve usability with simplified documentation and tutorials, fostering accessibility for education and practice, while encouraging community contributions on GitHub for ongoing development.

In conclusion, evclust is a powerful tool for evidence clustering and beyond, with significant potential for growth and impact in the data science community.

Part III

Conclusion

Discussion and perspectives

I Limitations and discussion

In this thesis, we have contributed to research on evidential clustering for trajectory analysis and, more specifically, to evidential clustering for sequential trajectory analysis in medicine and social science in general. We have addressed the challenges posed by the clustering of chronic pain care trajectories promoted by the eDOL project, whose data are collected by patient self-declarations of their health feelings via an mhealth application.

However, beyond the analytical complexity of the eDOL data presented at the beginning of our work in the Background chapter (chapter 2), certain limitations can also be highlighted regarding these data. Indeed, we can highlight a sampling bias mainly because requiring the use of a smartphone excludes patients who do not have or do not know how to use this tool. This could exclude the oldest or most precarious patients. The second limitation to highlight may be measurement bias, which frequently appears in observational studies [255]. In the eDOL study, measurement bias may have resulted from the patients themselves entering the data.

The contributions in this thesis provide complementary solutions to the complex problem of clustering of multidimensional trajectories, by exploiting the potential offered the belief functions. The aim was to propose robust evidential clustering methods adapted to discrete or categorical multidimensional trajectories, but the approaches developed can be extended to continuous or numerical temporal data. However, this work has some limitations.

Firstly, the feature-based approach introduced in chapter 4 is based on the extraction and selection of relevant features, a key step in transforming trajectories into data that can be exploited by traditional clustering algorithms. Although this method can yield clear, interpretable typologies, it depends greatly on the quality of the features extracted and selected. If inadequate or duplicate features are selected, this can introduce bias into the results and limit the effectiveness of the approach. In addition, this step introduces a significant dependency on extraction techniques, which may not be generalizable to all data, contexts and field.

To address these limitations, the multi-view evidential relational clustering algorithm (MECMdd) proposed in chapter 5 adopts a different perspective. By working directly on proximity data, this method overcomes the problem of extraction and selection of relevant features. However, MECMdd introduces other challenges. For example, the construction of proximity matrices from raw-data can be sensitive to the choice of dissimilarity metrics and

parameters, and still requires data transformation, which can have an impact on clustering quality.

Finally, in chapter 6, the Soft-ECM algorithm aims to overcome the limitations associated with raw data transformations, from feature extraction and selection and to the proximity matrix construction. By working directly on raw-data, Soft-ECM offers an adapted solution for processing complex data, including time series and mixed tabular data, without depending on the properties of Euclidean space. However, this flexibility comes with a cost, particularly in terms of the computational complexity associated with numerical optimization. The convergence of the algorithm and the evaluation of distances in a non-Euclidean space require significant computing resources, which can be problematic for large datasets.

The results obtained from the three contributions to clustering care trajectories for chronic pain in France, using data from the eDOL project, show very similar results but differing levels of robustness, interpretability and uncertainty management. The three approaches identify similar trajectory typologies, with two certain clusters related to distinct clinical profiles (**stress/fatigue vs. sleep/ mood**) and an uncertain intermediate cluster. However, the Soft-ECM method overcomes prior data transformations, providing better exploitation of temporal information. In contrast, approaches based on features or proximity matrices offer better interpretability, which is a weak point of Soft-ECM.

Table I.1: Overview of clustering methods and performance on eDOL.

Method	**Strengths**	**Limitations**	**Results on eDOL**	
			NS^1	ASW^2
Feature-based (ECM)	Yields interpretable typologies of trajectories.	Highly dependent on feature extraction and selection techniques, which may introduce bias. Requires data transformation. Based on Eucludian metric.	0.29	0.27
Raw data distance-based (MECMdd)	Yields features relevance and interpretable typologies of trajectories. Can use complex similarity metric, depending on trajectory characteristics.	Sensitive to the choice of dissimilarity metrics and parameters. Requires data transformation.	0.12	0.73
Raw data object-based (Soft-ECM)	Works directly on raw-data and preserves temporal information.	Involves high computational complexity. Object membership is not directly interpreatable.	0.37	0.34

[1]Performance using Non-specificity of the best partition.

[2]Performance using Average Silhouette Weight of the best partition.

Table I.1 summarizes the strengths and limitations of each method, as well as quantitative performance results regarding their applicability to clustering care trajectories from eDOL data. These three methods perform respectively with Euclidean, Optimal Matching (OM) and Soft-DTW similarity metrics. Despite quantitative differences in performance (NS and ASW), and differences in the distribution of patients within clusters (Table I.2), the similarity of results, in terms of the same number of clusters and subsets obtained for each method, highlights the reproducibility of these approaches.

Table I.2: Cluster sizes for the best partition obtained for each clustering method on eDOL.

Method	**Similarity Metric**	**Cluster sizes**		
		Cl_1	Cl_2	$\{Cl\_1, Cl\_2\}$
Feature-based (ECM)	Euclidean	216	157	291
Raw data distance-based (MECMdd)	OM	539	19	78
Raw data object-based (Soft-ECM)	Soft-DTW	99	95	442

II Perspectives and future research

Our research work opens up several perspectives, both for direct improvements of our contributions and for wider exploration of the field of evidential clustering and its applications to trajectory analysis.

A first immediately direction concerns the improvement of the Soft-ECM algorithm. We plan to integrate bi-level optimization schemes [256, 257], providing more efficient problem relaxation and enhancing the convergence properties of the algorithm. Furthermore, the implementation of feature or dimension weighting [221, 226, 232] could significantly improve the interpretability of results by highlighting the contribution of each attribute. In addition, our work can be directly improved using feature-based clustering of subsequences. Indeed, based on Subsequence time series clustering approaches [258], the objective is to extract features from interesting and relevant trajectory subsequences.

In a larger context, several research directions on this topic can be explored. Firstly, the development or use of non-linear dissimilarity measures adapted to MECMdd or Soft-ECM algorithms is an interesting direction for better capturing complex relationships in data, particularly in time series.
As one of the important steps of clustering, clustering validity is one of the important factors that determines the quality of results [259]. The evaluation and/or validation of credal partition and the Dempster-Shafer Theory in general are major challenge [94, 99]. While traditional clustering evaluation metrics have been widely studied, they may not be directly applicable to the specific challenges posed by evidential clustering. In credal partitions, each object can partially belong to several clusters, but also to sets of clusters, making the use of traditional measurement indices more complex. Extending existing hard or fuzzy internal measures to the evidential framework could provide a partial solution. Some

external evaluation measures are proposed from traditional clustering evaluation metrics to better suit the needs of evidential clustering [99], including evidential precision (EP), evidential recall (ER), evidential rank index (ERI), and Credal Rand Index (CRI). However, internal evaluation measures require prior research.
Finally, the application of the new Soft-ECM formulation to semi-supervised constrained learning problems could be an interesting perspective. For example, incorporating ideas from frameworks such as Constrained Evidential C-Means (CECM) [103] could enable a priori knowledge to be better integrated into the clustering process, improving both performance and interpretability.

General conclusion

This thesis contributes to research on evidential clustering of multidimensional trajectories, with a particular focus on the clustering and analysis of chronic pain care trajectories. Our work has allowed us to explore and introduce several clustering approaches and algorithms, addressing both theoretical and practical challenges related to trajectory data.

We reviewed existing clustering methods and, based on the identified challenges, proposed novel approaches—including feature-based evidential clustering, a multi-view relational evidential clustering algorithm (MECMdd) with adaptive weightings, the soft evidential clustering (Soft-ECM) for non-Euclidean data, and the evclust Python library—to effectively handle diverse and uncertain trajectory data. The results obtained from the application of these approaches to the clustering of chronic pain trajectories are discussed.

Overall, this thesis makes a significant contribution by advancing both the theoretical foundations and practical implementations of evidential clustering, thereby opening new perspectives for clustering multidimensional and uncertain trajectories in real-world applications.

Résumé étendu en français

Dans le chapitre 3, nous avons examiné et comparé les méthodes de classification automatique pour les trajectoires multidimensionnelles. Nous avons abordé les défis de regroupement des trajectoires multidimensionnelles caractérisées par des données longitudinales ou des séries temporelles discrètes ou catégorielles, mettant en évidence leurs applications et leurs différences. Les techniques de regroupement de trajectoires examinées incluent des approches basées sur les caractéristiques, les données brutes et les modèles. Une étude comparative a été menée sur les approches basées sur les données brutes en calculant les proximités ou relations (matrices de dissimilarité) à l'aide de l'analyse séquentielle et utilisant des algorithmes de regroupement dur et souple relationnels. Les méthodes additives se sont révélées les plus efficaces, malgré leur dépendance à des hypothèses majeures sur l'interdépendance des dimensions. Nous avons fourni une analyse critique des différences entre les approches étudiées et proposé l'exploration d'une nouvelle approche de regroupement de trajectoires multidimensionnelles dans un paradigme collaboratif.

Dans ce chapitre 4, nous avons exploré le regroupement basé sur les caractéristiques des données de trajectoires temporelles liées à la douleur chronique. En utilisant une approche de regroupement souple, nous avons cherché à découvrir des trajectoires de soins atypique ou uncertain. La méthode implique l'extraction et la sélection de caractéristiques clés à partir des données en utilisant des techniques avancées telles que Tsfresh et la sélection de caractéristiques non supervisée. Les caractéristiques extraites sont définies en suivant les recommandations du domaine médical. Au total, 28 caractéristiques uniques sont obtenues pour chaque série temporelle. Des algorithmes de sélection de caractéristiques, tels que la forêt aléatoire non supervisée (le plus performant), le score laplacien et la méthode spectrale non supervisée, ont été utilisés pour identifier les caractéristiques les plus pertinentes. Nous avons ensuite appliqué un algorithme de C-Moyennes évidentielle pour identifier les typologies de trajectoires de soins, en tenant compte de l'incertitude basée sur la subjectivité et la nature des données. Les résultats ont révélé deux clusters certains associés à l'inconfort et au bien-être, ainsi qu'un cluster incertain représentant des caractéristiques intermédiaires. Grâce à l'analyse descriptive, aux tests statistiques et à la régression multinomiale, nous avons identifié des profils de patients au sein de ces trajectoires, améliorant ainsi notre compréhension de la gestion de la douleur chronique.

Dans le chapitre 5 suivant, nous avons proposé un algorithme de regroupement relationnel multi-vue, particulièrement adapté aux données de proximité (ou relationnelles), qui sont souvent plus simples à collecter dans des applications du monde réel. Ces données permettent de gérer des objets caractérisés par des types de données complexes et variés. Cependant, la plupart des méthodes existantes de regroupement relationnel multi-vue

montrent des limites dans la gestion des ensembles de données incertaines et imprécises, en particulier pour des objets appartenant à des classes chevauchantes. Pour surmonter ces limitations, notre algorithme repose sur la théorie des fonctions de croyance, qui permet de caractériser partiellement les connaissances lors de l'attribution des objets aux classes. Ce nouvel algorithme, appelé Multi-View Evidential C-Medoid clustering with adaptive weightings (MECMdd), calcule des poids de pondération pour chaque vue dans un cadre collaboratif. Ces poids sont estimés localement pour chaque groupe, avec une contrainte de somme des poids (MECMdd-RWL-S) ou une contrainte de produit des poids (MECMdd-RWL-P), ainsi que globalement avec les mêmes types de contraintes (MECMdd-RWG-S et MECMdd-RWG-P). Nous avons évalué ces algorithmes à travers plusieurs expérimentations sur des ensembles de données réels, en comparant leurs performances et leurs avantages à ceux de méthodes similaires et de pointe décrites dans la littérature. De plus, nous les avons appliqués à l'étude de cas réelle de regroupement des trajectoires de la douleur chronique, démontrant leur efficacité et leur interprétabilité.

Le chapitre 6 aborde le défi du regroupement de données complexes basé sur les objets à l'aide des fonctions de croyance. Nous avons reformulé le problème de l'algorithme C-Moyennes évidentielle pour permettre la prise en charge des données non euclidiennes, telles que les données mixtes (numériques et catégorielles) et les données non tabulaires comme les séries temporelles et les données longitudinales. Ces types de données ne peuvent pas être traités par les algorithmes évidentielles traditionnels en raison de leur dépendance aux propriétés de l'espace euclidien. Le nouvel algorithme proposé, nommé Soft-ECM, est également capable de positionner les centroïdes de classes imprécises de manière cohérente en utilisant uniquement une semi-métrique, contrairement au C-Moyennes Évidentielles, qui utilise une moyenne arithmétique. Une semi-métrique est une mesure de dissimilarité qui est symétrique, positive et définie, mais qui ne satisfait pas nécessairement l'inégalité triangulaire. Nos expérimentations montrent que Soft-ECM obtient des résultats comparables à ceux des méthodes classiques de classification automatique évidentielle sur des données numériques, tout en étant également efficace pour les données mixtes. De plus, nous avons démontré ses avantages dans l'intégration de semi-métriques comme DTW pour les séries temporelles, illustrant ainsi sa polyvalence et son potentiel dans divers contextes de regroupement. Enfin, nous avons appliqué Soft-ECM à l'étude de cas de regroupement des trajectoires de la douleur chronique, mettant en évidence son efficacité.

Enfin, dans le chapitre 7, nous avons contribué à l'open source en développant la bibliothèque Python evclust, qui met à disposition plusieurs algorithmes de classification automatique basés sur les fonctions de croyance, parmi les plus avancés. Cette bibliothèque offre un environnement convivial de regroupement evidentiel tout en incluant des outils de visualisation et d'évaluation des resultats du regroupement.

Appendix and supplemental material

The appendix is organized as follows. Section A provides further definitions of some concepts used in the background chapter. In Section B.1, we present some more details of the synthetic data simulation procedure. In Section B.2, we outline related work in multi-view relational clustering and additional application results of Codist on eDOL dataset.

A Appendices for chapter 2

Model-Based Clustering – The basic idea is to select a particular model for each cluster and find the best fitting for that model. Model-based clustering algorithms presume data points are generated from a probabilistic model and seek to identify the most appropriate model to explain the data distribution. Diverse and well-developed models provide means to describe data adequately, and each model has its unique characteristics that may offer some notable benefits in some specific areas. However, overall, the time complexity of the models is relatively high, the premise is not entirely true, and the clustering result is dependent on the parameters of the models that are chosen.
Spectral-Based Clustering – Spectral clustering leverages the eigenvalues of similarity matrices derived from the data to perform dimensionality reduction before applying traditional clustering techniques. This method is particularly effective for datasets with non-convex clusters and complex structures. It excels in separating clusters that are not linearly separable in the input space. However, spectral clustering requires computationally expensive matrix operations and is sensitive to the choice of similarity measures, which can impact its robustness and scalability.
Density-Based Clustering – Density-based algorithms aim to identify clusters in high-dimensional regions of the feature space while also detecting noise points as outliers. These algorithms have high clustering efficiency, are sensitive to parameters, and are suitable for datasets of arbitrary shapes. In the case of uneven spatial data density, the quality of clustering results will decrease. Additionally, higher computing resources are required when processing large datasets.
Grid-Based Clustering – The fundamental principle of these clustering algorithms is to partition the initial data space into a grid structure of a predetermined size for clustering. While exhibiting low time complexity, high scalability, and compatibility with parallel processing and incremental updates, these algorithms do come with considerations. The clustering outcomes prove sensitive to the grid size, where the pursuit of heightened calculation efficiency may come at the expense of cluster quality and overall clustering accuracy.
Neural Network-Based Clustering – This approach combines traditional clustering methods with new deep-learning techniques to improve feature extraction and identify

complex patterns in data. These methods can learn complex data representations and handle high-dimensional datasets effectively. They offer the advantage of adaptability, as they can discover intricate patterns in data that traditional methods might miss. However, they require substantial computational resources, are sensitive to hyperparameter tuning, and often demand large amounts of data to achieve optimal performance.

B Appendices for chapter 3

B.1 Description of synthetic data simulation

The two datasets include: (1) two independent variables—a three-modality categorical variable and a four-modality ordered categorical variable; (2) four variables—two correlated ordered categorical variables with three and four modalities, and two independent categorical variables with three and four modalities. Each dataset comprises 150 points distributed across four classes of unequal sizes (50, 50, 15, and 35), simulated using multinomial distributions with category probabilities derived from a Dirichlet distribution with equal α parameters, ensuring equal likelihood for all modalities (see Table B.1).

To simulate longitudinal data, cross-sectional data are transformed into time-dependent data by incorporating variable observation times and intervals. The number of measurements per individual follows a non-zero Poisson distribution $nCount \sim \text{Pois}(\lambda = 6)$, while the average time interval between observations is gamma-distributed $mInterval \sim \Gamma(k = 30, \theta = 0.01)$, with a fixed variance $vInterval$.

Synthetic variable	Distributions	Mode of the variable	Class

Three-modality categorical variable: `var_c_3m`	0.36; 0.24; 0.4
	0.5; 0.23; 0.27
	0.43; 0.32; 0.25
	0.38; 0.32; 0.3
		Ibuprofen; Paracetamol; Aspirin	`cl_1`
			`cl_2`
			`cl_3`
			`cl_4`

Four-modality categorical variable: `var_c_4m`	0.28; 0.29; 0.18; 0.25
	0.24; 0.2; 0.37; 0.19
	0.25; 0.26; 0.34; 0.15
	0.28; 0.21; 0.28; 0.23
		Rhinitis; Hypertension; Migraine; Arthritis	`cl_1`
			`cl_2`
			`cl_3`
			`cl_4`

Four-mode ordinal categorical variable: var_ord_3m	0.3; 0.2; 0.2; 0.3
	0.2; 0.25; 0.25; 0.3
	0.3; 0.2; 0.24; 0.26
	0.28; 0.22; 0.27; 0.23
		NotClinic; ClinicNoTx; TxUncontrolled; TxControlled	`cl_1`
			`cl_2`
			`cl_3`
			`cl_4`

Tree-modality categorical variable: `var_c_4m`	Correlated with `var_ord_4m`	Mild; Moderate; Severe	

Table B.1: Configurations of categorical data sets: distribution of class modalities.

B.2 Related works on medoids-based relational multi-view clustering

In this work [125], F. de Carvalho and al. propose MRDCA, an extension of dynamic hard clustering for relational data, which uses multiple dissimilarity matrices. MRDCA has two variants, MRDCA-RWL and MRDCA-RWG, which can compute relevance weights for each dissimilarity matrix, respectively, locally for each cluster, or globally for the whole partition. However, MRDCA assumes a hard relationship between objects and clusters, which is unrealistic for real data that often contains uncertainty, imprecision, and inconsistency [229]. F. de Carvalho and al. [61] introduced multiple relational fuzzy c-medoids (MRFCMdd) based on FCMdd algorithm [64]. MRFCMdd is also designed to estimate weights in two ways: locally, i.e. a relevance weight for each matrix in each cluster (MRFCMdd-RWL), and globally, i.e. a global clustering weight for each matrix (MRDCA-RWG). In [g], Soubeiga, A. and al. proposed Evidential C-Medoid clustering with adaptive weightings (MECMdd) based on ECMdd [62], with two variants MECMdd-RWL and MECMdd-RWG. Numerous other non-Medoids multi-view relational clustering algorithms are available, illustrated in Figure B.1. For instance, in [60], Frigui and al. introduced the CARD algorithm based on fuzzy theory, which aggregates multiple dissimilarity matrices and learns a relevance weight for each matrix in each cluster.

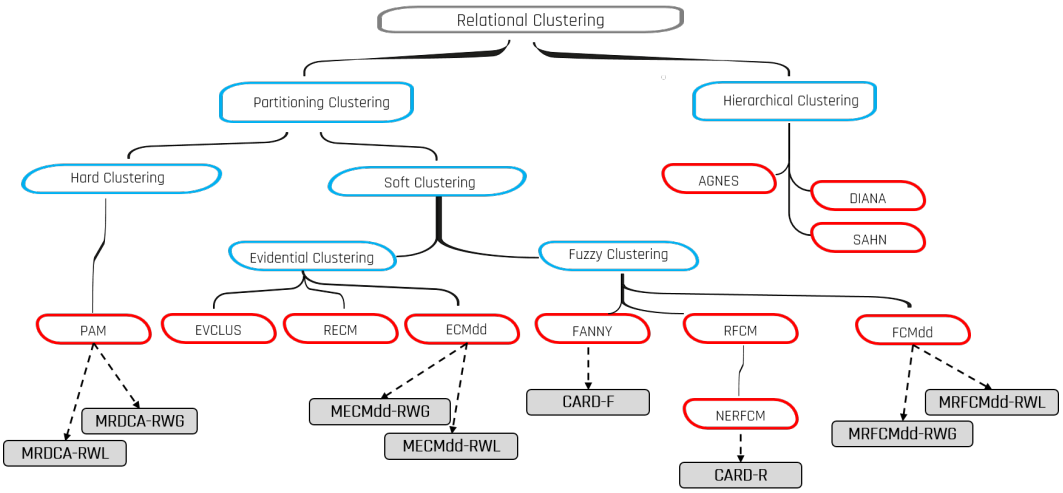


Figure B.1: Review of Multi-Relational clustering.

Bibliography

[1] Marie-Hélène Masson and Thierry Denœux. ECM: An evidential version of the fuzzy c-means algorithm. Pattern Recognition, 41(4):1384–1397, 2008.

[2] Moritz Herle, Nadia Micali, Mohamed Abdulkadir, Ruth Loos, Rachel Bryant-Waugh, Christopher Hübel, Cynthia M Bulik, and Bianca L De Stavola. Identifying typical trajectories in longitudinal data: modelling strategies and interpretations. European journal of epidemiology, 35(3):205–222, mar 2020.

[3] Nicolas Robette. The diversity of pathways to adulthood in france: Evidence from a holistic approach. Advances in Life Course Research, 15(2):89–96, 2010. Demographic Perspectives on the Transition to Adulthood.

[4] Richard A. Settersten and Karl Ulrich Mayer. The measurement of age, age structuring, and the life course. Annual Review of Sociology, 23(1):233–261, 1997.

[5] Matthias Studer and Gilbert Ritschard. What matters in differences between life trajectories: A comparative review of sequence dissimilarity measures. Journal of the Royal Statistical Society Series A: Statistics in Society, 179(2):481–511, 07 2015.

[6] Arnstein Aassve, Francesco C. Billari, and Raffaella Piccarreta. Strings of adulthood: A sequence analysis of young british women's work-family trajectories. European Journal of Population / Revue européenne de Démographie, 23:369–388, 10 2007.

[7] Jacques Antoine Gauthier, Eric D. Widmer, Philipp Bucher, and Cédric Notredame. Multichannel sequence analysis applied to social science data. http://dx.doi.org/10.1111/j.1467-9531.2010.01227.x, 40:1–38, 7 2010.

[8] Jana Klímová Chaloupková. The de-standardisation of early family trajectories in the czech republic: A cross-cohort comparison. Sociologický Časopis / Czech Sociological Review, 46:427–451, 01 2010.

[9] Kelsey M. Flint, Sarah J. Schmiege, Larry A. Allen, Timothy J. Fendler, John Rumsfeld, and David Bekelman. Health status trajectories among outpatients with heart failure. Journal of pain and symptom management, 53:224–231, 2 2017.

[10] Eleanor K. Seaton, Tiffany Yip, Antonio Morgan-Lopez, and Robert M. Sellers. Racial discrimination and racial socialization as predictors of african american adolescents' racial identity development using latent transition analysis. Developmental psychology, 48:448–458, 3 2012.

[11] Shuo Shang, Lisi Chen, Zhewei Wei, Christian Søndergaard Jensen, Kai Zheng, and Panos Kalnis. Trajectory similarity join in spatial networks. Proceedings of the VLDB Endowment, 10(11), 2017.

[12] Zachary Zimmer, Linda G. Martin, Bobby L. Jones, and Daniel S. Nagin. Examining late-life functional limitation trajectories and their associations with underlying onset, recovery, and mortality. Journals of Gerontology - Series B Psychological Sciences and Social Sciences, 69:275–286, 2014.

[13] Lei Chen, M. Tamer Ozsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories, 2005.

[14] Nicolas Robette, Xavier Bry, and Eva Leli'evre. A global interdependence approach to multidimensional sequence analysis. Sociological Methodology, 45(1):1–44, 2015.

[15] Gilbert Ritschard, Tim F. Liao, and Emanuela Struffolino. Strategies for multidomain sequence analysis in social research. Sociological Methodology, 53(2):288–322, 2023.

[16] Daniel S Nagin. Analyzing developmental trajectories: A semiparametric, group-based approach. Psychological Methods, 4:139–157, 06 1999.

[17] M. Hashem Pesaran. Time series and panel data econometrics. Oxford University Press, 10 2015.

[18] S. J. Kuramoto, A. S.B. Bohnert, and C. A. Latkin. Understanding subtypes of inner-city drug users with a latent class approach. Drug and alcohol dependence, 118:237–243, 11 2011.

[19] Matthew R. Lee, Laurie Chassin, and Ian K. Villalta. Maturing out of alcohol involvement: transitions in latent drinking statuses from late adolescence to adulthood. Development and psychopathology, 25:1137–1153, 11 2013.

[20] A. M. Elliott, B. H. Smith, K. I. Penny, W. C. Smith, and W. A. Chambers. The epidemiology of chronic pain in the community. Lancet, 354(9186):1248–1252, 1999.

[21] O. Van Hecke, N. Torrance, and B. H. Smith. Chronic pain epidemiology and its clinical relevance. British journal of anaesthesia, 111:13–18, 2013.

[22] D. J. Gaskin and P. Richard. The economic costs of pain in the united states. Journal of Pain, 13(8):715–724, 2012.

[23] L. J. Crofford. Adverse effects of chronic opioid therapy for chronic musculoskeletal pain. Nature Reviews Rheumatology, 6(4):191–197, 2010.

[24] J. S. Mogil. Animal models of pain: progress and challenges. Nature Reviews Neuroscience, 10(4):283–294, 2009.

[25] A. et al. Mouraux. Challenges and opportunities in translational pain research. European Journal of Pain, 25(4):731–756, 2021.

[26] D. et al. Bouhassira. Stratification of patients based on the neuropathic pain symptom inventory. Pain, 162(4):1038–1046, 2021.

[27] C. L. et al. Martin. The efficacy of mobile health interventions used to manage acute or chronic pain: a systematic review. Research in Nursing and Health, 44(1):111–128, 2021.

[28] N. Kerckhove, N. Delage, S. Cambier, N. Cantagrel, E. Serra, F. Marcaillou, C. Maindet, P. Picard, G. Martiné, R. Deleens, A.-P. Trouvin, L. Fourel, G. Espagne-Dubreuilh, L. Douay, S. Foulon, B. Dufraisse, C. Gov, E. Viel, F. Jedryka, S. Pouplin, C. Lestrade,

E. Combe, S. Perrot, D. Perocheau, V. De Brisson, P. Vergne-Salle, P. Mertens, B. Pereira, A. J. Djiberou Mahamadou, V. Antoine, A. Corteval, A. Eschalier, C. Dualé, N. Attal, and N. Authier. edol mhealth app and web platform for self-monitoring and medical follow-up of patients with chronic pain: Observational feasibility study. JMIR Form Res, 6(3):e30052, 03 2022.

[29] Laura Savaré, Francesca Ieva, Giovanni Corrao, and Antonio Lora. Mining and evaluation of patients' diagnostic therapeutic paths through state sequences analysis, 2022.

[30] Christos Chouaïd, Valentine Grumberg, Alexandre Batisse, Romain Corre, Matteo Giaj Levra, Anne-Françoise Gaudin, Martin Prodel, Joannie Lortet-Tieulent, Jean-Baptiste Assié, and Francois-Emery Cotté. Machine learning–based analysis of treatment sequences typology in advanced non–small-cell lung cancer long-term survivors treated with nivolumab. JCO Clinical Cancer Informatics, (6):e2100108, 2022. PMID: 35113656.

[31] Nolwenn Le Meur, Fei Gao, and Sahar Bayat. Mining care trajectories using health administrative information systems: The use of state sequence analysis to assess disparities in prenatal care consumption. BMC Health Services Research, 15:1–10, 12 2015.

[32] Nicolas Robette. L'analyse statistique des trajectoires: Typologies de séquences et autres approches. Ined Éditions, 1 2021.

[33] Hermine Lore Nguena Nguefack, M. Gabrielle Pagé, Joel Katz, Manon Choinière, Alain Vanasse, Marc Dorais, Oumar Mallé Samb, and Anaïs Lacasse. Trajectory modelling techniques useful to epidemiological research: A comparative narrative review of approaches. Clinical Epidemiology, 12:1205–1222, 2020.

[34] Alain Vanasse, Josiane Courteau, Mireille Courteau, Marc-André Roy, Emmanuel Stip, Marie-Josée Fleury, Alain Lesage, and Sébastien Brodeur. Multidimensional analysis of adult patients' care trajectories before a first diagnosis of schizophrenia. Schizophrenia, 8(1):52, 2022.

[35] Absalom E. Ezugwu, Abiodun M. Ikotun, Olaide O. Oyelade, Laith Abualigah, Jeffery O. Agushaka, Christopher I. Eke, and Andronicus A. Akinyelu. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. Engineering Applications of Artificial Intelligence, 110:104743, 2022.

[36] Rui Xu and Donald Wunsch. Survey of clustering algorithms. IEEE Transactions on neural networks, 16(3):645–678, 2005.

[37] Chris Fraley and Adrian E Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. The computer journal, 41(8):578–588, 1998.

[38] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. A review of clustering techniques and developments. Neurocomputing, 267:664–681, 2017.

[39] Sheng Zhou, Hongjia Xu, Zhuonan Zheng, Jiawei Chen, Zhao Li, Jiajun Bu, Jia Wu, Xin Wang, Wenwu Zhu, and Martin Ester. A comprehensive survey on deep clustering:

Taxonomy, challenges, and future directions. ACM Comput. Surv., 57(3), 11 2024.

[40] S.P. Lloyd. Least squares quantization in pcm. IEEE Transactions on Information Theory, 28(2):129–137, 1982.

[41] Leonard Kaufman and Peter J. Rousseeuw. Finding Groups in Data: An Introduction To Cluster Analysis. John Wiley & Sons, 01 1990.

[42] S.C. Johnson. Hierarchical clustering schemes. Psychometrika, 32(3):241–254, 1967.

[43] P.H. Sneath and R.R. Sokal. Numerical Taxonomy: The Principles and Practice of Numerical Classification. W. H. Freeman, 1973.

[44] S.M. Savaresi, D.L. Boley, S. Bittanti, and G. Gazzaniga. Cluster selection in divisive clustering algorithms. In Proceedings of the 2002 SIAM International Conference on Data Mining, pages 299–314. SIAM, 2002.

[45] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), pages 226–231. AAAI Press, 1996.

[46] R.J. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 160–172. Springer, 2013.

[47] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5):603–619, 2002.

[48] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, volume 27, pages 94–105. ACM, 1998.

[49] D. Birant and A. Kut. St-dbscan: An algorithm for clustering spatial–temporal data. Data & knowledge engineering, 60(1):208–221, 2007.

[50] C.E. Rasmussen. The infinite gaussian mixture model. In Advances in Neural Information Processing Systems, volume 12, pages 554–560, 1999.

[51] L. Rabiner and B. Juang. An introduction to hidden markov models. IEEE ASSP Magazine, 3(1):4–16, 1986.

[52] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. Journal of machine Learning research, 3:993–1022, 2003.

[53] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society: series B (methodological), 39(1):1–22, 1977.

[54] J. Shi and J. Malik. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):888–905, 2000.

[55] T. Kohonen. Self-organizing maps. Springer Series in Information Sciences, 30, 2001.

[56] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering

analysis, 2016.

[57] Richard J. Hathaway, John W. Davenport, and James C. Bezdek. Relational duals of the c-means clustering algorithms. Pattern Recognition, 22(2):205–212, 1989.

[58] Benjamin Moseley, Kirk Pruhs, Alireza Samadian, and Yuyan Wang. Relational algorithms for k-means clustering. Leibniz International Proceedings in Informatics, LIPIcs, 198, 8 2020.

[59] Richard J. Hathaway and James C. Bezdek. Nerf c-means: Non-euclidean relational fuzzy clustering. Pattern Recognition, 27(3):429–437, 1994.

[60] Hichem Frigui, Cheul Hwang, and Frank Chung-Hoon Rhee. Clustering and aggregation of relational data with applications to image database categorization. Pattern Recognition, 40(11):3053–3068, 2007.

[61] Francisco de A.T. de Carvalho, Yves Lechevallier, and Filipe M. de Melo. Relational partitioning fuzzy clustering algorithms based on multiple dissimilarity matrices. Fuzzy Sets and Systems, 215:1–28, 2013. Theme : Clustering.

[62] Kuang Zhou, Arnaud Martin, Quan Pan, and Zhunga Liu. Evidential relational clustering using medoids. 2015 18th International Conference on Information Fusion (Fusion), pages 413–420, 2015.

[63] Marie-Hélène Masson and Thierry Denœux. Recm: Relational evidential c-means algorithm. Pattern Recognition Letters, 30(11):1015–1026, 2009.

[64] Raghu Krishnapuram, Anupam Joshi, and Liyu Yi. A fuzzy relative of the k-medoids algorithm with application to web document and snippet clustering. In FUZZ-IEEE'99. 1999 IEEE International Fuzzy Systems. Conference Proceedings (Cat. No. 99CH36315), volume 3, pages 1281–1286. IEEE, 1999.

[65] Thomas Hofmann and Joachim Buhmann. Multidimensional scaling and data clustering. Advances in neural information processing systems, 7, 1994.

[66] Thierry Denœux and Mylène Masson. Clustering of proximity data using belief functions. pages 291–302, 2003.

[67] Thierry Denoeux and Marine H Masson. Evclus: Evidential clustering of proximity data. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 34(1):95–109, 2004.

[68] Zhi Su and Thierry Denœux. Bpec: Belief-peaks evidential clustering. IEEE Transactions on Fuzzy Systems, 27(1):111–123, 2019.

[69] James C. Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. Computers & Geosciences, 10(2):191–203, 1984.

[70] Josephine B. M. Benjamin and Ming-Syan Yang. Weighted multiview possibilistic c-means clustering with l2 regularization. IEEE Transactions on Fuzzy Systems, 30(5):1357–1370, 2022.

[71] Ming-Syan Yang and Kurnia P. Sinaga. Collaborative feature-weighted multi-view fuzzy c-means clustering. Pattern Recognition, 119:108064, 2021.

[72] Xiaodong Cai, Feiping Nie, and Heng Huang. Multi-view k-means clustering on

big data. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI), pages 2598–2604, 2013.

[73] Xiaochun Chen, Xiaowei Xu, Jianzhong Huang, and Yinyu Ye. Tw-k-means: Automated two-level variable-weighting clustering algorithm for multiview data. IEEE Transactions on Knowledge and Data Engineering, 25(4):932–944, 2013.

[74] Bing Jiang, Feng Qiu, and Liang Wang. Multi-view clustering via simultaneous weighting on views and features. Applied Soft Computing, 47:304–315, 2016.

[75] Guillaume Cleuziou, Marc Exbrayat, Laurent Martin, and Jean-Hugues Sublemontier. Cofkm: A centralized method for multiple-view clustering. In Proceedings of the 9th IEEE International Conference on Data Mining (ICDM), pages 752–757, 2009.

[76] Jian Han, Jianping Xu, Feiping Nie, and Xuelong Li. Multi-view k-means clustering with adaptive sparse memberships and weight allocation. IEEE Transactions on Knowledge and Data Engineering, 34(2):816–827, 2022.

[77] Yong Jiang, Francis L. Chung, Shitong Wang, Zhi Deng, Jun Wang, and Peide Qian. Collaborative fuzzy clustering from multiple weighted views. IEEE Transactions on Cybernetics, 45(4):688–701, 2015.

[78] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. ACM Computing Surveys, 31:264–323, 1999.

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

[80] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering – a decade review. Information Systems, 53:16–38, 10 2015.

[81] L. A. Zadeh. Fuzzy sets and systems, page 35–43. World Scientific Publishing Co., Inc., USA, 1996.

[82] Didier Dubois and Henri Prade. Possibility Theory, pages 2240–2252. Springer New York, New York, NY, 2012.

[83] Raghuram Krishnapuram and James M Keller. The possibilistic c-means algorithm: insights and recommendations. IEEE transactions on Fuzzy Systems, 4(3):385–393, 1996.

[84] N.R. Pal, K. Pal, J.M. Keller, and J.C. Bezdek. A possibilistic fuzzy c-means clustering algorithm. IEEE Transactions on Fuzzy Systems, 13(4):517–530, 2005.

[85] Vincent Barra and Jean-Yves Boire. Segmentation of fat and muscle from mr images of the thigh by a possibilistic clustering algorithm. Computer Methods and Programs in Biomedicine, 68(3):185–193, 2002.

[86] Zdzisław Pawlak. Rough sets. International journal of computer & information sciences, 11:341–356, 1982.

[87] Pawan Lingras. Rough set clustering for web mining. In 2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291), volume 2, pages 1039–1044. IEEE, 2002.

[88] Pawan Lingras and Chad West. Interval set clustering of web users with rough k-means. Journal of Intelligent Information Systems, 23(1):5–16, July 2004.

[89] Georg Peters, Fernando Crespo, Pawan Lingras, and Richard Weber. Soft clustering – fuzzy and rough approaches and their extensions and derivatives. International Journal of Approximate Reasoning, 54(2):307–322, 2013.

[90] D Maruthi Kumar, D Satyanarayana, and MN Giri Prasad. Improved rough-fuzzy c-means clustering and optimum fuzzy interference system for mri brain image segmentation. International Journal of Advanced Computer Science and Applications, 12(8), 2021.

[91] Glenn Shafer. A mathematical theory of evidence, volume 42. Princeton university press, 1976.

[92] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. The Annals of Mathematical Statistics, 38(2):325 – 339, 1967.

[93] Sally I. McClean. Data mining and knowledge discovery. In Robert A. Meyers, editor, Encyclopedia of Physical Science and Technology (Third Edition), pages 229–246. Academic Press, New York, third edition edition, 2003.

[94] Jean Dezert, Pei Wang, and Albena Tchamova. On the validity of dempster-shafer theory. In Fusion 2012 - 15th International Conference on Information Fusion, Singapour, Singapore, July 2012. 6.

[95] Frans Voorbraak. A computationally efficient approximation of dempster-shafer theory. International Journal of Man-Machine Studies, 30(5):525–536, 1989.

[96] P. Smets and R. Kennes. The transferable belief model. Artificial Intelligence, 66(2):191–234, 1994.

[97] Thierry Denœux. Decision-making with belief functions: A review. International Journal of Approximate Reasoning, 109:87–110, 2019.

[98] Thierry Denœux and Orakanya Kanjanatarakul. Beyond Fuzzy, Possibilistic and Rough: An Investigation of Belief Functions in Clustering, pages 157–164. 01 2017.

[99] Zuowei Zhang, Yiru Zhang, Hongpeng Tian, Arnaud Martin, Zhunga Liu, and Weiping Ding. A survey of evidential clustering: Definitions, methods, and applications. Information Fusion, 115:102736, 2025.

[100] Rajesh N Dave. Characterization and detection of noise in clustering. Pattern Recognition Letters, 12(11):657–664, 1991.

[101] Zhun-Ga Liu, Jean Dezert, Grégoire Mercier, and Quan Pan. Belief C-Means: An extension of fuzzy C-Means algorithm in belief functions framework. Pattern Recognition Letters, 33(3):291–300, 2012.

[102] Zhun ga Liu, Quan Pan, Jean Dezert, and Grégoire Mercier. Credal c-means clustering method based on belief functions. Knowledge-Based Systems, 74:119–132, 2015.

[103] V. Antoine, B. Quost, M.-H. Masson, and T. Denœux. CECM: Constrained evidential C-Means algorithm. Computational Statistics & Data Analysis, 56(4):894–914, 2012.

[104] Violaine Antoine, Jose A. Guerrero, and Jiarui Xie. Fast semi-supervised evidential

clustering. International Journal of Approximate Reasoning, 133:116–132, 2021.

[105] Abdoul Jalil Djiberou Mahamadou, Violaine Antoine, Gregory J. Christie, and Sylvain Moreno. Evidential clustering for categorical data. In Proceedings of the International Conference on Fuzzy Systems (FUZZ-IEEE), pages 1–6, 2019.

[106] Benoît Albert, Violaine Antoine, and Jonas Koko. ECM+: An improved evidential c-means with adaptive distance. Fuzzy Sets and Systems, 498:109168, 2025.

[107] Kuang Zhou, Arnaud Martin, Quan Pan, and Zhun ga Liu. Ecmdd: Evidential c-medoids clustering with multiple prototypes. Pattern Recognition, 60:239–257, 2016.

[108] Kuang Zhou, Arnaud Martin, Quan Pan, and Zhun ga Liu. Median evidential c-means algorithm and its application to community detection. Knowledge-Based Systems, 74:69–88, 2015.

[109] Jiang Bian, Dayong Tian, Yuanyan Tang, and Dacheng Tao. A survey on trajectory clustering analysis. ArXiv, abs/1802.06971, 2018.

[110] Sanjukta Krishnagopal. Multi-layer trajectory clustering: a network algorithm for disease subtyping. Biomedical Physics & Engineering Express, 6(6):065003, sep 2020.

[111] Jona Ruof, Max Bastian Mertens, Michael Buchholz, and Klaus Dietmayer. Real-time spatial trajectory planning for urban environments using dynamic optimization. In 2023 IEEE Intelligent Vehicles Symposium (IV), pages 1–7, 2023.

[112] Kevin Emery and André Berchtold. Comparison of two approaches in multichannel sequence analysis using the swiss household panel. Longitudinal and Life Course Studies, 14(4):592 – 623, 2023.

[113] Philippe Blanchard. Multi-dimensional biographies. explaining disengagement through sequence analysis. In 3rd ECPR Conference, Budapest, Hungary, pages 8–10, 2005.

[114] Musaab Riyadh, Norwati Mustapha, and Dina Riyadh. Review of trajectories similarity measures in mining algorithms. NTCCIT 2018 - Al Mansour International Conference on New Trends in Computing, Communication, and Information Technology, pages 36–40, 4 2019.

[115] Han Su, Shuncheng Liu, Bolong Zheng, Xiaofang Zhou, and Kai Zheng. A survey of trajectory distance measures and performance evaluation. The VLDB Journal, 29:3–32, 2020.

[116] L Spinsanti, M Berlingerio, and L Pappalardo. Mobility and geo-social networks. In Chiara Renso, Stefano Spaccapietra, and Esteban Zimanyi, editors, Mobility Data, pages 315–333, Cambridge, 10 2013. Cambridge University Press.

[117] Jean Damascene Mazimpaka. Trajectory data mining: A review of methods and applications. Journal of Spatial Information Science, 13, 12 2016.

[118] Andrew Abbott. Sequences of social events: Concepts and methods for the analysis of order in social processes. Historical Methods: A Journal of Quantitative and Interdisciplinary History, 16(4):129–147, 1983.

[119] Neil Vaughan and Bogdan Gabrys. Comparing and combining time series trajectories using dynamic time warping. Procedia Computer Science, 96:465–474, 1 2016.

[120] Dianfeng Qiao, Xinyu Yang, Yan Liang, and Xiaohui Hao. Rapid trajectory clustering based on neighbor spatial analysis. Pattern Recognition Letters, 156:167–173, 2022.

[121] Sharmila Selvaraj and B Sabarish. Analysis of distance measures in spatial trajectory data clustering. IOP Conference Series: Materials Science and Engineering, 1085:012021, 02 2021.

[122] Mohd Yousuf Ansari, Mainuddin, Amir Ahmad, and Gopal Bhushan. Spatiotemporal trajectory clustering: A clustering algorithm for spatiotemporal data. Expert Systems with Applications, 178:115048, 2021.

[123] Dongzhi Zhang, Kyungmi Lee, and Ickjai Lee. Hierarchical trajectory clustering for spatio-temporal periodic pattern mining. Expert Systems with Applications, 92:1–11, 2018.

[124] T. W. Liao. Clustering of time series data—a survey. Pattern Recognition, 38:1857–1874, 11 2005.

[125] Francisco de A.T. de Carvalho, Yves Lechevallier, and Filipe M. de Melo. Partitioning hard clustering algorithms based on multiple dissimilarity matrices. Pattern Recognition, 45(1):447–464, 2012.

[126] Kevin Toohey and Matt Duckham. Trajectory similarity measures. ACM SIGSPATIAL Special, 7:43–50, 5 2015.

[127] Nehal Magdy, Mahmoud A. Sakr, Tamer Mostafa, and Khaled El-Bahnasy. Review on trajectory similarity measures. 2015 IEEE 7th International Conference on Intelligent Computing and Information Systems, ICICIS 2015, pages 613–619, 2 2016.

[128] Yaguang Tao, Alan Both, Rodrigo I. Silveira, Kevin Buchin, Stef Sijben, Ross S. Purves, Patrick Laube, Dongliang Peng, Kevin Toohey, and Matt Duckham. A comparative analysis of trajectory similarity measures. https://doi.org/10.1080/15481603.2021.1908927, 58:643–669, 2021.

[129] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. SIGMOD Rec., 23(2):419–429, may 1994.

[130] Pierre François Marteau. Time warp edit distance with stiffness adjustment for time series matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31:306–318, 2009.

[131] Helmut Alt. The computational geometry of comparing shapes. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5760 LNCS:235–248, 2009.

[132] Lei Chen and Raymond Ng. On the marriage of lp-norms and edit distance, 2004.

[133] Zhenni Feng and Yanmin Zhu. A survey on trajectory data mining: Techniques and applications. IEEE Access, 4:1–1, 01 2016.

[134] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. Discovering similar multidimensional trajectories. Proceedings 18th International Conference on Data Engineering, pages 673–684, 2002.

[135] Daniel G. Zelterman. Longitudinal and time-series analysis. In Studying Human Populations: An Advanced Course in Statistics, pages 335–370. Springer New York,

New York, NY, 2008.

[136] Robert E. Weiss. Introduction to longitudinal data. Springer New York, pages 1–26, 12 2005.

[137] Nadia Deville-Stoetzel, Isabelle Gaboury, Djamal Berbiche, and Mylaine Breton. Profiling patterns of patient experiences of access and continuity at team-based primary healthcare clinics (canada): a latent class analysis. International Journal for Equity in Health, 23(1), October 2024.

[138] Shi Zhong and Joydeep Ghosh. A unified framework for model-based clustering. The Journal of Machine Learning Research, 4:1001–1037, 2003.

[139] Ángel López-Oriona and José A. Vilar. Analyzing categorical time series with the r package ctsfeatures. Journal of Computational Science, 76:102233, 2024.

[140] Luca De Angelis and José G. Dias. Mining categorical sequences from data using a hybrid clustering method. European Journal of Operational Research, 234(3):720–730, 2014.

[141] Christophe Genolini and Bruno Falissard. Kml: A package to cluster longitudinal data. Computer Methods and Programs in Biomedicine, 104(3):e112–e121, 2011.

[142] Chongming Gao, Zhong Zhang, Chen Huang, Hongzhi Yin, Qinli Yang, and Junming Shao. Semantic trajectory representation and retrieval via hierarchical embedding. Information Sciences, 538:176–192, 2020.

[143] Janet Giele and Glen Elder. Methods of life course research : qualitative and quantitative approaches. Contemporary Sociology, 1 1998.

[144] Gilbert Ritschard, Alexis Gabadinho, Matthias Studer, and Nicolas Müller. Converting between various sequence representations. Studies in Computational Intelligence, 223:155–175, 07 2009.

[145] Alexis Gabadinho, Gilbert Ritschard, Nicolas S Müller, and Matthias Studer. Analyzing and visualizing state sequences in r with traminer. Journal of Statistical Software, 40(4):1–37, 2011.

[146] Arnstein Aassve, Francesco C. Billari, and Raffaella Piccarreta. Strings of adulthood: A sequence analysis of young british women's work-family trajectories (parcours de la vie adulte: Une analyse par séquence des trajectoires travail-famille des jeunes femmes britanniques). European Journal of Population / Revue Européenne de Démographie, 23(3/4):369–388, 2007.

[147] Brendan Halpin. Multiple imputation for categorical time series. The Stata Journal, 16(3):590–612, 2016.

[148] Joseph B. Kruskal. An overview of sequence comparison: Time warps, string edits, and macromolecules. SIAM Review, 25(2):201–237, 1983.

[149] J.-C. Deville and G. Saporta. Correspondence analysis, with an extension towards nominal time series. Journal of Econometrics, 22(1):169–189, 1983.

[150] Patrick Rousset, Jean-François Giret, and Yvette Grelet. Typologies de parcours et dynamique longitudinale, 2012.

[151] Cees H. Elzinga. Sequence similarity - a non-aligning technique. Sociological Methods & Research, 32:3–29, 8 2003.

[152] R. W. Hamming. Error detecting and error correcting codes. The Bell System Technical Journal, 29(2):147–160, 1950.

[153] Andrew Abbott and Jon Forrest. Optimal matching methods for historical sequences. Journal of Interdisciplinary History, 16:471, 1986.

[154] Andrew Abbott and Angela Tsay. Sequence analysis and optimal matching methods in sociology: Review and prospect. Sociological Methods & Research, 29(1):3–33, 2000.

[155] Laurent Lesnard. Setting cost in optimal matching to uncover contemporaneous socio-temporal patterns. Sociological Methods & Research, 38(3):389–419, 2010.

[156] Matissa Hollister. Is optimal matching suboptimal? Sociological Methods & Research, 38(2):235–264, 2009.

[157] Brendan Halpin. Optimal matching analysis and life-course data: The importance of duration. Sociological Methods & Research, 38(3):365–388, 2010.

[158] Torsten Biemann. A transition-oriented approach to optimal matching. Sociological Methodology, 41(1):195–221, 2011.

[159] Hui Wang, Zhiwei Lin, Sally McClean, and Jun Liu. Measuring similarity for multidimensional sequences. In 2010 IEEE International Conference on Data Mining Workshops, pages 281–287, 2010.

[160] Katariina Salmela-Aro, Noona Kiuru, Jari Erik Nurmi, and Mervi Eerola. Mapping pathways to adulthood among finnish university students: Sequences, patterns, variations in family- and work-related roles. Advances in Life Course Research, 16:25–41, 3 2011.

[161] Philippe Blanchard. Analyse séquentielle et carrières militantes, 4 2010. Rapport de recherche.

[162] Lawrence Hubert and Phipps Arabie. Comparing partitions. Journal of classification, 2:193–218, 1985.

[163] Matthias Studer. Weightedcluster library manual: A practical guide to creating typologies of trajectories in the social sciences with r. LIVES Working papers, 24, 01 2013.

[164] Nicolas Muller, Matthias Studer, and Gilbert Ritschard. Classification de parcours de vie à l'aide de l'optimal matching, 09 2007.

[165] Bureau of Labor Statistics, U.S. Department of Labor. National longitudinal survey of youth 1979 cohort, 1979-2016 (rounds 1-27). Produced and distributed by the Center for Human Resource Research (CHRR), The Ohio State University. Columbus, OH, 2019.

[166] Marc Scott and Matthew Ziedenberg. Order or chaos? understanding career mobility using categorical clustering and information theory. Longitudinal and Life Course Studies, 7, 10 2016.

[167] Eduardo Marín-López, Gustavo Rodríguez-Leal, and Florencia Vargas. Primary biliary

cirrhosis: prognostic models. Revista de gastroenterología de México, 60:S72–4, 10 1995.

[168] Sébastien Massoni, Madalina Olteanu, and Nathalie Villa-Vialaneix. Which dissimilarity is to be used when extracting typologies in sequence analysis: A comparative study. In Advances in computational intelligence, volume 7902, pages 69–79, Berlin, Heidelberg, 06 2013. Springer Berlin Heidelberg.

[169] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. The Journal of Machine learning research, 7:1–30, 2006.

[170] C. Chenaf, J. Delorme, N. Delage, D. Ardid, A. Eschalier, and N. Authier. Prevalence of chronic pain with or without neuropathic characteristics in france using the capture-recapture method: a population-based study. Pain, 159(11):2394—2402, 11 2018.

[171] M. D. K. Jenssen, P. A. Bakkevoll, P. D. Ngo, A. Budrionis, A. J. Fagerlund, M. Tayefi, J. G. Bellika, and F. Godtliebsen. Machine learning in chronic pain research: A scoping review. Applied Sciences, 11(7), 2021.

[172] T. C. Fu. A review on time series data mining. Engineering Applications of Artificial Intelligence, 24:164–181, 2 2011.

[173] I. Prakaisak and P. Wongchaisuwat. Hydrological time series clustering: A case study of telemetry stations in thailand. Water 2022, Vol. 14, Page 2095, 14:2095, 6 2022.

[174] B. D. Fulcher. Feature-based time-series analysis. CoRR, abs/1709.08055, 2017.

[175] Muhammad Abu Bakar Siddik, Md. Mahfuzul Islam Mazumder, Rakibul Alam, and Musharrat Khan. Performance comparison between dimension reduction and feature selection approaches for data classification. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), pages 893–898, 2021.

[176] I. Guyon and A. Elisseeff. An introduction of variable and feature selection. J. Machine Learning Research Special Issue on Variable and Feature Selection, 3:1157 – 1182, 01 2003.

[177] Chonghui Guo, Menglin Lu, and Jingfeng Chen. An evaluation of time series summary statistics as features for clinical prediction tasks. BMC Med. Inform. Decis. Mak., 20(1):48, 03 2020.

[178] R. Suganya and R. M. Shanthi. Fuzzy c- means algorithm- a review. International Journal of Scientific and Research Publications, 2, 2012.

[179] James C. Bezdek, Mikhil R. Pal, James Keller, and Raghu Krisnapuram. Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Kluwer Academic Publishers, USA, 1999.

[180] Pierpaolo D'Urso, Livia De Giovanni, and Riccardo Massari. Robust fuzzy clustering of multivariate time trajectories. Int. J. Approx. Reasoning, 99(C):12–38, 08 2018.

[181] P. D'Urso. Fuzzy clustering for data time arrays with inlier and outlier time trajectories. IEEE Transactions on Fuzzy Systems, 13(5):583–604, 2005.

[182] Ángel López-Oriona, Pierpaolo D'Urso, José A. Vilar, and Borja Lafuente-Rego. Quantile-based fuzzy c-means clustering of multivariate time series: Robust techniques. International Journal of Approximate Reasoning, 150:55–82, 2022.

[183] Ángel López-Oriona, José A. Vilar, and Pierpaolo D'Urso. Quantile-based fuzzy clustering of multivariate time series in the frequency domain. Fuzzy Sets and Systems, 443:115–154, 2022. From Learning to Modeling and Control.

[184] José Vilar, Borja Lafuente Rego, and Pierpaolo D'Urso. Quantile autocovariances: A powerful tool for hard and soft partitional clustering of time series. Fuzzy Sets and Systems, 340, 03 2017.

[185] Pierpaolo D'Urso, Livia De Giovanni, and Vincenzina Vitale. Robust DTW-based entropy fuzzy clustering of time series. Ann. Oper. Res., 12 2023.

[186] Pierpaolo D'Urso, Luis García-Escudero, Livia De Giovanni, Vincenzina Vitale, and Agustin Mayo. Robust fuzzy clustering of time series based on b-splines. International Journal of Approximate Reasoning, 136, 06 2021.

[187] Sandeep Panda, Sanat Kumar Sahu, Pradeep Kumar Jena, and Subhagata Chattopadhyay. Comparing fuzzy-c means and k-means clustering techniques: A comprehensive study. In Advances in Intelligent and Soft Computing, pages 451–460. Springer Berlin Heidelberg, 2012.

[188] A. S. Zigmond and R. P. Snaith. The hospital anxiety and depression scale. Acta psychiatrica Scandinavica, 67:361–370, 1983.

[189] M. J. L. Sullivan, W. Stanish, H. Waite, M. Sullivan, and D. A. Tripp. Catastrophizing, pain, and disability in patients with soft-tissue injuries. Pain, 77:253–260, 9 1998.

[190] M. F. Scheier, C. S. Carver, and M. W. Bridges. Distinguishing optimism from neuroticism (and trait anxiety, self-mastery, and self-esteem): a reevaluation of the life orientation test. Journal of personality and social psychology, 67:1063–1078, 1994.

[191] Pugazhenthi Annadurai and L.s Kumar. Selection of optimal number of clusters and centroids for k-means and fuzzy c-means clustering: A review. In 2020 5th International conference on computing, communication and security (ICCCS), pages 1–4. IEEE, 10 2020.

[192] T. C. Fu, F. L. Chung, V. Ng, and R. Luk. Pattern discovery from stock time series using self-organizing maps. Citeseer, 1, 2001.

[193] J. Wilpon and L. Rabiner. A modified k-means clustering algorithm for use in isolated work recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 33(3):587–594, 1985.

[194] C. Goutte, P. Toft, E. Rostrup, F. Å. Nielsen, and L. K. Hansen. On clustering fmri time series. NeuroImage, 9(3):298–310, 1999.

[195] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr. Time series feature extraction on basis of scalable hypothesis tests (tsfresh – a python package). Neurocomputing, 307:72–77, 9 2018.

[196] J. Van Der Donckt, J. Van Der Donckt, E. Deprost, N. Vandenbussche, M. Rademaker, G. Vandewiele, and S. Van Hoecke. Do not sleep on traditional machine learning: Simple and interpretable techniques are competitive to deep learning for sleep scoring. Biomedical Signal Processing and Control, 81:104429, 2023.

[197] J. Wiens, E. Horvitz, and J. Guttag. Patient risk stratification for hospital-associated c.

diff as a time-series classification task. Curran Associates, Inc., 25, 2012.

[198] J. G. Dy and C. E. Brodley. Feature selection for unsupervised learning. J. Mach. Learn. Res., 5:845–889, 2004.

[199] J. Hua, W. D. Tembe, and E. R. Dougherty. Performance of feature-selection methods in the classification of high-dimension data. Pattern Recognition, 42(3):409–424, 2009.

[200] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. Bioinformatics, 23(19):2507–2517, 08 2007.

[201] J. Tang, S. Alelyani, and H. Liu. Feature selection for classification: A review. Data Classification: Algorithms and Applications, pages 37–64, 01 2014.

[202] D. Devakumari and T. Kuttiyannan. Unsupervised adaptive floating search feature selection based on contribution entropy. Proceedings of 2010 International Conference on Communication and Computational Intelligence, INCOCCI-2010, 01 2010.

[203] S. Solorio-Fernández, J. Carrasco-Ochoa, and J. F. Martínez-Trinidad. A review of unsupervised feature selection methods. Artificial Intelligence Review, 53, 02 2020.

[204] B. Remeseiro and V. Bolon-Canedo. A review of feature selection methods in medical applications. Computers in Biology and Medicine, 112:103375, 2019.

[205] T. Shi and S. Horvath. Unsupervised learning with random forest predictors. Journal of Computational and Graphical Statistics, 15(1):118–138, 2006.

[206] H. Elghazel and A. Aussem. Unsupervised feature selection with ensemble learning. Machine Learning, 98:157–180, 1 2015.

[207] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In Y. Weiss, B. Schölkopf, and J. Platt, editors, Advances in Neural Information Processing Systems, volume 18. MIT Press, 2005.

[208] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. ACM International Conference Proceeding Series, 227:1151–1157, 2007.

[209] G. J. Klir and M. J. Wierman. Uncertainty-based information: elements of generalized information theory. Springer-Verlag, New-York, 15, 1999.

[210] F. Batool and C. Hennig. Clustering with the average silhouette width. Computational Statistics & Data Analysis, 158:107190, 2021.

[211] H. Y. Teh, K. I. Wang, and A. W. Kempa-Liehr. Expect the unexpected: Unsupervised feature selection for automated sensor anomaly detection. IEEE Sensors Journal, 21:18033–18046, 8 2021.

[212] A. Kennedy, G. Nash, N. J. Rattenbury, and A. W. Kempa-Liehr. Modelling the projected separation of microlensing events using systematic time-series feature engineering. Astronomy and Computing, 35:100460, 4 2021.

[213] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

[214] Batunacun, R. Wieland, T. Lakes, and C. Nendel. Using shapley additive explanations to interpret extreme gradient boosting predictions of grassland degradation in xilingol, china. Geoscientific Model Development, 14(3):1493–1510, 2021.

[215] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. A unified view of kernel k-means, spectral clustering and graph cuts. Citeseer, 2004.

[216] Benjamin A. Antunes and David R. C. Hill. Reproducibility, replicability, and repeatability: A survey of reproducible research with a focus on high performance computing, 2024.

[217] Zhong dong Wu, Wei xin Xie, and Jian ping Yu. Fuzzy c-means clustering algorithm based on kernel method. In Proceedings Fifth International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2003, pages 49–54, 2003.

[218] Yizong Cheng. Mean shift, mode seeking, and clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(8):790–799, 1995.

[219] Li Zheng and Tao Li. Semi-supervised hierarchical clustering. In 2011 IEEE 11th International Conference on Data Mining, pages 982–991, 2011.

[220] Thomas A Runkler. Relational fuzzy clustering. Advances in Fuzzy Clustering and its Applications, pages 31–52, 2007.

[221] Jian-Ping Mei and Lihui Chen. Fuzzy clustering with weighted medoids for relational data. Pattern Recognition, 43(5):1964–1974, 2010.

[222] R.N. Dave and S. Sen. Robust fuzzy clustering of relational data. IEEE Transactions on Fuzzy Systems, 10(6):713–727, 2002.

[223] Thierry Denœux and Orakanya Kanjanatarakul. Evidential clustering: A review. In Integrated Uncertainty in Knowledge Modelling and Decision Making, pages 24–35. Springer International Publishing, 11 2016.

[224] Renê Pereira de Gusmão and Francisco de A.T. de Carvalho. Clustering of multi-view relational data based on particle swarm optimization. Expert Systems with Applications, 123:34–53, 2019.

[225] Zhe Liu, Haojian Huang, and Sukumar Letchmunan. Adaptive weighted multi-view evidential clustering. In Artificial Neural Networks and Machine Learning – ICANN 2023, pages 265–277. Springer Nature Switzerland, 06 2023.

[226] Zhe Liu, Haojian Huang, Sukumar Letchmunan, and Muhammet Deveci. Adaptive weighted multi-view evidential clustering with feature preference. Knowledge-Based Systems, 294:111770, 2024.

[227] Guoqing Chao, Shiliang Sun, and Jinbo Bi. A survey on multiview clustering. IEEE transactions on artificial intelligence, 2(2):146–168, 2021.

[228] Francisco de A.T. de Carvalho, Marc Csernel, and Yves Lechevallier. Clustering constrained symbolic data. Pattern Recognition Letters, 30(11):1037–1045, 2009.

[229] Patric Bosc and Henri Prade. An introduction to the fuzzy set and possibility theory-based treatment of flexible queries and uncertain or imprecise databases. In Uncertainty Management in Information Systems: From Needs to Solutions, pages

285–324. Springer, 1997.

[230] Enrique H. Ruspini, James C. Bezdek, and James M. Keller. Fuzzy clustering: A historical perspective. IEEE Computational Intelligence Magazine, 14(1):45–55, 2019.

[231] Bin Yu, Zijian Zheng, Mingjie Cai, Witold Pedrycz, and Weiping Ding. Frcm: A fuzzy rough c-means clustering method. Fuzzy Sets and Systems, 480:108860, 2024.

[232] Armel Soubeiga, Violaine Antoine, and Sylvain Moreno. Multi-view relational evidential c-medoid clustering with adaptive weighted. In 2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA), pages 1–10. IEEE, 2024.

[233] Thierry Denœux, Shoumei Li, and Songsak Sriboonchitta. Evaluating and comparing soft partitions: An approach based on dempster–shafer theory. IEEE Transactions on Fuzzy Systems, 26(3):1231–1244, 2018.

[234] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[235] Maria Brbić and Ivica Kopriva. Multi-view low-rank sparse subspace clustering. Pattern Recognition, 73:247–258, 2018.

[236] João C. Xavier, Anne M. P. Canuto, Noriedson D. Almeida, and Luiz M. G. Gonçalves. A comparative analysis of dissimilarity measures for clustering categorical data. In The 2013 International Joint Conference on Neural Networks (IJCNN), pages 1–8, 2013.

[237] J. C. Gower. A general coefficient of similarity and some of its properties. Biometrics, 27(4):857–871, 1971.

[238] Anthony J. Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn J. Keogh. The UEA multivariate time series classification archive, 2018. CoRR, abs/1811.00075, 2018.

[239] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In Proceedings of the 3rd international conference on knowledge discovery and data mining, pages 359–370, 1994.

[240] N. Karthikeyani Visalakshi and Dr. Thangavel Kuttiyannan. Impact of normalization in distributed k-means clustering. International Journal of Soft Computing, 4:168–172, 01 2009.

[241] Philippe Smets. Decision making in the tbm: the necessity of the pignistic transformation. International Journal of Approximate Reasoning, 38(2):133–147, 2005.

[242] François Petitjean, Germain Forestier, Geoffrey I Webb, Ann E Nicholson, Yanping Chen, and Eamonn Keogh. Dynamic time warping averaging of time series allows faster and more accurate classification. In Proceedings of the International conference on data mining (ICDM), pages 470–479, 2014.

[243] Stephan Dempe and Alain Zemkoho. Bilevel optimization. In Optimization and its applications, volume 161. Springer, 2020.

[244] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In Proceedings of the Conference on Neural Information Processing Systems (NIPS), 2017.

[245] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. IEEE Transactions on Emerging Topics in Computing, 2(03):267–279, 2014.

[246] Marco Cuturi and Mathieu Blondel. Soft-DTW: a differentiable loss function for time-series. In Proceedings of the International Conference on Machine Learning (ICML), volume 70, pages 894–903, 2017.

[247] Manjunatha Veerappa, Mathias Anneken, Nadia Burkart, and Marco F. Huber. Validation of xai explanations for multivariate time series classification in the maritime domain. Journal of Computational Science, 58:101539, 2022.

[248] Sarthak Manas Tripathy, Ashish Chouhan, Marcel Dix, Arzam Kotriwala, Benjamin Klöpper, and Ajinkya Prabhune. Explaining anomalies in industrial multivariate time-series data with the help of explainable ai. In 2022 IEEE International Conference on Big Data and Smart Computing (BigComp), pages 226–233, 2022.

[249] Joshua Warner et al. Scikit-fuzzy: Fuzzy logic toolbox for python. https://github.com/scikit-fuzzy/scikit-fuzzy, 2018.

[250] Miin-Shen Yang and Kuo-Lung Wu. Unsupervised possibilistic clustering. Pattern Recognition, 39(1):5–21, 2006.

[251] Georg Peters. Rough clustering utilizing the principle of indifference. Information Sciences, 277:358–374, 2014.

[252] Thierry Denoeux. evclust: An R Package for Evidential Clustering. The Comprehensive R Archive Network (CRAN), 2023. R package version 2.0.3.

[253] Thierry Denœux, Songsak Sriboonchitta, and Orakanya Kanjanatarakul. Evidential clustering of large dissimilarity data. Knowledge-Based Systems, 106:179–195, 2016.

[254] Lianmeng Jiao, Thierry Denœux, Zhun ga Liu, and Quan Pan. Egmm: An evidential version of the gaussian mixture model for clustering. Applied Soft Computing, 129:109619, 2022.

[255] Alaa Althubaiti. Information bias in health research: definition, pitfalls, and adjustment methods. Journal of Multidisciplinary Healthcare, page 211, May 2016.

[256] Y. Abo-Elnaga and S. Nasr. K-means cluster interactive algorithm-based evolutionary approach for solving bilevel multi-objective programming problems. Alexandria Engineering Journal, 61(1):811–827, 2022.

[257] El-Ghazali Talbi. A Taxonomy of Metaheuristics for Bi-level Optimization, page 1–39. Springer Berlin Heidelberg, 2013.

[258] Seyedjamal Zolhavarieh, Saeed Aghabozorgi, and Ying Wah Teh. A review of subsequence time series clustering. The Scientific World Journal, 2014(1):312521, 2014.

[259] Hong-Yu Wang, Jie-Sheng Wang, and Guan Wang. A survey of fuzzy clustering validity evaluation methods. Information Sciences, 618:270–297, 2022.