RÉPUBLIQUE FRANÇAISE
Liberté Égalité Fraternité

Université de Toulouse

Doctorat de l'Université de Toulouse

préparé à l'Université Toulouse - Jean Jaurès

Evaluation et recommandation des mesures de similarité pour le clustering de données mixtes

Thèse présentée et soutenue, le 21 janvier 2025 par

Abdoulaye DIOP

École doctorale
EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse

Spécialité
Informatique et Télécommunications

Unité de recherche
IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par
Max CHEVALIER et Olivier TESTE

Composition du jury
M. Cédric WEMMERT, Président, Université de Strasbourg
Mme Armelle BRUN, Rapporteure, Université de Lorraine
M. Mustapha LEBBAH, Rapporteur, Université Paris-Saclay
Mme Maguelonne TEISSEIRE, Examinatrice, INRAE Occitanie-Montpellier
M. Max CHEVALIER, Directeur de thèse, Université de Toulouse
M. Olivier TESTE, Co-directeur de thèse, Université Toulouse - Jean Jaurès

Membres invités
M. Geoffrey ROMAN-JIMENEZ, Co-encadrant du monde socio-économique, Solution Data Group
M. André PENINOU, Co-encadrant, Université Toulouse - Jean Jaurès

Abstract: Clustering algorithms are essential in data mining, offering powerful tools to uncover hidden patterns and structures within datasets. aim to divide data points into coherent groups based on similarities or dissimilarities, making it easier to explore and understand complex data. A critical component of clustering algorithms is the similarity measure, which significantly affects their ability to identify meaningful patterns. Thus, selecting suitable similarity measures for clustering algorithms is a crucial challenge addressed in this thesis.

Our research focuses on clustering mixed data—datasets containing both numerical and categorical attributes—which are increasingly common in fields such as healthcare, finance, marketing, and social sciences. Traditional clustering algorithms, designed for homogeneous data, cannot be directly applied to mixed data due to the differing nature of numerical and categorical attribute types. This necessitates specialized approaches for mixed data clustering.

We categorize mixed data clustering methods into two groups: conversion-based approaches (referred to as homogenization methods) and non-conversion-based approaches (mixed methods). Through extensive experiments, we demonstrate that mixed methods are generally more effective, as they handle different data types directly without altering the dataset's inherent structure. In contrast, homogenization methods, which convert one data type into another, often lead to sub-optimal clustering results.

Focusing on mixed methods, we further investigate the impact of similarity measures on clustering performance. Unlike clustering algorithms for homogeneous data, mixed methods typically combine two similarity measures—one for numerical attributes and one for categorical attributes. Our experiments show that the choice of these similarity measures has a significant impact on the clustering results, highlighting the importance of selecting the appropriate measures for each dataset.

However, selecting the right similarity measures can be challenging, especially for non-experts, due to the wide range of available measures for each data type and their performance dependency on the dataset, clustering algorithm, and cluster validity index. To address this, we propose SIMREC, a similarity measure recommendation system for mixed data clustering. SIMREC leverages meta-learning to identify relationships between dataset characteristics and the performance of similarity measures for different mixed data clustering algorithms and cluster validity indices. Given a mixed dataset, clustering algorithm, and validity index, the system recommends optimal pairs of numerical and categorical similarity measures based on the dataset characteristics. This system aims to assist both expert and non-expert users in efficiently selecting similarity measures, avoiding time-consuming trial-and-error and search-based strategies.

Keywords: Mixed Data Clustering, Similarity measures, Recommendation Systems, Meta-Learning

Résumé : Le clustering est une tâche importante pour l'exploration de données. Il permet de découvrir de manière non-supervisée des tendances ou des structures cachées au sein de grands ensembles de données. Les algorithmes de clustering visent à regrouper un ensemble d'observations en plusieurs groupes ou clusters de telle sorte que les observations au sein d'un même groupe soient similaires entre elles et différentes des observations dans les autres groupes. Un composant clé de ces algorithmes est la mesure de similarité qui a un impact direct sur la construction des clusters et, par conséquent, sur les performances des algorithmes. Le choix d'une mesure de similarité adaptée en fonction des données et de l'algorithme de clustering considéré est donc primordial et constitue l'objet principal de cette thèse.

Notre recherche se concentre sur le clustering de données mixtes qui sont des données hétérogènes présentant à la fois des attributs numériques et catégoriels. Elles sont très courantes dans des domaines tels que la santé, la finance, le marketing et les sciences sociales. Les algorithmes de clustering traditionnels, conçus pour des données homogènes, ne peuvent pas être appliqués directement aux données mixtes, d'où la nécessité de méthodes spécialisées. Nous classons ces méthodes en deux catégories : les approches basées sur la conversion (appelées méthodes d'homogénéisation) et celles qui considèrent les données mixtes directement sans conversion (méthodes mixtes). Nous montrons dans ce manuscrit, que les méthodes mixtes sont généralement préférables aux méthodes d'homogénéisation, car elles conservent la structure originale des données et utilisent un traitement adapté pour chaque type d'attribut. Nos travaux se focalisent donc principalement sur les méthodes mixtes.

Dans un premier temps, nous avons mené une étude expérimentale afin d'évaluer l'impact des mesures de similarité sur les performances des méthodes mixtes. Ces méthodes combinent généralement deux mesures de similarité : l'une pour les attributs numériques et l'autre pour les attributs catégoriels. Nos expérimentations montrent que le choix de ces mesures de similarité influence de manière significative les performances des différents algorithmes considérés, soulignant ainsi l'importance de choisir des mesures appropriées.

Trouver les meilleures ou de bonnes mesures de similarité est difficile, en particulier pour les utilisateurs non experts, en raison du grand nombre de mesures qui existent dans la littérature et de leurs performances variables en fonction du jeu de données, de l'algorithme de clustering et de la mesure de performance. Afin de répondre à cette problématique, nous avons proposé SIMREC, un système de recommandation de mesures de similarité pour les algorithmes de clustering de données mixtes. SIMREC utilise le meta-learning (ou méta-apprentissage) pour identifier les relations entre les caractéristiques des jeux de données et les performances des différentes mesures de similarité, et ce pour différents algorithmes de clustering et mesures de performance. SIMREC prend en entrée un triplet composé d'un jeu de données mixtes, d'un algorithme de clustering et d'une mesure de performance à optimiser. Il recommande ensuite les paires optimales de mesures de similarité numérique et catégorielle en fonction des caractéristiques du jeu de données d'entrée. Ce système permet à la fois à des utilisateurs experts et non experts de choisir de façon efficace des mesures de similarité adaptées à leur cas d'usage, évitant ainsi les stratégies d'essais-erreurs qui sont souvent chronophages

et coûteuses.

Mots clés : Clustering de données mixtes, Mesures de similarité, Système de recommendation, Méta-apprentissage

Contents

List of figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Research Context

1.1.1 Clustering

Clustering is a very common unsupervised machine learning task that plays a pivotal role in data mining, offering powerful tools for uncovering hidden patterns and structures within datasets. Clustering algorithms aim to divide data points into coherent groups or clusters based on *similarities* or *dissimilarities* [Ezugwu 2022], making it easier to explore and understand complex data. The clustering process mirrors how the human brain organizes information. Just as the brain identifies patterns and groups similar items together without predefined labels, clustering algorithms group data points based on their similarities. This cognitive parallel underscores the intuitive nature of clustering and highlights its critical role in data exploration, summarization, understanding, etc. Clustering algorithms find a wide range of applications in various domains:

- **Medicine.** The healthcare sector is a critical field that demands continuous improvement, particularly with the advancement of contemporary society. Clustering analysis has become a key tool in healthcare. For instance, it has been applied to personalized medicine by identifying subgroups of complex patients who may benefit from tailored care management strategies [Magoev 2018, Newcomer 2011]. Additionally, clustering algorithms are widely used for medical image segmentation, which involves dividing an image into regions that are homogeneous in certain properties. This technique plays a significant role in detecting various diseases that can be identified through medical imaging such as tumors and some ocular diseases [Waheed 2015, Rahman 2024].

- **Finance.** Clustering techniques play a crucial role in enhancing customer service, profitability, and security within the financial sector [Cai 2016]. These techniques enable institutions to group entities (banks, customers, territorial entities, etc.) into homogeneous clusters, facilitating a more accurate assessment of risk profiles and profitability, as demonstrated in [Dardac 2009]. In addressing security concerns, clustering is particularly valuable for anti-money laundering efforts, where algorithms like DBSCAN are employed to detect suspicious transactions [Ezugwu 2022]. Furthermore, clustering aids in identifying at-risk customers, such as those vulnerable to fraud or phishing [Alkhasov 2015], and supports strategic business decisions like selecting optimal locations for ATMs and branches based on regional economic activity [Raghu Kisore 2017].

- **Marketing and sales.** Marketing is another sector where clustering techniques have been widely adopted to automate and optimize tasks such as market segmentation, new product development, and product positioning. In customer segmentation, clustering is particularly valuable as it processes large volumes of customer data to identify groups of customers with similar

Table 1.1: Example of a mixed dataset

Weight (kg)	Height (m)	Age	Blood Group	Profession
80.6	1.85	35	B+	Teaching
73.6	1.72	47	A+	Teaching
70.8	1.79	32	B+	Medical
85.9	1.91	27	A-	Sportsman
83.4	1.65	51	A+	Medical

behaviors and preferences. This allows businesses to design more personalized and effective marketing strategies [Kansal 2018, Tabianan 2022]. Another key application is customer review analysis, where clustering groups reviews based on shared sentiments or preferences. This enables more nuanced market insights, facilitating better-informed product development and positioning decisions [Jamil 2021, Jardim 2022].

There are many other application domains using clustering, such as energy [Vialetto 2020], aeronautics [Li 2015, Mangortey 2020], manufacturing [Subramaniyan 2020], information retrieval [Chifu 2015], text mining [Lydia 2018], urban development [Peng 2013], weather forecasting [Chakraborty 2014], etc.

As clustering techniques continue to demonstrate their value across diverse domains, the increasing complexity of real-world data presents new challenges for clustering algorithms. One such challenge is the clustering of heterogeneous data, particularly mixed data, which are prevalent in many practical applications and are the focus of this research.

1.1.2 Mixed-data

In many recent applications and clustering tasks in health [Halawani 2012, McParland 2017], business and marketing [Kassi 2015], finance [Hennig 2013, Caruso 2021], or social studies [Niu 2015], we face data with attributes that exhibit heterogeneous properties [Abdullin 2012, Ahmad 2019]. There are different types of attribute heterogeneity [Cao 2013]: type heterogeneity (e.g., attributes are of different types such as numeric, categorical, textual...), distribution heterogeneity, and domain heterogeneity. This work focuses on type heterogeneity and particularly on *mixed data*, i.e., data with *numeric* and *categorical* attributes. An example of a mixed dataset is given in table 1.1.

Numerical data consist of numerical values representing quantities or measurements. They are characterized by an order relation between values and can be continuous (e.g., Height, Weight) or discrete (e.g., Age) [Barcelo-Rico 2012]. They also allow for mathematical operations such as addition, subtraction, multiplication, division, and statistical calculations (e.g., mean, median, standard deviation).

Similarity measures between numeric data have been extensively studied and are well-defined in the literature [Deza 2013, Shirkhorshidi 2015, Abu Alfeilat 2019]. They are usually based on distance metrics such as Euclidean distance, Mahalanobis distance, cosine similarity, etc.

In contrast, categorical data comprise a finite number of labels or categories (e.g., Blood Group, Profession) without an order relation and lacking inherent numerical meaning [Barcelo-Rico 2012]. Categorical data do not support mathematical operations in the same way numerical data does. Instead, operations like counting occurrences, frequency distribution, and mode calculation are more relevant. Although the notion of similarity or distance for categorical data is not as straightforward as for continuous data due to the non-ordered aspect of categorical values, different measures have been proposed in the literature [Alves 2019, Boriah 2008] that exploit other information such as the frequency and distribution of categorical values.

Clustering mixed datasets is challenging due to the different nature of numerical and categorical attributes. It raises the question of how to jointly handle and integrate these two types of attributes so that one could efficiently group objects without loss of information [Behzadi 2020]. Existing approaches in the literature can be divided into two groups [Barcelo-Rico 2012, Ji 2013, Wei 2015]: methods based on data conversion called *homogenization methods* and methods that handle mixed data directly without any conversion named *mixed methods*.

Given a mixed dataset, homogenization methods convert all attributes to a single type (either numerical or categorical) and use standard clustering algorithms with known similarity measures for the target type. Despite providing a convenient way to directly apply standard clustering algorithms such as K-Means, homogenization methods have several drawbacks. First, converting numerical attributes to categorical ones ignores the ordering relation between numerical values, implying a loss of information [van de Velden 2019, Wei 2015]. Second, converting categorical attributes to numerical results in creating order between categorical values or increasing data dimension in order to represent categorical values [Barcelo-Rico 2012].

To better solve the problem, mixed methods rely on strategies that allow to jointly handle numerical and categorical attributes without needing conversion. The main strategy to do this is to use a specific similarity measure for each data type and then combine the two measures to define a global similarity for mixed data [Ahmad 2019]. An example of such a method is the K-Prototypes algorithm [Huang 1998] which extends the K-Means algorithm with a new definition of cluster centers (using mean value for numerical attributes and mode value for categorical ones) and a new similarity for mixed data based on a linear combination of the *squared Euclidean distance* for numerical attributes and *Hamming distance* for categorical ones.

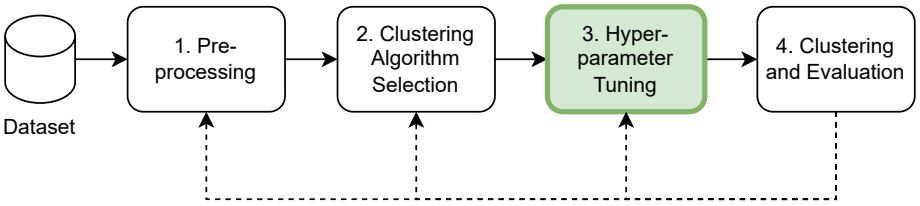


Figure 1.1: A typical clustering pipeline

1.2 Research problem

Given a dataset, figure 1.1 shows the typical clustering pipeline: 1) preprocessing, e.g., normalization, feature selection, etc. 2) clustering algorithm selection 3) hyper-parameter tuning, e.g., number of clusters, similarity measures, etc. 4) clustering and evaluation. The user has to make several decisions at each step of this process such as the choice of the preprocessing techniques and clustering algorithm, the tuning of the algorithm parameters, and so on. Since these choices highly impact the quality of obtained clusterings, important efforts have been made in the literature to support data scientists by providing recommendations for the different steps. In our case, we are interested in the third step, i.e., hyper-parameter tuning. We are interested in one parameter in particular: the similarity measure. Similarity measures play a crucial role in clustering algorithms since these algorithms rely on similarity between observations to build clusters. Their choice significantly influences clustering results, impacting the algorithms' ability to uncover meaningful patterns and structures within the data. In the context of mixed data, non-conversion-based methods (i.e., *mixed methods*) generally define similarity for mixed data by combining two similarity measures: one for numerical attributes and one for categorical ones. *How to choose the right pair of numerical and categorical similarity measures is a critical question for mixed data clustering algorithms and is the problem we address in this thesis.*

To better illustrate the problem, we consider the K-Prototypes algorithm [Huang 1998], one of the most popular clustering algorithms for mixed data. Figure 1.2 shows, for 10 different datasets, the distribution of obtained clustering accuracy scores when using 120 different pairs of similarity measures. The clustering accuracy score is a cluster validity index, i.e., a performance measure for clustering algorithms. It is similar to the accuracy score in classification. It varies in $[0, 1]$ and higher values indicate better performances. We can see from figure 1.2 that the choice of the pair of similarity measures highly impacts the performance of the clustering algorithm. The difference between the accuracy of the best and the worst pairs is greater than 0.25 for the different datasets and close to 0.5 in some cases. We also observe that for the different datasets, there exist several pairs of similarity measures that outperform the default pair used in K-Prototypes [Huang 1998]. Furthermore, for some datasets using the best similarity pair can result in more 40%

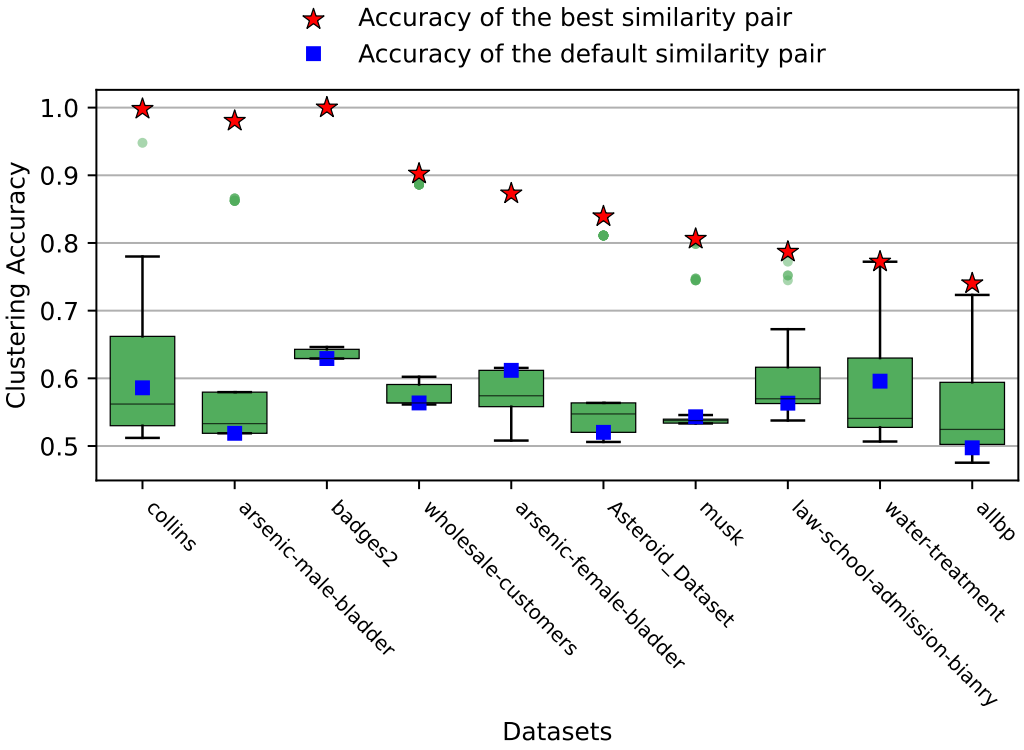


Figure 1.2: Distribution of the clustering accuracy for the K-Prototypes algorithm when using different pairs of similarity measures. Each box plot represents the clustering accuracy scores obtained by 120 different pairs of similarity measures (10 measures for numerical attributes and 12 for categorical ones) on the same dataset. Red stars indicate the accuracy of the best pairs of similarity measures. Blue squares indicate the accuracy obtained with the default pair of similarity measures for the K-Prototypes algorithm (*squared Euclidean distance*, *Hamming distance*)

of improvement compared to the default one. We obtain similar results for other mixed data clustering (MDC) algorithms as shown in chapter 4, confirming the importance of appropriately choosing the pair of similarity measures when clustering mixed datasets.

However, this selection process is complex, especially for non-experts. This complexity arises from two main factors. First, there is "*no free lunch*", i.e. there is no universal pair of similarity measures that performs optimally across all datasets, clustering algorithms, and cluster validity indices. Therefore, when selecting the pair of similarity measures, one needs to consider several factors such as the dataset in question, the clustering algorithm being used, and the cluster validity index to be optimized. Understanding how these factors affect the performance of the different pairs of similarity measures is a significant challenge, even for seasoned experts. Second, there are many alternative choices for the pair of similarity measures, with

a large number of similarity measures that have been proposed in the literature for numerical [Abu Alfeilat 2019] and categorical data types [Boriah 2008, Choi 2009]. Furthermore, in the context of mixed data, the search space is larger than homogeneous data, as two similarity measures must be selected instead of just one. This adds another layer of complexity to the task.

A possible solution is to adopt trial-and-error strategies like grid search and random search which evaluate several possible pairs of similarity measures to identify the most suitable ones. However, due to the high number of pairs of similarity measures, these strategies can be prohibitively slow and expensive, especially if they are applied each time clustering is performed. More advanced search-based approaches such as Bayesian optimisation and evolutionary algorithms that have been used in the context of clustering hyper-parameters optimisation [Poulakis 2024] can also be considered. Although they are more efficient compared to grid or random search, these approaches suffer from the same drawback concerning the computational cost because they need to explore and evaluate the performances of several solutions in order to find the best one. Furthermore, the long wait before receiving feedback on the obtained solution and its performance limits users' productivity, while the high computational cost limits the broad accessibility of these approaches [Garouani 2022].

More recently, approaches based on meta-learning have been proposed in the literature to automatically select the similarity measure when clustering homogeneous datasets (i.e., purely numerical or purely categorical datasets) [Zhu 2020b, Alves 2019]. Meta-Learning is a learning paradigm that studies how learning systems can learn from their experience. Concretely, given a clustering algorithm, these approaches exploit the previous learning experience to learn the relationship between dataset characteristics and similarity measure performances. This way, given a new dataset, the system will be able to recommend suitable similarity measures based only on the characteristics of the dataset. Meta-learning is an efficient alternative for the automatic recommendation of similarity measures for clustering. However, existing strategies are limited to homogeneous datasets and the case of mixed data has not been addressed yet.

The main objective of this thesis is to address the limitations mentioned above and design an efficient approach for the automatic selection of similarity measures when clustering mixed datasets. This objective is accomplished by proposing a meta-learning-based recommendation system able to recommend suitable pairs of similarity measures according to a given MDC algorithm, mixed dataset, and cluster validity index (e.g. clustering accuracy, silhouette, etc.).

1.3 Contributions

This thesis is structured around the following contributions:

- **A comparative study of mixed and homogenization approaches for mixed data clustering.** Homogenization methods are very common in prac-

tice when facing mixed data. However, as stated above, despite providing a convenient way to directly apply standard clustering algorithms, this strategy has several drawbacks, the main one being the modification of the original structure of the converted data which can affect clustering performances. Mixed methods, on the other hand, aim to better handle mixed data. However, integrating the information from the two data types is challenging due to their different nature. The two approaches are very different in how they handle mixed data, and while it is suggested in the literature that mixed methods are preferable because they do not have the same drawbacks as homogenization ones [Ding 2017, Harikumar 2015, Huang 1998, Ji 2013], to the best of our knowledge no comparison study has been proposed yet in the literature in order to either confirm or refute this suggestion. This is the reason why we propose in chapter 3 an experimental framework in which mixed and homogenization methods are compared on various mixed datasets to determine the best approach.

- **Impact of similarity measures on mixed data clustering algorithms.** We propose to evaluate how MDC algorithms, mixed methods in particular, are impacted by the choice of the similarity measures for numerical and categorical attributes. In fact, to define similarity for mixed data, these methods rely on the combination of two similarity measures: one for numerical attributes and another for categorical attributes. There are several possible similarity measures in the literature for each data type. Furthermore, using different similarity measures lead to different clusterings because different similarity measures exploit different information and capture different relations in the considered dataset. So a particular attention should be paid to the choice of the similarity measures. We aim, through this contribution, to evaluate the impact of the choice of the similarity measures on clustering performances and show the importance of choosing the right ones.

- **SIMREC: A similarity measure recommendation system for mixed data clustering algorithms.** This is the main contribution of our work. We propose a similarity measure recommendation system for mixed data clustering algorithms, based on meta-learning. Such a system will help the end users, both non-expert and expert users, select appropriate similarity measures depending on their use cases (dataset, algorithm, and validity index) while avoiding expensive trial-and-error and search-based strategies. To the best of our knowledge this is the first recommendation system proposed in the literature in the context of mixed data clustering. Furthermore, since meta-learning uses datasets characteristics, known as meta-features, to predict the performances of the different pairs of similarity measures, we propose new meta-features that provide more information about the similarity measures and complement those existing in the literature.

1.4 Publications and resources

Several papers have been published during the development of the research for this thesis. These publications present parts of the research contributions detailed in this thesis, particularly focusing on the impact of similarity measures and the development of the SIMREC recommendation system:

- Diop, A., El Malki, N., Chevalier, M., Péninou, A., & Teste, O. (2022, July). Impact of similarity measures on clustering mixed data. In Proceedings of the 34th International Conference on Scientific and Statistical Database Management (pp. 1-12).
- Diop, A., El Malki, N., Chevalier, M., Péninou, A., Geoffrey, R. J., & Teste, O. (2024, July). Similarity Measures Recommendation for Mixed Data Clustering. In Proceedings of the 36th International Conference on Scientific and Statistical Database Management (pp. 1-10).

Additionally, we provide in the following repository [Diop 2024] access to the code for using SIMREC in practice to perform similarity measure recommendations. The repository is thoroughly documented and includes examples demonstrating how SIMREC can be applied to your datasets.

During the thesis, extensive experimental work has been conducted, resulting in a range of resources, including code, experimental results, and more. These materials are available in the same repository ([Diop 2024]) for reproducing experiments or extending SIMREC with new algorithms, similarity measures, and datasets.

1.5 Outline

This manuscript is organized as follows. Chapter 2 presents a detailed literature review on similarity measures, clustering, mixed-data clustering, and hyper-parameter optimization. Chapter 3 presents the comparison between mixed and homogenization methods. In chapter 4, we evaluate the impact of the choice of the pair of numerical and categorical similarity measures on mixed data clustering algorithms. Chapter 5 presents SIMREC, the proposed similarity measure recommendation system for mixed data clustering. Finally, chapter 6 presents a summary of this thesis, its limitations and future research directions.

Chapter 2

Related Works

Contents

2.1 Introduction

The research presented in this manuscript involves several fields in machine learning including similarity measures, mixed-data clustering, hyper-parameter optimization, and meta-learning. This chapter presents the related works in these different fields. Section 2.2 introduces some theory on similarity measures and presents existing similarity measures for different data types (numerical and categorical data in particular). Section 2.3 defines the clustering task and presents a taxonomy of the main clustering methods proposed in the literature. In particular, section 2.4 presents the related works on mixed data clustering. Finally, section 2.5 describes the main approaches for hyper-parameter optimization in the context of clustering, especially meta-learning-based approaches.

2.2 Similarity measures

Measuring similarity or dissimilarity between observations is a core requirement for several data mining and knowledge discovery tasks, especially for clustering [Boriah 2008, Zhu 2020b]. For hierarchical clustering algorithms, the similarity measure determines which clusters should be merged or how a cluster should be split each time. For centroid-based clustering algorithms, the similarity measure determines to which cluster an observation should be assigned. For density-based clustering algorithms, the similarity measure determines the density of objects in the data space. For the most well-known distribution-based clustering algorithms, EM [Ambroise 1998], the similarity measure is used to compute the similarity between an observation and the mean value of a Gaussian distribution, thereby obtaining the probability that the observation is derived for this Gaussian distribution. For spectral clustering algorithms, the similarity measure determines the weights of the edges in the constructed graph. This shows the importance of similarity and dissimilarity measures for clustering algorithms.

2.2.1 Definition

Similarity is an amount that reflects the strength of the relationship between two observations [Irani 2016]. Inversely, a dissimilarity or distance deals with the measurement of divergence between observations. Formally [Deza 2013]:

Definition 1 *Let X be a set. A function $s : X \times X \to \mathbb{R}$ is called a **similarity** on X if, for all $x, y \in X$, there holds:*

- $s(x, y) \geq 0$ *(**non-negativity**)*
- $s(x, y) = s(y, x)$ *(**symmetry**)*
- $s(x, x) \geq s(x, y)$ *with equality if and only if $x = y$ (**Maximality**)*

Definition 2 *Let X be a set. A function $d : X \times X \to \mathbb{R}$ is called a **distance (or dissimilarity)** on X if, for all $x, y \in X$, there holds:*

- $d(x, y) \geq 0$ *(**non-negativity**)*
- $d(x, y) = d(y, x)$ *(**symmetry**)*
- $d(x, x) = 0$ *(**reflexivity**)*

Definition 3 *Let X be a set. A distance or dissimilarity $d : X \times X \to \mathbb{R}$ is called a **metric** on X if for all $x, y, z \in X$, d satisfies the **triangle inequality**: $d(x, z) \leq d(x, y) + d(y, z)$. In this case, X is called a metric space.*

Usually, the similarity is defined in the interval $[0, 1]$ with $s(x, x) = 1$ for all $x \in X$. A dissimilarity measure can be easily transformed into a similarity measure and vice-versa. Let d be a dissimilarity measure. We can define a similarity s from d using the following formula for example:

$$s(x, y) = e^{-\frac{d(x,y)}{\lambda}}, \text{with } \lambda > 0 \tag{2.1}$$

$$s(x, y) = \frac{1}{1 + d(x, y)} \tag{2.2}$$

It is easy to demonstrate that if d is a dissimilarity, then s verifies all the properties of a similarity measure for both equations. Similarly, given a similarity measure s (that takes values in $[0, 1]$), we can define a dissimilarity measure d from s by:

$$d(x, y) = 1 - s(x, y) \tag{2.3}$$

$$d(x, y) = \frac{1 - s(x, y)}{s(x, y)} \tag{2.4}$$

In the following, unless otherwise indicated, the term "similarity measure" is used interchangeably for similarity and dissimilarity measures.

The definition of similarity in practice is highly problem-dependent. Let us consider two different problems which are patient segmentation in medicine and user segmentation in social networks. In patient segmentation, two patients are considered similar if they have similar clinical data, biological data, etc. However, in users segmentation in social networks similarity is based on different notions such as the topics of interest, friends, community, etc. The definition of similarity also highly depends on the considered data type. In fact, different data types exhibit different properties that should be considered in the definition of the similarity measure. For example, similarity measures for numerical data should exploit the order relation between numerical values, similarity measures for image data should exploit the spatial relation in images, similarity measures for time series data should exploit the temporal relation in time series, and so on. Each data type presents some specific properties that can be exploited to define similarity for this data type.

The following sections present various similarity measures for numerical and categorical data. These measures are those that will be considered in the different experiments conducted in the following chapters. They are not exhaustive but have been selected to be as diverse and representative as possible.

2.2.2 Similarity measures for numerical data

Several similarity measures have been proposed in the literature for numerical data. In their study on the effects of similarity measures on the K-Nearest Neighbors (KNN) algorithm, Abu Alfeilat *et al.* [Abu Alfeilat 2019] reported 54 measures, and this was not exhaustive. They also proposed a categorisation of the different similarity measures. Rather than presenting all the measures discussed, we focus here on a subset that represents the main categories identified by the authors. For more examples of similarity measures for numerical data, the reader can refer to the study of Abu Alfeilat *et al.* [Abu Alfeilat 2019] or the *"Encyclopedia of Distances"* of Deza *et al.* [Deza 2013].

2.2.2.1 Euclidean distance

The Euclidean distance is the most commonly used distance measure [Kumar 2014]. It is also known as the L_2 norm. The Euclidean distance, between two numerical data points x and y is defined as:

$$d(x,y) = \left(\sum_{k=1}^{p} (x_k - y_k)^2\right)^{\frac{1}{2}} \tag{2.5}$$

where x_k and y_k represent the k^{th} attributes of x and y respectively. p is the number of numerical attributes. The Euclidean distance is a metric since it satisfies all metric properties. It tends to form hyperspherical clusters. The main strength of this measure is that clusters formed are invariant to translation and rotation in the feature space. One of its disadvantages is that if one of the input attributes has a relatively large range, then it can overcome the other attributes. Therefore, the use of the Euclidean distance generally needs to preprocess the data by normalizing or standardizing all attributes such that they have the same or similar scales.

2.2.2.2 Manhattan distance

The Manhattan distance, also known as L_1 norm, Taxicab norm, rectilinear distance, or City block distance, between two data points is defined as the sum of the absolute differences of their attributes. Given two numerical data points x and y, the Manhattan distance between x and y is defined as:

$$d(x,y) = \sum_{k=1}^{p} |x_k - y_k| \tag{2.6}$$

The Manhattan distance is a metric. It tends to form rectangular-shaped clusters. The advantage of over Euclidean distance is the reduced computation time as shown in [Bora 2014]. However, it has the same drawback concerning attribute scales.

2.2.2.3 Chebyshev distance

The Chebyshev distance, equation 2.7, calculates the maximum of the absolute differences between the attributes of a pair of data points. It is a metric also known as maximum value distance, Lagrange, and chessboard distance. This distance is appropriate in cases when two observations are to be defined as different if they are different for any one of the attributes [Abu Alfeilat 2019]. However, the drawback concerning attribute scales is even more important here because only the attribute with the maximum absolute difference is considered in the distance computation.

$$d(x, y) = \max_{1 \leq k \leq p} |x_k - y_k| \tag{2.7}$$

2.2.2.4 Squared Euclidean distance

The squared Euclidean distance between two observations is the sum of the squared differences between their attribute values without taking the square root as in the Euclidean distance. It is mathematically defined as:

$$d(x, y) = \sum_{k=1}^{p} (x_k - y_k)^2 \tag{2.8}$$

The squared Euclidean distance is not a metric since it does not satisfy the triangle inequality. Compared to the Euclidean distance, the squared Euclidean distance can be used to strengthen (exaggerate) the effect of longer distances [Spencer 2013]. For example, this can have the effect of discouraging the joining of two clusters that have some dissimilar observations even if most of their observations are similar.

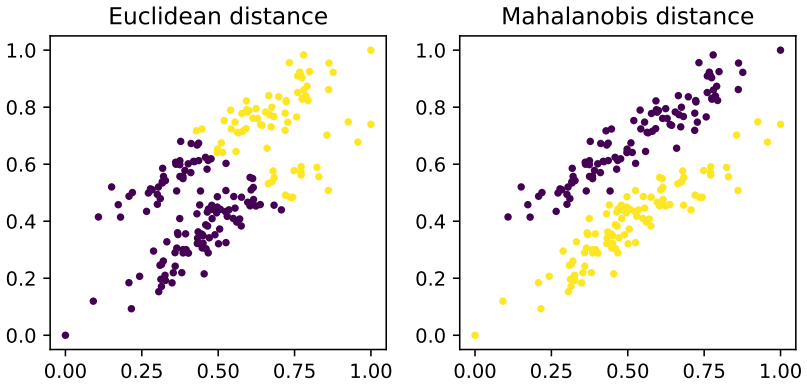
2.2.2.5 Mahalanobis distance

The Mahalanobis distance (or quadratic distance) is a distance introduced by Mahalanobis [Mahalanobis 2018] that is based on the correlations between variables by which different patterns can be identified and analyzed [Kumar 2014]. The Mahalanobis distance between two numerical observations x and y is defined as:

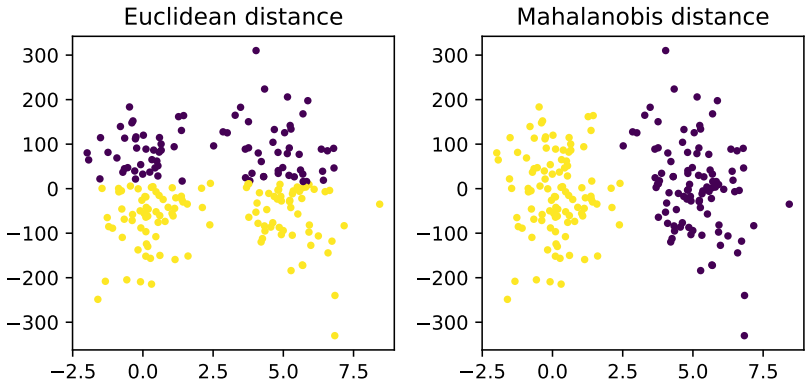
$$d(x, y) = \sqrt{(x - y)\Sigma^{-1}(x - y)^T} \tag{2.9}$$

where Σ is the covariance matrix of the considered dataset. If Σ is the identity matrix, the Mahalanobis distance reduces to the Euclidean distance. If the attributes are independent, then Σ is a diagonal matrix and the Mahalanobis distance reduces to the standardized (or weighted) Euclidean distance where each attribute is weighted by the inverse of its variance. The advantage of the Mahalanobis distance

is that it takes into account the correlations between the dataset attributes (Figure 2.1a) and is scale-invariant (Figure 2.1b). It also tends to form ellipsoidal clusters [Kumar 2014]. Finally, it is a metric if Σ^{-1} is positive-definite [Deza 2013].



(a) Correlated attributes



(b) Attributes of different scales

Figure 2.1: Illustrating how the Mahalanobis distance can help to handle datasets with correlated attributes (a) or with attributes of different scales (b) compared to the Euclidean distance. Each point is an observation. K-Medoids is used for clustering. Colors represent the obtained clusters.

2.2.2.6 Cosine distance

The cosine distance evaluates the dissimilarity between two vectors based on the cosine of the angle between them. Given two vectors x and y, we denote θ the angle between them. The cosine distance is defined as:

$$d(x, y) = 1 - \cos(\theta) = 1 - \frac{\langle x, y \rangle}{\|x\| \cdot \|y\|} \tag{2.10}$$

The cosine distance is not a metric. It varies between 0 and 2. The distance is 0 when the angle between the two vectors is 0. When the angle increases, the distance also increases (because the cosine decreases). The distance is maximum when the

angle is maximum, i.e., the two vectors are opposite.

2.2.2.7 Canberra distance

Given two observations, the Canberra distance measures the sum of absolute fractional differences between their attributes. It is mathematically defined as follows:

$$d(x,y) = \sum_{k=1}^{p} \frac{|x_k - y_k|}{|x_k| + |y_k|} \tag{2.11}$$

The Canberra distance is a metric and is a weighted version of the Manhattan distance. Compared to other distances, it is very sensitive to small changes near 0 due to the normalization [Kumar 2014].

2.2.2.8 Pearson correlation similarity

The Pearson correlation similarity or Pearson's correlation coefficient, is a similarity that measures the linear relationship between two vectors. It is defined by:

$$s(x,y) = \frac{Cov(x,y)}{\sqrt{Var(x) \cdot Var(y)}} = \frac{\sum_{k=1}^{p}(x_k - \bar{x})(y_k - \bar{y})}{\sqrt{(\sum_{k=1}^{p}(x_k - \bar{x})^2) \cdot (\sum_{k=1}^{p}(y_k - \bar{y})^2)}} \tag{2.12}$$

where $\bar{x}$ and $\bar{y}$ are the mean values of x and y respectively. This measure varies between 0 and 1. It tends to disclose the similarity in shapes rather than to detect the closeness of attribute values [Kumar 2014].

The *Pearson distance* is defined from this similarity measure by subtracting it from 1.

$$d(x,y) = 1 - \frac{Cov(x,y)}{\sqrt{Var(x) \cdot Var(y)}} \tag{2.13}$$

2.2.2.9 Divergence

The divergence distance is a weighted version of the squared Euclidean distance. It is defined by:

$$d(x,y) = 2\sum_{k=1}^{p} \frac{(x_k - y_k)^2}{(x_k + y_k)^2} \tag{2.14}$$

2.2.2.10 Lorentzian distance

The Lorentzian distance between two observations is defined as:

$$d(x,y) = \sum_{k=1}^{p} \ln\left(1 + |x_k - y_k|\right) \tag{2.15}$$

This distance corresponds to the natural *log* of the absolute difference between two observations. It is sensitive to small changes since the log scale expands the lower range and compresses the higher range [Abu Alfeilat 2019].

2.2.3 Similarity measures for categorical data

Computing similarity for categorical data instances is not as straightforward as for numerical data due to the fact that there is no explicit notion of ordering between categorical values. The simplest similarity measure is the *overlap similarity* or equivalently the *Hamming distance* which is defined by:

$$d(x,y) = \sum_{k=1}^{q} \delta(x_k, y_k) \text{ such that } \delta(x_k, y_k) = \begin{cases} 0 & \text{if } x_k = y_k \\ 1 & \text{if } x_k \neq y_k \end{cases} \tag{2.16}$$

where x and y are two categorical data points. q is the number of categorical attributes. x_k and y_k are the values of the k^{th} attribute for x and y respectively. Given an attribute, if the two values are identical, the distance is 0, if they are different the distance is 1. There is no intermediate value. However, in many cases, we would expect different levels of similarities. For example, consider the example dataset given in table 1.1, we would expect compatible blood groups such as A+ and A-, to be more similar than non-compatible ones such as A+ and B+. This is not the case with the Hamming distance which gives the same distance in the two situations. Therefore, more complex measures have been proposed in the literature to better capture the similarity between the different values of a categorical attribute. The challenge of these measures is how to define similarity between two values of a categorical attribute when there is no inherent order between these values. To do so, these measures exploit different kinds of information from the dataset such as frequencies, distributions, co-occurrences of categorical values, etc. In this sense, these are data-driven measures. We present in the following subsections, the measures that have been considered in this manuscript. For more examples of similarity measures for categorical data, the reader can refer the the study of Boriah *et al.* [Boriah 2008].

Before that, let us introduce some notations and definitions. Given a categorical dataset X with n observations and q attributes, we denote A^k its k^{th} attribute. The cardinality or number of unique values in A^k is denoted c_k. The frequency of a value $u \in A^k$ is defined as the number of times attribute A^k takes the value u:

$$f_k(u) = |\{x \in X, x_k = u\}| \tag{2.17}$$

2.2.3.1 Eskin distance

This distance has been proposed by Eskin et al. [Eskin 2002]. It exploits the information about the cardinality of the categorical attributes. It considers that the dissimilarity between two values of a given attribute is more important if the attribute has a small cardinality. Inversely, this dissimilarity should be lower for attributes with high cardinality. More formally, the dissimilarity δ between the

values x_k and y_k of the k^{th} attribute of two observation x and y is defined as:

$$\delta(x_k, y_k) = \begin{cases} 0 & \text{if } x_k = y_k \\ \frac{2}{c_k^2} & \text{if } x_k \neq y_k \end{cases} \tag{2.18}$$

The distance between x and y is the defined by $d(x, y) = \sum_{k=1}^{q} \delta(x_k, y_k)$.

2.2.3.2 Inverse Occurrence Frequency (IOF) similarity

The IOF similarity defines the similarity between attribute values based on their frequency. It assigns lower similarity to values that have a high frequency. Inversely, attribute values that have a low frequency are considered more similar. Mathematically, the IOF similarity s_k between the values x_k and y_k of the k^{th} attribute of two observation x and y is defined as:

$$s_k(x_k, y_k) = \begin{cases} 1 & \text{if } x_k = y_k \\ \frac{1}{1+\log f_k(x_k) \cdot \log f_k(y_k)} & \text{if } x_k \neq y_k \end{cases} \tag{2.19}$$

Finally, the similarity between x and y is defined by $s(x, y) = \frac{1}{q} \sum_{k=1}^{q} s_k(x_k, y_k)$. For clustering algorithms that take as input a dissimilarity measure instead of a similarity, we define the IOF dissimilarity between x and y by:

$$d(x, y) = \sum_{k=1}^{q} 1 - s_k(x_k, y_k) \tag{2.20}$$

2.2.3.3 Occurrence Frequency (OF) similarity

The OF similarity is the opposite of the IOF similarity. It assigns higher similarity to values that have a high frequency, and lower similarity to values that have a low frequency. Mathematically, the similarity s_k between the values x_k and y_k of the k^{th} attribute of two observation x and y is defined as:

$$s_k(x_k, y_k) = \begin{cases} 1 & \text{if } x_k = y_k \\ \frac{1}{1+\log \frac{n}{f_k(x_k)} \cdot \log \frac{n}{f_k(y_k)}} & \text{if } x_k \neq y_k \end{cases} \tag{2.21}$$

2.2.3.4 Ahmad and Dey distance

A common drawback in the previous similarity measures is that they do not exploit the relationship between attributes to compute the similarity. The different attributes are considered independently. Ahmad and Dey [Ahmad 2007b] proposed a method to compute the distance between two values of a given attribute based on the fact that the similarity of two attribute values is dependent on their relationship with other attributes. Concretely, two values x_k and y_k of a given attribute A^k are considered similar if they co-occur with similar values in other attributes. Let us consider the example in table 2.1. Values a and b in attribute A^1 occur with the

Table 2.1: Example of a categorical dataset

A^1	A^2
a	e
b	e
b	e
c	f
a	e
c	g

same value (e) in attribute A^2. Therefore, a and b of A^1 are similar according to the attribute A^2. This is the main concept behind the distance proposed by Ahmad and Dey.

Let A^k and A^l be two categorical attributes of a given dataset. Let w denote a subset of values of the attribute A^l. Using set-theoretic notation, ($\sim w$) denotes the complementary set of values occurring for attribute A^l. Let $P(w/A^k = x_k)$ denote the conditional probability that an element having value x_k for A_k, has a value belonging to w for A^l. Using the same notation, $P(\sim w/A^k = x_k)$ denotes the conditional probability that an element having value y_k for A^k has a value belonging to $\sim w$ for A^l. The dissimilarity $\delta(x_k, y_k, A^l)$ between the values x_k and y_k of A^k with respect to the attribute A^l is defined as:

$$\delta^l(x_k, y_k, A^l) = \max_{w \subset A^l} \Big(P(w/A^k = x_k) + P(\sim w/A^k = y_k) - 1 \Big) \tag{2.22}$$

This formula needs some explanation. On the one hand, when x_k and y_k always occur with the same values in A^l their distance with respect to A^l is 0. In fact, in this case we have $\forall w \subset A^l, P(w/A^k = x_k) = P(w/A^k = y_k)$. Therefore, $\forall w \subset A^l, P(w/A^k = x_k) + P(\sim w/A^k = y_k) = 1$, which finally leads to $\delta^l(x_k, y_k, A^l) = 0$.

On the other hand, the distance $\delta^l(x_k, y_k, A^l)$ is maximal when x_k and y_k never occur with the same values in A^l. In fact, $\delta^l(x_k, y_k, A^l)$ is maximal when $P(w/A^k = x_k)$ and $P(\sim w/A^k = y_k)$ are both maximal, i.e. equal to 1. This is only possible if w contains all the values of A^l that occur with x_k and no value of A^l that occur with y_k.

The distance δ between x_k and y_k is defined as the average of δ^l across all attributes A^l different fro A^k:

$$\delta(x_k, y_k) = \frac{1}{q-1} \sum_{l \neq k} \delta^l(x_k, y_k, A^l) \tag{2.23}$$

And the final distance between x and y is defined as:

$$\delta(x, y) = \sum_{k=1}^{q} \delta(x_k, y_k) \tag{2.24}$$

Table 2.2: Binary similarity measures for categorical data. Given two categorical data points that have been transformed into binary vectors, a is the number of dimensions where the binary vectors are both 1; b is the number of dimensions where the first is 1 and the second is 0; c is the number of dimensions where the first is 0 and the second is 1; d is the number of dimensions where they are both 0

Name	Formula
Jaccard	$\frac{a}{a+b+c}$
Dice	$\frac{2a}{2a+b+c}$
Klusinski	$\frac{a}{b+c}$
Rogerstanimoto	$\frac{a+d}{a+2(b+c)+d}$
Russellrao	$\frac{a}{a+b+c+d}$
Sokalmichener	$\frac{a+d}{a+b+c+d}$
Sokalsneath	$\frac{a}{a+2(b+c)}$

2.2.3.5 Binary similarity measures

Categorical data can be represented in a binary form using one-hot encoding. This technique transforms each categorical attribute into several binary attributes such that each binary attribute is used to encode the presence (1) or absence (0) of one of the values of the attribute. Therefore, the number of binary attributes equals the number of unique values of the transformed attribute. In this representation, binary similarity measures [Choi 2009] can be used to measure similarity between observations. We consider 7 binary similarity measures which are presented in table 2.2.

2.2.4 Synthesis and dicussion

Similarity is an important component for several machine learning tasks, especially for clustering. It highly depends on the considered data type since different data types exhibit different properties that can be exploited to define similarity. This section introduced several similarity measures for numerical and categorical data respectively. It also shows the diversity of these measures in how they exploit different properties of the considered data type in order to define similarity. The following section presents the related works on clustering.

2.3 Clustering

Clustering is one of the most important tasks in data mining. It allows the discovery of patterns and structures in datasets without supervision. Concretely, given an unlabeled dataset $X = \{x_1, ..., x_{n_X}\}$, clustering consists in grouping the data into k groups (clusters) $C_1, ..., C_k$ that contain data that are *similar* to each other and *dissimilar* from those in other clusters [Ahmad 2019] with $X = \bigsqcup_{i=1}^{k} C_i$. Clustering has been extensively studied in the literature. As a result, a large number of clustering algorithms have been proposed based on different clustering paradigms, including centroid-based clustering, hierarchical clustering, density-based clustering, model-based, graph-based clustering, etc. We describe these algorithms in the following sections.

2.3.1 Centroid-based algorithms

Centroid-based clustering algorithms group data points into clusters based on their distance from a central point called *centroid*. The general idea is to partition the data into a specified number of clusters, where each cluster is represented by its centroid, typically the mean of the data points in that cluster (for numerical data). The goal is to minimize the distance between data points and their respective centroids while maximizing the distance between the centroids of the different clusters.

Figure 2.2 illustrates the principle of the K-Means algorithm [Macqueen 1967], one of the most used centroid-based algorithms for numerical data. Let $U = (u_{i,j}) \in \mathbb{R}^{n_X \times k}$ be the clustering assignment matrix defined by $u_{i,j} = 1$ if $x_i \in C_j$ and 0 otherwise. The objective of the K-Means algorithm is to find the matrix U that minimizes the intra-cluster distance:

$$\sum_{j=1}^{k} \sum_{i=1}^{n_X} u_{i,j} \cdot d(x_i, c_j) \tag{2.25}$$

where c_j is the centroid of the j^{th} cluster. For the K-Means algorithm, the centroid corresponds to the mean of the observations in the cluster (see equation 2.26). d is the distance measure. Typically the *Euclidean distance* is traditionally used.

$$c_j = \frac{1}{|C_j|} \sum_{i=1}^{n_X} u_{i,j} \cdot x_i \tag{2.26}$$

To minimize the intra-cluster distance, the K-Means algorithm adopts an iterative strategy:

1. **Initialization.** The algorithm starts by initializing the centroids with k randomly chosen observations in the input dataset.
2. **Iteration.** At each iteration, the algorithm performs two operations:
   - **Update of the clustering assignments (U).** The algorithm computes the distances between all observations and the k centroids. Each

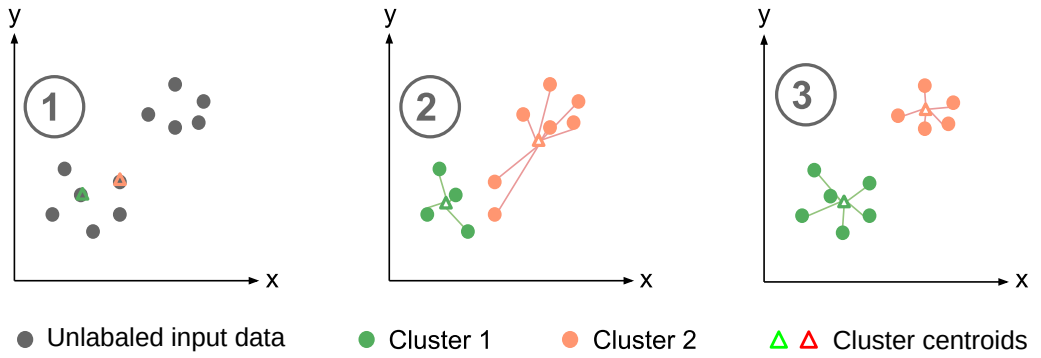


Figure 2.2: Illustration of the K-Means algorithm. Given an unlabeled dataset, the algorithm starts by initializing cluster centers (1). Then each observation is assigned to the cluster with the closest centroid and the position of each centroid is updated as the mean of the observations that have been assigned to its cluster (2). This process iterates (3) until a stopping criterion is met.

observation is then assigned to the cluster with the closest centroid.

- **Update of the centroids.** The centroid of each cluster is updated according to the new clustering assignments using equation 2.26.

3. **End.** The algorithm ends when a stopping criterion is met such as reaching a maximum number of iterations or when a local minimum is reached, i.e. the intra-cluster distance (2.25) does not decrease any more.

Another well-known centroid-based clustering algorithm is K-Medoids [Park 2009, Schubert 2021]. Its main difference with the K-Means algorithm relies on how cluster centers are defined. Instead of the mean, K-Medoids define the center of a cluster as the observation in the cluster that minimizes its distance to other observations in the same cluster (see equation 2.27). Therefore, in K-Medoids, cluster centers correspond to existing observations in the dataset, which is not true for K-Means. Furthermore, compared to K-Means, the K-Medoids algorithm can be used for any data type by considering an adapted dissimilarity measure d.

$$c_j = \underset{x_i \in C_j}{\arg\min} \sum_{x_k \in C_j} d(x_k, x_i) \tag{2.27}$$

Our final example of centroid-based algorithms is K-Modes. It is another variant of the K-Means algorithm used for categorical data. It differs from the K-Means algorithm in the definition of cluster centers and the definition of the dissimilarity between observations. Since the notion of "mean" does not exist for categorical data, centers are represented using mode values. Given a cluster C_j, the mode value of C_j for the categorical attribute A^l is the value of A^l that occurs the most for the

observations in C_j:

$$c_{j,l} = \underset{u \in set(A^l)}{\arg\max} |x_i \in C_j : x_{i,l} = u| \tag{2.28}$$

where $set(A^l)$ is the set of unique values of the attribute A^l.

Centroid-based algorithms are straightforward to implement and easy to interpret. Algorithms like K-Means are particularly efficient, making them well-suited for handling large datasets. However, these methods come with notable limitations. First, the number of clusters must be specified by the user, as it is not automatically determined. Additionally, these algorithms are highly sensitive to the initialization of the cluster centers. Because they use a greedy approach, they often converge to local optima. Poor initialization can trap the algorithm in suboptimal clustering solutions. To mitigate this, multiple runs with different random initializations are typically performed, and the solution that minimizes intra-cluster distances is selected. Another drawback is their sensitivity to outliers, though methods like K-Medoids and K-Modes are more robust in this regard compared to K-Means. Lastly, centroid-based algorithms are not well-suited for clusters with arbitrary shapes or varying sizes [El Malki 2020].

2.3.2 Hierarchical Clustering algorithms

In Hierarchical Clustering algorithms, clusters are represented hierarchically, at different granularity levels, through a tree structure called a dendrogram [Malki 2021]. Hierarchical methods can be categorized into *agglomerative* and *divisive* [Reddy 2014]. Agglomerative methods start with clusters containing each a single observation. These clusters are continuously merged based on their similarity to form larger clusters and build a *bottom-up* hierarchy of the clusters. In contrast, divisive methods begin with a single cluster (containing all observations) that is continuously split into smaller clusters generating a *top-down* hierarchy of clusters. One, quite basic, motivation for using hierarchical clustering is to have a large number of partitions with different granularity levels (corresponding to different levels or cut points in the built hierarchy) [Murtagh 2017]. A different motivation might be to structure one's data in a manner that would be relevant for interpretation as genealogy or as a concept hierarchy or taxonomy.

2.3.2.1 Agglomerative methods

Agglomerative methods consist of consecutive merging of the most similar clusters until a single cluster containing all observations is obtained. The similarity between clusters is measured using a *linkage* criterion that generalizes the similarity between individual observations to a similarity between sets of observations. Let C_1 and C_2 be two clusters. There are three main linkage criteria used in agglomerative methods [Ezugwu 2022]:

- **Single-linkage.** The single-linkage is also referred to as the nearest neighbor or minimum or connectedness method. It measures the nearest distance from

any member of one cluster to any other cluster member. It measures the similarity between two clusters by measuring the closest distance between pairs of cluster elements. The single linkage clustering has a chaining effect with the tendency to produce elongated clusters.

- **Average linkage.** The average linkage is also regarded as the minimum-variance linkage. It finds the mean or median of the distances among all the data points between clusters.

- **Complete linkage.** The complete linkage, also known as the maximum or diameter or farthest neighbor method, determines the distance between two clusters by measuring the longest distance from any member of one cluster to any member of the other cluster. The complete-linkage algorithm clusters are more compact and tightly bound than single-linkage clustering.

2.3.2.2 Divisive methods

Divisive hierarchical clustering is a reverse of the agglomerative clustering process that effectively divides every cluster into smaller chunks beginning with every object in a single cluster until the required number of clusters is attained. The standard method of splitting a cluster into two subsets that contain one or more elements requires the consideration of every likely bi-partition. Though it is normal to analyze all the likely bi-partitions, it is evident that the full enumeration process offers a universal optimum but is very expensive in terms of computation cost [Ezugwu 2022]. Therefore, various divisive clustering approaches that do not consider all bi-partitions have been investigated. These approaches include [Reddy 2014]:

- **Bisecting K-Means.** In the bisecting K-Means, clusters are divided using the K-Means algorithm with $k = 2$. The choice of the cluster to be split at each iteration can be made by considering the cluster with the highest squared error (i.e. the squared within cluster distance).

- **Minimum Spanning Tree-based approach.** In a weighted graph, a minimum spanning tree (MST) is an acyclic subgraph that covers all the vertices with the minimum edge weights. Using the MST in a graph where vertices represent observations and edge weights represent the dissimilarities between observations, a divisive clustering method can be developed which removes the largest weighted edge to get two clusterings and subsequently removes the next largest edge to get three clusterings and so on. This process of removing edges from an MST gives rise to an effective divisive clustering method. The major advantage of this method is that it is able to detect clusters with non-spherical shapes effectively.

2.3.3 Density-based clustering

In density-based clustering, a density-based cluster is a set of data objects spread in the data space over a contiguous region of high-density of objects, separated from other density-based clusters by contiguous regions of low-density of objects [Kriegel 2011].

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [Ester 1996] is the most well-known density-based clustering algorithm [Xu 2015]. Given a distance threshold ϵ and a density threshold $minPts$, the density of a point x_i is defined as the number of points k_i that are within a radius ϵ around x_i. If $k_i > minPts$, the corresponding point x_i is considered a *core point*. Two points are *connected* if they have a distance of less than ϵ. Two points are *density-connected* if they are connected to core points and these core points are, in turn, density-connected. These definitions allow to define the transitive hull of density-connected points, forming density-based clusters [Kriegel 2011]. Points not belonging to any density-based cluster (i.e. not connected to any core point) are considered outliers or noise points.

DBSCAN can extract clusters of arbitrary shapes with filtration of noises [Bhattacharjee 2020] and is very efficient in terms of computational complexity allowing it to handle large data sets [Xu 2015]. However, it is highly sensitive to the values of the 2 parameters ϵ and $minPts$ and cannot find clusters of variable densities. These limitations have been extensively addressed in the literature, leading to several extensions such as OPTICS [Ankerst 1999] and HDBSCAN [Campello 2013]. A comprehensive survey of density-based clustering algorithms is proposed in [Bhattacharjee 2020], for more detail on these algorithms and other density-based clustering algorithms.

2.3.4 Model-based clustering

In model-based clustering [Gormley 2023], data are assumed to be generated by an underlying probability distribution or a model [Ezugwu 2022]. This underlying probability distribution is modelled as a *mixture model*, i.e. a mixture of several component distributions representing each a different cluster. Let f be the underlying probability distribution of the data and $f_1, ...f_K$ components distributions representing the clusters. We have:

$$f(x;\theta) = \sum_{k=1}^{K} \lambda_k f_k(x;\theta_k) \tag{2.29}$$

with the λ_k being the mixing weights, $\lambda_k > 0$, $\sum_{k=1}^{K} \lambda_k = 1$. λ_k represents the probability for a random data point to be drawn from the k^{th} component. θ_k is the parameter vector of the k^{th} component. $\theta = [\theta_1, ...\theta_K, \lambda_k, ...\lambda_K]$ is the over-all parameter vector of the mixture model.

The first step when using a mixture model is to determine its architecture: proper distributions for the different components and the number of components

in the mixture. The mixtures can be constructed with any type of component, but more commonly, multivariate Gaussian densities are used due to its complete theory and analytical tractability [Xu 2005]. The number of components is generally defined as the number of clusters.

After these design choices have been made, one needs to estimate the parameters of the mixture model. Maximum likelihood (ML) estimation is an important statistical approach for parameter estimation. It considers the best estimate as the one that maximizes the likelihood or equivalently the log-likelihood of the data. The likelihood is the probability of observing the data, as a function of the parameters. Let $X = \{x_1, ..., x_{n_X}\}$ denote the considered dataset. Assuming independent samples, the likelihood is defined as:

$$p(X|\theta) = \prod_{i=1}^{n_X} f(x_i; \theta) \tag{2.30}$$

The log-likelihood is defined as:

$$\mathcal{L}(X, \theta) = \log p(X|\theta) = \sum_{i=1}^{n_X} \log f(x_i; \theta) \tag{2.31}$$

Several algorithms can be used to find the parameter vector that maximizes the log-likelihood including the expectation-maximization (EM) algorithm and Bayesian inference [Gormley 2023]. When the gradient of the log-likelihood (w.r.t. the parameters) can be computed, techniques such as gradient descent can also be used.

After the parameters have been estimated, the posterior probability for assigning a data point to a cluster can be easily calculated with Bayes's theorem:

$$p(C_k|x, \theta) = \frac{\lambda_k f_k(x; \theta_k)}{\sum_{l=1}^{K} \lambda_l f_l(x; \theta_l)} \tag{2.32}$$

A popular model-based algorithm is GMM (Gaussian Mixture Models) which is based on the principle described below, using multivariate Gaussian distributions for the different components and the EM algorithm for parameter estimation. Several algorithms have been proposed using different component distributions and parameter estimation techniques. These methods are presented in a recent survey of model-based clustering algorithms proposed in [Gormley 2023].

Model-based clustering algorithms have several advantages including a strong statistical and theoretical background and their ability to produce soft clustering assignments using cluster membership probabilities [Xu 2015]. Furthermore, a particularity of mixture models is that they can generate new data based on the learned density functions. A major limitation of model-based clustering relies on the assumption about component densities and the difficulty in modelling clusters with complex shapes [Gormley 2023, Ezugwu 2022]. Another limitation is the high computational complexity which limits applications to high dimensional and large

datasets [Gormley 2023, Ezugwu 2022].

2.3.5 Graph-based clustering

The concepts and properties of graph theory make it very convenient to describe clustering problems using graphs [Xu 2005]. A graph structure is a data structure comprising nodes and edges connecting the nodes [Ezugwu 2022]. In graph-based clustering, a dataset is represented as a graph where nodes correspond to individual observations and edges reflect the similarities between these observations. Edges are generally associated with weights representing the similarity or dissimilarity between the 2 nodes of the edge. Graph-based clustering algorithms divide nodes into clusters so that the edge density across clusters is smaller compared to the edge density within clusters [Ezugwu 2022]. Nodes are grouped into clusters based on the graph topology so that the output clusters are characterized by high intra-connectivity and low inter-connectivity among the generated clusters.

Zahn [Zahn 1971] proposes a new algorithm based on minimum spanning trees (MST). Given a dataset and its dissimilarity graph (i.e., graph edges are weighted based on objects dissimilarities), the algorithm constructs the graph's minimum spanning tree. After that, it identifies inconsistent edges and removes them from the MST. The remaining connected components are then considered as the clusters provided by the algorithm. An edge is inconsistent if the associated length (weight) is substantially larger than the nearby edges' average length [Ezugwu 2022]. Hartuv and Shamir [Hartuv 2000] treated clusters as highly connected sub-graphs, where "highly connected" means the connectivity (the minimum number of edges needed to disconnect a graph) of the sub-graph is at least half as great as the number of the vertices. A minimum cut (mincut) procedure, which aims to separate a graph with a minimum number of edge cuts, is used to find these highly connected sub-graphs recursively. Cluster Identification via Connectivity Kernels (CLICK) [Sharan 2000] is another example of a graph-based clustering algorithm. In CLICK, graph edges are weighted and the edge weights are assigned a new interpretation, by combining probability and graph theory [Xu 2005]. The minimum weight division is then performed on the graph to generate clusters [Ezugwu 2022].

Graph-based methods typically operate by searching for balanced graph cuts, sometimes invoking notions from spectral graph theory, i.e., using the spectral decomposition of the adjacency or Laplacian matrices of the graph [Liu 2020]. Spectral clustering is an example of graph-based clustering algorithms that rely on spectral graph theory. Spectral clustering embeds the data observations into a vector space spanned by the k eigenvectors corresponding to the k smallest eigenvalues of the similarity graph's normalized Laplacian matrix [Kim 2020]. Local and nonlinear structures can then be accurately grouped by clustering in the embedded space. From the perspective of graph partitioning using cuts, spectral clustering corresponds to solving the relaxation of an NP-hard discrete graph cut problem based on spectral graph theory. For data given in the form of a similarity graph, the problem is to identify an optimal cut such that the edges between different groups

have significantly low weights, and the edges within a group have high weights.

2.3.6 Synthesis and dicussion

This section presented the basic concepts and principles behind the main clustering paradigms. These paradigms correspond to different ways of defining a cluster. For centroid-based clustering, a cluster is considered is a compact group of data points located near the cluster-centroid point and far from the centroids of other clusters. For density-based clustering, clusters are regions of high density of points separated by lower density regions. For graph-based clustering, a cluster is defined as a highly interconnected group of objects that have low connectivity with objects in other clusters. In model-based clustering, clusters are defined as the components of a mixture model representing the data distribution. Which clustering paradigm to consider highly depends on the target application. One should consider the paradigm that better aligns with a cluster means in the context of the target application.

Besides this diversity of definitions, we wanted to point out the core role of similarity and dissimilarity measures in these different paradigms. In centroid-based clustering, similarity measures are used to evaluate the similarity between the objects and the centroids. In hierarchical clustering, similarity measures determine how clusters are merged or split. In density-based clustering, object densities are computed based on their similarities with other objects. Finally, in graph-based clustering, connections between objects are determined based on their similarity. So, for the different paradigms, similarity measures have a central role and a direct impact on the created clusters.

From the section 2.2 on similarity measures, we observed the diversity of existing similarity measures and how different measures exploit different properties of the data. These observations and the central role of similarity measures for the different clustering paradigms raise the question of the appropriate selection of these measures. This thesis addresses this question in the context of mixed data clustering. We present the related works on mixed data clustering in the following.

2.4 Mixed-data clustering

When clustering mixed datasets, the main concern relies on how to jointly handle and integrate numerical and categorical attributes. This is challenging because of the different natures and the different properties of the two types of attributes. A taxonomy of mixed data clustering (MDC) algorithms is proposed in figure 2.3 based on how these algorithms handle and integrate mixed numerical and categorical attributes. Existing approaches in the literature can be broadly divided into two groups [Barcelo-Rico 2012, Ji 2013, Wei 2015]:

- Methods based on data conversion called *homogenization methods*

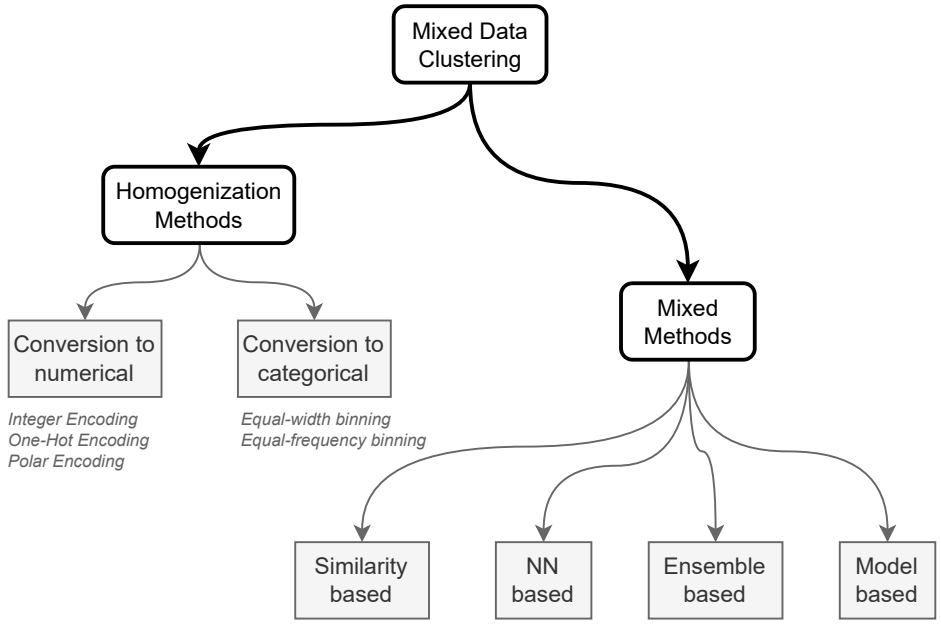


Figure 2.3: A taxonomy of mixed data clustering algorithms

- Methods that handle mixed data directly without any conversion named *mixed methods*.

2.4.1 Homogenization methods

Homogenization methods are composed of two main steps: 1) the conversion step and 2) the clustering step. During the conversion step, the attributes are converted to a single target type that is either numerical or categorical. Once the data are converted, the clustering step simply applies a standard clustering algorithm for the target type to compute clusters. This algorithm can be any of those presented earlier in section 2.3. Therefore, the main difference between homogenization methods relies on the used conversion technique. We distinguish conversion techniques that convert categorical attributes into numerical and those that convert numerical attributes into categorical. Different conversion techniques have been proposed in the two cases. We present these techniques in sections 2.4.1.1 and 2.4.1.2 respectively referring to the work of Barcelo-Rico and Diez [Barcelo-Rico 2012] and Liu *et al.* [Liu 2002]. Figure 2.4 provides graphical illustrations of the different techniques for a better understanding.

2.4.1.1 Conversion of categorical attributes into numerical ones

Integer encoding. This is the simplest strategy to convert categorical attributes into numerical ones. Given a categorical attribute, the conversion is done by associating an integer number to each category of the attributes. For example, the categories {A+, B+, A-, B-} of the attribute *Blood Group* are replaced by the integer values {0, 1, 2, 3} respectively. Despite its simplicity, this conversion technique has a major drawback which is the creation of an inexistent order between categorical values which can bias the similarities between these values.

One-hot encoding. Given a categorical attribute, this conversion technique transforms the attribute into several binary attributes. Each binary attribute corresponds to one of the values of the original attribute. When the original attribute takes a given value, the binary attribute corresponding to this value equals 1 and all other binary attributes equal 0. This is illustrated in figure 2.4. *One-hot encoding* has been used in [Ralambondrainy 1995] to convert mixed data into numerical before using K-Means to perform clustering. Its advantage compared to *integer encoding* is that it does not introduce an order between categorical values. However, its main drawback is the important augmentation of the dimension of the data. For example, if a categorical attribute has 10 categories, 10 binary attributes will be created. Besides the augmentation of the dimension of the data, this can also result in giving more importance to categorical attributes compared to numerical ones during the distance computations.

Polar and spherical codification. In [Barcelo-Rico 2012], authors propose a method that uses polar or spherical coordinates to convert categorical attributes into numerical ones and then uses K-Means clustering on the new numerical data set. The categories of each categorical attribute are transformed into 2D (resp. 3D) vectors by placing them on the unit circle (resp. sphere) such that they are regularly spaced. This strategy has the advantage of not increasing a lot the data set dimension. However, it still creates an order since the obtained representations for categories are not equidistant in the target space.

2.4.1.2 Conversion of numerical attributes into categorical ones

Discretization techniques are generally used to convert numerical data into categorical. Discretization of unsupervised data has attracted less attention in the literature compared to supervised data [Liu 2002]. Given a numerical attribute, the main adopted technique in the unsupervised framework is to divide the attribute range into a finite number of intervals based on a user-specified width (range of values) or frequency (number of instances in each interval) [van de Velden 2019, Liu 2002]. The obtained intervals are then considered as the categories of the newly created categorical attribute. The issue here is how to define the intervals, their width, frequency, etc. In some fields and for some attributes (such as *age*), we may have some

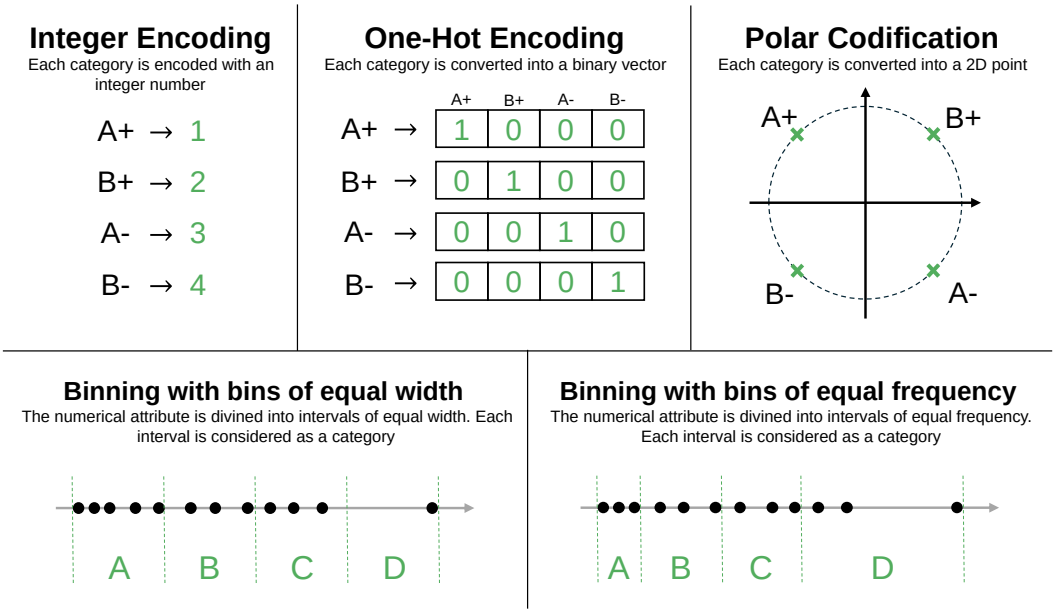


Figure 2.4: Illustration of different conversion techniques to transform categorical attributes into numerical (top) and vice versa (bottom).

predefined intervals (e.g., using age groups; non-adult, young adults, elderly etc.) that give meaningful categories. However, in many cases, such intervals are not present, complicating the discretization process and the rationale behind choosing specific interval widths or frequencies [van de Velden 2019]. Besides this complexity, the obtained interval yields less precise information than the original numerical values [van de Velden 2019], leading to important information loss. Additionally, treating intervals as categories often disregards the inherent order among them, thus ignoring the similarity implied by this ordinal relationship.

2.4.2 Mixed methods

In contrast with homogenization methods, mixed methods do not rely on data conversion and aim to consider the specific properties of the different data types. Therefore, the main challenge of mixed methods is how to integrate or combine mixed numerical and categorical attributes effectively without converting the data. We distinguish 4 categories of mixed methods in the literature depending on how the two data types are combined during the clustering process: 1) similarity-based methods that combine the two data types at the similarity level, i.e. when defining the similarity measure for mixed data 2) methods that use deep neural networks to learn a latent representation of mixed data and perform clustering [Zhu 2020a, Lee 2023, Jian 2018, Balaji 2020], 3) ensemble-based methods that perform clustering separately for numerical and categorical attributes and combine the two obtained clustering [He 2005, Suguna 2012], and 4) model-based methods that use a specific probabilistic model for each data type

[Cheeseman 1997, Moustaki 2005, McParland 2016].

2.4.2.1 Similarity-based methods

These methods define similarity for mixed data by combining two numerical and categorical similarity measures. The numerical similarity measure is used for numerical attributes while the categorical similarity measure is used for categorical attributes. The combination is generally done using a linear combination while a weight is assigned to each attribute.

In [Philip 1983], the Gower similarity [Bishnoi 2020] is used with the hierarchical clustering algorithm. The Gower similarity is a similarity measure for mixed data defined as a weighted combination of the *Manhattan distance* for numerical attributes and the *Hamming distance* for categorical attributes. Numerical attributes are weighted inversely to their interval range to give the same importance to all attributes.

In [Huang 1998], Huang introduced K-Prototypes which extends the K-Means algorithm with a new representation of cluster centers and a new definition of similarity. The similarity is defined as a weighted sum of the *squared Euclidean distance* for numerical attributes and the *Hamming distance* for categorical ones. A unique weight is associated with all categorical attributes giving different importance to numerical and categorical attributes. However, this weight needs to be defined manually. Later, several works [Huang 2005, Ahmad 2007a, Ji 2013] based on K-Prototypes have been proposed in the literature to automatically compute the weights of the different attributes during the clustering. In addition, in [Ahmad 2007a, Ji 2013] the *Ahmad and Dey distance* [Ahmad 2007b] is used for the categorical data instead of the *Hamming distance*, and in [Ji 2013] the *Manhattan distance* is used for numerical data instead of the *squared Euclidean distance*. Based on the same idea of combining two numerical and categorical similarity measures, several other clustering algorithms have been extended to mixed data such as K-Medoids [Harikumar 2015, Budiaji 2019], Fuzzy C-Medoids [D'Urso 2019], Spectral Clustering [Mbuga 2022], and Density-Based Clustering [Du 2017, Ding 2017].

Hsu et al. [Hsu 2007] propose a new similarity measure for mixed data based on distance hierarchy. A tree-based representation is adopted for each attribute such that the tree nodes correspond to the attribute values and link weights represent the similarity between two nodes. The similarity between two values of a given attribute is measured by the total link weight of the path between the corresponding nodes in the tree or hierarchy. This allows to unify how similarity is computed for numerical and categorical values. However, the main drawback is that the link weights must be defined by the data analyst or a domain expert. A similar approach has been proposed recently by [Zhang 2023] using graphs instead of trees to represent attributes. The link weights are computed automatically however the approach can be time-intensive.

Similarity-based methods allow simple and comprehensive integration of mixed numerical and categorical attributes while considering the specific properties of

each attribute type through the use of adapted similarity measures. However, the physical interpretation of the obtained similarity measure for mixed data can be problematic as the physical meanings of numerical and categorical similarity measures are different [Wei 2015].

2.4.2.2 Neural network-based methods

These methods exploit the representation ability of deep neural networks to learn a latent representation of mixed data and perform clustering. Some methods learn the latent representation independently of the clustering process while others jointly optimize the latent representation and the clustering. Learned data representation in an unsupervised framework is challenging due to the absence of ground truth labels to guide the learning process. However, existing strategies address this gap using self-supervised learning. This learning paradigm uses the data itself to generate supervisory signals, rather than relying on external labels provided by humans. For example, Auto-Encoders (AE) are based on this paradigm. Auto-encoders are composed of two models: an encoder that takes the input data and learns a lower-dimensional latent representation and a decoder that learns to regenerate the original input data from the latent representation. The two models are trained jointly, the objective being to learn a latent representation that captures the essential features or structure of the data such that the original input data can be reconstructed from it. Another challenge these methods face is the heterogeneity of numerical and categorical attributes. The neural network should be adapted to handle mixed data, especially categorical attributes suitably.

Zhu *et al.* [Zhu 2020a] introduces Mix2Vec, an unsupervised deep learning model designed to learn high-quality representations of mixed data that can be used for mixed data visualization and in several downstream tasks including clustering and K-Nearest Neighbors classification. Mix2Vec use different encoding layers for numerical and categorical attributes before learning a joint representation. The model is trained through mechanisms like random shuffling prediction, prior distribution matching, and structural informativeness maximization, which are different forms of self-supervision.

Balaji *et al.* [Balaji 2020] introduce a Constraint-Based Deep Convolutional Generative Adversarial Network (CB-DCGAN) framework designed for mixed data, which generates synthetic data to enhance the training set and improve clustering algorithm performance. This augmented data is then input into a Deep Convolutional Neural Network (DCNN) encoder, which jointly learns both a low-dimensional representation of the data and the cluster centroids. However, the paper does not clearly explain how the numerical and categorical attributes are represented or handled by the network. Additionally, a significant limitation of this approach is the absence of a regularization mechanism during the clustering step (i.e., the learning of the low-dimensional representation of the data and cluster centroids) to prevent the model from distorting the original data space and producing clusters that do not reflect the original structure of the data.

Lee *et al.* [Lee 2023] propose a novel deep embedded clustering framework for mixed data. The proposed approach jointly learns low-dimensional feature representations and optimizes the clustering goals. A deep auto-encoder model is used to learn the low-dimensional feature representations. Besides the Auto-Encoder, the model also has additional trainable parameters indicating cluster centers in the latent space. The proposed approach has two main steps: 1) a pre-training step where only the reconstruction loss is optimized and 2) a clustering step where the reconstruction loss is jointly optimized with a clustering loss defined in the latent space. To handle mixed data suitably, adapted activation functions and reconstruction losses are used for numerical (linear activation and mean squared error loss) and categorical (softmax activation and cross-entropy loss) attributes.

The representation ability of deep neural networks makes them powerful in improving clustering accuracy [Zhang 2023]. Furthermore, the learned representation of mixed data can be used for several tasks such as mixed data visualization or determining semantic similarities between mixed data. However, despite these advantages, the inherent weakness of deep neural networks in terms of interpretability may somewhat limit their applications, especially for clustering-based data understanding, knowledge acquisition, and so on [Zhang 2023].

2.4.2.3 Ensemble-based methods

These algorithms consider numerical and categorical attributes separately. Given a mixed dataset, clustering is first performed on numerical attributes using adapted clustering algorithms such as K-Means. Then, another clustering algorithm adapted for categorical data such as K-Modes is used to perform clustering on categorical attributes. The two clustering results are then combined to obtain the final clusters.

He *et al.* [He 2005] propose a novel divide-and-conquer technique to solve the mixed data clustering problem. First, the original mixed dataset is divided into two sub-datasets: the pure categorical dataset and the pure numerical dataset. Next, existing well-established clustering algorithms designed for different types of datasets are employed to produce corresponding clusters. Last, the clustering results on the categorical and numerical datasets are combined as a categorical dataset, on which the categorical data clustering algorithm is used to get the final clusters.

Suguna and Selvi [Suguna 2012] propose an unsupervised ensemble fuzzy clustering approach that permits the exploitation of both the flexibility of the fuzzy sets and the robustness of the ensemble methods. They use an ensemble clustering approach that merges the results of two different fuzzy clustering algorithms namely Fuzzy C-Means and Fuzzy C-Modes.

Ensemble-based methods provide a framework for directly applying standard clustering algorithms for numerical and categorical data respectively, without converting the data. However, since they integrate the information from the two data types after performing clustering on each data type, they cannot consider the intercorrelation between numerical and categorical attributes which may be essential for

identifying certain clusters.

2.4.2.4 Model-based methods

These methods are based on the model-based clustering paradigm (section 2.3.4). They combine numerical and categorical attributes by using adapted probabilistic models for each attribute type. Cheeseman and Stutz [Cheeseman 1997] propose AUTOCLASS, an algorithm that performs clustering by integrating finite mixture distribution and Bayesian methods with a prior distribution of each attribute. Different distributions are used depending on the attribute type. Gaussian distributions are used for real-valued numerical attributes (such as weight), Mises-Fisher distributions for circular or angular real-valued numerical attributes, and Poisson distribution is used for integer count-valued attributes. For categorical attributes, the generalized Bernoulli distribution is used on the cross-product of individual attribute values. Clusters are then formed by finding the parameters of the joint distribution that better fit the data for each cluster. In [Moustaki 2005] a latent class mixture model is used for clustering Archaeological data with mixed numerical and categorical attributes. Categorical attributes are assumed to follow a multinomial distribution while numerical attributes are assumed to follow a normal distribution.

Foss *et al.* [Foss 2016] develop KAMILA (KAy-means for MIxed LArge data sets), a semi-parametric method for clustering mixed data. The KAMILA algorithm integrates two different kinds of clustering algorithms; the K-means algorithm and Gaussian-multinomial mixture models. Like the K-means clustering algorithm, no strong parametric assumptions are made for numerical features in the KAMILA algorithm. Like Gaussian-multinomial mixture models (using Gaussian and multinomial distributions for numerical and categorical attributes respectively), KAMILA can successfully balance the contribution of continuous and categorical variables without specifying weights.

The main limitation of model-based methods is that they cannot represent complex data distributions, making them unsuitable for detecting clusters of complex shapes [Gormley 2023, Ezugwu 2022]. Furthermore, their high time complexity limits their applications for large or high-dimensional datasets [Gormley 2023, Ezugwu 2022].

2.4.3 Synthesis and dicussion

In this section, we presented the related work in the context of mixed data clustering. A taxonomy of the different mixed data clustering algorithms is proposed based on how they handle or integrate mixed attribute types. This taxonomy is composed of 2 main groups, namely conversion-based and non-conversion-based algorithms, which we refer to as *homogenisation* and *mixed methods* respectively. Homogenization methods are suitable for applying standard clustering algorithms for single-type or homogeneous data. However, they present several drawbacks that pertain to the modification of the original structure of the data and may lead to sub-optimal

Table 2.3: Characteristics of the presented similarity-based methods.

Paper	Year	Algorithm Family	Similarity for numerical attributes	Similarity for categorical attributes	Definition of Attributes' weights
[Philip 1983]	1983	Hierarchical	Manhattan distance	Hamming distance	Inverse attribute range
[Huang 1998]	1998	Partitional	squared Euclidean distance	Hamming distance	By the user
[Huang 2005]	2005	Partitional	squared Euclidean distance	Hamming distance	Automatic based on within cluster distances
[Ahmad 2007a]	2007	Partitional	squared Euclidean distance	Ahmad and Dey distance	Automatic
[Hsu 2007]	2007	Hierarchical	Distance hierarchy	Distance hierarchy	By the user
[Ji 2013]	2013	Partitional	Manhattan distance	Ahmad and Dey distance	Automatic based on within cluster distances
[Harikumar 2015]	2015	Partitional	Manhattan distance	Ahmad and Dey distance	1 for all attributes
[Du 2017]	2017	Density based	Euclidean distance	Hamming distance	Automatic based on entropy
[Ding 2017]	2017	Density based	Euclidean distance	Hamming distance	Automatic based on entropy
[Budiaji 2019]	2019	Partitional	Any	Any	By the user
[D'Urso 2019]	2019	Fuzzy Clustering	Any	Any	Automatic
[Mbuga 2022]	2022	Spectral Clustering	Euclidean distance	Hamming distance	By the user
[Zhang 2023]	2023	Partitional	Graph-based similarity	Graph-based similarity	Automatic

performances.

On the other hand, mixed methods rely on adapted strategies that can handle mixed data directly while considering the properties of each data type. We distinguish similarity-based, neural-network-based, ensemble-based, and model-based methods. As shown above, each method has its advantages and limitations. In this thesis, we are mainly interested in similarity-based mixed methods which offer a better compromise between clustering quality and interpretability compared to other methods.

A summary of similarity-based methods and their characteristics is available in table 2.3. The main challenge of similarity-based methods is how to appropriately choose or define the similarity measure for each data type and how to integrate these measures to define a global similarity measure for mixed data. This is illustrated in the table by the diversity of used similarity measures across the different papers, which underscores again the importance of choosing the right similarity measures when performing clustering, especially in the context of mixed data. Since the numerical and categorical similarity measures are among the different hyper-parameters of similarity-based mixed data clustering algorithms, the problem of similarity measures selection can be seen as a particular case of the hyper-parameter optimization (HPO) problem. Therefore, existing HPO techniques can be used to select optimal similarity measures. We present these techniques in the following.

2.5 Hyper-parameter optimization for clustering

Hyper-parameter optimization is an important task for clustering and machine learning algorithms since hyper-parameter values highly affect algorithm performances. Common hyper-parameters in clustering include the number of clusters, the similarity measures, the linkage criterion for hierarchical algorithms, etc.

Let A be a clustering algorithm and Q be a cluster validity index. Given a dataset X, we denote $Q(A, X, \theta)$ the measured clustering performance when running A on dataset X with the hyper-parameter vector θ. The goal of hyper-parameter optimization is to find the optimal parameters θ^* that maximize the performance of the considered algorithm for the given dataset:

$$\theta^* = \max_{\theta \in \Theta} Q(A, X, \theta) \tag{2.33}$$

where Θ is the set of possible parameter values. This problem poses several challenges. First the objective function $\theta \mapsto Q(A, X, \theta)$ has no analytical form. So, the only way to evaluate it is to run the clustering algorithm on the dataset with the specified parameter values and then evaluate the result. Consequently, many traditional optimization methods designed to solve convex or differentiable optimization problems are unsuitable in this case. Furthermore, the evaluation of the objective function can be computationally expensive depending on the dataset size and clustering algorithm complexity. This can highly slow down the search for the optimal hyper-parameters, especially when the hyper-parameter space is large. Second, the hyper-parameters may have various types including continuous, discrete, categorical, etc. Thus, many traditional numerical optimization methods that only aim to tackle numerical or continuous variables are unsuitable for solving the HPO problem.

In the following, we present different methods used in the literature to solve the HPO problem and show how meta-learning can be used as an alternative to traditional HPO techniques.

2.5.1 Traditional hyper-parameter optimization techniques

2.5.1.1 Grid search

Grid search (GS) is one of the most commonly used methods to explore hyper-parameter configuration space [Yang 2020]. GS can be considered an exhaustive search or a brute-force method that evaluates all the hyper-parameter combinations given to the grid of configurations. GS works by evaluating the Cartesian product of a user-specified finite set of values. It is the most straightforward search algorithm that leads to the most accurate predictions [Yu 2020]. As long as sufficient resources are given, the user can always find the optimal combination. GS can be easily implemented and parallelized. However, the main drawback of GS is its inefficiency for high-dimensionality hyper-parameter configuration space, since the number of evaluations increases exponentially as the number of hyper-parameters

grows [Yang 2020]. This exponential growth is referred to as the curse of dimensionality. For GS, assuming that there are k parameters, and each of them has n distinct values, its computational complexity increases exponentially at a rate of $O(n^k)$. Thus, GS can be an effective HPO method only when the hyper-parameter configuration space is small.

2.5.1.2 Random search

Random search (RS) [Bergstra 2012] is a basic improvement on grid search. Instead of testing all values in the search space, RS randomly selects a pre-defined number of samples between the upper and lower bounds as candidate hyper-parameter values, and then trains these candidates until the defined budget (e.g., resources or time constraint) is exhausted [Yang 2020]. The theoretical basis of RS is that if the configuration space is large enough, then the global optimums, or at least their approximations, can be detected. With a limited budget (execution), RS is able to explore a larger search space than GS [Bergstra 2012].

Unlike GS, RS samples a fixed number of parameter combinations from the specified distribution, which improves system efficiency by reducing the probability of wasting much time on a small poor-performing region. Since the number of total evaluations in RS is set to a fixed value n before the optimization process starts, the computational complexity of RS is $O(n)$. Although RS is more efficient than GS for large search spaces, there are still a large number of unnecessary function evaluations since it does not exploit the previously well-performing regions.

2.5.1.3 Bayesian optimization

Bayesian Optimization (BO) [Snoek 2012] is an iterative algorithm widely used for hyper-parameter optimization. This optimization technique targets the global optimization of black-box functions that are expensive to evaluate [Poulakis 2024]. Unlike grid search (GS) and random search (RS), BO selects future evaluation points based on past results, using two key components: a surrogate model and an acquisition function [Yang 2020]. The surrogate model approximates the objective function using the observed data, while the acquisition function balances exploration (testing unexplored areas) and exploitation (focusing on promising regions) to choose the next configuration. Common surrogate models include Gaussian Process, Random Forest, and Tree Parzen Estimator.

The basic procedures of BO are as follows [Yang 2020]:

1. Build a probabilistic surrogate model of the objective function.
2. Identify optimal hyper-parameters on the surrogate model.
3. Evaluate these hyperparameters on the real objective function.
4. Update the surrogate model with new results.
5. Repeat until a stopping criterion is met (e.g., max iterations).

This approach is more efficient than GS and RS because it reuses past results and running a surrogate model is cheaper than evaluating the full objective function. However, BO is harder to parallelize as it depends on sequential evaluations; but it can usually detect near-optimal hyper-parameter combinations within a few iterations [Yang 2020].

2.5.1.4 Meta-heuristic algorithms

The term meta-heuristic describes higher-level heuristics that are proposed for solving a wide range of optimization problems [Dokeroglu 2019]. Unlike traditional optimization techniques, meta-heuristic algorithms can solve non-convex, non-continuous, and non-smooth optimization problems. Furthermore, they are able to obtain the best/optimal solutions even for very large problem sizes in small amounts of time.

Meta-heuristic algorithms are based on two main components [Gandomi 2013]: exploration and exploitation. Exploration (or diversification) means generating diverse solutions to explore the search space on a global scale while exploitation (or intensification) means focusing on the search in a local region by exploiting the information that a current good solution is in this region. The overall efficiency of meta-heuristic algorithms is mainly influenced by a fine balance between these two components.

A major type of meta-heuristic algorithms are population-based optimization algorithms (POAs) including genetic algorithms and particle swarm optimization which are popularly used for HPO problems [Yang 2020]. POAs operate by iteratively evolving a population of candidate solutions across multiple generations until the global optimum is identified. The population of each generation is generated based on the previous generation by balancing between exploration and exploitation. The main distinction between different POAs lies precisely in how they generate, update, and select individuals within the population across generations. POAs are easy to parallelize, as each member of the population can be evaluated independently on separate threads or machines.

2.5.1.5 Limitations

From the previous sections, it can be observed that most HPO techniques are search-based algorithms in the sense that they need to explore the hyper-parameter space and evaluate the performances of several solutions to find the best one. Search-based approaches have an inherent drawback which is their high computational cost, especially in our context where the evaluation of each single solution involves running the clustering algorithm on the entire dataset. This high computational cost can be an important obstacle to using HPO in practice when enough computational resources are not available or for tasks that involve important interactions with the user such as exploratory data analysis. To address these limitations, alternative approaches [Poulakis 2024, Feurer 2014,

Pimentel 2020, Alves 2019, Zhu 2020b, Garouani 2022] have been proposed in the literature using meta-learning [Vanschoren 2018]. We present these approaches in the following section.

2.5.2 Meta-Learning

In traditional HPO techniques, there is no mechanism to create knowledge and learn from previously faced datasets. The same optimization process is repeated for each new dataset. However, one could exploit the similarity between the new dataset and previously faced ones to find the optimal hyper-parameters for the new dataset more efficiently. This can be done using Meta-learning.

Meta-learning is a machine-learning paradigm that learns how learning systems can increase in efficiency through experience. Meta-learning takes inspiration from how we humans exploit our previous learning experiences in our learning of new tasks. In fact, when we learn new skills, we rarely - if ever - start from scratch. We start from skills learned earlier in related tasks, reuse approaches that worked well before, and focus on what is likely worth trying based on experience [Vanschoren 2018]. With every skill learned, learning new skills becomes easier.

We describe in the following how meta-learning works and how it has been used in the literature for clustering algorithm recommendation and hyper-parameter optimization.

2.5.2.1 Meta-learning principle: case of algorithm recommendation

Meta-learning methods have been widely studied in the field of algorithm selection as support tools for machine learning practitioners [de Souto 2008, Pimentel 2020]. Suppose we have to perform a machine learning task, let's say clustering, on a given dataset. There may exist several candidate algorithms for this learning task. However, depending on the dataset and its characteristics, some algorithms may be preferable compared to others. Without initial experience, it is necessary to evaluate all candidate algorithms to find the most appropriate for the current dataset. The more datasets we face, the more experience we gain on which algorithms are most suitable for which datasets. Meta-learning allows to exploit this experience to learn how to select a suitable algorithm depending on the faced dataset or more precisely, its characteristics also known as *meta-features*.

The first step for meta-learning is to create a knowledge database named *meta-dataset* that stores all the knowledge obtained from prior learning experiences. This includes information about the datasets, their meta-features, and the performances of the candidate algorithms on these datasets. The second step is to use the created meta-dataset to train a machine learning model named *meta-model* that learns the relationship between datasets' meta-features and algorithm performances. Once trained, this model can be used when facing a new dataset, to predict the most suitable algorithm based only on the meta-features of the dataset.

The first application of meta-learning to clustering is from [de Souto 2008], in

the context of clustering algorithm recommendation. Given a dataset, the proposed approach provides a ranking of the 7 candidate algorithms. They used 8 meta-features based on statistical measures and evaluated their framework using 32 micro-array datasets about cancer gene expression. Their results suggest that the proposed approach performs better than a baseline based on average ranking. Later studies have mainly worked on designing new meta-features to enhance predictive capability. In [Ferrari 2015], a new set of meta-features based on the distribution of similarity between observations is proposed to extract more information about the internal structure of the datasets. Vukicevic *et al.* [Vukicevic 2016] introduce meta-features based on internal CVIs (cluster validity indices). Pimentel and de Carvalho [Pimentel 2019] proposed new meta-features based on correlation and dissimilarity measures. Gabbay *et al.* [Gabbay 2021] propose two new sets of meta-features. The first is based on the combination of isolation forest embedding and singular value decomposition. The second set is based on the landmarking approach. Landmarking is a technique used to indirectly characterize a dataset, by applying landmarkers on the dataset. A landmarker is a simple and efficient learning algorithm whose performance on the dataset is used to characterize it. In [Poulakis 2020], new landmark meta-features are proposed based on the performances of the MeanShift algorithm measured by internal cluster validity indices. The similarity based on the values in these internal cluster validity indices is expected to reveal datasets with similar structural properties. In [ElShawi 2022] meta-learning is used to initialize the population of a genetic algorithm for the selection of the optimal clustering pipeline (preprocessing and clustering algorithm). As in [Poulakis 2020] landmark meta-features are proposed by considering DBSCAN and OPTICS algorithms in addition to MeanShift. In [Cohen-Shapira 2021] a different paradigm is proposed to define the meta-features. Instead of using hand-crafted meta-features, authors propose to learn the meta-features directly from the datasets using neural networks. This allows to automatically extract characteristics that better describe the datasets for the considered meta-learning task. Concretely, first, a graph representation is adopted to represent datasets. Then a graph convolutional neural network (GCNN) technique is used to generate an embedding representation of each graph. The neural network weights are learned such that datasets with the same best-performing clustering algorithm have similar embeddings. However, the main drawback of this approach, inherent to deep neural networks, is how to interpret the learned meta-features.

2.5.2.2 Meta-learning for hyper-parameter optimization

Meta-learning can be used in 2 ways for hyper-parameter optimization. The first usage is to warm-start traditional hyper-parameter optimization algorithms [Poulakis 2024]. This is likely to provide benefits, for example, by providing better initializations or reducing the hyper-parameter search space to the most promising solutions. From our research, this use case has been only considered in the context of hyper-parameter optimization for supervised algorithms [Feurer 2014],

which is out of the scope of this manuscript. However, results showed that using meta-learning to warm-start the optimization can highly increase the convergence speed. Therefore, it could be interesting to use such strategies to enhance HPO techniques for clustering algorithms.

The second usage is to use meta-learning to directly recommend the optimal hyper-parameters of a given algorithm according to the faced dataset. This is similar to how meta-learning has been used for clustering algorithm recommendation. The idea is to use prior HPO tasks to learn how to automatically recommend optimal hyper-parameters for newly faced datasets (without running HPO on these new datasets) based only on their meta-features. In [Pimentel 2020], meta-learning is used for recommending the number of clusters (for the K-Means algorithm). New meta-features based on density distribution are introduced since they may convey information about the number of clusters. The meta-model is built as an ensemble model composed of 10 sub-models. The results show the interest of the proposed meta-features for recommending the number of clusters, as well as the fact of combining several learners. In [Alves 2019], meta-learning is used for similarity measure recommendation for clustering categorical data. The authors used statistical meta-features about the datasets and their attributes. They considered 10 similarity measures for categorical data and validated their approach using one clustering algorithm (Hierarchical Clustering) and 60 synthetic datasets. Zhu *et al.* [Zhu 2020b] propose a meta-learning approach to recommend similarity measures for clustering numerical data. Besides statistical meta-features, they also use structural information-based and distance-based (distribution of the Euclidean distance between pairs of observations) meta-features. They considered 9 similarity measures for numerical data and validated their approach using two clustering algorithms (K-Means and CURE) and 199 datasets.

2.5.3 Synthesis and dicussion

In this section, we reviewed the related works on hyper-parameter optimization (HPO) for clustering algorithms. We highlighted the primary approaches used in the literature, which predominantly rely on search-based algorithms such as Grid Search, Random Search, Bayesian Optimization, Genetic Algorithms, and other nature-inspired methods. The main limitation of these techniques is that they are computationally expensive.

Meta-learning offers a more efficient alternative to traditional HPO algorithms, especially when computational efficiency and interactivity are key concerns. By leveraging prior HPO tasks, meta-learning can automatically recommend optimal hyper-parameters for new datasets based on their characteristics (meta-features) without rerunning the full HPO process.

Table 2.4: Meta-learning for clustering algorithm recommendation and clustering hyper-parameter optimization

Paper	**Year**	**Recommendation task**	**Meta-features type**	**Data type**
[de Souto 2008]	2008	Clustering algorithm recommendation	Statistical	Numerical
[Ferrari 2015]	2015	Clustering algorithm recommendation	Distance based	Numerical
[Vukicevic 2016]	2016	Clustering algorithm recommendation	Clustering evaluation with internal CVIs	Numerical
[Pimentel 2019]	2019	Clustering algorithm recommendation	Correlation based, Distance based	Numerical
[Gabbay 2021]	2021	Clustering algorithm recommendation	Isolation forest, Landmark	Numerical
[Poulakis 2020]	2020	Clustering algorithm recommendation	Landmark	Numerical
[ElShawi 2022]	2022	Clustering algorithm recommendation	Landmark	Numerical
[Cohen-Shapira 2021]	2021	Clustering algorithm recommendation	Graph embedding with graph neural networks	Numerical
[Pimentel 2020]	2020	Number of clusters recommendation	Density based	Numerical
[Alves 2019]	2019	Similarity measure recommendation	Statistical	Categorical
[Zhu 2020b]	2020	Similarity measure recommendation	Statistical, Distance based, Structural information-based	Numerical

2.6 Conclusion and contributions

In this chapter, we presented a comprehensive review of the literature on similarity measures, clustering, with a particular focus on mixed data clustering, and hyper-parameter optimization. Through this discussion, we have demonstrated the critical role that similarity measures play in various clustering paradigms, underscoring their importance, especially in the context of mixed data clustering. Similarity-based mixed data clustering algorithms define the similarity between data points as a combination of numerical and categorical measures. However, the wide range of similarity measures proposed in the literature, each with different properties, raises several unresolved questions:

1. *How does the choice of similarity measures affect the performance of mixed data clustering algorithms?*
2. *How to automatically select suitable similarity measures according to the considered mixed dataset and mixed data clustering algorithm?*

These open questions remain unaddressed in the existing literature, and this thesis aims to fill this gap. To address the first question, an experimental frame-

work is proposed in chapter 4 to study the impact of similarity measures on mixed data clustering algorithms. The problem of the automatic selection of the similarity measures is a particular case of the more general hyper-parameter optimization (HPO) problem. While traditional HPO techniques have been widely studied, they often suffer from high computational costs, making them impractical in scenarios where efficiency and responsiveness are crucial. Meta-learning emerges as a promising alternative, enabling the efficient recommendation of optimal or near-optimal hyper-parameters by leveraging dataset characteristics. However, current meta-learning research to support data scientists when performing clustering tasks has primarily focused on homogeneous data (see table 2.4), leaving a significant gap in the study of mixed data clustering. We propose in chapter 5, to extend the application of meta-learning to the automatic recommendation of similarity measures in the context of mixed data clustering, thereby addressing a critical need in the field.

Chapter 3

A comparative study of mixed and homogenization approaches for mixed data clustering

Contents

3.1 Introduction

Heterogeneous data - comprising various data types such as numerical, categorical, and even textual data - are increasingly prevalent across a wide range of real-world applications, including healthcare, marketing, finance, and business analytics [Abdullin 2012, Ahmad 2019]. As data complexity continues to grow, the occurrence of purely homogeneous datasets has become increasingly rare. This shift presents new challenges, particularly in the field of clustering because traditional clustering algorithms are designed primarily for homogeneous datasets and cannot be directly applied to heterogeneous datasets due to the different nature of involved data types.

In the case of mixed data, several methods have been proposed for addressing this challenge. These methods fall into two groups [Barcelo-Rico 2012, Wei 2015, Ding 2017]: homogenization methods and mixed methods. Homogenization methods involve converting one data type to another, enabling the application of traditional clustering algorithms. Although convenient, this approach risks distorting the original data structure, potentially leading to sub-optimal clustering outcomes. On the other hand, mixed methods strive to directly handle the heterogeneity of the data, preserving the distinct nature of each data type. Despite their theoretical advantages, these methods often face difficulties in integrating the different types of data effectively.

While the literature acknowledges the potential benefits of mixed methods over homogenization approaches, particularly in terms of maintaining the integrity of the original data [van de Velden 2019, Wei 2015, Barcelo-Rico 2012, Ding 2017, Harikumar 2015, Huang 1998], there remains a notable gap. Specifically, no comprehensive comparative study has been conducted to empirically evaluate these methods under various conditions. This chapter addresses this gap by proposing an experimental framework to systematically compare the performance of homogenization and mixed methods across a diverse set of mixed datasets. **The goal is to provide empirical evidence that either validates or challenges the prevailing assumption that mixed methods are superior to homogenization ones for clustering mixed data and offer a deeper understanding of the practical implications of choosing one strategy over the other. The results of this comparative study will not only contribute to the ongoing discourse on mixed data clustering but also provide valuable insights for researchers and practitioners seeking the most effective techniques for their specific datasets.**

The remainder of this chapter is organized as follows. Section 3.2 presents our methodology for comparing the two strategies and introduces some notations. Section 3.3 describes the experimental setup. The experiment results are presented and discussed in Section 3.4. Finally, Section 3.5 summarizes the work presented in this chapter.

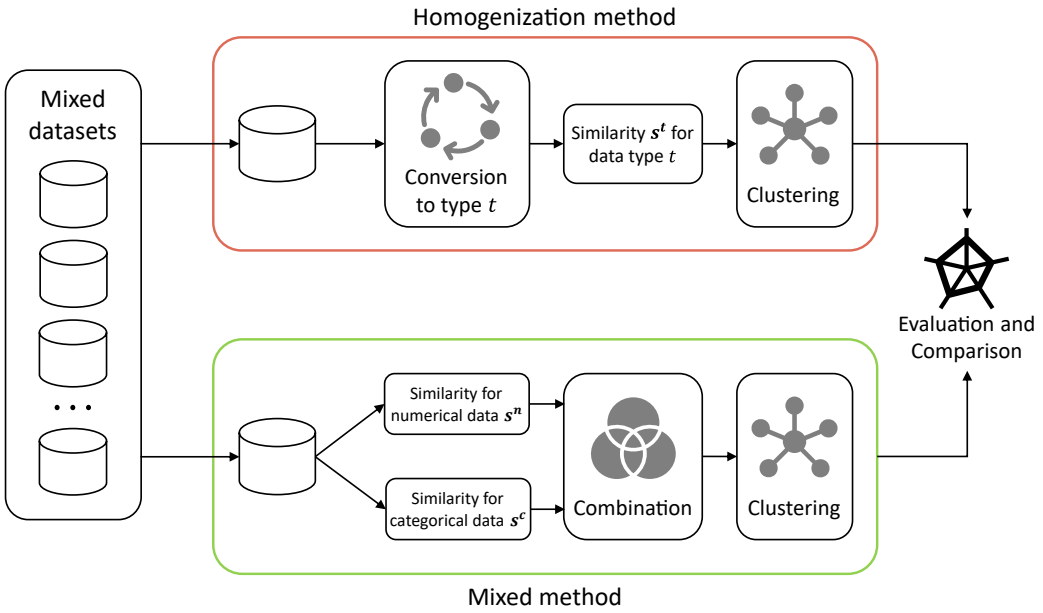


Figure 3.1: Experimental framework

3.2 Comparison methodology

We aim to find which strategy between homogenization and mixed methods is better when clustering mixed datasets. In other words, does the conversion used in homogenization methods lead to poorer results compared to mixed methods which do not involve data conversion?

Figure 3.1 shows the experimental framework. We evaluate and compare the performances of different mixed and homogenization methods on various mixed datasets.

- Homogenization methods have two main steps: a conversion step and a clustering step. In the conversion step, the data are converted into a target type ($t \in \{n, c\}$) that can be either numerical (n) or categorical (c). In the clustering step, any known clustering algorithm for the converted data can be used to compute the clusters.

- For the mixed methods, we focus on similarity-based methods, i.e. methods that define similarity for mixed data as a combination of two numerical (s^n) and categorical (s^c) similarity measures. The newly defined similarity measure is then used with an existing clustering algorithm (or an adaptation of an existing clustering algorithm) to compute the clusters.

We only compare mixed and homogenization methods based on the same underlying clustering algorithm. This approach ensures that any observed differences between the methods are solely due to how they handle the different data types rather than

Table 3.1: Datasets Description

Id	Name	# of attributes	# of numerical attributes	# of categorical attributes	# of samples	# of classes
AUS	Australian Credit Approval	14	6	8	690	2
BANK	Bank Marketing	16	7	9	4521	2
CLEV	Cleveland Heart Disease	13	6	7	299	2
CRED	Credit Approval	15	6	9	690	2
LPH	Lymphography	18	3	15	148	4
HEART	South Africa Heart Disease	9	8	1	462	2
ZOO	Zoo	16	1	15	101	7

variations in the considered algorithm in the clustering procedure. Additionally, for a fair comparison, each time we evaluate a given method on a dataset, a grid search strategy is used to find its optimal parameters according to the considered cluster validity index. These parameters include the similarity measures (two for mixed methods, one for homogenization ones), the combination weights of the two similarity measures for mixed methods, and other algorithm-specific parameters.

3.3 Experimental setup

We describe our experimental setup in the following. Section 3.3.1 describes the datasets used in our experiments. Sections 3.3.2 and 3.3.3 present the considered homogenization and mixed methods respectively. Finally, section 3.3.4 presents the considered cluster validity indices.

3.3.1 Datasets

We use 7 publicly available mixed datasets (table 3.1) from various application domains. We consider these datasets because they are widely used in the mixed data clustering literature. They are publicly available in the UCI machine learning repository [1].

- The **Australian Credit Approval (AUS)** dataset, originating from the finance domain, is commonly utilized for research in credit approval decision systems. It consists of 690 instances, each described by 14 attributes, out of which 6 are numerical and 8 are categorical. The dataset is structured to classify each instance into one of two possible classes, representing the credit approval (positive) and denial (negative) decisions.

[1] https://archive.ics.uci.edu/datasets

- The **Bank Marketing (BANK)** dataset is related to direct marketing campaigns (phone calls) conducted by a Portuguese banking institution. The goal is to predict whether a client will subscribe to a term deposit. It consists of 4521 instances, each described by 16 attributes, out of which 7 are numerical and 9 are categorical.
- The **Cleveland Heart Disease (CLEV)** dataset is a well-known dataset in the domain of healthcare, specifically for heart disease diagnosis. It consists of 299 instances (patients), each described by 13 attributes, out of which 6 are numerical and 7 are categorical. The classification task involves predicting the presence or absence of heart disease in the patient.
- The **Credit Approval (CRED)** dataset is similar to the Australian Credit Approval dataset but contains 15 attributes, out of which 6 are numerical and 9 are categorical.
- The **Lymphography (LPH)** dataset falls within the domain of health and medicine. It represents patients divided into four classes according to radiological examination results. It consists of 148 instances, each described by 18 attributes, out of which 3 are ordinal and considered as numerical and 15 are categorical (nominal).
- The **South Africa Heart Disease (HEART)** dataset is about coronary heart disease (CHD) obtained from the Coronary Risk Factor Study conducted in South Africa. It consists of 462 instances, each described by 18 attributes, out of which 8 are numerical and 1 is categorical. The dataset has two classes corresponding the presence or absence of a myocardial infarction at the time of the survey.
- The **Zoo** dataset falls under the Biology domain. The dataset is designed to classify animals into seven classes. It consists of 101 instances, each described by 16 attributes, out of which 1 is numerical and 15 are categorical.

3.3.2 Homogenization methods

Homogenization methods differ mainly in their conversion step. Some methods transform numerical attribute values into categorical ones, while others perform the reverse, converting categorical values into numerical ones. The approach that usually gives better results is to convert categorical attribute values into numerical values and use a numerical clustering algorithm [Barcelo-Rico 2012]. Therefore, we focus only on this approach in our experiments and we consider two techniques for the conversion of categorical data to numerical data. The first one encodes the categories of a categorical attribute as ordered integers and the second uses one-hot encoding. These techniques are described in the related works, 2 section 2.4.1.1.

In the clustering step, we consider clustering algorithms that have been extended to mixed data clustering (such as K-Means) or that can take any similarity measure as input (such as K-Medoids). Here are the selected algorithms:

- Centroid-based algorithms: K-Means [Macqueen 1967], K-Medoids (PAM [Schubert 2021] and SFKM [Park 2009] variants)
- Hierarchical algorithms: Agglomerative hierarchical clustering with average linkage (H-AVG) [Reddy 2014]
- Density-based algorithms: DBSCAN [Ester 1996]
- Graph-based algorithms: Spectral clustering [Ng 2001]

They cover the main families of clustering algorithms and are commonly used in practice. Centroid-based clustering algorithms like K-Means and K-Medoids use a random initialization of cluster centers. So, at each call, they are run 10 times with different random initializations, and the best result is kept. For the number of clusters, we simply use the number of classes in the ground truth labels. For other parameters, as stated above, a grid search strategy is used to find the parameters that optimize the considered cluster validity index. We consider the following 10 numerical similarity measures described in chapter 2: *Euclidean distance*, *Manhattan distance*, *Chebyshev distance*, *Squared Euclidean distance*, *Canberra distance*, *Mahalanobis distance*, *Cosine dissimilarity*, *Pearson dissimilarity*, *Loretzian distance* and *Divergence distance*. They are described in more detail in chapter 2 section 2.2. For the K-Means algorithm, only the Euclidean distance is used.

3.3.3 Mixed methods

As argued in section 3.2, we consider mixed methods that adapt existing clustering algorithms with the definition of a new similarity for mixed data. These methods are listed below. We denote $X = \{x_1, ..., x_{n_X}\}$ a mixed dataset with p numeric attributes and q categorical ones. For each $x_i \in X$ such that $x_i = (x_{i,1}^n, ...x_{i,p}^n, x_{i,1}^c, ...x_{i,q}^c)$, we denote $x_i^n = (x_{i,1}^n, ...x_{i,p}^n)$ and $x_i^c = (x_{i,1}^c, ...x_{i,q}^c)$ the numerical and categorical parts of x_i respectively.

- **K-Prototypes (K-PROTO) [Huang 1998].** It is the extension of the K-Means algorithm to mixed data. Given two mixed data samples x_i and x_j, their similarity is defined by:

$$s(x_i, x_j) = s^n(x_i^n, x_j^n) + \gamma \cdot s^c(x_i^c, x_j^c) \tag{3.1}$$

  Where $\gamma \in \mathbb{R}_+$, s^n and s^c are two similarity measures for numerical and categorical data respectively. In our experiments, the K-Prototypes algorithm is compared to homogenization methods using K-Means in their clustering step.

- **(Mixed) K-Medoids.** A mixed version of the K-Medoids algorithm. We use the PAM [Schubert 2021] and SFKM [Park 2009] variants of K-Medoids as for homogenization methods and the similarity for mixed data defined in equation 3.2.

- **(Mixed) Hierarchical Clustering with average linkage (H-AVG).** Hierarchical Clustering has been used in [Philip 1983] with the Gower similarity. However, since the Gower similarity yields poor results because it gives the same importance to numerical and categorical attributes [Ahmad 2019], we use the similarity for mixed data defined in equation 3.2.

- **(Mixed) Spectral Clustering [Mbuga 2022].** It extends the spectral clustering algorithm to mixed data using as similarity for mixed data a linear combination of two numerical and categorical similarity measures. Given two mixed data samples x_i and x_j, their similarity is defined by:

$$s(x_i, x_j) = (1 - w) \cdot s^n(x_i^n, x_j^n) + w \cdot s^c(x_i^c, x_j^c) \tag{3.2}$$

  Where $w \in [0, 1]$ is a positive weight, s^n and s^c are two similarity measures for numerical and categorical data respectively.

- **(Mixed) DBSCAN.** We extend the DBSCAN algorithm to mixed data by using the similarity for mixed data defined in equation 3.2.

As for homogenization methods, mixed methods using centroid-based clustering algorithms are run 10 times, at each call, with different random initializations of the cluster centers. For the number of clusters, here too, we simply use the number of classes in the ground truth labels, and the other parameters, including the two similarity measures and their combination weight (γ in equation 3.1 and w in equation 3.2), are selected using a grid search strategy over their value range. The similarity measures for numerical data are the same as for homogenization methods. For categorical data, we consider the 12 following measures described in chapter 2: *Hamming distance* or *Overlap* similarity, *Eskin*, *Occurrence Frequency*, *Inverse Occurrence Frequency*, *co-occurence* based similarity, *Jaccard*, *Dice*, *Klusinski*, *Rogerstanimoto*, *Russellrao*, *Sokalmichener*, and *Sokalsneath*.

3.3.4 Cluster validity indices

Cluster validity indices are used to evaluate the performances of clustering algorithms. We consider three different validity indices in this comparison study. We use external validity indices, i.e., validity indices that rely on ground truth labels to evaluate clustering quality. Let $\hat{y} = (\hat{y}_1, ..., \hat{y}_{n_X})$ be the computed cluster labels for a given dataset $X = \{x_1, ..., x_{n_X}\}$ and $C_1, ..., C_k$ the corresponding clusters. Let $y = (y_1, ..., y_{n_X})$ be the ground truth labels and $W_1, ..., W_l$ the corresponding classes (in our case $k = l$). We use 3 external cluster validity indices, i.e. based on ground truth labels.

- **Clustering Accuracy (CA).** A cluster label does not correspond exactly to the class with the same label in the ground truth. So, in order to compute the accuracy score for clustering, one start by finding the permutation σ of cluster labels with the best match with the ground truth. Then, the accuracy

is computed as for the classification task considering the found permutation as the predicted classes and the ground truth as true labels.

$$CA(\hat{y}, y) = \max_{\sigma \in \mathcal{P}} \frac{1}{N} \sum_{i=1}^{N} 1\left(\sigma(\hat{y}_i) = y_i\right) \tag{3.3}$$

Where $\mathcal{P}$ is the set of permutations. $1\left(\sigma(\hat{y}_i) = y_i\right) = 1$ if $\sigma(\hat{y}_i) = y_i$, 0 otherwise.

- **Adjusted Rand Index (ARI).** The rand index [Hubert 1985] is a function that measures the similarity between cluster labels and ground truth labels, ignoring permutations. Let a the number of samples pairs (x_i, x_j) such that $\hat{y}_i = \hat{y}_j$ and $y_i = y_j$. let b the number of samples pairs (x_i, x_j) such that $\hat{y}_i \neq \hat{y}_j$ and $y_i \neq y_j$. The rand index is defined as follows.

$$RI(\hat{y}, y) = \frac{a + b}{C_2^N} \tag{3.4}$$

Where C_2^N is the total number of sample pairs. $a + b$ represents the number of pairs for which the clustering and the ground truth agree. The adjusted rand index is a correction of the rand index which ensures that random label assignments will get a value close to zero.

$$ARI(\hat{y}, y) = \frac{RI(\hat{y}, y) - E[RI]}{RI_{max} - E[RI]} \tag{3.5}$$

Where $E[RI]$ is the expected rand index of random labeling and RI_{max} is the maximal rand index $RI(y, y)$

- **Purity (PURITY).** It measures the purity of clusters. A cluster is considered pure if it contains labeled objects from one and only one class [Harikumar 2015].

$$PURITY = \frac{1}{N} \sum_{i=1}^{k} \max_{1 \leq j \leq l} |C_i \cap W_j| \tag{3.6}$$

3.4 Experiment results and discussion

This section presents our results for the comparison between mixed and homogenization methods on the different datasets.

Global comparison. In order to observe some global tendencies, we start with a comparison including all mixed and homogenization methods regardless of their underlying clustering algorithm. To do so, for each dataset, we compare the performance of the best mixed method to the performance of the best homogenization method. Figure 3.2 shows for each dataset, the difference between the best score

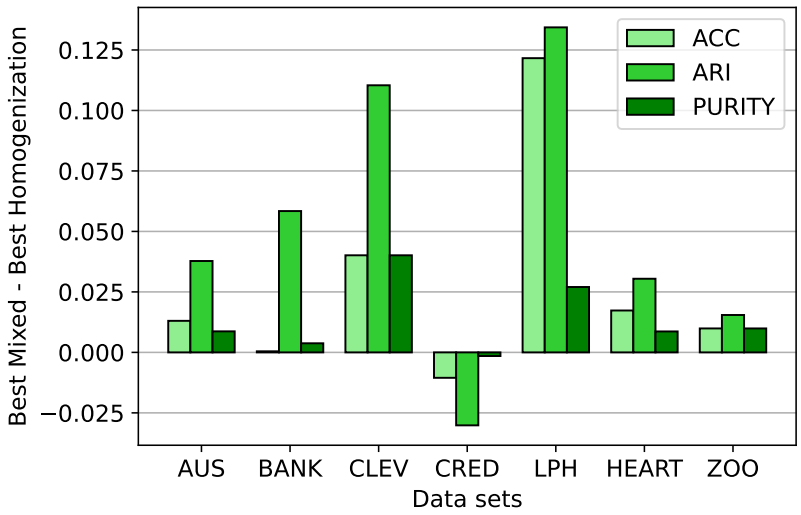


Figure 3.2: Difference between the best result for all mixed methods and the best results for all homogenization methods on each dataset.

obtained by the mixed methods and the best score obtained by the homogenization methods on that dataset using different CVIs. The figure shows that for all datasets except CRED, mixed methods perform better than homogenization ones for the different cluster validity indices. In other words, for most of the considered datasets, we can find at least one mixed method that performs better than all homogenization methods.

Comparison of methods based on the same algorithm. The previous results provide a first insight into the comparison of mixed and homogenization methods. However, for a more precise comparison between the two strategies, we need to remove the effect of the underlying clustering algorithm such that the difference between the compared methods will be only explained by the employed strategy (i.e., mixed or homogenization). As stated in section 3.2, to remove the effect of the underlying clustering algorithm, we only compare each mixed and homogenization methods that are based on the same underlying clustering algorithm.

Tables 3.2, 3.3, and 3.4 show for the performances of mixed and homogenization methods on the different datasets for CA, ARI, and Purity cluster validity indices respectively. Given a clustering algorithm, the tables show the performances of the mixed and homogenization methods based on that algorithm on the different datasets. We observe that for the different CVIs and clustering algorithms, the mixed strategy performs better than the homogenization one on most datasets. Interestingly, for H-AVG, Spectral Clustering, and DBSCAN algorithms for the CA index (table 3.2), the performance of the mixed strategy is greater or equal to the performance of the homogenization strategy on all datasets. This means that for these algorithms the homogenization strategy never outperforms the mixed

Table 3.2: Comparison of the performances of mixed and homogenization methods on the different datasets using the CA index

Algorithm	Method	AUS	BANK	CLEV	CRED	LPH	HEART	ZOO	Total Wins
K-PROTO	Homo	**0.855**	0.584	0.816	**0.863**	0.554	0.654	0.733	2
	Mixed	0.838	**0.798**	**0.853**	0.838	**0.716**	**0.695**	**0.950**	**5**
H-AVG	Homo	0.754	0.885	0.809	0.554	0.601	0.658	0.921	0
	Mixed	**0.868**	0.885	**0.839**	**0.742**	**0.797**	**0.662**	**0.941**	**6**
SC	Homo	0.846	0.884	0.823	0.799	0.601	0.695	0.822	0
	Mixed	**0.862**	**0.885**	**0.863**	**0.853**	**0.791**	**0.699**	**0.911**	**7**
DBSCAN	Homo	0.674	0.885	0.605	0.704	0.676	0.671	0.941	0
	Mixed	**0.755**	0.885	**0.763**	**0.733**	0.676	**0.684**	**0.950**	**5**
PAM	Homo	0.838	**0.871**	0.813	0.814	0.588	0.654	0.733	1
	Mixed	**0.854**	0.786	**0.86**	**0.847**	**0.642**	**0.686**	**0.822**	**6**
SFKM	Homo	0.813	**0.885**	0.813	0.835	**0.595**	0.654	0.644	2
	Mixed	**0.854**	0.702	**0.86**	**0.841**	0.568	**0.712**	**0.703**	**5**

strategy on the considered datasets. We have similar results for the H-AVG, Spectral Clustering, DBSCAN, and PAM algorithms for the ARI index (table 3.3), and H-AVG, Spectral Clustering, PAM, and SFKM algorithms for the Purity index (table 3.4).

For the CRED dataset, we observed in the general comparison that when comparing all homogenization and mixed methods regardless of the underlying clustering algorithm, homogenization methods slightly outperform mixed ones. However, tables 3.2, 3.3, and 3.4 show that, for the same dataset, when we isolate the effect of the underlying clustering algorithm by comparing only mixed and homogenization methods based on the same underlying clustering algorithm, the mixed strategy is preferable in most cases. For instance, for the CA index (table 3.2) the homogenization strategy outperforms the mixed one only for the K-Prototypes algorithm, but for all other 5 algorithms, the mixed strategy is better.

To confirm that the observed superiority of the mixed strategy over the homogenization one is significant, we use the *Wilcoxon signed-rank test* [Wilcoxon 1945] which is a non-parametric statistical hypothesis test used to compare two related paired samples. Non-parametric means that no assumption is made about the distribution of samples. For each clustering algorithm, we consider the performances obtained respectively by the mixed and homogenization strategies on the different datasets and with the different CVIs. Using these performances, we test the null hypothesis H_0 = *"The difference between the performances of the mixed and*

Table 3.3: Comparison of the performances of mixed and homogenization methods on the different datasets using the ARI index

Algorithm	Method	AUS	BANK	CLEV	CRED	LPH	HEART	ZOO	Total Wins
K-PROTO	Homo	**0.504**	0.018	0.398	**0.527**	0.24	0.090	0.687	2
	Mixed	0.455	**0.112**	**0.496**	0.456	**0.367**	**0.148**	**0.962**	**5**
H-AVG	Homo	0.256	0.005	0.381	0.001	0.146	0.090	0.950	0
	Mixed	**0.541**	**0.112**	**0.459**	**0.233**	**0.411**	0.090	**0.966**	**6**
SC	Homo	0.479	0.113	0.415	0.356	0.286	0.142	0.758	0
	Mixed	**0.524**	**0.170**	**0.525**	**0.497**	**0.420**	**0.152**	**0.937**	**7**
DBSCAN	Homo	0.221	0.172	0.191	0.237	0.138	0.105	0.958	0
	Mixed	**0.298**	**0.231**	**0.273**	0.237	**0.166**	**0.116**	**0.973**	**6**
PAM	Homo	0.455	0.032	0.389	0.393	0.178	0.090	0.672	0
	Mixed	**0.499**	**0.111**	**0.515**	**0.480**	**0.249**	**0.135**	**0.802**	**7**
SFKM	Homo	0.391	**0.075**	0.389	0.448	0.196	0.090	0.523	1
	Mixed	**0.499**	0.037	**0.515**	**0.464**	**0.221**	**0.173**	0.523	**5**

homogenization strategies for the considered algorithm is zero" at the 0.05 level of significance.

Table 3.5 shows the *statistics* and *p-values* of the test for the different clustering algorithms. We can observe that the obtained *p-values* are less 0.05 for all clustering algorithms. So, we reject the null hypothesis for the different algorithms. This indicates that the difference between the performances of the mixed and homogenization strategies is significant for the different algorithms. Since we observed above that mixed methods have globally better performances than homogenization ones, *we can conclude that for all considered clustering algorithms, adopting a mixed strategy is better than adopting a homogenization one and the difference between the two strategies is statistically significant.*

Why do mixed methods perform better? To better understand why mixed methods perform better than homogenization ones, we propose to study the effect of the conversion on the internal structure of the datasets. Given a dataset, we consider that the internal structure of the dataset is represented by its ground truth classes. Then, to evaluate the effect of the conversion, we measure the coherence between these ground truth classes and the converted data compared to the original mixed data. We expect that the conversion will result in less coherence between the ground truth classes and the data. We use the silhouette score to measure the coherence between the ground truth classes and the data. Given a dataset, the silhouette score can be computed for each sample of the dataset. Given a sample, let a be its average dissimilarity with all other samples in the same class (in the ground truth) and b its average dissimilarity with samples in the next closest class.

Table 3.4: Comparison of the performances of mixed and homogenization methods on the different datasets using the Purity index

Algorithm	Method	AUS	BANK	CLEV	CRED	LPH	HEART	ZOO	Total Wins
K-PROTO	Homo	**0.855**	0.885	0.816	**0.863**	0.770	0.654	0.921	2
	Mixed	0.838	0.885	**0.853**	0.838	**0.838**	**0.695**	**0.950**	**4**
H-AVG	Homo	0.754	0.885	0.809	0.554	0.622	0.658	0.931	0
	Mixed	**0.868**	0.885	**0.839**	**0.742**	**0.804**	**0.662**	**0.950**	**6**
SC	Homo	0.846	0.885	0.823	0.799	0.811	0.695	0.931	0
	Mixed	**0.862**	0.885	**0.863**	**0.853**	**0.838**	**0.699**	**0.950**	**6**
DBSCAN	Homo	0.859	0.888	0.739	**0.872**	0.709	0.703	0.941	1
	Mixed	0.859	**0.892**	**0.799**	0.871	**0.736**	**0.71**	**0.950**	**5**
PAM	Homo	0.838	0.885	0.813	0.814	0.757	0.654	0.921	0
	Mixed	**0.854**	0.885	**0.86**	**0.847**	**0.804**	**0.686**	0.921	**5**
SFKM	Homo	0.813	0.885	0.813	0.835	0.757	0.654	0.832	0
	Mixed	**0.854**	0.885	**0.86**	**0.841**	**0.777**	**0.712**	0.832	**5**

Table 3.5: *Statistic* and *p-value* of the Wilcoxon signed-rank test between the mixed strategy and the homogenization one for each clustering algorithm. A *p-value* ≤ 0.05 indicates that the mixed strategy is significantly better than the homogenization one.

	K-PROTO	H-AVG	SC	DBSCAN	PAM	SFKM
statistic	177	210	210	186	175	140
p-value	0.004	4e-05	4e-05	0.0001	0.0006	0.009

The silhouette score of the considered sample is defined by:

$$silhouette = \frac{b - a}{max(a, b)} \tag{3.7}$$

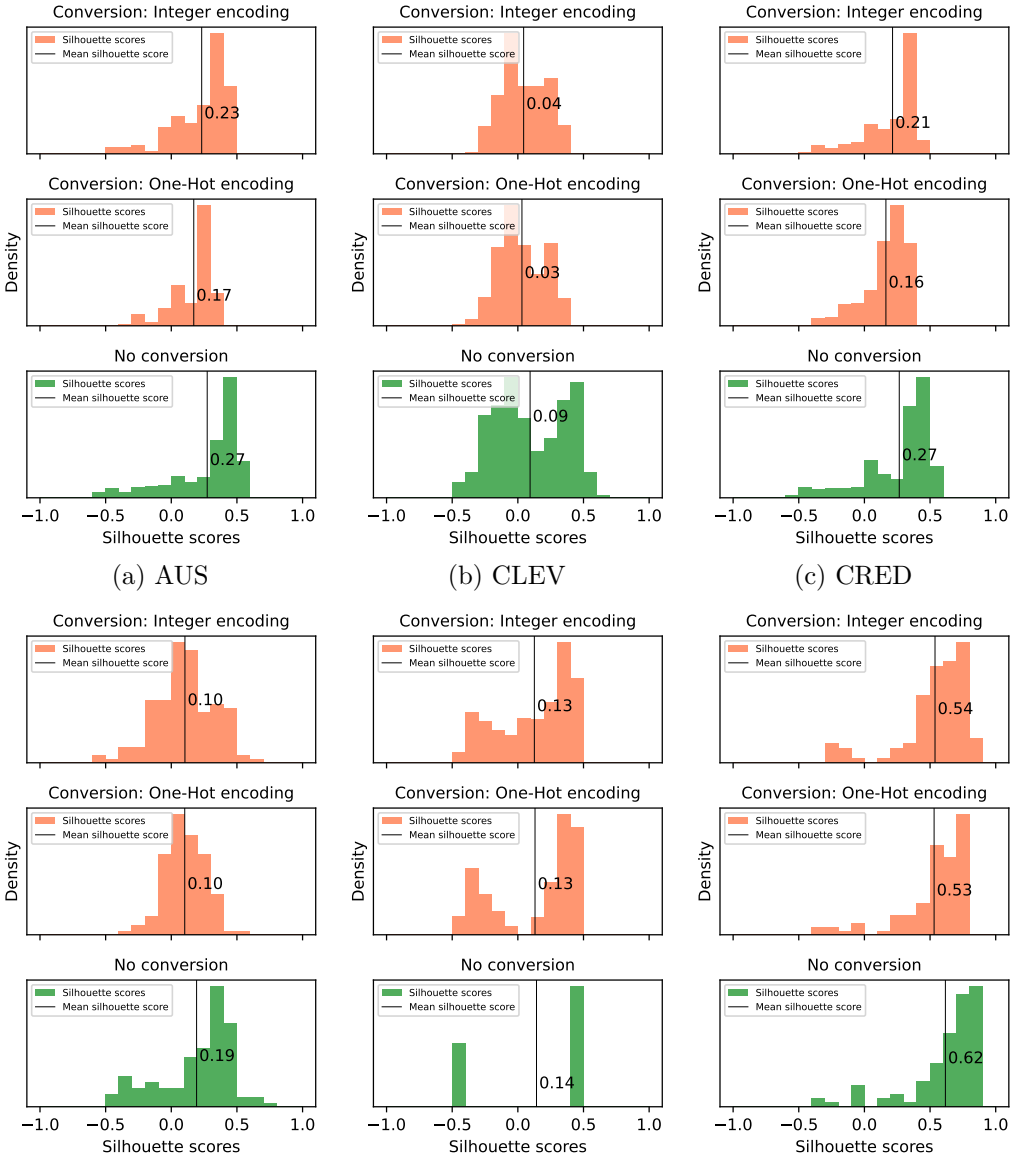
The silhouette score takes values in $[-1, 1]$ and evaluates how similar a sample is to its own class compared to other classes. If the data are coherent with the ground truth classes, then we will obtain high silhouette scores for the different samples. We consider the two conversion techniques presented earlier: *integer encoding* and *one-hot encoding*. For the choice of the similarity measure to be used to compute the silhouette scores on the converted data, we evaluate all similarity measures and keep the one that yields the highest average silhouette over all samples. To compute the silhouette scores on mixed data directly (i.e. without conversion), we use the

similarity defined in equation 3.2 (the numerical and categorical similarity measures and the weight w that give the highest average silhouette across all samples are used).

Figure 3.3 shows for each dataset, the silhouette scores distribution on converted (orange) and not-converted data (green). Globally we observe that for the different datasets, the silhouette scores measured on the converted data are less than those measured on the mixed data directly. These observations confirm that the conversion effectively affects the original data structure, making the data less coherent with the initial (ground truth) classes, leading then to worse clustering results compared to non-conversion-based methods.

3.5 Conclusion

In this chapter, a comparison study is conducted between mixed and homogenization methods for mixed data clustering. An experimental framework is proposed to compare the two approaches based on how they handle mixed data, i.e., (i) the homogenization methods' reliance on data conversion to standardize the data and (ii) the mixed methods' utilization of techniques capable of handling mixed data directly, without necessitating conversion. Experiments are conducted on 7 real-world mixed datasets with several mixed and homogenization methods. The obtained results suggest that, in the considered experimental framework, mixed methods are preferable to homogenization ones since they use adapted similarity measures for each data type while homogenization methods alter the original structure of the converted attributes leading to poorer performances. **These results show that when facing mixed data and more generally data with heterogeneous types, it is important to consider the specific properties of the different data types and consequently use adapted treatment for each type.** In the case of the considered mixed methods, this is done by using an adapted similarity measure for each data type. However, the high number of existing similarity measures for each data type raises the question of which similarity measures are the most suitable. To answer this question, we propose to first study in the next chapter, the impact of the choice of these similarity measures on mixed data clustering algorithms, mixed methods in particular.



(a) AUS (b) CLEV (c) CRED

(d) LPH (e) HEART (f) ZOO

Figure 3.3: Silhouette scores distribution on converted (orange) and not-converted data (green) for the different datasets. Higher silhouette scores indicate higher coherence between the data and the ground truth classes.

Chapter 4

Impact of similarity measures on mixed data clustering algorithms

Contents

4.1 Introduction

Similarity is a fundamental component of all clustering algorithms. Whether in centroid-based, hierarchical, density-based, graph-based, or model-based approaches, the assessment of similarity or dissimilarity between data points is dependent on the chosen similarity measures [Zhu 2020b]. Consequently, the quality of the clustering is directly influenced by the choice of these measures.

In the previous chapter, we showed that when dealing with mixed data, mixed methods—those that handle mixed data directly without conversion—are generally preferable to homogenization or conversion-based methods. This chapter, and the following ones, will focus on these mixed methods. To define similarity for mixed data, these methods typically rely on the combination of two similarity measures: one for numerical attributes and another for categorical attributes, ensuring that each measure is appropriately applied to the corresponding data type.

The importance of similarity measures becomes even more pronounced in the context of mixed data, where two distinct measures must be effectively integrated. Despite this significance, the impact of these similarity measures on the performance of mixed data clustering (MDC) algorithms has not been thoroughly explored in the literature. This oversight may contribute to the common practice of neglecting the careful selection of similarity measures in favor of default or classical choices, such as Euclidean distance for numerical attributes and Hamming distance for categorical ones, regardless of the dataset in question.

This chapter aims to address this gap by evaluating, through extensive experimentation, various pairs of numerical and categorical similarity measures and their influence on different MDC algorithms and mixed datasets. Our goal is to raise awareness among practitioners about the critical impact of similarity measures and the importance of selecting the most appropriate ones, rather than relying on default or classical options. Additionally, we seek to provide insights into the potential benefits of developing new methods to assist data scientists in identifying suitable pairs of similarity measures based on their specific use cases, including the chosen MDC algorithms and datasets.

The remainder of this chapter is organized as follows. Section 4.2 presents our methodology for comparing the two strategies and introduces some notations. Section 4.2 presents the description and goals of the experiments. The experiment results are presented and discussed in Section 4.3. Finally, Section 4.4 summarizes the work presented in this chapter.

4.2 Experiments description

This section describes the conducted experiments in this chapter. We have 3 objectives in these experiments:

1. **Impact of the similarity measures on MDC algorithms.** First, we eval-

Table 4.1: Experimental setup

	Number	Enumeration
Datasets	7	Australian Credit Approval (AUS), Bank Marketing (BANK), Cleveland Heart Disease (CLEV), Credit Approval (CRED), Lymphography (LPH), South Africa Heart Disease (HEART), ZOO
Clustering Algorithms	6	K-Prototypes [Huang 1998] (K-PROTO), Spectral Clustering [Mbuga 2022] (SC), K-Medoids - PAM [Schubert 2021] K-Medoids - SFKM [Park 2009]), Hierarchical Clustering with average linkage (H-AVG), DBSCAN
Numerical similarity measures	10	Euclidean distance, Manhattan distance, Chebyshev distance, Canberra distance, Squared Euclidean distance, Mahalanobis distance, Cosine dissimilarity, Pearson dissimilarity, Loretzian distance, Divergence distance
Categorical similarity measures	12	Hamming distance, Eskin, Occurrence Frequency, Inverse Occurrence Frequency, co-occurence based similarity, Jaccard, Dice, Klusinski, Rogerstanimoto, Russellrao, Sokalmichener, Sokalsneath
CVIs	3	Clustering Accuracy (CA) Adjusted Rand Index (ARI) Purity

uate the effect of the choice of the pair of numerical and categorical similarity measures on the performances of MDC algorithms. We consider several MDC algorithms, mixed datasets, cluster validity indices, and similarity measures for numerical and categorical data. For each triplet of one MDC algorithm, one mixed dataset, and one CVI, we evaluate the obtained performances when using all possible pairs of numerical and categorical similarity measures. Then, the effect of the choice of the similarity measures is determined by evaluating the variation in the obtained performances.

2. **Comparison of the best similarity pair and the literature baseline.** We compare for each considered MDC algorithm, the performance of the default similarity pair used in the literature (literature baseline) to the performance of the best similarity pair. Since it is widespread in practice to use

the default similarity pair, this comparison will allow us to see if this default pair is always a good choice and evaluate the potential gain one could obtain by searching for the best similarity pair.

3. **Variability of the best performing similarity pairs.** Finally, we evaluate the variability of the best-performing similarity pairs according to the datasets and according to the clustering algorithms. This is important to evaluate the difficulty of the task of finding the best-performing similarity pairs.

About the experimental setup, the considered MDC algorithms, mixed datasets, similarity measures and cluster validity indices are the same as in the previous chapter and are summarized in table 4.1. We have 7 mixed datasets widely used in the MDC literature, 6 MDC algorithms, 10 similarity measures for numerical data, 12 similarity measures for categorical data, and 3 cluster validity indices. This correspond to a total of 15120 clustering evaluations, when we ignore the conducted grid search to optimize the other algorithm parameters such as the weight used to combine the two similarity measures.

4.3 Results and discussion

4.3.1 Impact of the similarity measures

Given a mixed dataset, a clustering algorithm, and a cluster validity index, we measure the performances of the different pairs of similarity measures. Then, the variation of performances is evaluated as the difference between the highest and lowest performances.

Figure 4.1 shows the performance variation due to the choice of the pair of similarity measures for the different clustering algorithms, datasets, and CVIs. We observe important variations in the performances of the pairs of similarity measures for the different algorithms, datasets, and CVIs. For some algorithms and datasets, the difference between the best and worst pair is greater than 0.3 for the clustering accuracy, 0.5 for ARI, and 0.3 for Purity.

Furthermore, these performance variations vary according to the considered dataset showing that some datasets are more impacted by the choice of the pair of similarity measures than others. Similarly, we observe that, given a dataset, some clustering algorithms are more affected by the choice of the similarity pair exhibiting higher performance variation (e.g., H-AVG compared to other algorithms for the Australian dataset).

4.3.2 Best similarity pair VS literature baseline

The previous results show the importance of choosing a suitable pair of similarity measures given a clustering algorithm and a mixed dataset. However, it is widespread in practice to use the default pair of similarity measures used in the literature for the considered algorithm regardless of the dataset on which clustering

Table 4.2: Default similarity pair for each clustering algorithm.

Mixed Method	Default similarity pair (s^n, s^c)
K-PROTO [Huang 1998]	*(squared Euclidean distance, Hamming distance)*
PAM [Budiaji 2019]	*(Manhattan distance, Hamming distance)*
SFKM [Budiaji 2019]	*(Manhattan distance, Hamming distance)*
H-AVG [Philip 1983]	*(Manhattan distance, Hamming distance)*
SC [Mbuga 2022]	*(Euclidean distance, Hamming distance)*
DBSCAN	*(Euclidean distance, Hamming distance)*

Table 4.3: Literature baseline (LB) VS Best similarity pair for the K-Prototypes algorithm. CA(LB) is the accuracy of the literature baseline and CA(Best) is the accuracy of the best similarity pair. The percentage to the right of the arrow indicates the improvement of the best pair compared to the literature baseline.

Dataset	Best similarity pair	CA(LB)	CA(Best)
AUS	(euclidean, co-oc)	0.813	0.838 ↑3%
BANK	(mahalanobis, russellrao)	0.591	0.798 ↑35%
CLEV	(chebyshev, co-oc)	0.826	0.853 ↑3.2%
CRED	(cosine, co-oc)	0.811	0.838 ↑3.3%
LPH	(canberra, of)	0.588	0.716 ↑22%
HEART	(mahalanobis, eskin)	0.656	0.695 ↑5.9%
ZOO	(chebyshev, iof)	0.901	0.95 ↑5.5%

Table 4.4: Literature baseline (LB) VS Best similarity pair for PAM algorithm.

Dataset	Best similarity pair	CA(LB)	CA(Best)
AUS	(cosine, co-oc)	0.842	0.854 ↑1.4%
BANK	(divergence, of)	0.631	0.786 ↑25%
CLEV	(lorentzian, co-oc)	0.823	0.86 ↑4.5%
CRED	(chebyshev, co-oc)	0.812	0.847 ↑4.3%
LPH	(divergence, kulsinski)	0.554	0.642 ↑16%
HEART	(cosine, eskin)	0.684	0.686 ↑0.32%
ZOO	(canberra, kulsinski)	0.772	0.822 ↑6.4%

Table 4.5: Literature baseline (LB) VS Best similarity pair for SFKM algorithm.

Dataset	Best similarity pair	CA(LB)	CA(Best)
AUS	(cosine, co-oc)	0.822	0.854 ↑3.9%
BANK	(mahalanobis, jaccard)	0.653	0.702 ↑7.6%
CLEV	(cosine, co-oc)	0.823	0.86 ↑4.5%
CRED	(sqeuclidean, sokalsneath)	0.715	0.841 ↑18%
LPH	(chebyshev, eskin)	0.527	0.568 ↑7.7%
HEART	(euclidean, hamming)	0.695	0.712 ↑2.5%
ZOO	(sqeuclidean, iof)	0.644	0.703 ↑9.2%

Table 4.6: Literature baseline (LB) VS Best similarity pair for H-AVG algorithm.

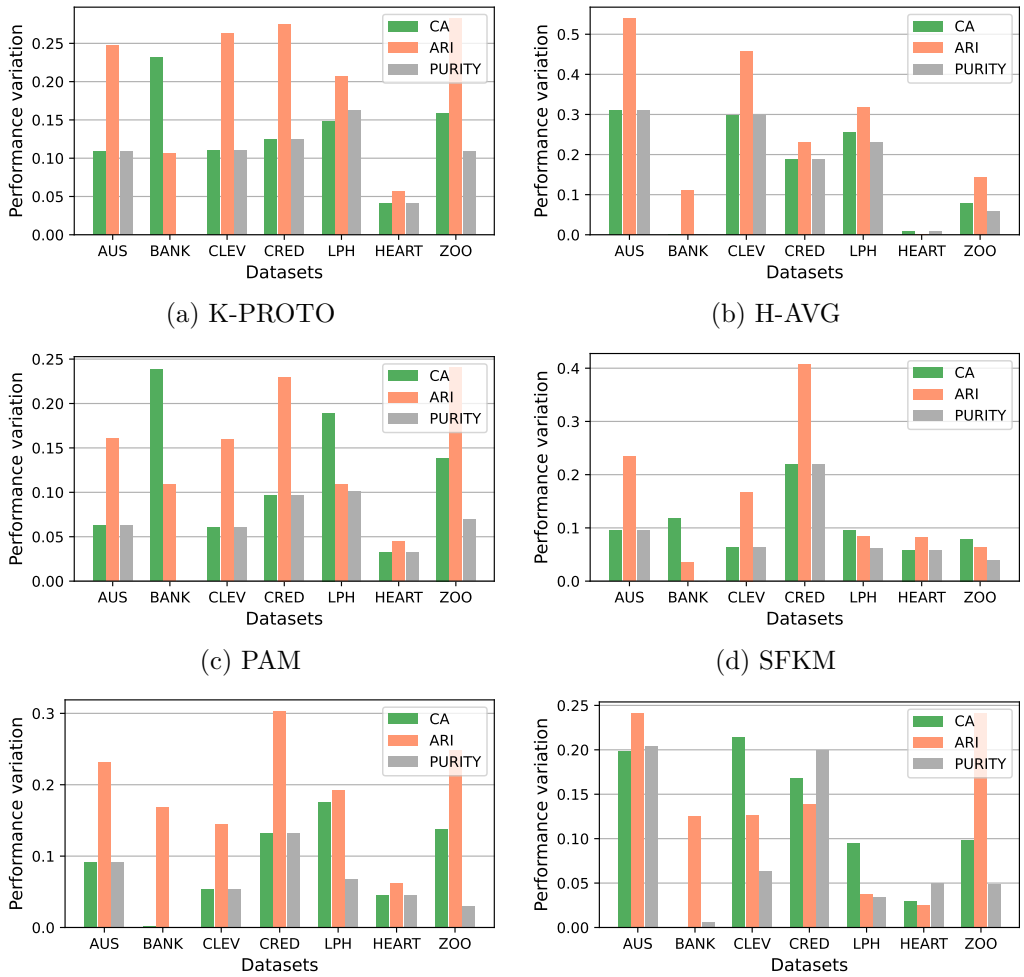
Dataset	Best similarity pair	CA(LB)	CA(Best)
AUS	(chebyshev, iof)	0.703	0.868 ↑24%
BANK	(sqeuclidean, jaccard)	0.885	0.885
CLEV	(lorentzian, iof)	0.666	0.839 ↑26%
CRED	(canberra, eskin)	0.554	0.742 ↑34%
LPH	(sqeuclidean, sokalsneath)	0.635	0.797 ↑26%
HEART	(euclidean, of)	0.662	0.662
ZOO	(euclidean, of)	0.931	0.941 ↑1.1%

Table 4.7: Literature baseline (LB) VS Best similarity pair for the SC algorithm.

Dataset	Best similarity pair	CA(LB)	CA(Best)
AUS	(chebyshev, sokalmichener)	0.857	0.862 ↑0.68%
BANK	(euclidean, eskin)	0.884	0.885 ↑0.05%
CLEV	(sqeuclidean, iof)	0.856	0.863 ↑0.78%
CRED	(chebyshev, eskin)	0.806	0.853 ↑5.8%
LPH	(mahalanobis, sokalsneath)	0.696	0.791 ↑14%
HEART	(canberra, hamming)	0.69	0.699 ↑1.3%
ZOO	(sqeuclidean, sokalsneath)	0.822	0.911 ↑11%

Table 4.8: Literature baseline (LB) VS Best similarity pair for DBSCAN.

Dataset	Best similarity pair	CA(LB)	CA(Best)
AUS	(mahalanobis, eskin)	0.723	0.755 ↑4.4%
BANK	(pearson, sokalsneath)	0.885	0.885
CLEV	(pearson, co-oc)	0.682	0.763 ↑12%
CRED	(lorentzian, iof)	0.704	0.733 ↑4.1%
LPH	(canberra, hamming)	0.655	0.676 ↑3.1%
HEART	(euclidean, of)	0.684	0.684
ZOO	(sqeuclidean, hamming)	0.931	0.95 ↑2.1%



(a) K-PROTO (b) H-AVG

(c) PAM (d) SFKM

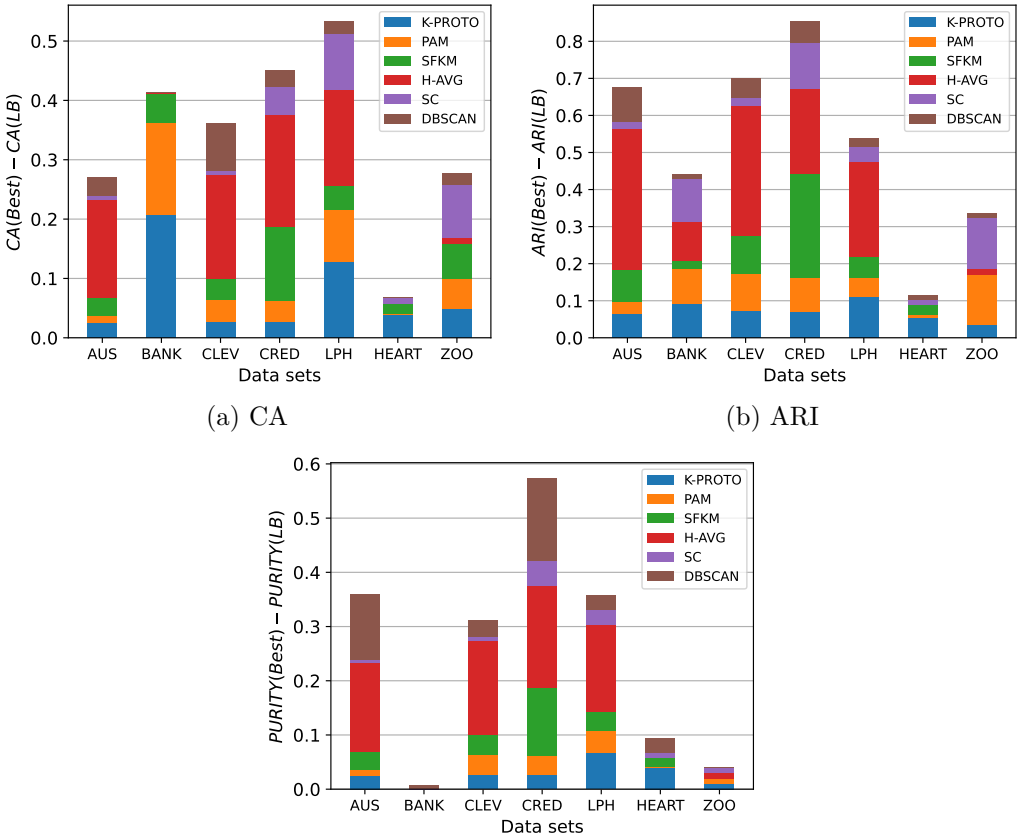
(e) SC (f) DBSCAN

Figure 4.1: Variation of clustering performances when using different pairs of similarity measures.

is performed. Therefore, in the following, we compare for each algorithm, the performance of the default similarity pair to the performance of the best similarity pair that could be used, aiming to show the interest of searching for this best similarity pair.

The default similarity pairs of the considered MDC algorithms are presented in table 4.2 and called *literature baselines* (LB) in the following. For the particular case of DBSCAN which has not been used in the literature for clustering mixed data, we define as literature baseline the most popular similarity measures for numeric and categorical data, i.e. the pair *(Euclidean distance, Hamming distance)*.

For each clustering algorithm, we compare the performances of the literature baseline to those of the best similarity pair on the different datasets. Table 4.3 shows the obtained for the K-Prototypes algorithm when using the clustering accuracy metric. First of all, we notice that for the different datasets, the best similarity



(a) CA

(b) ARI

(c) Purity

Figure 4.2: Best similarity pair VS literature baseline. For each evaluation metric, results are presented as a stacked bar plot where for each colored box, the color correspond to a clustering algorithm and the height corresponds to the difference between the best similarity pair and the literature baseline.

pair is always different from the literature baseline. Furthermore, we observe that using the best similarity pair can lead to significant improvement compared to the literature baseline. For example, we obtain up to 35% and 22% of improvement on BANK and LPH datasets respectively. The same observations yield for the other clustering algorithms, see tables 4.4 to 4.8.

We also obtain similar results for other CVIs as shown in figure 4.2. The figure shows for each CVI, the difference between the performances of the best similarity pair and the literature baseline for the different algorithms and datasets. For each CVI, we observe that there are several algorithms and datasets for which using the best similarity pair can significantly enhance clustering performances compared to using the literature baseline. Furthermore, we can notice that the difference between the performances of the best similarity pair and the literature baseline is higher for certain algorithm such as H-AVG than for other. This indicates that the search of the best similarity pair is potentially more interesting for these algorithms.

4.3.3 Variability of the best-performing similarity pairs

Beside the potential improvement we can get by using the best similarity pair compared to the literature baseline, we also observe from the previous results (table 4.3 to 4.8) a high variability of the best similarity pair according to the dataset and clustering algorithm. This indicates that there is no single similarity pair that performs optimally across all datasets and algorithms. In the following, we study in more detail this variability of the best similarity pair and more generally the k best pairs (with $k = 10$) according to the datasets and clustering algorithms. Let us start with the variability according to the datasets. Given a clustering algorithm, for each pair of datasets, we measure the number of common similarity pairs among their 10 best-performing pairs. Figure 4.3 shows for each pair of datasets, the average across all clustering algorithms, of the number of common pairs among the 10 best-performing similarity pairs for each of the 2 datasets. We observe very low similarities between the top-performing similarity pairs of the different datasets. So the best-performing highly vary according to the considered dataset.

We conducted a similar study to evaluate the variability of the best similarity pairs according to the clustering algorithms. Given a dataset, for each pair of 2 clustering algorithms, we measure the number of common pairs among their 10 best-performing similarity pairs for the considered dataset. Figure 4.4 shows for each pair of algorithms, their average number of common best-performing similarity pairs across all datasets. Here too, we observe very low similarities between the best-performing similarity pairs of the different algorithms. This shows a high variability of the best-performing pairs of similarity measures according to the considered clustering algorithm.

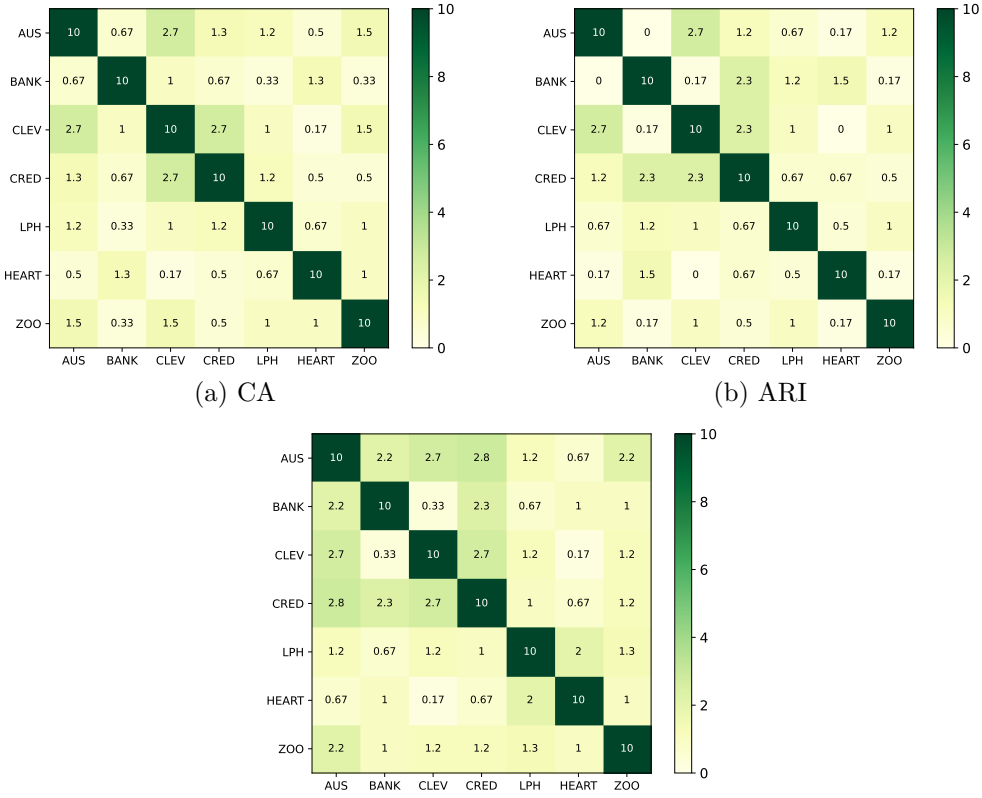
4.4 Conclusion

In this chapter, we have focused on the role of similarity measures in mixed data clustering (MDC). We have conducted experiments to assess the impact of choosing different similarity measures (concretely different pairs of numeric and categorical similarity measures) on the performances of similarity-based MDC algorithms.

Our findings indicate that the choice of the pair of similarity measures can significantly affect the performance of clustering, with the degree of variability (either low or high) depending on the specific algorithm and dataset. We have also compared the performance of each algorithm when using the most effective pair of similarity measures against the default pair typically used in the literature for this algorithm.

The results of this comparison reveal that significant improvements can be achieved over the default pair by using the most effective pair of similarity measures. We have also noted that the most effective pair, or even the top-performing pairs, can vary greatly depending on the algorithm and dataset in question.

These findings highlight the potential for future research into determining the most effective pair of similarity measures for a given MDC algorithm and dataset. Yet, the task of automatically identifying this pair, without having to test all possi-



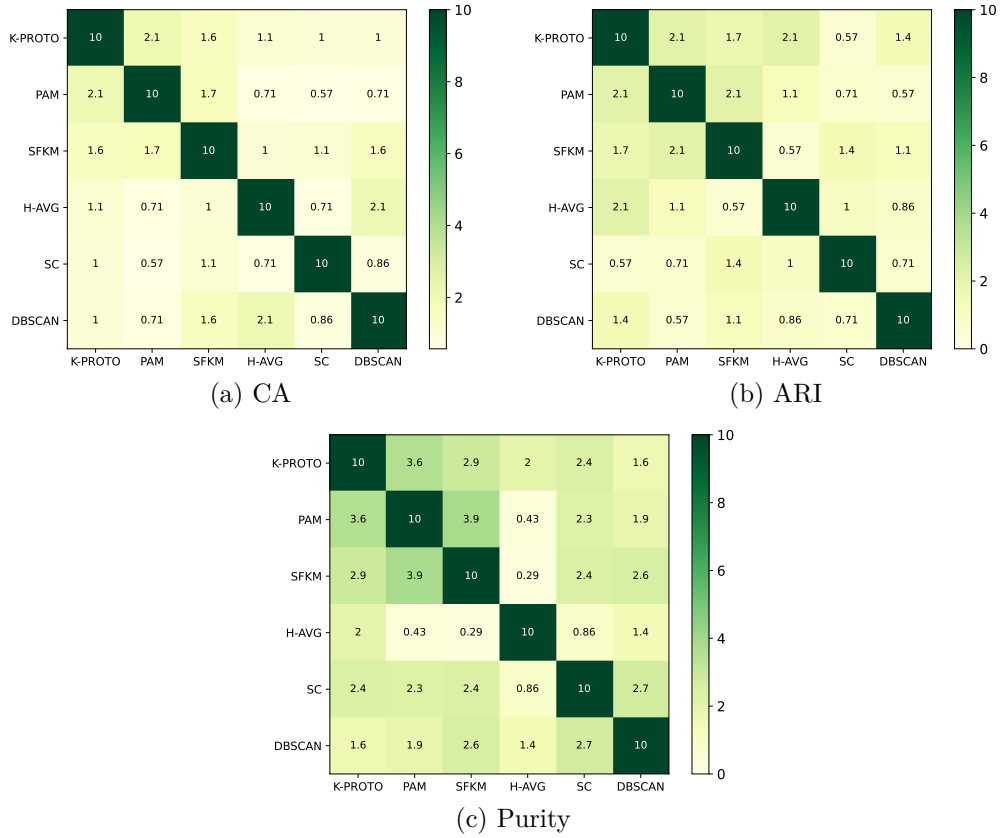
(a) CA

(b) ARI

(c) Purity

Figure 4.3: Number of common best-performing similarity pairs between datasets. Each box corresponds to a pair of datasets and indicates the average across all clustering algorithms, of the number of common pairs among the 10 best-performing similarity pairs for each of the 2 datasets.

ble combinations and while considering the specific properties of the algorithm and dataset, presents a significant challenge. This will be the focus of our next chapter.



(a) CA (b) ARI

(c) Purity

Figure 4.4: Number of common best-performing pairs of similarity measures between MDC algorithms. Each box corresponds to a pair of algorithms and indicates their average number of common best-performing pairs of similarity measures across all datasets.

Chapter 5

SIMREC: A similarity measure recommendation system for mixed data clustering algorithms

Contents

5.1 Introduction

We showed in the previous chapter that the selection of an appropriate pair of numerical and categorical similarity measures is fundamental for MDC algorithms to obtain good clustering results. However, this selection process is complex, especially for non-experts. This complexity arises from two main factors. First, there is "*no free lunch*", i.e. there is no universal pair of similarity measures that performs optimally across all datasets, clustering algorithms, and cluster validity indices. This is confirmed by our results in chapter 4 where we can observe a high variability of the top-performing pairs of similarity measures according to the datasets and clustering algorithms. Therefore, when selecting the pair of similarity measures, one needs to consider several factors such as the dataset in question, the clustering algorithm being used, and the cluster validity index to be optimized. Understanding how these different factors affect the performance of the different pairs of similarity measures is a significant challenge, even for seasoned experts. Second, there are many alternative choices for the pair of similarity measures, with a large number of similarity measures that have been proposed in the literature for numerical [Abu Alfeilat 2019] and categorical data types [Boriah 2008, Choi 2009]. Furthermore, in the context of mixed data, the search space is larger compared to homogeneous data, as two similarity measures need to be selected instead of just one. This adds another layer of complexity to the task.

A possible solution is to adopt trial-and-error strategies such as grid search or random search which consist in evaluating several possible pairs of similarity measures in order to identify the most suitable ones. However, these strategies can be very costly due to the high number of pairs of similarity measures, especially if they are applied each time clustering is performed. More advanced search strategies such as Bayesian optimization and evolutionary algorithms also have the same drawbacks and require additional knowledge from the user. As a result, in practice, popular similarity measures like *Euclidean distance* and *Hamming distance*, or the similarity measures used in the literature for the considered algorithm are commonly used. However, our results in chapter 4 show that, depending on the dataset, these measures might not be appropriate, leading to poor clustering performances. So, given that in most cases (both in practice and in the literature) the default pair of similarity measures is used when applying a clustering algorithm to a given dataset, this illustrates the importance of an efficient solution for the automatic selection of suitable pairs of similarity measures.

To solve this problem, we provide in this chapter, a similarity measure recommendation system for mixed data clustering named SIMREC. Given an input query composed of a mixed dataset, a mixed data clustering algorithm, and a cluster validity index to be optimized, SIMREC predicts the ranking of the pairs of similarity measures according to their performances for the targeted cluster validity index. The proposed method involves two phases: a **learning phase** and a **recommendation phase** (figure 5.1).

During the learning phase, we learn how to predict the ranking of the pairs

of similarity measures given an input query. Empirical studies have shown that the performance of the similarity measures is related to the characteristics of the datasets to be clustered [Zhu 2020b]. Therefore, given an MDC algorithm and a cluster validity index, meta-learning [Vanschoren 2018] can be used to mine this relationship, by using results obtained from prior experience, and thus learn a model able to rank the pairs of similarity measures based on the characteristics of the datasets. By prior experience, we mean prior evaluations of the pairs of similarity measures on different mixed datasets with the considered algorithm and cluster validity index. In the meta-learning field, the characteristics of the datasets are called *meta-features*. Examples of meta-features include the number of instances, number of attributes, distribution of numerical attributes, entropy of categorical attributes, etc. The learned model, that predicts the ranking of the pairs of similarity measures based on the characteristics of the datasets, is called *meta-model*. A specialized meta-model is learned for each pair of an MDC algorithm and a cluster validity index.

The recommendation phase corresponds to the practical use of SIMREC to recommend suitable pairs of similarity measures for newly encountered datasets thanks to the meta-models learned during the learning phase. Specifically, each time a user inputs a new mixed dataset, an MDC algorithm, and a cluster validity index into SIMREC, the system first computes the meta-features of the new dataset. It then uses the relevant meta-model (for the MDC algorithm and cluster validity index given by the user) to predict the ranking of pairs of similarity measures. This ranking is finally provided to the user as the recommendation. It is important to note that the learning phase is performed only once; no additional learning occurs when SIMREC is used to make recommendations during the recommendation phase.

The main contributions of the work presented in this chapter are as follows:

1. A similarity measure recommendation system for mixed data clustering is proposed.

2. Meta-learning is extended to the context of mixed data clustering while previous works on meta-learning for clustering focus on homogeneous data, i.e. data with only one type.

3. New statistical meta-features of datasets that complement those existing in the literature are proposed in order to provide more information about the similarity measures.

The remainder of this chapter is organized as follows. Section 5.2 presents SIMREC and its different components. Section 5.3 describes our choices for the current implementation of SIMREC. Section 5.4 describes the conducted experiments. Experiment results are presented in section 5.5. Section 5.6 discusses some important points of the proposed work.

5.2 SIMREC: a similarity measure recommendation system for mixed data clustering

This section presents SIMREC, the proposed similarity measure recommendation system for MDC algorithms, in particular algorithms based on the combination of a pair of numerical and categorical similarity measures. The remainder of the section is as follows. We formally state the problem in section 5.2.1. Section 5.2.2 gives an overview of SIMREC. Sections 5.2.3 and 5.2.4 present the learning and recommendation phases respectively.

5.2.1 Problem Statement

Let $\mathcal{X}$ be the set of mixed datasets, $\mathcal{A} = \{A_1, ..., A_U\}$ be a finite set of MDC algorithms, and $\mathcal{Q} = \{Q_1, ..., Q_V\}$ be a finite set of cluster validity indices. Let $\mathcal{S}^n$ and $\mathcal{S}^c$ be the sets of numerical and categorical similarity measures respectively. Let $\mathcal{S} = \{s_1, ..., s_K\} \subset \mathcal{S}^n \times \mathcal{S}^c$ be a finite subset of pairs of similarity measures. We denote $Q_v(A_u, s_k, X)$ the performance of the pair $s_k \in \mathcal{S}$ on the mixed dataset $X \in \mathcal{X}$ when using the MDC algorithm $A_u \in \mathcal{A}$ and the cluster validity index $Q_v \in \mathcal{Q}$.

We aim to create a similarity measure recommendation system that takes as input a mixed dataset $X \in \mathcal{X}$, an MDC algorithm $A_u \in \mathcal{A}$, and a cluster validity index $Q_v \in \mathcal{Q}$ and predicts the ranking of the pairs of similarity measures $s_k \in \mathcal{S}$ according to their performances $Q_v(A_u, s_k, X)$.

5.2.2 Overview

To address the stated problem, we need to answer the following questions: How can we predict the ranking of the pairs of similarity measures according to the considered dataset, MDC algorithm, and cluster validity index (CVI)? Which information should be exploited to make such a prediction?

Empirical studies have demonstrated that the performances of the similarity measures are related to the meta-features of the datasets to be clustered [Zhu 2020b]. An example of these meta-features is the dimension of the dataset. It is known that in high dimensions, similarity measures such as the Minkowski distances (Euclidean, Manhattan, etc.) tend to lose their discrimination power giving uniform similarities between observations which can lead to poor clustering performances. We also know, as shown in chapter 4, that the performances of the pairs of similarity measures depend on the considered MDC algorithm and CVI. So, by exploiting the relationship between the meta-features of the datasets and the performance of the pairs of similarity measures, we can learn, for each algorithm and CVI, how to predict the ranking of the pairs of similarity measures according to the faced dataset.

The proposed method involves two phases as illustrated in figure 5.1:

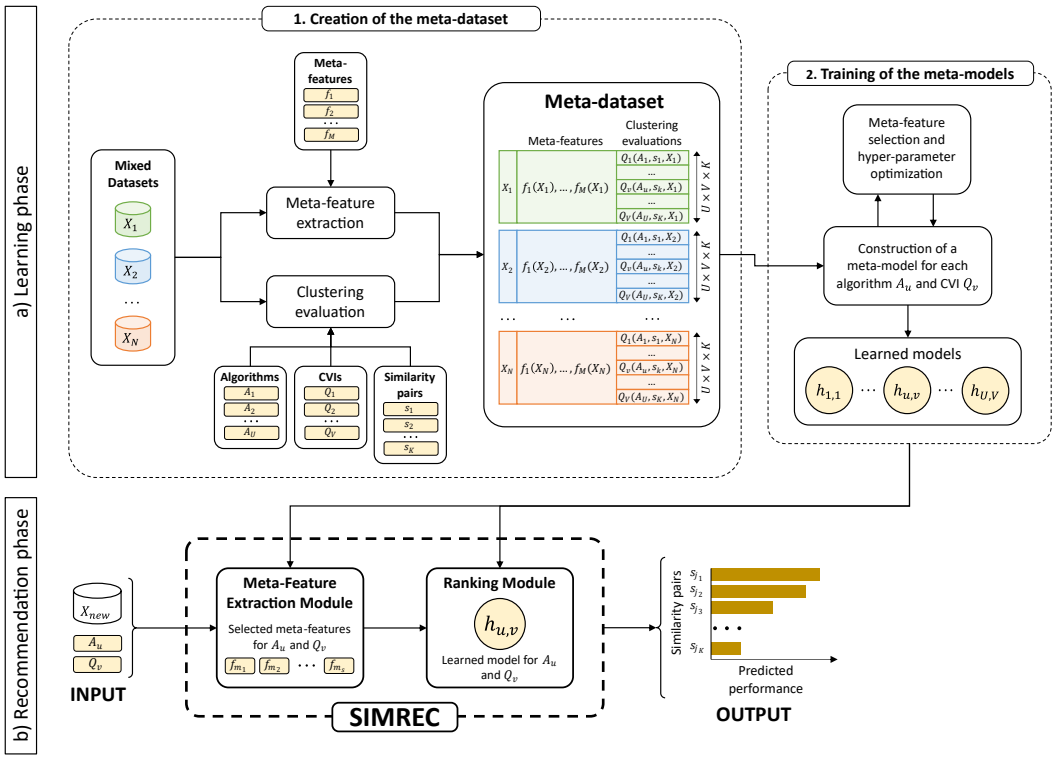


Figure 5.1: Learning and recommendation phases of SIMREC.

a) **The learning phase** (figure 5.1.a): In this phase, we use meta-learning to mine the relationship between the meta-features of the datasets and the performances of the pairs of similarity measures. For each algorithm $A_u \in \mathcal{A}$ and CVI $Q_v \in \mathcal{Q}$, the principle is to exploit prior learning experience, i.e. prior evaluations of the pairs of similarity measures on different mixed datasets with A_u and Q_v, to train a machine-learning model that learns to predict the ranking of the pairs of similarity measures based on the meta-features of the datasets.

b) **The recommendation phase** (figure 5.1.b): This phase corresponds to the use of SIMREC to recommend suitable pairs of similarity measures for new and unknown datasets, thanks to the models learned during the learning phase.

These two phases are described in more detail in the following sections. We introduce here some notations. Let $X = \{x_1, ..., x_{n_X}\} \in \mathcal{X}$ be a given mixed dataset such that $n_X = |X|$. Let p and q be the number of numerical and categorical attributes respectively. For a given observation $x_i \in X$, we denote $x_i^n = (x_{i,1}^n, ..., x_{i,p}^n)$ and $x_i^c = (x_{i,1}^c, ..., x_{i,q}^c)$ the numerical and categorical parts of x_i respectively. Finally, we denote $A_j^n = (x_{1,j}^n, ..., x_{n_X,j}^n)$ the j^{th} numerical attribute of

X and $A_l^c = (x_{1,l}^c, ..., x_{n_X,l}^c)$ the l^{th} categorical attribute of X with $1 \leq j \leq p$ and $1 \leq l \leq q$.

5.2.3 The Learning phase

Creation of the meta-dataset. The first step during the learning phase is to create a knowledge database, known as *meta-dataset*, containing the meta-data about different mixed datasets and prior learning experience on these datasets. These meta-data include the meta-features of the different datasets and the performances of the pairs of similarity measures on these datasets for the different algorithms and CVIs. In more detail, let us consider that we have a set of mixed datasets $\{X_i\}_{i=1}^N$. Let $\{f_m : \mathcal{X} \rightarrow \mathbb{R}\}_{m=1}^M$ be the considered meta-features (presented in section 5.2.3.1). For each dataset X_i, we first compute its meta-features $f_1(X_i), ..., f_M(X_i)$. Then for each algorithm $A_u \in \mathcal{A}$ and each pair of similarity measures $s_k \in \mathcal{S}$, we run A_u on X_i using the pair of similarity measures s_k. The obtained clustering is then evaluated using the different cluster validity indices $Q_v \in \mathcal{Q}$. Hence, for each dataset, we obtain $U \times V \times K$ evaluations with U, V, and K the number of algorithms, CVIs, and pairs of similarity measures respectively. Finally, the meta-features and clustering evaluations computed for each dataset are stored in the meta-dataset.

Training of the meta-models. The next step of the learning phase is to use the created meta-dataset to train, for each pair $(A_u, Q_v) \in \mathcal{A} \times \mathcal{Q}$, a meta-model that learns the mapping between the meta-features of the datasets and the ranking of the pairs of similarity measures for A_u and Q_v. We describe the meta-models in section 5.2.3.2. In addition, for each meta-model, we introduce a meta-feature selection and hyper-parameter optimization strategy based on a Genetic Algorithm (GA) (section 5.2.3.3) to select the optimal subset of meta-features and optimal hyper-parameters for the meta-model.

5.2.3.1 Meta-features

Our main hypothesis is that the performances of the pairs of similarity measures on a given dataset can be predicted based on the meta-features of the dataset. So, it is crucial to define meta-features that describe well the datasets and embed useful information for accurate prediction of the performances of the pairs of similarity measures.

Meta-features extracted from the literature Although several meta-features have been proposed in the literature [de Souto 2008, Ferrari 2015, Vukicevic 2016, Pimentel 2019, Pimentel 2020], we consider only meta-features based on statistical measures about the datasets and their attributes. The first reason for this choice is that other meta-features based on similarity, density, clustering evaluation, and landmarking have been designed for homogeneous (numerical) data and cannot be directly applied to mixed data. Second, they need a predefined similarity measure,

Table 5.1: Meta-features extracted from the literature

Type	Name	Description	Variants
GEN	Samples	Number of samples (N)	-
	Attributes	Number of attributes (d)	-
	Dim	Dimensionality (d/N)	-
	NumAtt	Number of numerical attributes (p)	-
	CatAtt	Number of categorical attributes (q)	-
	NumOnCat	p/q	-
NUM	MeansNumAtt	Means of numerical attributes	$min, q_1, mean, q_3, max$
	StdNumAtt	Standard deviations of numerical attributes	$min, q_1, mean, q_3, max$
	Covariance	Covariance between numerical attributes	$min, q_1, mean, q_3, max$
CAT	CardCatAtt	Cardinal of categorical attributes	$min, q_1, mean, q_3, max$
	EntropyCatAtt	Entropy of categorical attributes	$min, q_1, mean, q_3, max$

Table 5.2: Proposed Meta-features

Type	Name	Description	Variants
NUM	MeansSqNumAtt	Means of squared numerical attributes	$min, q_1, mean, q_3, max$
	StdSqNumAtt	Standard deviations of squared numerical attributes	$min, q_1, mean, q_3, max$
	MeansIntProdNumAtt	Means of internal product of numerical attributes	$min, q_1, mean, q_3, max$
	StdIntProdNumAtt	Std of internal product of numerical attributes	$min, q_1, mean, q_3, max$
CAT	StdFreqCatAtt	Std of frequencies of categorical attribute values	$min, q_1, mean, q_3, max$
	MutualInfoCatAtt	Mutual information between categorical attributes	$min, q_1, mean, q_3, max$

which is not trivial in the context of similarity measure recommendation. Furthermore, even if we can arbitrarily select one similarity measure (pair for mixed data) to use as in [Zhu 2020b], the computation of these meta-features can be very time-consuming since they are based on pairwise similarities between observations and evaluations of clustering results. We divide the 31 selected statistical meta-features into three categories: General statistics about the datasets (GEN), statistics about numerical attributes (NUM), and statistics about categorical attributes (CAT). They are presented in table 5.1.

Proposed new meta-features To complete the meta-features extracted from the literature, we propose 30 new meta-features (in table 5.2) that extract information about diverse notions exploited by the similarity measures. The first 10 meta-features are based on squared numerical attributes since several similarity measures for numerical data, such as *Euclidean distance* and *squared Euclidean distance*, use squared attribute values. For each numerical attribute A_j^n of the considered dataset, we compute the mean and standard deviation (std) of its squared values: $mean(\{u^2 : u \in A_j^n\})$ and $std(\{u^2 : u \in A_j^n\})$. Then, the meta-features are defined using the min, q_1, $mean$, q_3, and max values of the computed mean and standard deviation across all numerical attributes. Based on the same idea, the next 10 meta-features consider the internal products of numerical attribute values ($\{u.v : u, v \in A_j^n\}$) instead of the squared values.

The next 5 meta-features are based on the frequency of categorical attribute values. The aim is to provide some information about the balance between categories within the same attribute. This can give important insights into frequency-based similarity measures. Given a categorical attribute A_l^c, we compute the frequency of each category (u) within the attributes $\{\frac{\#u}{card(A_l^c)} : u \in set(A_l^c)\}$, where $\#u$ is the number of occurrences of u in A_l^c . Then, we use the standard deviation of these frequencies to estimate the balance between the categories. A low standard deviation indicates that the different categories within the attribute are well-balanced (i.e. they have similar frequencies) while a high standard deviation indicates unbalanced categories. Finally, the meta-features are defined as the min, q_1, $mean$, q_3, and max values of the standard deviations across all categorical attributes.

The last 5 meta-features are based on the mutual information between categorical attributes. They provide information about the relationships between the categorical attributes (in terms of shared information) and can give important insights for co-occurrence-based similarity measures such as the *Ahmad and Dey distance* [Ahmad 2007b]. We compute the mutual information [Cover 1999], I, between all pairs of categorical attributes $\{I(A_k^c, A_l^c) : 1 \leq k < l \leq q\}$. Then, the meta-features are defined as the min, q_1, $mean$, q_3, and max values of the computed mutual information values.

5.2.3.2 The Meta-models

For each pair $(A_u, Q_v) \in \mathcal{A} \times \mathcal{Q}$ of one MDC algorithm and one CVI, we want to learn a machine-learning model able to predict the ranking of the pairs of similarity measures according to the meta-features of the datasets. This problem is known as *label ranking* in the literature [Zhou 2014, de Sá 2017]. In our case, we transform the label ranking task into a multi-output regression task where the goal is simply to predict the performances of the different pairs of similarity measures. The predicted performances are then used to create the ranking. Let $h_{u,v} : \mathbb{R}^M \rightarrow \mathbb{R}^K$ be the meta-model for algorithm A_u and CVI Q_v. We recall that M is the number of meta-features and K is the number of pairs of similarity measures. $h_{u,v}$ is trained by minimizing the following loss:

$$\min \frac{1}{N} \sum_{i=1}^{N} \|h_{u,v}(\tilde{x}_i) - \tilde{y}_i^{u,v}\|_2^2 \tag{5.1}$$

where $\tilde{x}_i = [f_1(X_i), ..., f_M(X_i)] \in \mathbb{R}^M$ is the meta-feature vector of the dataset X_i. $\tilde{y}_i^{u,v} = [Q_v(A_u, s_1, X_i), ..., Q_v(A_u, s_K, X_i)] \in \mathbb{R}^K$ is the vector containing the performances of the different pairs of similarity measures on the dataset X_i for A_u and Q_v. $h_{u,v}$ can be any regression model that supports multiple outputs. We tested several models including k-Nearest Neighbors (KNN), ELasticNet, Decision Tree, Random Forest, and Neural Networks with different architectures. Results are shown only for the KNN model which gave the best results.

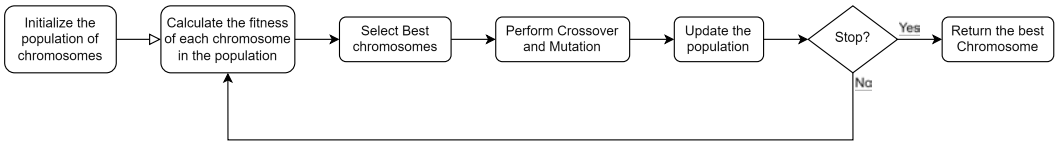


Figure 5.2: Flowchart of the used genetic algorithm

5.2.3.3 Meta-feature selection and hyper-parameter optimization

The 61 meta-features presented in section 5.2.3.1 have been considered because they are expected to carry useful information for accurate prediction of the performances of the pairs of similarity measures. However, there is no evidence about which meta-features are relevant or not. Therefore, instead of using all 61 meta-features, we introduce for each meta-model $h_{u,v}$, a meta-feature selection strategy based on a Genetic Algorithm (GA) [Kabir 2011].

The traditional approaches in feature selection can be broadly categorized into three approaches: filter, wrapper, and hybrid approaches [Kabir 2011]. The filter approach requires the statistical analysis of the feature set only to perform feature selection independently of the considered machine learning model. The wrapper approach selects the feature subset that optimizes the performance of the considered learning model [Liu 2005]. The hybrid approach attempts to take advantage of the filter and wrapper approaches. The filter approaches are faster to implement. However, the wrapper approaches generally give better results since they directly optimize the performance of the learning model. GA is one of the most advanced and successful techniques for feature selection due to its ability to search for global optima in large search space [Kabir 2011]. Given $(A_u, Q_v) \in \mathcal{A} \times \mathcal{Q}$, we use GA as a wrapper technique to find the meta-features subset that optimizes the performance of the corresponding meta-model $h_{u,v}$. We use the same algorithm to jointly optimize the hyper-parameters of $h_{u,v}$. The principle of the GA is illustrated in figure 5.2. Here are the main steps:

1. **Initialize the population.** The algorithm starts by randomly initializing a *chromosome* population. A chromosome (see figure 5.3) is a vector containing a binary value for each meta-feature (1 if the meta-feature is selected, 0 otherwise) and other values representing the hyper-parameters of the meta-model. Each value of the chromosome vector is called *gene*. Since we use KNN as the meta-model, we consider the following hyper-parameters: the number of nearest neighbors (that varies from 1 to 30), the distance metric (possible values are Euclidean distance, Manhattan distance, and Cosine dissimilarity), and the weights associated with the neighbors (possible values are "uniform" for uniform weights, and "distance" to weight each neighbor according to its distance to the considered observation).

2. **Compute fitness.** The algorithm computes the *fitness* of each chromosome in the population. The fitness of a chromosome is measured as the

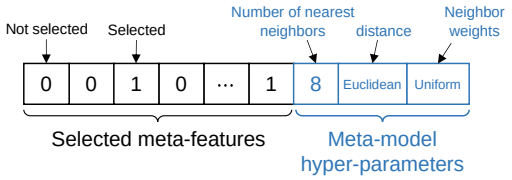


Figure 5.3: Illustration of a chromosome

cross-validation performance of the meta-model while using the selected meta-features and hyper-parameter values in the chromosome.

3. **Update the population.** Several chromosomes are selected according to their fitness values to create the next generation of chromosomes using *crossover* and *mutation* operations. A crossover operation is performed between two chromosomes (*parents*) and produces two new chromosomes (*children*) which inherit genes from both parent chromosomes. We use uniform crossover meaning that each gene of the children chromosomes is randomly copied from one of the two parents. A mutation operation takes one chromosome and randomly modifies one of its genes. These operations are performed with a certain probability defined by the user.

4. **Iterate.** This process continues with the newly created population until a stopping criterion is met (e.g. a maximum number of generations is reached). Then the best chromosome in the last generation is returned.

5.2.4 The recommendation phase

Figure 5.1.b illustrates the recommendation phase corresponding to the use of SIMREC in practice to perform similarity measure recommendation on new datasets thanks to the learned meta-model during the learning phase. SIMREC is composed of two modules: the **meta-feature extraction module** and the **ranking module**. Given a new and unknown mixed dataset X_{new}, an algorithm $A_u \in \mathcal{A}$, and a CVI $Q_v \in \mathcal{Q}$:

- The meta-feature extraction module computes the meta-feature vector of the dataset $\tilde{x}_{new} = [f_{m_1}(X_{new}), ..., f_{m_s}(X_{new})]$, with $f_{m_1}, ..., f_{m_s}$ the meta-features that have been selected by the meta-feature selection algorithm for A_u and Q_v, during the learning phase.

- The ranking module takes as input the computed meta-feature vector $\tilde{x}_{new}$. This vector is then fed to the meta-model $h_{u,v}$ learned during the learning phase for A_u and Q_v which predicts the performances $h_{u,v}(\tilde{x}_{new})$ of the pairs of similarity measures for the given input. Finally, the ranking of the pairs of similarity measures is determined according to the predicted performances

Algorithm 1 illustrates the recommendation procedure.

Algorithm 1 Recommendation algorithm

Input: X_{new}, A_u, Q_v
Output: The ranking of the pairs of similarity measures
$listFeats \leftarrow SelectedFeatures(A_u, Q_v)$
$\tilde{x}_{new} \leftarrow MetaFeatureExtraction(X_{new}, listFeats)$
$model \leftarrow GetMetaModel(A_u, Q_v)$
$prediction \leftarrow model(\tilde{x}_{new})$
$ranking \leftarrow GenerateRanking(prediction)$
return *ranking*

5.3 Implementation

This section introduces the current implementation of SIMREC and presents the considered MDC algorithms (section 5.3.1), cluster validity indices (section 5.3.2), and candidate numerical and categorical similarity measures (section 5.3.3). It is important to note that SIMREC is not limited to the MDC algorithms, CVIs, and similarity measures presented in this section. It can be extended with new MDC algorithms, CVIs, and similarity measures.

5.3.1 Clustering algorithms

We consider the following algorithms:

- **K-Prototypes (K-PROTO)** [Huang 1998]
- **Hierarchical Clustering with average linkage (H-AVG)** [Philip 1983]. we replace the Gower similarity used in [Philip 1983] by the following similarity measure for mixed data:

$$s(x_i, x_j) = (1 - w) \cdot s^n(x_i^n, x_j^n) + w \cdot s^c(x_i^c, x_j^c), \text{ with } w \in [0, 1] \tag{5.2}$$

- **K-Medoids (K-MED)** [Budiaji 2019]. We use the PAM version implemented in [Schubert 2021] and the same similarity measure as for H-AVG.

We chose these algorithms since they are commonly used and are among the most impacted by the choice of the pair of similarity measures as shown in chapter 4. Among the numerical and categorical similarity measures, the main parameters of these algorithms are the number of clusters and the combination weight of the two similarity measures. For the number of clusters, we use the number of classes in the ground truth. The combination weight is determined using a grid search strategy to find the weight that optimizes the clustering performances.

5.3.2 Cluster validity indices

Cluster validity indices (CVIs) are used to evaluate the performance of clustering algorithms. We distinguish 2 types of CVIs: external indices that rely on ground

truth labels (e.g. Clustering Accuracy) and internal ones that do not need ground truth labels (e.g. Silhouette). On the one hand, internal CVIs can be used if we want to recommend pairs of similarity measures that better capture some internal properties of the datasets such as compactness, separability, and homogeneity of the clusters. On the other hand, external metrics can be used to recommend pairs of similarity measures that better reflect the choice of an expert. Therefore, the choice of the CVI mainly depends on the user and its use cases.

We consider the following CVIs in SIMREC: the silhouette score (SIL) [Rousseeuw 1987], the clustering accuracy (CA) [Ahmad 2019], and the adjusted rand index (ARI) [Hubert 1985]. The silhouette score is an internal CVI while the other two are external indices. Let $(\hat{y}_1, ..., \hat{y}_{n_X})$ and $(y_1, ..., y_{n_X})$ be respectively the found cluster labels and ground truth labels for a given dataset X. Let k be the number of clusters. The metrics are defined below:

- **SIL.** The silhouette score measures the cohesion within clusters and the separation between clusters. Let a_i be the mean dissimilarity between the i^{th} observation and all other observations in the same cluster, and b_i the mean dissimilarity between the i^{th} observation and all other observations in the next closest cluster. The silhouette score is defined by:

$$SIL = \frac{1}{n_X} \sum_{i=1}^{n_X} \frac{b_i - a_i}{\max(a_i, b_i)} \tag{5.3}$$

  It takes values in $[-1, 1]$, where 1 indicates compact and well-separated clusters, and values around 0 (or less) indicate overlapping clusters.

- **CA.** The clustering accuracy is one of the most commonly used cluster validity indices for MDC. It is similar to the accuracy score used in classification tasks and is defined by:

$$CA(\hat{y}, y) = \max_{\sigma} \frac{1}{n_X} \sum_{i=1}^{n_X} 1\left(\sigma(\hat{y}_i) = y_i\right) \tag{5.4}$$

  Where σ is a permutation of $\{1, ..., k\}$ that maps each cluster label to a corresponding class label in the ground truth. $1\left(\sigma(\hat{y}_i) = y_i\right) = 1$ if $\sigma(\hat{y}_i) = y_i$, 0 otherwise. The accuracy score takes values in $[0, 1]$, and greater values indicate a better match between the found clusters and the ground truth labels.

- **ARI.** It measures the similarity between cluster labels and ground truth labels based on pairwise comparison. Let a be the number of samples pairs (x_i, x_j) such that $\hat{y}_i = \hat{y}_j$ and $y_i = y_j$. Let b the number of samples pairs (x_i, x_j) such that $\hat{y}_i \neq \hat{y}_j$ and $y_i \neq y_j$. The rand index (RI) is defined as follows.

$$RI(\hat{y}, y) = \frac{a + b}{C_2^{n_X}} \tag{5.5}$$

  Where $C_2^{n_X}$ is the total number of sample pairs. $a + b$ represents the number

of pairs for which the clustering and the ground truth agree. The ARI is a correction of the rand index (RI) which ensures that a random label assignment gets a value close to zero.

$$ARI(\hat{y}, y) = \frac{RI(\hat{y}, y) - E[RI]}{RI_{max} - E[RI]} \tag{5.6}$$

Where $E[RI]$ is the expected rand index for a random clustering and RI_{max} is the maximal rand index $RI(y, y)$. The ARI score takes values in $[-0.5, 1]$, where 1 indicates a perfect match between the found clusters and the ground truth labels, and values around 0 (or less) indicate random clustering.

When evaluating clustering performance using external indices, the best scenario would involve using labels assigned by experts, as these reflect expert judgments of good clusters. However, since expert labels are often unavailable, we typically use class labels as a substitute for ground truth labels. The issue with this approach is that class labels may not always represent good clusters. A class label might group data points that an expert or a good clustering method would not. Therefore, in our analysis, we consider for external indices, only datasets for which at least one of the considered MDC algorithms has a high clustering accuracy ($CA \geq 0.7$), indicating that the ground truth labels (class labels) are likely to represent good clusters.

5.3.3 Candidate similarity measures

As in the previous chapters, we use the same similarity measures presented in the related works (section 2.2). These measures consist of 10 similarity measures for numerical data and 12 for categorical data, leading to 120 pairs of similarity measures. They represent well-known and representative measures for the two data types.

5.4 Experiments

To show the relevance of the proposed recommendation system, we conducted experiments to answer the following questions.

- **RQ1 (Effectiveness and Generalization).** How does SIMREC perform on new and unknown datasets for the considered MDC algorithms and cluster validity indices?
- **RQ2 (Impact of the meta-feature selection).** How does the meta-feature selection contribute to the performance of SIMREC?
- **RQ3 (Importance of the meta-features).** How are the different meta-features, especially the proposed ones, involved in the predictions of the meta-models?

Table 5.3: Datasets description: for each statistic, we show its minimum and maximum values as well as its three quartiles

	# of attributes	# of samples	# of numerical attributes	# of categorical attributes	# of classes
min	3	34	1	2	2
25%	9	205	3	3	2
50%	16	690	7	6	2
75%	30	5.28K	15	15	3
max	1.64K	5.1M	1.6K	1.56K	102

- **RQ4 (Efficiency).** What about the efficiency of the proposed method, in terms of computation time of the learning and recommendation phases?

5.4.1 Datasets

The datasets have been selected from the OpenML platform [Vanschoren 2014]. We considered all the available mixed datasets in the platform. They all have ground truth labels. We manually filtered the datasets to remove redundant datasets (since several datasets have duplicates in OpenML). Thus, we ended with 185 datasets. Table 5.3 presents some descriptive statistics about these datasets. Finally, due to computational constraints, datasets with a large number of observations have been down-sampled to a maximum of 10000 observations.

5.4.2 Baselines

The performances of the learned meta-models are compared to the performances obtained with the following baselines:

- **Random Baseline (RB).** This baseline uses a random pair of similarity measures.
- **Literature Baseline (LB).** This baseline uses the pair of similarity measures used in the literature for the considered clustering algorithm and can be considered as the default choice of a data scientist when there is no tool to select suitable similarity measures automatically. For K-Prototypes [Huang 1998], the pair (*Squared Euclidean*, *Hamming*) is used. For H-AVG, the Gower similarity used in [Philip 1983] is equivalent to the pair (*Manhattan*, *Hamming*). For K-Medoids, we choose the pair (*Euclidean*, *Hamming*) from those used in [Budiaji 2019].
- **Average Ranking Baseline (AR).** This baseline ranks the pairs of similarity measures according to their average performance on all datasets in the meta-dataset. This baseline is more complex than the previous ones since it partially exploits the knowledge in the meta-dataset.

5.4.3 Evaluation of the predictions of SIMREC

Let $s_{\hat{\pi}_1} \succ ... \succ s_{\hat{\pi}_K}$ and $s_{\pi_1} \succ ... \succ s_{\pi_K}$ be respectively the ranking predicted by SIMREC and the true ranking of the pairs of similarity measures for a given dataset X, MDC algorithm A_u, and CVI Q_v. $\hat{\pi}$ and π are two permutations of the set $\{1, .., K\}$. $s_k \succ s_l$ indicate that the pair of similarity measures s_k is better than s_l. We consider two metrics to evaluate the predicted ranking:

- **top-r**: it evaluates the quality of the r top-ranked pairs of similarity measures.

$$\text{top-}r = \frac{\max_{k=1}^{r} Q_v(A_u, s_{\hat{\pi}_k}, X) - Q_v^{min}}{Q_v(A_u, s_{\pi_1}, X) - Q_v^{min}} \tag{5.7}$$

  where Q_v^{min} is the lower bound of Q_v. We have $Q_v^{min} = -1$, -0.5, and 0 for SIL, ARI, and CA respectively.

- **NDCG** (Normalized Discounted Cumulative Gain): A metric based on the notion of Discounted Cumulative Gain (DCG), which evaluates the quality and the ranking of the top-ranked pairs of similarity measures. The DCG at rank r is defined by:

$$DCG@r = \sum_{k=1}^{r} \frac{rel(s_{\hat{\pi}_k}, X)}{log_2(k+1)}, \text{ where } rel(s_k, X) = \left(\frac{Q_v(A_u, s_k, X) - Q_v^{min}}{Q_v(A_u, s_{\pi_1}, X) - Q_v^{min}} \right)^{\alpha} \tag{5.8}$$

  $rel(s_k, X)$ is the relevance of s_k for X. α is a positive number that controls how the relevance decreases when the performance of s_k decreases relative to the performance of the best pair. We use $\alpha = 4$ in the experiments such that the relevance drops quickly when the performance of s_k deviates from that of the best pair. The NDCG is then defined by normalizing the DCG with the Ideal DCG (IDCG), which corresponds to the DCG of the true ranking:

$$NDCG@r = \frac{DCG@r}{IDCG@r} \tag{5.9}$$

5.4.4 Experimental protocol

1. **Preprocessing.** The retrieved datasets are preprocessed by removing constant attributes, handling missing values, and normalizing the numerical attributes to the range $[0, 1]$. Missing values are handled by removing attributes with more than 50% missing values. Then, if the number of observations containing at least one missing value is greater than 20% of the total number of observations, the missing values are replaced using the mean value for numerical attributes and the majority value for categorical attributes. However, if this number is less than 20%, the corresponding observations are dropped.

2. **Meta-dataset creation.** We create the meta-dataset as described in section 5.2.3. For each dataset, we compute its meta-features and evaluate the per-

formances of the pairs of similarity measures on the dataset for all considered MDC algorithms and CVIs. As stated in section 5.3.2, using ground truth labels (class labels) as a reference for external CVIs can be problematic since these labels may not align well with what an expert would consider a good clustering. So, for external CVIs, only datasets for which at least one of the considered MDC algorithms has a high clustering accuracy ($CA \geq 0.7$) are considered (i.e. 102 datasets out of 185), indicating that the ground truth labels are likely to represent good clusters.

3. **Training and evaluation.** We train the different meta-models and evaluate their generalization performances using a 10-fold procedure. There are 9 meta-models since we have 3 MDC algorithms and 3 CVIs. Given one meta-model, the datasets are divided into 10 folds. We realize 10 iterations such that at each iteration the meta-model is trained on the datasets in the 9 folds and tested on the datasets in the one remaining fold. At each iteration, the rankings predicted by the meta-model on the test datasets are evaluated using the top-r metric for $r = 1$ and $r = 10$ and the $NDCG@r$ metric for $r = 10$. At the end of the iterations, each dataset has been used once for testing. Therefore, we have the generalization performance of the meta-model for each dataset.

4. **Meta-feature selection and hyper-parameter optimization.** For each meta-model, the presented genetic algorithm is used to find the meta-feature subset and the hyper-parameter configuration that optimizes its generalization performance. This performance is computed as the mean of the top-1 scores measured during the 10-fold procedure. The genetic algorithm runs over 300 generations with a population size of 16 chromosomes, a steady state selection scheme, a crossover probability of 95%, and a mutation probability of 5%. The results presented in the following are those obtained with the found meta-feature subset and hyper-parameters.

5.5 Experiment Results

5.5.1 RQ1. Effectiveness and Generalization

In this section, we assess the quality of the recommendations of SIMREC on new and unknown datasets compared to the different baselines. Table 5.4 shows for each MDC algorithm and CVI, the mean and standard deviation ($mean \pm std$) of the top-1 scores across all datasets. The best results (of SIMREC and the baselines except the ORACLE) are represented in **bold**. The percentage right to the SIMREC result indicates the improvement of SIMREC relative to the literature baseline (LB). We observe that SIMREC outperforms the baselines for the all MDC algorithms and CVIs. Furthermore, we obtain important improvements compared to the literature

Table 5.4: (*mean* ± *std*) of the top-1 scores across all datasets, for the different algorithms and CVIs. The best results are represented in **bold**. The percentage right to the SIMREC result indicates the corresponding improvement relative to the literature baseline (LB).

Algorithm	CVI	LB	RB	AR	SIMREC	ORACLE
K-PROTO	SIL	0.473±0.23	0.443±0.22	0.570±0.2	**0.608±0.20 ↑28%**	0.685±0.20
	ARI	0.236±0.24	0.24±0.24	0.238±0.24	**0.283±0.26 ↑19%**	0.37±0.26
	CA	0.709±0.12	0.716±0.12	0.744±0.11	**0.753±0.11 ↑6%**	0.809±0.09
K-MED	SIL	0.481±0.27	0.478±0.25	0.614±0.23	**0.656±0.22 ↑36%**	0.735±0.2
	ARI	0.231±0.23	0.224±0.23	0.228±0.23	**0.287±0.27 ↑24%**	0.347±0.27
	CA	0.727±0.12	0.724±0.11	0.723±0.12	**0.758±0.12 ↑4%**	0.795±0.10
H-AVG	SIL	0.523±0.2	0.550±0.2	0.654±0.18	**0.726±0.18 ↑38%**	0.785±0.17
	ARI	0.217±0.24	0.169±0.22	0.245±0.27	**0.268±0.27 ↑23%**	0.329±0.27
	CA	0.789±0.11	0.776±0.11	0.797±0.11	**0.804±0.10 ↑1%**	0.833±0.09

Table 5.5: Obtained *p-values* when comparing SIMREC to the baselines using the top-1 metric. Given a baseline, an algorithm A_u, and a CVI Q_v, a *p-value* ≤ 0.05 indicates that SIMREC significantly outperforms the baseline for algorithm A_u and CVI Q_v.

(a) SIMREC vs AR

	K-PROTO	**K-MED**	**H-AVG**
SIL	0.000	0.000	0.000
ARI	0.000	0.002	0.004
CA	0.003	0.000	0.037

(b) SIMREC vs LB

	K-PROTO	**K-MED**	**H-AVG**
SIL	0.000	0.000	0.000
ARI	0.000	0.000	0.001
CA	0.000	0.001	0.011

(c) SIMREC vs RB

	K-PROTO	**K-MED**	**H-AVG**
SIL	0.000	0.000	0.000
ARI	0.000	0.000	0.000
CA	0.000	0.000	0.000

Table 5.6: ($mean \pm std$) of the top-10 and $NDCG$@10 scores across all datasets, for the different algorithms and CVIs. The best results are represented in **bold**

Algorithm	CVI	top-10		$NDCG$@10	
		AR	SIMREC	AR	SIMREC
K-PROTO	SIL	0.626±0.21	**0.645±0.2**	0.529±0.27	**0.627±0.29**
	ARI	0.291±0.25	**0.318±0.26**	0.362±0.28	**0.463±0.30**
	CA	0.766±0.11	**0.777±0.10**	0.491±0.33	**0.536±0.32**
K-MED	SIL	0.684±0.21	**0.699±0.21**	0.538±0.25	**0.632±0.28**
	ARI	0.288±0.26	**0.308±0.27**	0.387±0.26	**0.459±0.27**
	CA	0.758±0.11	**0.773±0.12**	0.493±0.31	**0.567±0.29**
H-AVG	SIL	0.728±0.19	**0.749±0.17**	0.511±0.27	**0.685±0.29**
	ARI	0.285±0.27	**0.293±0.27**	0.478±0.3	**0.504±0.31**
	CA	**0.818±0.1**	0.817±0.09	0.693±0.28	**0.698±0.28**

baseline. For the silhouette score, we obtain more than 30% of improvement for K-MED and H-AVG algorithms. For ARI, we obtain more than 20% of improvement for K-MED and H-AVG algorithms. For the CA index, we obtain up to 6% of improvement for the K-Prototypes algorithm.

To confirm these observations and check if the superiority of SIMREC over the baselines is significant, we use the Wilcoxon signed-rank test [Wilcoxon 1945]. It is a non-parametric statistical hypothesis test used to compare two related paired samples. Given a baseline, an algorithm A_u, and a CVI Q_v, we consider the paired performances (measured with the top-1 metric) of SIMREC and the baseline on all datasets for A_u and Q_v". Then, we test the null hypothesis H_0=*"The difference between the top-1 scores of SIMREC and the baseline is zero for A_u and Q_v"* at the 0.05 level of significance.

Tables 5.5a, 5.5b, and 5.5c show the obtained *p-values* for the AR, LB, and RB baselines respectively. All *p-values* are smaller than 0.05. So, we can reject the null hypothesis for all baselines, algorithms, and CVIs. This means that the difference between the top-1 scores of SIMREC and the baselines is statistically significant for all considered algorithms and CVIs. And since SIMREC obtains higher scores according to table 5.4, we can finally conclude that for the top-1 metric, SIMREC significantly outperforms the baselines for all considered algorithms and CVIs. **These results indicate that, for all the considered clustering algorithms and CVIs, the top-ranked pair of similarity measures in the ranking predicted by SIMREC significantly outperforms the literature and random baselines as well as the top-ranked pair of the AR baseline.**

Table 5.6 shows for each MDC algorithm and CVI, the mean and standard deviation ($mean \pm std$) of the top-10 and $NDCG$@10 across all datasets. The best

(a) top-10

	K-PROTO	K-MED	H-AVG
SIL	0.000	0.000	0.000
ARI	0.000	0.002	0.762
CA	0.034	0.004	0.583

(b) $NDCG$@10

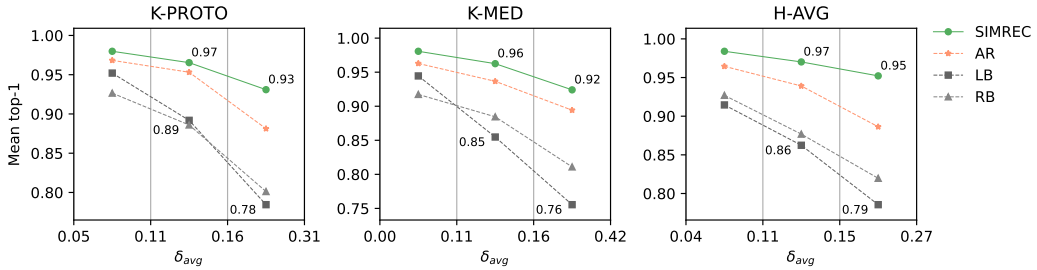
	K-PROTO	K-MED	H-AVG
SIL	0.000	0.000	0.000
ARI	0.007	0.002	0.222
CA	0.041	0.023	0.184

Table 5.7: Obtained *p-values* when comparing SIMREC to the AR baseline for the top-10 and $NDCG$@10 metrics. Given an algorithm A_u, and a CVI Q_v, a *p-value* ≤ 0.05 indicates that SIMREC significantly outperforms the AR baseline for algorithm A_u and CVI Q_v.

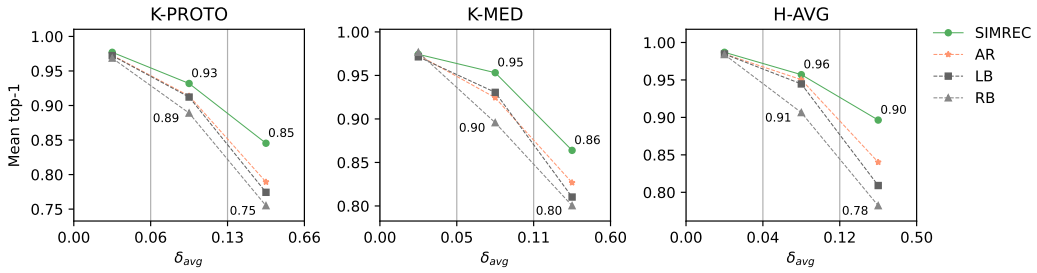
results for each metric are represented in **bold**. For the top-10 metrics, SIMREC outperforms the AR baseline in most cases but yields similar results for the H-AVG algorithm when using the CA index. Table 5.7a shows for each algorithm A_u and CVI Q_v, the obtained *p-value* when comparing the top-10 scores of SIMREC to those of the AR baseline. We use the Wilcoxon signed-rank test with the null hypothesis H_0=*"The difference between the top*-10 *scores of SIMREC and the AR baseline is zero for* A_u *and* Q_v*"*. The table confirms that, in most cases, SIMREC significantly outperforms the AR baseline for the top-10 metric (*p-values* ≤ 0.05), except for the H-AVG algorithm when using the external cluster validity indices CA and ARI for which the null hypothesis cannot be rejected. **This shows that for most of the considered algorithms and CVI, the 10 best similarity pairs recommended by SIMREC contain significantly higher quality pairs than the 10 best similarity pairs of the AR baseline.**

For the $NDCG$@10 metric, we can observe from table 5.6 that SIMREC outperforms the AR baseline for all algorithms and CVIs. Table 5.7b shows for each algorithm A_u and CVI Q_v, the obtained *p-value* when comparing the $NDCG$@10 scores of SIMREC to those of the AR baseline. We use the Wilcoxon signed-rank test with the null hypothesis H_0=*"The difference between the* $NDCG$@10 *scores of SIMREC and the AR baseline is zero for* A_u *and* Q_v*"*. The test results confirm that, for the $NDCG$@10 metric, SIMREC significantly outperforms the AR baseline (*p-values* ≤ 0.05) for the different algorithms and CVIs except for the H-AVG algorithm when using the external cluster validity indices CA and ARI. **To conclude, for most of the considered algorithms and CVI, SIMREC is significantly better at identifying and ranking the top-performing similarity pairs than the AR baseline.**

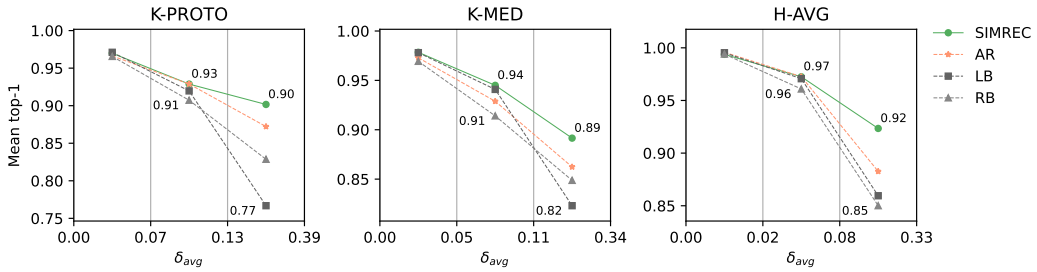
The previous results show the ability of the meta-model to identify and rank the top-performing pairs of similarity measures. However, it is easy to recommend a suitable pair of similarity measures when all pairs yield similar performances. So, it would be interesting to observe the behavior of SIMREC on datasets for which there is a large difference between the performances of the different pairs of similarity measures. We introduce the notion of the *difficulty* of a dataset. Given an MDC algorithm A_u and a cluster validity index Q_v, we consider a dataset X as



(a) SIL



(b) ARI



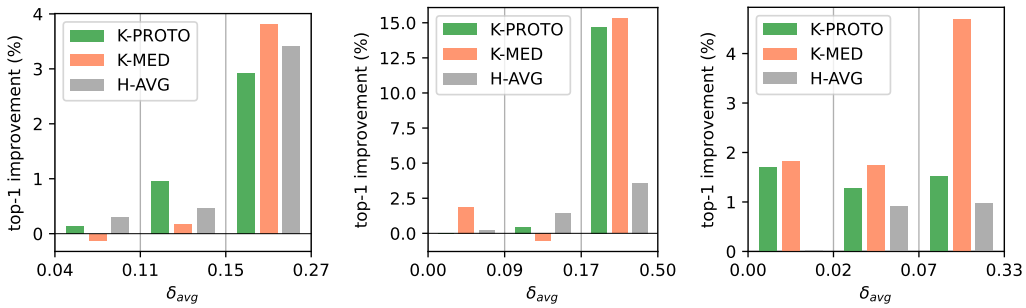
(c) CA

Figure 5.4: Mean top-1 scores of SIMREC and the baselines according to the values of δ_{avg}. Given a dataset, δ_{avg} indicates the variation of clustering performance due to the choice of the pair of similarity measures. The higher is δ_{avg}, the more important to choose the right similarity pair.

difficult when a randomly chosen pair has a low probability of being close (in terms of clustering performance) to the best pair. This means that the average value δ_{avg} (equation 5.10) of the difference of performance δ_k between any given pair s_k and the best pair for that dataset is large. Inversely, the dataset is considered easy when δ_{avg} is small (i.e. all pairs of similarity measures perform similarly to the best pair). The more difficult the dataset is, the more it is important to correctly choose the pair of similarity measures.

$$\delta_{avg} = \frac{1}{K}\sum_{k=1}^{K}\delta_k, \text{ where } \delta_k = 1 - \frac{Q_v(A_u, s_k, X) - Q_v^{min}}{\max_l Q_v(A_u, s_l, X) - Q_v^{min}} \tag{5.10}$$

For each pair of a clustering algorithm and a cluster validity index, we divide the values of δ_{avg} into three intervals of equal number of datasets to simulate different difficulty levels. Figure 5.4 shows the mean top-1 scores of SIMREC and the



(a) SIL (b) ARI (c) CA

Figure 5.5: Obtained improvement when using meta-feature selection compared to using all meta-features. Given a dataset, δ_{avg} indicates the variation of clustering performance due to the choice of the pair of similarity measures. The higher is δ_{avg}, the more important it is to suitably choose the similarity pair.

baselines for each interval. As expected, for small values of δ_{avg}, SIMREC and the baselines yield similar results. This is less the case for the cluster validity index SIL for which we can see that SIMREC significantly outperforms the baselines even if these datasets have small δ_{avg}. When δ_{avg} increases, i.e. when the choice of the pair of similarity measures is more important, we observe that the baselines are more impacted, and their performances decrease more rapidly compared to the performances of SIMREC. **So, the more difficult the dataset is (i.e. the more it is important to correctly choose the pair of similarity measures), the more interesting it is to use SIMREC compared to the different baselines.** Interestingly, when we consider the literature baseline for example, for datasets with high δ_{avg} we obtain up to 19%, 26%, and 22% of improvement for K-Prototypes, K-Medoids, and H-AVG algorithms respectively when using SIL as the cluster validity index. For ARI, we obtain up to 15%, 17%, and 14% of improvement respectively. For CA we obtain up to 17%, 12%, and 6% of improvement respectively.

5.5.2 RQ2. Impact of the meta-feature selection

To evaluate the impact of the meta-feature selection, the meta-models are trained using all 61 meta-features, i.e. meta-feature selection is not done. We use a 10-fold procedure to evaluate the models as in the previous section. Hyper-parameter optimization is done using a grid search strategy (since we do not perform meta-feature selection, the GA is not used). Then, we compare the obtained performances when using all meta-features to those obtained with the meta-feature selection algorithm. Figure 5.5 shows the obtained improvement due to the meta-feature selection. We observe important improvements for the different clustering algorithms and cluster validity indices, particularly for datasets with high δ_{avg}, i.e. datasets for which the choice of the pair of similarity measures is more important. Interestingly, for these datasets, we obtain up to 15%, 5%, and 3% improvement for the cluster validity indices ARI (with K-PROTO and K-MED), CA (with K-MED), and SIL, respec-

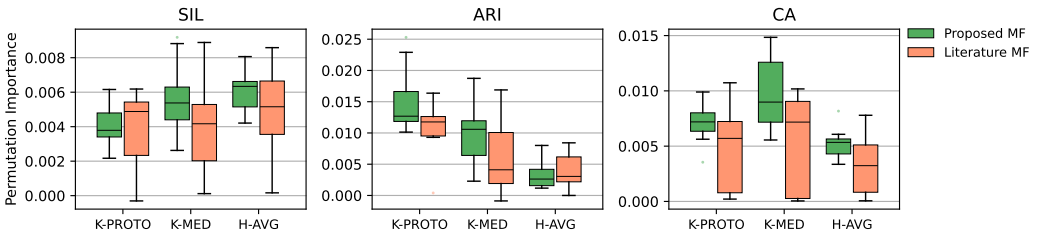


Figure 5.6: Permutation Importance of the proposed meta-features (MF) compared to the literature meta-features

tively. These results show that, by identifying the most relevant meta-features for the considered algorithm and cluster validity index, meta-feature selection can improve the performances of SIMREC, especially for datasets highly impacted by the choice of the pair of similarity measures.

5.5.3 RQ3. Importance of the meta-features

In this section, we conduct a meta-feature importance study to analyze how the different meta-features, especially our proposed ones, influence the predictions of the meta-models.

We use the permutation importance to evaluate the importance of the different meta-features. Only meta-features selected by the meta-feature selection algorithm are considered for each meta-model. The permutation importance of a given meta-feature for a given meta-model is computed as follows. First, the meta-dataset is divided into 10 folds as in previous sections. At each iteration, the meta-model is trained on the nine folds and tested on the remaining fold (using only the meta-features selected by the meta-feature selection algorithm). The performance of the meta-model on the test fold is measured using the top-1 metric. Next, the values of the considered meta-feature are randomly permuted in the test fold and the meta-model is evaluated on the permuted test fold. If the permutation of the meta-feature decreases the performance of the meta-model, then the meta-feature is considered important. The difference between the obtained performances before and after the permutation is defined as the permutation importance. Since we use random permutations, the process is repeated 10 times and the average value is used as the permutation importance of the meta-feature for the considered test fold. Finally, the average permutation importance on all folds is defined as the permutation importance of the meta-feature.

Figure 5.6 shows the permutation importance of the proposed meta-features (from the meta-features that have been selected by the meta-feature selection algorithm) compared to the literature meta-features. We observe that the proposed meta-features are globally more important than the literature ones, for most clustering algorithms and cluster validity indices. Furthermore, we can see from table

Table 5.8: Number and proportion of meta-features selected by the meta-feature selection algorithm in each meta-feature subset

(a) SIL

	GEN	NUM	CAT	Proposed NUM	Proposed CAT	TOT
K-PROTO	1 (4%)	5 (19%)	6 (23%)	10 (38%)	4 (15%)	26
K-MED	2 (7%)	7 (24%)	4 (14%)	11 (38%)	5 (17%)	29
H-AVG	3 (12%)	6 (24%)	6 (24%)	5 (20%)	5 (20%)	25

(b) ARI

	GEN	NUM	CAT	Proposed NUM	Proposed CAT	TOT
K-PROTO	3 (13%)	3 (13%)	7 (29%)	4 (16%)	7 (29%)	24
K-MED	1 (3%)	7 (22%)	7 (22%)	10 (31%)	7 (22%)	32
H-AVG	2 (7%)	6 (22%)	3 (11%)	9 (33%)	7 (26%)	27

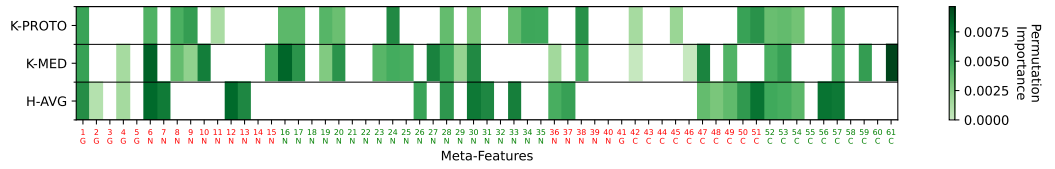
(c) CA

	GEN	NUM	CAT	Proposed NUM	Proposed CAT	TOT
K-PROTO	4 (13%)	8 (26)	5 (16%)	10 (32%)	4 (13%)	31
K-MED	1 (4%)	8 (30%)	6 (22%)	6 (22%)	6 (22%)	27
H-AVG	3 (11%)	8 (30%)	6 (22%)	10 (37%)	0 (0%)	27

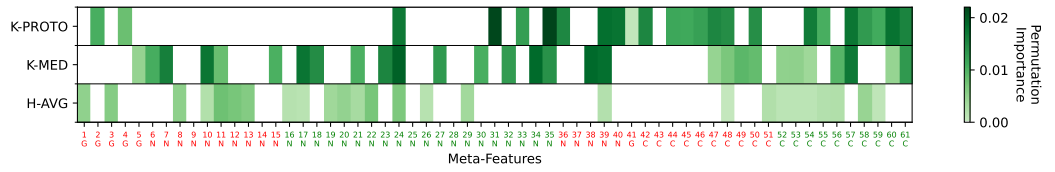
5.8 that the proposed meta-features represent a significant part (between 37% and 59%) of the meta-features selected by the meta-feature selection algorithm for the different clustering algorithm and cluster validity indices. These results show that the proposed meta-features embed useful information for predicting the ranking of the pairs of similarity measures, and show their interest for the similarity measure recommendation task.

Figure 5.7 gives more detail about the permutation importance of the different meta-features. Given one algorithm and one CVI, the figure shows the permutation importance of the meta-features selected by the feature selection algorithm. The 61 meta-features are represented in abscissa by their indices $(1, 2, ..., 61)$. The proposed new meta-features are colored in green while the meta-features selected from the literature are colored in red. The importance of a given meta-feature is determined by the color of the corresponding box. If a meta-feature has not been selected by the feature selection algorithm, the corresponding box is white.

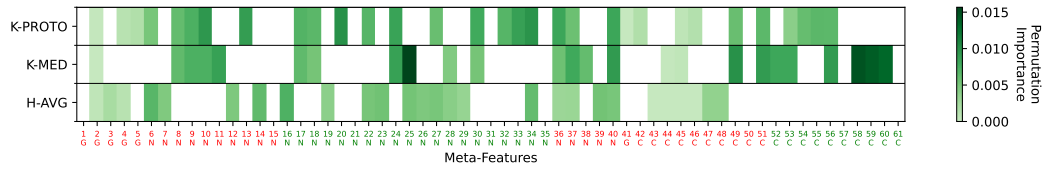
First of all, we observe that each meta-feature has been selected at least for one of the nine possible pairs of clustering algorithms and CVIs. This shows that all meta-features, including the proposed ones, are important and can be useful depending on



(a) SIL



(b) ARI



(c) CA

Figure 5.7: Permutation importance of the meta-features according to the considered clustering algorithm and cluster validity index. The meta-features are represented in abscissa by their indices $(1, 2, ..., 61)$. Proposed new meta-features are colored in green while the meta-features extracted from the literature are colored in red. The letters below the meta-features denote the type of meta-feature (G for GEN, N for NUM, and C for CAT). White boxes indicate meta-features that have not been selected by the meta-feature selection algorithm.

the considered algorithm and CVI. Furthermore, we observe an important variation in the importance of the meta-feature according to the considered algorithm and CVI. This confirms (again) the interest of the feature-selection strategy to identify the most relevant meta-features according to the faced algorithm and CVI.

5.5.4 RQ4. Efficiency of the proposed recommendation approach

In this section, we study the efficiency of the proposed approach, in terms of the computation time of the learning and recommendation phases. The learning phase is the most time-intensive. Table 5.9 shows the total computation times of the different tasks during the learning phase. The first step of the learning phase, i.e. the creation of the meta-dataset, involves the computation of the meta-features and the evaluation of the considered MDC algorithms with the different pairs of similarity measures (and the search, for each pair, of the best combination weight) for all the retrieved mixed datasets. The meta-feature extraction took 7 min 25 s. The evaluation of the clustering algorithms took about 15, 8, and 5 days respectively for K-Prototypes, K-Medoids, and H-AVG. These computations have been carried out on a platform with 16 CPUs - Intel Xeon Gold 6226R CPU@2.90GHz - using parallel execution over the datasets. The second step of the learning phase, i.e. the

Table 5.9: Time analysis of the creation of the meta-dataset

Task	Description	Time
Meta-feature extraction	Computation of the 61 meta-features for the 185 datasets	7 min 25 s
Clustering benchmark with K-PROTO	Evaluation of K-PROTO on all 185 datasets with the 120 similarity measure pairs and 40 different values for the combination weight, using the 3 cluster validity indices: For a total of **2.64M evaluations**.	15 days
Clustering benchmark with K-MED	Evaluation of K-MED on all 185 datasets with the 120 similarity measure pairs and 51 different values for the combination weight, using the 3 cluster validity indices: For a total of **3.4M evaluations**.	8 days
Clustering benchmark with H-AVG	Evaluation of H-AVG on all 185 datasets with the 120 similarity measure pairs and 51 different values for the combination weight, using the 3 cluster validity indices: For a total of **3.4M evaluations**.	5 days
Training of the meta-models	Training of the 9 meta-models on the hole meta-dataset. The training time includes the GA based meta-feature selection	9 min 50 s

training of the meta-models, involves the training of the 9 meta-models on the whole meta-dataset, including the GA-based meta-feature selection and hyper-parameter optimization for each meta-model. The step lasted 9 min 50 s in total, i.e., 1 min 5 s per meta-model on average. We can notice that creating the meta-dataset, especially evaluating the performances of the different MDC algorithms and pairs of similarity measures on the retrieved mixed datasets using different CVIs, is the most computational task during the learning phase. However, it is important to note that these computations are done only once and the computational cost is not a major problem since it only concerns the (offline) learning phase.

The recommendation phase only involves the computation of the meta-features of the considered dataset and the prediction of the ranking of the pairs of similarity measures. In figure 5.8, we show the inference time of SIMREC for the different datasets. For comparison, we consider, for each dataset, the clustering time of K-Prototypes on that dataset averaged over all pairs of similarity measures. The clustering time includes the search for the best combination weight of the pair of similarity measures. Note that we use a logarithmic scale for the time axis. The average clustering time varies from nearly 1 second to 10^4 seconds depending on the

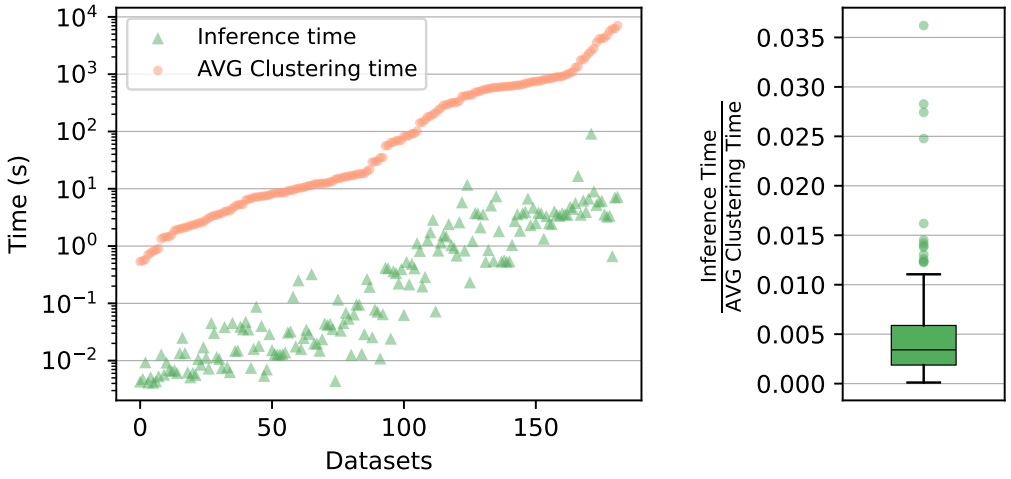


Figure 5.8: **Left:** Inference time of SIMREC compared to the average clustering time for the K-Prototypes algorithm. Each index on abscissa corresponds to one dataset, the datasets are ordered per average clustering time. Log scale is used for the ordinates. **Right:** Distribution of the ratio between the inference time and the average clustering time over all datasets.

dataset. We can see that the inference time is less than the average clustering time for all datasets. The ratio between the inference time and the average clustering time is less than 0.04 for all datasets. Therefore, we can conclude that the overhead added by the similarity measure recommendation is negligible compared to the duration of the clustering process, for most datasets.

5.6 Discussion

Our experiments show that the proposed similarity measure recommendation system (SIMREC) can be used as an effective and efficient solution for the recommendation of similarity measures in the context of MDC. This study complements existing studies in the literature to support data scientists in tasks such as clustering algorithm selection, algorithm parameter setting, similarity measures selection, and so on. SIMREC focuses on this later task and allows, once the user has selected a clustering algorithm, to drastically reduce the needed effort to find suitable pairs of similarity measures. However, once the user has identified one pair of similarity measures to use, she/he will need, as usual, to search for the optimal parameters of the clustering algorithm. Therefore, an interesting direction for future work is to design a unified framework that allows the data scientist to jointly perform all the mentioned tasks.

One might perceive meta-learning as a costly solution since the creation of the meta-dataset needs prior evaluations of the considered clustering algorithms on several datasets using different similarity measures and parameter configurations.

However, this is not a major problem since these evaluations are computed only during the learning phase. Moreover, after this phase, the use in practice of the recommendation system helps avoid expensive trial-and-error strategies, leading to significant time and energy savings. Nevertheless, the cost of the learning phase can be mitigated by considering several strategies. For example, compared to the grid search strategy we used in the experiments, more efficient strategies (such as Bayesian Optimisation) can be considered for the search of the optimal parameters of the clustering algorithm. Also, fewer similarity measures can be considered by exploiting similarity between similarity measures.

It is also important to note that SIMREC can be easily extended with other mixed data clustering algorithms and cluster validity indices to address more use cases in practice by simply adding (i.e. training) new meta-models for these algorithms and cluster validity indices. Furthermore, it turns out that, in addition to similarity measure recommendation, SIMREC can be used to recommend the clustering algorithm by comparing the predictions of the meta-models that have been trained for different algorithms. However, this needs further research to evaluate the efficiency of SIMREC in such a task.

One of the core components of SIMREC is the meta-feature extraction. We considered very simple meta-features based on statistical measures of the datasets and their attributes. Although our experiments have demonstrated that these meta-features are sufficient to create an effective similarity measure recommendation system, it would be interesting to explore other (more complex) meta-features (e.g. distance-based, landmarking, etc.) that have been proposed in the literature and extend these meta-features to the context of mixed data clustering. Future studies inspired by the works of [Cohen-Shapira 2021, Jomaa 2021] on designing and learning new meta-features that better characterize the datasets for our specific meta-learning task would also be of high interest.

5.7 Conclusion

This chapter presents SIMREC, a similarity measure recommendation system for mixed data clustering algorithms. Using meta-learning to mine the relationship between dataset characteristics and the performances of the pairs of similarity measures for different mixed data clustering algorithms and cluster validity indices, SIMREC is able to recommend suitable pairs of numerical and categorical similarity measures for new and unknown datasets. SIMREC is implemented using 120 pairs of similarity measures (10 numerical and 12 categorical), three commonly used MDC algorithms (K-Prototypes, K-Medoids, and Hierarchical Clustering), and three cluster validity indices (Silhouette, Clustering Accuracy, and ARI). Our experiments show that the pairs of similarity measures recommended by SIMREC outperform the considered baselines (including the default pairs of similarity measures used in the literature for the considered algorithms), especially for datasets that are highly impacted by the choice of the pair of similarity measures. The code

for using SIMREC is available here [Diop 2024].

We believe that SIMREC is a valuable contribution to clustering that can enhance clustering quality and efficiency when facing mixed data. However, there are several promising research directions and opportunities for further improvements, which are elaborated upon in the following chapter.

Chapter 6

Conclusion

Contents

6.1 Summary

As real-world applications increasingly involve heterogeneous data—encompassing numerical, categorical, and textual types—purely homogeneous datasets have become less common. This shift introduces new challenges in data mining, machine learning, and various industry sectors, such as healthcare and finance, where the ability to handle heterogeneous data is critical for making accurate decisions. In this thesis, we have aimed to address some of these emerging challenges in the context of mixed data clustering.

As presented in this work, mixed data clustering algorithms can be divided into two main groups: conversion-based methods, which we refer to as *homogenization methods*, and non-conversion-based methods, which we refer to as *mixed methods*. In our first contribution, we proposed an experimental framework to compare the two approaches based on how they handle mixed data (chapter 3). The results indicate that mixed methods are generally more effective than homogenization methods, as they are able to suitably handle each data type without relying on conversion (e.g., using an adapted similarity measure for each type). In contrast, homogenization methods alter the original structure of the data through conversion, leading to suboptimal performance. These results show that when facing data with heterogeneous types it is important to consider the specific properties of each type.

Next, focusing on mixed methods, and similarity-based methods in particular, we addressed the important question of the choice of the similarity measures. Similarity-based mixed methods rely on the combination of two similarity measures, one for numerical attributes and another for categorical ones, enabling them to effectively handle each data type. However, with the diversity of existing similarity

measures in the literature for each data type, which ones to use becomes a critical question. This led us to conduct several experiments to evaluate the impact of the choice of the pair of numerical and categorical similarity measures on various mixed data clustering algorithms (chapter 4). Our findings indicate that the choice of the pair of similarity measures can significantly affect clustering performances, emphasizing the importance of choosing the right similarity measures when clustering mixed datasets.

However, choosing the right similarity measures for the current dataset and algorithm can be very complex. First, we observed through our results in chapter 4 that the default pairs similarity measures used in the literature for various mixed data clustering algorithms, are not the most suitable and yield poor results on several datasets. This indicates that one cannot simply rely on the default pair of similarity measures. Unfortunately, this is often ignored in practice. Furthermore, we also noted a high variability of the top-performing pairs of similarity measures depending on the considered clustering algorithm, mixed dataset, and cluster validity index. Therefore, the selection of the right pair of similarity measures would need a deep understanding of the relationship between these different factors and the performances of the different pairs of similarity measures. To support data scientists, especially non-experts, in this complex and time-consuming task, we proposed a recommendation system named SIMREC, that can recommend suitable similarity measures according to the considered dataset, mixed data clustering algorithm, and cluster validity index (chapter 5). For each clustering algorithm and cluster validity index, SIMREC uses meta-learning to learn the relationship between dataset characteristics and the performance of the different pairs of similarity measures for the considered algorithm and validity index. This learning is possible thanks to the knowledge obtained from prior evaluations of the performances of the pairs of similarity measures on various datasets. SIMREC is currently implemented using 120 pairs of similarity measures (10 numerical and 12 categorical), three commonly used MDC algorithms (K-Prototypes, K-Medoids, and Hierarchical Clustering), and three cluster validity indices (Silhouette, Clustering Accuracy, and ARI). Our results show that the recommendations can positively help users select the most appropriate pairs of similarity measures depending on their use cases (i.e. clustering algorithm, dataset, and cluster validity index). These recommendations outperform the traditionally used similarity measures in the literature, particularly for datasets where the choice of the similarity measures has a significant impact. Furthermore, they allow to avoid time-expensive trial-and-error strategies to manually select the similarity measures.

6.2 Limitations

Although SIMREC demonstrates strong performance, it has several limitations. The primary challenge lies in the cost of the offline learning phase. While this cost does not affect the end user since the recommendation phase operates with-

out learning, it may hinder the integration of additional clustering algorithms and similarity measures into SIMREC.

Furthermore, for clustering algorithms that have additional hyper-parameters to the similarity measures, SIMREC only partially addresses the challenge of hyper-parameter selection. While it significantly reduces the search space and speeds up the selection process by recommending similarity measures, users are still required to search for the remaining hyper-parameters.

Lastly, like many meta-learning approaches, SIMREC relies on hand-crafted meta-features. We selected these meta-features for their simplicity and interpretability. However, there is no complete guarantee that these meta-features capture all the relevant information from datasets for accurate recommendations. Although they have been validated as important for the recommendation model, and a selection strategy is included in SIMREC to identify the most relevant meta-features, there may exist other meta-features that describe the datasets more effectively for the similarity measure recommendation task.

These various limitations naturally present avenues for future work which is presented in the following section.

6.3 Future Work

We have identified several directions for future research to address the limitations mentioned above and extend the work proposed in this manuscript to a larger context. We categorize these future work into short-term, mid-term and long-term, which are outlined in sections 6.3.1, 6.3.2, and 6.3.3, respectively.

6.3.1 Short-term Future Work

In the short term, our focus is on enhancing SIMREC while minimizing changes to the core methodology. One key direction involves extending SIMREC with additional mixed data clustering (MDC) algorithms, cluster validity indices (CVIs), and similarity measures. These additions will broaden its applicability to more diverse practical scenarios.

Another priority is reducing the computational cost of the learning phase, which would facilitate the integration of more MDC algorithms, CVIs, and similarity measures. A potential approach to tackle this in the short term is to use more efficient hyper-parameter optimization. Currently, we rely on grid search to determine the best combinations of similarity measures and hyper-parameters. A more efficient approach, such as Bayesian Optimization, could reduce the number of evaluations needed to find optimal or near-optimal configurations. However, this might result in a smaller meta-dataset, which could impact the accuracy of recommendations.

6.3.2 Mid-term Future Work

For mid-term future work, we are interested in improving the proposed meta-learning-based recommendation methodology. Meta-features are one of the main components of this methodology. Designing meta-features that extract relevant information for the considered meta-learning task is crucial to ensure good recommendations. As stated above, one of the limitations of our methodology relies on the use of hand-crafted meta-features. Since these meta-features are manually defined, we do not have a guarantee that these meta-features are optimal or fully relevant to our specific recommendation task. Furthermore, figure 5.7 in chapter 5 showed that different meta-learning tasks (or equivalently different meta-models) may have different important meta-features.

To address these limitations, a key direction for mid-term research is the development of *task-driven* meta-features. By task-driven, we mean meta-features that are defined to optimize the recommendation performances for a specific recommendation task. Pioneering work in this area has been conducted in studies such as [Cohen-Shapira 2021, Jomaa 2021], where a novel paradigm is introduced for defining meta-features. Instead of using hand-crafted meta-features, the meta-features are learned directly from datasets using neural networks to ensure optimal performance on the recommendation task. Further research in this area holds great promise and could have a positive impact on meta-learning applications.

6.3.3 Long-term works

In the long term, we have identified several strategic directions for further research. One promising avenue involves enhancing support for data scientists during the clustering process. Clustering involves numerous steps - such as data preprocessing, feature selection, algorithm selection, hyperparameter tuning, and clustering evaluation - all of which can drastically influence the quality of the clustering results. Given the growing complexity of these components, developing a unified framework that assists data scientists throughout the entire clustering pipeline would be highly valuable. Such a framework could help both non-expert and expert users by recommending appropriate preprocessing methods, clustering algorithms, and hyperparameters (including similarity measures) tailored to their data. Ultimately, this would aid in tasks like exploratory data analysis and data comprehension, helping non-experts make informed decisions while boosting productivity for experts. Similar work has been proposed by Parmezan *et al.* [Parmezan 2021] and Garouani [Garouani 2022] in the context of supervised learning using meta-learning (for the automatic recommendation of feature selection algorithms and of machine-learning pipelines), and can inspire future work in this direction.

Another important direction for future work relies on how to better define similarity when performing clustering on mixed datasets. We have shown that when facing mixed data, it is important to consider the heterogeneity of numerical and categorical attribute types, by using an adapted similarity measure for each data

type. However, even within a single data type, attributes may have heterogeneous properties requiring distinct similarity measures. Therefore important directions for future work are to investigate this attribute level heterogeneity and study how to choose an appropriate similarity for each attribute instead of using the same similarity measures for all attributes of the same type.

Bibliography

[Abdullin 2012] Abdullin, A. and Nasraoui, O. *Clustering Heterogeneous Data Sets*. pages 1–8, October 2012. (Cited in pages 3 and 48.)

[Abu Alfeilat 2019] Abu Alfeilat, H. A., Hassanat, A. B. A., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S. and Prasath, V. B. S. *Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review*. Big data, vol. 7, no. 4, pages 221–248, December 2019. [Online]. Available: `https://doi.org/10.1089/big.2018.0175`. (Cited in pages 4, 7, 14, 15, 17, and 74.)

[Ahmad 2007a] Ahmad, A. and Dey, L. *A k-mean clustering algorithm for mixed numeric and categorical data*. Data & Knowledge Engineering, vol. 63, no. 2, pages 503–527, November 2007. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0169023X0700050X`. (Cited in pages 33 and 37.)

[Ahmad 2007b] Ahmad, A. and Dey, L. *A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set*. Pattern Recognition Letters, vol. 28, no. 1, pages 110–118, January 2007. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0167865506001759`. (Cited in pages 19, 33, and 80.)

[Ahmad 2019] Ahmad, A. and Khan, S. S. *Survey of State-of-the-Art Mixed Data Clustering Algorithms*. IEEE Access, vol. 7, pages 31883–31902, 2019. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2019.2903568. Conference Name: IEEE Access. (Cited in pages 3, 4, 22, 48, 53, and 84.)

[Alkhasov 2015] Alkhasov, S. S., Tselykh, A. N. and Tselykh, A. A. *Application of cluster analysis for the assessment of the share of fraud victims among bank card holders*. In Proceedings of the 8th International Conference on Security of Information and Networks, SIN '15, pages 103–106, New York, NY, USA, September 2015. Association for Computing Machinery. (Cited in page 2.)

[Alves 2019] Alves, G., Couceiro, M. and Napoli, A. Similarity Measure Selection for Categorical Data Clustering. December 2019. (Cited in pages 4, 7, 41, 43, and 44.)

[Ambroise 1998] Ambroise, C. *The EM Algorithm and Extensions, by G.M. McLachlan and T. Krishnan*. Journal of Classification, vol. 15, no. 1, pages 154–156, January 1998. [Online]. Available: `https://doi.org/10.1007/s003579900027`. (Cited in page 12.)

[Ankerst 1999] Ankerst, M., Breunig, M. M., Kriegel, H.-P. and Sander, J. *OPTICS: ordering points to identify the clustering structure.* SIGMOD Rec., vol. 28, no. 2, pages 49–60, June 1999. [Online]. Available: `https://dl.acm.org/doi/10.1145/304181.304187`. (Cited in page 26.)

[Balaji 2020] Balaji, K., Lavanya, K. and Mary, A. G. *Clustering of mixed datasets using deep learning algorithm.* Chemometrics and Intelligent Laboratory Systems, vol. 204, page 104123, September 2020. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0169743920303695`. (Cited in pages 32 and 34.)

[Barcelo-Rico 2012] Barcelo-Rico, F. and Diez, J.-L. *Geometrical codification for clustering mixed categorical and numerical databases.* Journal of Intelligent Information Systems, vol. 39, no. 1, pages 167–185, August 2012. [Online]. Available: `https://doi.org/10.1007/s10844-011-0187-y`. (Cited in pages 3, 4, 29, 30, 31, 48, and 51.)

[Behzadi 2020] Behzadi, S., Müller, N. S., Plant, C. and Böhm, C. *Clustering of mixed-type data considering concept hierarchies: problem specification and algorithm.* International Journal of Data Science and Analytics, vol. 10, no. 3, pages 233–248, September 2020. [Online]. Available: `https://doi.org/10.1007/s41060-020-00216-2`. (Cited in page 4.)

[Bergstra 2012] Bergstra, J. and Bengio, Y. *Random Search for Hyper-Parameter Optimization.* Journal of Machine Learning Research, vol. 13, no. 10, pages 281–305, 2012. [Online]. Available: `http://jmlr.org/papers/v13/bergstra12a.html`. (Cited in page 39.)

[Bhattacharjee 2020] Bhattacharjee, P. and Mitra, P. *A survey of density based clustering algorithms.* Frontiers of Computer Science, vol. 15, no. 1, page 151308, September 2020. [Online]. Available: `https://doi.org/10.1007/s11704-019-9059-3`. (Cited in page 26.)

[Bishnoi 2020] Bishnoi, S. and Hooda, B. K. *A survey of distance measures for mixed variables.* International Journal of Chemical Studies, vol. 8, pages 338–343, July 2020. [Online]. Available: http://dx.doi.org/10.22271/chemi.2020.v8.i4f.10087. (Cited in page 33.)

[Bora 2014] Bora, M. D. J. and Gupta, D. A. K. *Effect of Different Distance Measures on the Performance of K-Means Algorithm: An Experimental Study in Matlab.* May 2014. [Online]. Available: `http://arxiv.org/abs/1405.7471`. arXiv:1405.7471 [cs]. (Cited in page 15.)

[Boriah 2008] Boriah, S., Chandola, V. and Kumar, V. *Similarity Measures for Categorical Data: A Comparative Evaluation.* In Proceedings of the 2008 SIAM International Conference on Data Mining (SDM), Proceedings, pages

243–254. Society for Industrial and Applied Mathematics, April 2008. (Cited in pages 4, 7, 12, 18, and 74.)

[Budiaji 2019] Budiaji, W. and Leisch, F. *Simple K-Medoids Partitioning Algorithm for Mixed Variable Data.* Algorithms, vol. 12, no. 9, page 177, September 2019. [Online]. Available: `https://www.mdpi.com/1999-4893/12/9/177`. Number: 9 Publisher: Multidisciplinary Digital Publishing Institute. (Cited in pages 33, 37, 65, 83, and 86.)

[Cai 2016] Cai, F., Le-Khac, N.-A. and Kechadi, T. *Clustering Approaches for Financial Data Analysis: a Survey.* September 2016. [Online]. Available: `http://arxiv.org/abs/1609.08520`. arXiv:1609.08520 [q-fin]. (Cited in page 2.)

[Campello 2013] Campello, R. J. G. B., Moulavi, D. and Sander, J. *Density-Based Clustering Based on Hierarchical Density Estimates.* In Pei, J., Tseng, V. S., Cao, L., Motoda, H. and Xu, G., editors, Advances in Knowledge Discovery and Data Mining, pages 160–172, Berlin, Heidelberg, 2013. Springer. (Cited in page 26.)

[Cao 2013] Cao, L. *Non-IIDness Learning in Behavioral and Social Data.* The Computer Journal, vol. 57, pages 1358–1370, August 2013. [Online]. Available: http://dx.doi.org/10.1093/comjnl/bxt084. (Cited in page 3.)

[Caruso 2021] Caruso, G., Gattone, S. A., Fortuna, F. and Di Battista, T. *Cluster Analysis for mixed data: An application to credit risk evaluation.* Socio-Economic Planning Sciences, vol. 73, page 100850, February 2021. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0038012119305440`. (Cited in page 3.)

[Chakraborty 2014] Chakraborty, S., Nagwani, N. K. and Dey, L. *Weather Forecasting using Incremental K-means Clustering.* June 2014. [Online]. Available: `https://arxiv.org/abs/1406.4756v1`. (Cited in page 3.)

[Cheeseman 1997] Cheeseman, P. and Stutz, J. *Bayesian Classification(AutoClass):Theory and Results.* Advances in Knowledge Discovery and Data Mining, May 1997. (Cited in pages 33 and 36.)

[Chifu 2015] Chifu, A.-G., Hristea, F., Mothe, J. and Popescu, M. *Word sense discrimination in information retrieval: A spectral clustering-based approach.* Information Processing & Management, vol. 51, no. 2, pages 16–31, March 2015. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0306457314001046`. (Cited in page 3.)

[Choi 2009] Choi, S., Cha, S.-H. and Tappert, C. *A Survey of Binary Similarity and Distance Measures.* J. Syst. Cybern. Inf., vol. 8, November 2009. (Cited in pages 7, 21, and 74.)

[Cohen-Shapira 2021] Cohen-Shapira, N. and Rokach, L. *Automatic selection of clustering algorithms using supervised graph embedding.* Information Sciences, vol. 577, pages 824–851, October 2021. [Online]. Available: `http://arxiv.org/abs/2011.08225`. arXiv:2011.08225 [cs, stat]. (Cited in pages 42, 44, 99, and 104.)

[Cover 1999] Cover, T. M. Elements of information theory. John Wiley & Sons, 1999. (Cited in page 80.)

[Dardac 2009] Dardac, N. and Boitan, I. A. *Cluster analysis approach for banks' risk profile : the Romanian evidence.* 2009. [Online]. Available: `https://www.um.edu.mt/library/oar/handle/123456789/31853`. Accepted: 2018-07-17T10:28:50Z Publisher: University of Piraeus. International Strategic Management Association. (Cited in page 2.)

[de Souto 2008] de Souto, M., Prudêncio, R., Soares, R., Araujo, D., Costa, I., Ludermir, T. and Schliep, A. *Ranking and Selecting Clustering Algorithms Using a Meta-Learning Approach.* pages 3729–3735, June 2008. (Cited in pages 41, 44, and 78.)

[de Sá 2017] de Sá, C. R., Soares, C., Knobbe, A. and Cortez, P. *Label Ranking Forests.* Expert Systems, vol. 34, no. 1, page e12166, 2017. [Online]. Available: `https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.12166`. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/exsy.12166. (Cited in page 80.)

[Deza 2013] Deza, M. M. and Deza, E. Encyclopedia of Distances. Springer, Berlin, Heidelberg, 2013. (Cited in pages 4, 12, 14, and 16.)

[Ding 2017] Ding, S., Du, M., Sun, T., Xu, X. and Xue, Y. *An entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood.* Knowledge-Based Systems, vol. 133, pages 294–313, October 2017. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0950705117303490`. (Cited in pages 8, 33, 37, and 48.)

[Diop 2024] Diop, A. *SImilarity Measure RECommendation system for mixed data clustering (SIMREC).* 2024. [Online]. Available: `https://github.com/AbdoulayeDiop/simrec`. Accessed: 2024-06-07. (Cited in pages 9 and 100.)

[Dokeroglu 2019] Dokeroglu, T., Sevinc, E., Kucukyilmaz, T. and Cosar, A. *A survey on new generation metaheuristic algorithms.* Computers & Industrial Engineering, vol. 137, page 106040, November 2019. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0360835219304991`. (Cited in page 40.)

[Du 2017] Du, M., Ding, S. and Xue, Y. *A novel density peaks clustering algorithm for mixed data.* Pattern Recognition Letters, vol. 97, pages 46–53, October

2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865517302337. (Cited in pages 33 and 37.)

[D'Urso 2019] D'Urso, P. and Massari, R. *Fuzzy clustering of mixed data.* Information Sciences, vol. 505, pages 513–534, December 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025519307194. (Cited in pages 33 and 37.)

[El Malki 2020] El Malki, N., Cugny, R., Teste, O. and Ravat, F. *DECWA: Density-Based Clustering using Wasserstein Distance.* In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM '20, pages 2005–2008, New York, NY, USA, October 2020. Association for Computing Machinery. (Cited in page 24.)

[ElShawi 2022] ElShawi, R. and Sakr, S. *TPE-AutoClust: A Tree-based Pipline Ensemble Framework for Automated Clustering.* In 2022 IEEE International Conference on Data Mining Workshops (ICDMW), pages 1144–1153, November 2022. ISSN: 2375-9259. (Cited in pages 42 and 44.)

[Eskin 2002] Eskin, E., Arnold, A., Prerau, M., Portnoy, L. and Stolfo, S. *A Geometric Framework for Unsupervised Anomaly Detection.* In Barbará, D. and Jajodia, S., editors, Applications of Data Mining in Computer Security, pages 77–101. Springer US, Boston, MA, 2002. (Cited in page 18.)

[Ester 1996] Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. *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, Portland, Oregon, August 1996. AAAI Press. (Cited in pages 26 and 52.)

[Ezugwu 2022] Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I. and Akinyelu, A. A. *A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects.* Engineering Applications of Artificial Intelligence, vol. 110, page 104743, April 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095219762200046X. (Cited in pages 2, 24, 25, 26, 27, 28, and 36.)

[Ferrari 2015] Ferrari, D. G. and de Castro, L. N. *Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods.* Information Sciences, vol. 301, pages 181–194, April 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025514011967. (Cited in pages 42, 44, and 78.)

[Feurer 2014] Feurer, M., Springenberg, J. T. and Hutter, F. *Using meta-learning to initialize bayesian optimization of hyperparameters.* In Proceedings of the 2014 International Conference on Meta-learning and Algorithm Selection -

Volume 1201, MLAS'14, pages 3–10, Aachen, DEU, September 2014. CEUR-WS.org. (Cited in pages 41 and 42.)

[Foss 2016] Foss, A., Markatou, M., Ray, B. and Heching, A. *A semiparametric method for clustering mixed data.* Machine Learning, vol. 105, no. 3, pages 419–458, December 2016. [Online]. Available: https://doi.org/10.1007/s10994-016-5575-7. (Cited in page 36.)

[Gabbay 2021] Gabbay, I., Shapira, B. and Rokach, L. *Isolation forests and landmarking-based representations for clustering algorithm recommendation using meta-learning.* Information Sciences, vol. 574, pages 473–489, October 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025521006241. (Cited in pages 42 and 44.)

[Gandomi 2013] Gandomi, A., Yang, X.-S., Talatahari, S. and Alavi, A. *Metaheuristic Algorithms in Modeling and Optimization.* In Metaheuristic Applications in Structures and Infrastructures, pages 1–24. December 2013. Journal Abbreviation: Metaheuristic Applications in Structures and Infrastructures. (Cited in page 40.)

[Garouani 2022] Garouani, M. *Towards Efficient and Explainable Automated Machine Learning Pipelines Design : Application to Industry 4.0 Data.* phdthesis, Université du Littoral Côte d'Opale ; Université Hassan II (Casablanca, Maroc), September 2022. (Cited in pages 7, 41, and 104.)

[Gormley 2023] Gormley, I. C., Murphy, T. B. and Raftery, A. E. *Model-Based Clustering.* Annual Review of Statistics and Its Application, vol. 10, no. Volume 10, 2023, pages 573–595, March 2023. [Online]. Available: https://www.annualreviews.org/content/journals/10.1146/annurev-statistics-033121-115326. Publisher: Annual Reviews. (Cited in pages 26, 27, 28, and 36.)

[Halawani 2012] Halawani, S. M., Alhaddad, M. and Ahmad, A. *A study of digital mammograms by using clustering algorithms.* JSIR Vol.71(09) [September 2012], September 2012. [Online]. Available: http://nopr.niscair.res.in/handle/123456789/14628. Accepted: 2012-08-31T09:32:15Z Publisher: NISCAIR-CSIR, India. (Cited in page 3.)

[Harikumar 2015] Harikumar, S. and Pv, S. *K-Medoid Clustering for Heterogeneous DataSets.* Procedia Computer Science, vol. 70, pages 226–237, January 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187705091503241X. (Cited in pages 8, 33, 37, 48, and 54.)

[Hartuv 2000] Hartuv, E. and Shamir, R. *A clustering algorithm based on graph connectivity.* Information Processing Letters, vol. 76, no. 4, pages 175–181, December 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020019000001423. (Cited in page 28.)

[He 2005] He, Z., Xu, X. and Deng, S. *Clustering Mixed Numeric and Categorical Data: A Cluster Ensemble Approach.* arXiv:cs/0509011, September 2005. [Online]. Available: `http://arxiv.org/abs/cs/0509011`. arXiv: cs/0509011. (Cited in pages 32 and 35.)

[Hennig 2013] Hennig, C. and Liao, T. F. *How to Find an Appropriate Clustering for Mixed-Type Variables with Application to Socio-Economic Stratification.* Journal of the Royal Statistical Society Series C: Applied Statistics, vol. 62, no. 3, pages 309–369, May 2013. [Online]. Available: `https://doi.org/10.1111/j.1467-9876.2012.01066.x`. (Cited in page 3.)

[Hsu 2007] Hsu, C.-C., Chen, C.-L. and Su, Y.-W. *Hierarchical clustering of mixed data based on distance hierarchy.* Information Sciences, vol. 177, no. 20, pages 4474–4492, October 2007. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0020025507002319`. (Cited in pages 33 and 37.)

[Huang 1998] Huang, Z. *Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values.* Data Mining and Knowledge Discovery, vol. 2, no. 3, pages 283–304, September 1998. [Online]. Available: `https://doi.org/10.1023/A:1009769707641`. (Cited in pages 4, 5, 8, 33, 37, 48, 52, 63, 65, 83, and 86.)

[Huang 2005] Huang, J., Ng, M., Rong, H. and Li, Z. *Automated variable weighting in k-means type clustering.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 5, pages 657–668, May 2005. [Online]. Available: `https://ieeexplore.ieee.org/abstract/document/1407871`. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. (Cited in pages 33 and 37.)

[Hubert 1985] Hubert, L. and Arabie, P. *Comparing partitions.* Journal of Classification, vol. 2, no. 1, pages 193–218, December 1985. [Online]. Available: `https://doi.org/10.1007/BF01908075`. (Cited in pages 54 and 84.)

[Irani 2016] Irani, J., Pise, N. and Phatak, M. *Clustering Techniques and the Similarity Measures used in Clustering: A Survey.* International Journal of Computer Applications, vol. 134, pages 9–14, January 2016. [Online]. Available: http://dx.doi.org/10.5120/ijca2016907841. (Cited in page 12.)

[Jamil 2021] Jamil, M. S. J., Naim, F. A., Ahamed, B. and Huda, M. N. *Customer Review Analysis by Hybrid Unsupervised Learning Applying Weight on Priority Data.* In Uddin, M. S. and Bansal, J. C., editors, Proceedings of International Joint Conference on Advances in Computational Intelligence, pages 333–342, Singapore, 2021. Springer. (Cited in page 3.)

[Jardim 2022] Jardim, S. and Mora, C. *Customer reviews sentiment-based analysis and clustering for market-oriented tourism services and products development or positioning.* Procedia Computer Science, vol. 196, pages 199–206,

January 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921022298. (Cited in page 3.)

[Ji 2013] Ji, J., Bai, T., Zhou, C., Ma, C. and Wang, Z. *An improved k-prototypes clustering algorithm for mixed numeric and categorical data.* Neurocomputing, vol. 120, pages 590–596, November 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231213004773. (Cited in pages 4, 8, 29, 33, and 37.)

[Jian 2018] Jian, S., Hu, L., Cao, L. and Lu, K. *Metric-Based Auto-Instructor for Learning Mixed Data Representation.* Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, April 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11597. Number: 1. (Cited in page 32.)

[Jomaa 2021] Jomaa, H. S., Schmidt-Thieme, L. and Grabocka, J. *Dataset2Vec: Learning Dataset Meta-Features.* January 2021. [Online]. Available: http://arxiv.org/abs/1905.11063. arXiv:1905.11063 [cs, stat]. (Cited in pages 99 and 104.)

[Kabir 2011] Kabir, M. M., Shahjahan, M. and Murase, K. *A new local search based hybrid genetic algorithm for feature selection.* Neurocomputing, vol. 74, no. 17, pages 2914–2928, October 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231211002748. (Cited in page 81.)

[Kansal 2018] Kansal, T., Bahuguna, S., Singh, V. and Choudhury, T. *Customer Segmentation using K-means Clustering.* In 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), pages 135–139, December 2018. (Cited in page 3.)

[Kassi 2015] Kassi, M. L., Berrado, A., Benabbou, L. and Benabdelkader, K. *Towards a new framework for clustering in a mixed data space: Case of gasoline service stations segmentation in Morocco.* In 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), pages 1–6, November 2015. ISSN: 2161-5330. (Cited in page 3.)

[Kim 2020] Kim, Y., Do, H. and Kim, S. B. *Outer-Points shaver: Robust graph-based clustering via node cutting.* Pattern Recognition, vol. 97, page 107001, January 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320319303048. (Cited in page 28.)

[Kriegel 2011] Kriegel, H.-P., Kröger, P., Sander, J. and Zimek, A. *Density-based clustering.* WIREs Data Mining and Knowledge Discovery, vol. 1, no. 3, pages 231–240, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.30.

_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.30. (Cited in page 26.)

[Kumar 2014] Kumar, V., Chhabra, J. K. and Kumar, D. *Performance Evaluation of Distance Metrics in the Clustering Algorithms*. INFOCOMP Journal of Computer Science, vol. 13, no. 1, pages 38–52, September 2014. [Online]. Available: `https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/21`. Number: 1. (Cited in pages 14, 15, 16, and 17.)

[Lee 2023] Lee, Y., Park, C. and Kang, S. *Deep Embedded Clustering Framework for Mixed Data*. IEEE Access, vol. 11, pages 33–40, 2023. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2022.3232372. Conference Name: IEEE Access. (Cited in pages 32 and 35.)

[Li 2015] Li, L., Das, S., John Hansman, R., Palacios, R. and Srivastava, A. N. *Analysis of Flight Data Using Clustering Techniques for Detecting Abnormal Operations*. Journal of Aerospace Information Systems, vol. 12, no. 9, pages 587–598, 2015. [Online]. Available: `https://doi.org/10.2514/1.I010329`. Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.I010329. (Cited in page 3.)

[Liu 2002] Liu, H., Hussain, F., Tan, C. L. and Dash, M. *Discretization: An Enabling Technique*. Data Mining and Knowledge Discovery, vol. 6, no. 4, pages 393–423, October 2002. [Online]. Available: `https://doi.org/10.1023/A:1016304305535`. (Cited in pages 30 and 31.)

[Liu 2005] Liu, H. and Yu, L. *Toward integrating feature selection algorithms for classification and clustering*. IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 4, pages 491–502, April 2005. [Online]. Available: `https://ieeexplore.ieee.org/abstract/document/1401889`. Conference Name: IEEE Transactions on Knowledge and Data Engineering. (Cited in page 81.)

[Liu 2020] Liu, Z. and Barahona, M. *Graph-based data clustering via multiscale community detection*. Applied Network Science, vol. 5, no. 1, pages 1–20, December 2020. [Online]. Available: `https://appliednetsci.springeropen.com/articles/10.1007/s41109-019-0248-7`. Number: 1 Publisher: SpringerOpen. (Cited in page 28.)

[Lydia 2018] Lydia, L., Govindasamy, P., Lakshmanaprabu, S. and Ramya, D. *Document Clustering Based On Text Mining K-Means Algorithm Using Euclidean Distance Similarity*. Journal of Advanced Research in Dynamical and Control Systems, vol. 10, April 2018. (Cited in page 3.)

[Macqueen 1967] Macqueen, J. *Some methods for classification and analysis of multivariate observations*. In Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press, 1967. (Cited in pages 22 and 52.)

[Magoev 2018] Magoev, K., Krzhizhanovskaya, V. V. and Kovalchuk, S. V. *Application of clustering methods for detecting critical acute coronary syndrome patients*. Procedia Computer Science, vol. 136, pages 370–379, January 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918315837. (Cited in page 2.)

[Mahalanobis 2018] Mahalanobis, P. C. *On the Generalized Distance in Statistics*. Sankhyā: The Indian Journal of Statistics, Series A (2008-), vol. 80, pages S1–S7, 2018. [Online]. Available: https://www.jstor.org/stable/48723335. Publisher: [Springer, Indian Statistical Institute]. (Cited in page 15.)

[Malki 2021] Malki, N. E. *New partition-based and density-based approaches for improving clustering*. phdthesis, Université Toulouse le Mirail - Toulouse II, January 2021. (Cited in page 24.)

[Mangortey 2020] Mangortey, E., Monteiro, D., Ackley, J., Gao, Z., Puranik, T., Kirby, M., Pinon Fischer, O. and Mavris, D. Application of Machine Learning Techniques to Parameter Selection for Flight Risk Identification. January 2020. (Cited in page 3.)

[Mbuga 2022] Mbuga, F. and Tortora, C. *Spectral Clustering of Mixed-Type Data*. Stats, vol. 5, no. 1, pages 1–11, March 2022. [Online]. Available: https://www.mdpi.com/2571-905X/5/1/1. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute. (Cited in pages 33, 37, 53, 63, and 65.)

[McParland 2016] McParland, D. and Gormley, I. C. *Model based clustering for mixed data: clustMD*. Advances in Data Analysis and Classification, vol. 10, no. 2, pages 155–169, June 2016. [Online]. Available: https://doi.org/10.1007/s11634-016-0238-x. (Cited in page 33.)

[McParland 2017] McParland, D., Phillips, C. M., Brennan, L., Roche, H. M. and Gormley, I. C. *Clustering high-dimensional mixed data to uncover sub-phenotypes: joint analysis of phenotypic and genotypic data*. Statistics in Medicine, vol. 36, no. 28, pages 4548–4569, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.7371. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sim.7371. (Cited in page 3.)

[Moustaki 2005] Moustaki, I. and Papageorgiou, I. *Latent class models for mixed variables with applications in Archaeometry*. Computational Statistics & Data Analysis, vol. 48, no. 3, pages 659–675, March 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167947304000374. (Cited in pages 33 and 36.)

[Murtagh 2017] Murtagh, F. and Contreras, P. *Algorithms for hierarchical clustering: an overview, II*. WIREs Data Mining and Knowledge Discovery, vol. 7, no. 6, page e1219, 2017. [Online]. Available: http

s://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1219. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1219. (Cited in page 24.)

[Newcomer 2011] Newcomer, S. R., Steiner, J. F. and Bayliss, E. A. *Identifying subgroups of complex patients with cluster analysis.* The American journal of managed care, vol. 17, no. 8, pages e324–32, August 2011. (Cited in page 2.)

[Ng 2001] Ng, A., Jordan, M. and Weiss, Y. *On Spectral Clustering: Analysis and an algorithm.* In Advances in Neural Information Processing Systems, volume 14. MIT Press, 2001. (Cited in page 52.)

[Niu 2015] Niu, K., Niu, Z., Su, Y., Wang, C., Lu, H. and Guan, J. *A Coupled User Clustering Algorithm Based on Mixed Data for Web-Based Learning Systems.* Mathematical Problems in Engineering, vol. 2015, page e747628, August 2015. [Online]. Available: https://www.hindawi.com/journals/mpe/2015/747628/. Publisher: Hindawi. (Cited in page 3.)

[Park 2009] Park, H.-S. and Jun, C.-H. *A simple and fast algorithm for K-medoids clustering.* Expert Systems with Applications, vol. 36, no. 2, Part 2, pages 3336–3341, March 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741740800081X. (Cited in pages 23, 52, and 63.)

[Parmezan 2021] Parmezan, A. R. S., Lee, H. D., Spolaôr, N. and Wu, F. C. *Automatic recommendation of feature selection algorithms based on dataset characteristics.* Expert Systems with Applications, vol. 185, page 115589, December 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421009908. (Cited in page 104.)

[Peng 2013] Peng, X., Zhou, C., Hepburn, D. M., Judd, M. D. and Siew, W. H. *Application of K-Means method to pattern recognition in on-line cable partial discharge monitoring.* IEEE Transactions on Dielectrics and Electrical Insulation, vol. 20, no. 3, pages 754–761, June 2013. [Online]. Available: https://ieeexplore.ieee.org/document/6518945. Conference Name: IEEE Transactions on Dielectrics and Electrical Insulation. (Cited in page 3.)

[Philip 1983] Philip, G. and Ottaway, B. S. *Mixed Data Cluster Analysis: An Illustration Using Cypriot Hooked-Tang Weapons.* Archaeometry, vol. 25, no. 2, pages 119–133, 1983. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1475-4754.1983.tb00671.x. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1475-4754.1983.tb00671.x. (Cited in pages 33, 37, 53, 65, 83, and 86.)

[Pimentel 2019] Pimentel, B. A. and de Carvalho, A. C. P. L. F. *A new data characterization for selecting clustering algorithms using meta-learning.* Information Sciences, vol. 477, pages 203–219, March 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025518308624. (Cited in pages 42, 44, and 78.)

[Pimentel 2020] Pimentel, B. A. and de Carvalho, A. C. P. L. F. *A Meta-learning approach for recommending the number of clusters for clustering algorithms.* Knowledge-Based Systems, vol. 195, page 105682, May 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705120301209. (Cited in pages 41, 43, 44, and 78.)

[Poulakis 2020] Poulakis, Y., Doulkeridis, C. and Kyriazis, D. *AutoClust: A Framework for Automated Clustering Based on Cluster Validity Indices.* In 2020 IEEE International Conference on Data Mining (ICDM), pages 1220–1225, November 2020. ISSN: 2374-8486. (Cited in pages 42 and 44.)

[Poulakis 2024] Poulakis, Y., Doulkeridis, C. and Kyriazis, D. *A Survey on AutoML Methods and Systems for Clustering.* ACM Transactions on Knowledge Discovery from Data, vol. 18, no. 5, pages 120:1–120:30, February 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/3643564. (Cited in pages 7, 39, 41, and 42.)

[Raghu Kisore 2017] Raghu Kisore, N. and B Koteswaraiah, C. *Improving ATM coverage area using density based clustering algorithm and voronoi diagrams.* Information Sciences, vol. 376, pages 1–20, January 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025516310878. (Cited in page 2.)

[Rahman 2024] Rahman, J. S. U., Hussain, S. M., Anjum, F., Naz, T. and Sathish, K. S. *Brain image segmentation using K mean segmentation and fuzzy C-means (FCM) algorithm to improve efficiency of tumor detection.* AIP Conference Proceedings, vol. 3161, no. 1, page 020155, August 2024. [Online]. Available: https://doi.org/10.1063/5.0229431. (Cited in page 2.)

[Ralambondrainy 1995] Ralambondrainy, H. *A conceptual version of the K-means algorithm.* Pattern Recognition Letters, vol. 16, no. 11, pages 1147–1157, November 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/016786559500075R. (Cited in page 31.)

[Reddy 2014] Reddy, C. K. and Vinzamuri, B. *A Survey of Partitional and Hierarchical Clustering Algorithms.* In Data Clustering. Chapman and Hall/CRC, 2014. Num Pages: 24. (Cited in pages 24, 25, and 52.)

[Rousseeuw 1987] Rousseeuw, P. J. *Silhouettes: A graphical aid to the interpretation and validation of cluster analysis.* Journal of Computational and

Applied Mathematics, vol. 20, pages 53–65, November 1987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377042787901257. (Cited in page 84.)

[Schubert 2021] Schubert, E. and Rousseeuw, P. J. *Fast and eager k-medoids clustering: O(k) runtime improvement of the PAM, CLARA, and CLARANS algorithms.* Information Systems, vol. 101, page 101804, November 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306437921000557. (Cited in pages 23, 52, 63, and 83.)

[Sharan 2000] Sharan, R. and Shamir, R. *CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis.* Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology, vol. 8, pages 307–16, February 2000. (Cited in page 28.)

[Shirkhorshidi 2015] Shirkhorshidi, A. S., Aghabozorgi, S. and Wah, T. Y. *A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data.* PLOS ONE, vol. 10, no. 12, page e0144059, December 2015. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0144059. Publisher: Public Library of Science. (Cited in page 4.)

[Snoek 2012] Snoek, J., Larochelle, H. and Adams, R. P. *Practical Bayesian Optimization of Machine Learning Algorithms.* August 2012. [Online]. Available: http://arxiv.org/abs/1206.2944. arXiv:1206.2944 [cs, stat]. (Cited in page 39.)

[Spencer 2013] Spencer, N. H. Essentials of Multivariate Data Analysis. CRC Press, December 2013. Google-Books-ID: EG3SBQAAQBAJ. (Cited in page 15.)

[Subramaniyan 2020] Subramaniyan, M., Skoogh, A., Muhammad, A. S., Bokrantz, J., Johansson, B. and Roser, C. *A generic hierarchical clustering approach for detecting bottlenecks in manufacturing.* Journal of Manufacturing Systems, vol. 55, pages 143–158, April 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612520300315. (Cited in page 3.)

[Suguna 2012] Suguna, J. and Selvi, M. *Ensemble Fuzzy Clustering for Mixed Numeric and Categorical Data.* International Journal of Computer Applications, vol. 42, pages 19–23, March 2012. [Online]. Available: http://dx.doi.org/10.5120/5672-7705. (Cited in pages 32 and 35.)

[Tabianan 2022] Tabianan, K., Velu, S. and Ravi, V. *K-Means Clustering Approach for Intelligent Customer Segmentation Using Customer Purchase Behavior Data.* Sustainability, vol. 14, no. 12, page 7243, January 2022. [Online].

Available: https://www.mdpi.com/2071-1050/14/12/7243. Number: 12 Publisher: Multidisciplinary Digital Publishing Institute. (Cited in page 3.)

[van de Velden 2019] van de Velden, M., Iodice D'Enza, A. and Markos, A. *Distance-based clustering of mixed data.* WIREs Computational Statistics, vol. 11, no. 3, page e1456, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/wics.1456. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wics.1456. (Cited in pages 4, 31, 32, and 48.)

[Vanschoren 2014] Vanschoren, J., van Rijn, J. N., Bischl, B. and Torgo, L. *OpenML: networked science in machine learning.* ACM SIGKDD Explorations Newsletter, vol. 15, no. 2, pages 49–60, June 2014. [Online]. Available: http://arxiv.org/abs/1407.7722. arXiv:1407.7722 [cs]. (Cited in page 86.)

[Vanschoren 2018] Vanschoren, J. *Meta-Learning: A Survey.* October 2018. [Online]. Available: http://arxiv.org/abs/1810.03548. arXiv:1810.03548 [cs, stat]. (Cited in pages 41 and 75.)

[Vialetto 2020] Vialetto, G. and Noro, M. *An innovative approach to design cogeneration systems based on big data analysis and use of clustering methods.* Energy Conversion and Management, vol. 214, page 112901, June 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0196890420304398. (Cited in page 3.)

[Vukicevic 2016] Vukicevic, M., Radovanovic, S., Delibasic, B. and Suknovic, M. *Extending meta-learning framework for clustering gene expression data with component-based algorithm design and internal evaluation measures.* International Journal of Data Mining and Bioinformatics, vol. 14, no. 2, pages 101–119, January 2016. [Online]. Available: https://www.inderscienceonline.com/doi/abs/10.1504/IJDMB.2016.074682. Publisher: Inderscience Publishers. (Cited in pages 42, 44, and 78.)

[Waheed 2015] Waheed, A., Waheed, Z., Akram, M. U. and Shaukat, A. *Removal of False Blood Vessels Using Shape Based Features and Image Inpainting.* Journal of Sensors, vol. 2015, no. 1, page 839894, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2015/839894. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2015/839894. (Cited in page 2.)

[Wei 2015] Wei, M., Chow, T. W. S. and Chan, R. H. M. *Clustering Heterogeneous Data with k-Means by Mutual Information-Based Unsupervised Feature Transformation.* Entropy, vol. 17, no. 3, pages 1535–1548, March 2015. [Online]. Available: https://www.mdpi.com/1099-4300/17/3/1535. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute. (Cited in pages 4, 29, 34, and 48.)

[Wilcoxon 1945] Wilcoxon, F. *Individual Comparisons by Ranking Methods.* Biometrics Bulletin, vol. 1, no. 6, pages 80–83, 1945. [Online]. Available: `https://www.jstor.org/stable/3001968`. Publisher: [International Biometric Society, Wiley]. (Cited in pages 56 and 90.)

[Xu 2005] Xu, R. and Wunsch, D. *Survey of Clustering Algorithms.* Neural Networks, IEEE Transactions on, vol. 16, pages 645–678, June 2005. [Online]. Available: http://dx.doi.org/10.1109/TNN.2005.845141. (Cited in pages 27 and 28.)

[Xu 2015] Xu, D. and Tian, Y. *A Comprehensive Survey of Clustering Algorithms.* Annals of Data Science, vol. 2, no. 2, pages 165–193, June 2015. [Online]. Available: `https://doi.org/10.1007/s40745-015-0040-1`. (Cited in pages 26 and 27.)

[Yang 2020] Yang, L. and Shami, A. *On hyperparameter optimization of machine learning algorithms: Theory and practice.* Neurocomputing, vol. 415, pages 295–316, November 2020. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0925231220311693`. (Cited in pages 38, 39, and 40.)

[Yu 2020] Yu, T. and Zhu, H. *Hyper-Parameter Optimization: A Review of Algorithms and Applications.* March 2020. [Online]. Available: `http://arxiv.org/abs/2003.05689`. arXiv:2003.05689 [cs, stat]. (Cited in page 38.)

[Zahn 1971] Zahn, C. *Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters.* IEEE Transactions on Computers, vol. C-20, no. 1, pages 68–86, January 1971. [Online]. Available: `https://ieeexplore.ieee.org/document/1671676`. Conference Name: IEEE Transactions on Computers. (Cited in page 28.)

[Zhang 2023] Zhang, Y. and Cheung, Y.-M. *Graph-Based Dissimilarity Measurement for Cluster Analysis of Any-Type-Attributed Data.* IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 9, pages 6530–6544, September 2023. [Online]. Available: `https://ieeexplore.ieee.org/abstract/document/9887970`. Conference Name: IEEE Transactions on Neural Networks and Learning Systems. (Cited in pages 33, 35, and 37.)

[Zhou 2014] Zhou, Y., Liu, Y., Yang, J., He, X. and Liu, L. *A Taxonomy of Label Ranking Algorithms.* Journal of Computers, vol. 9, March 2014. [Online]. Available: http://dx.doi.org/10.4304/jcp.9.3.557-565. (Cited in page 80.)

[Zhu 2020a] Zhu, C., Zhang, Q., Cao, L. and Abrahamyan, A. *Mix2Vec: Unsupervised Mixed Data Representation.* In 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), pages 118–127, October 2020. (Cited in pages 32 and 34.)

[Zhu 2020b] Zhu, X., Li, Y., Wang, J., Zheng, T. and Fu, J. *Automatic Recommendation of a Distance Measure for Clustering Algorithms.* ACM Transactions on Knowledge Discovery from Data, vol. 15, no. 1, pages 7:1–7:22, December 2020. [Online]. Available: `https://dl.acm.org/doi/10.1145/3418228`. (Cited in pages 7, 12, 41, 43, 44, 62, 75, 76, and 79.)

Titre : Evaluation et recommandation des mesures de similarité pour le clustering de données mixtes

Mots clés : Clustering de données mixtes, Mesures de similarité, Système de recommandation, Méta-apprentissage

Résumé : Le clustering est une tâche importante pour l'exploration de données. Il permet de découvrir de manière non-supervisée des tendances ou des structures cachées au sein de grands ensembles de données. Les algorithmes de clustering visent à regrouper un ensemble d'observations en plusieurs groupes ou clusters de telle sorte que les observations au sein d'un même groupe soient similaires entre elles et différentes des observations dans les autres groupes. Un composant clé de ces algorithmes est la mesure de similarité qui a un impact direct sur la construction des clusters et, par conséquent, sur les performances des algorithmes. Le choix d'une mesure de similarité adaptée en fonction des données et de l'algorithme de clustering considéré est donc primordial et constitue l'objet principal de cette thèse.

Notre recherche se concentre sur le clustering de données mixtes qui sont des données hétérogènes présentant à la fois des attributs numériques et catégoriels. Elles sont très courantes dans des domaines tels que la santé, la finance, le marketing et les sciences sociales. Les algorithmes de clustering traditionnels, conçus pour des données homogènes, ne peuvent pas être appliqués directement aux données mixtes, d'où la nécessité de méthodes spécialisées. Nous classons ces méthodes en deux catégories : les approches basées sur la conversion (appelées méthodes d'homogénéisation) et celles qui considèrent les données mixtes directement sans conversion (méthodes mixtes). Nous montrons dans ce manuscrit, que les méthodes mixtes sont généralement préférables aux méthodes d'homogénéisation, car elles conservent la structure originale des données et utilisent un traitement adapté pour chaque type d'attribut. Nos travaux se focalisent donc principalement sur les méthodes mixtes.

Dans un premier temps, nous avons mené une étude expérimentale afin d'évaluer l'impact des mesures de similarité sur les performances des méthodes mixtes. Ces méthodes combinent généralement deux mesures de similarité : l'une pour les attributs numériques et l'autre pour les attributs catégoriels. Nos expérimentations montrent que le choix de ces mesures de similarité influence de manière significative les performances des différents algorithmes considérés, soulignant ainsi l'importance de choisir des mesures appropriées.

Trouver les meilleures ou de bonnes mesures de similarité est difficile, en particulier pour les utilisateurs non experts, en raison du grand nombre de mesures qui existent dans la littérature et de leurs performances variables en fonction du jeu de données, de l'algorithme de clustering et de la mesure de performance. Afin de répondre à cette problématique, nous avons proposé SIMREC, un système de recommandation de mesures de similarité pour les algorithmes de clustering de données mixtes. SIMREC utilise le meta-learning (ou méta-apprentissage) pour identifier les relations entre les caractéristiques des jeux de données et les performances des différentes mesures de similarité, et ce pour différents algorithmes de clustering et mesures de performance. SIMREC prend en entrée un triplet composé d'un jeu de données mixtes, d'un algorithme de clustering et d'une mesure de performance à optimiser. Il recommande ensuite les paires optimales de mesures de similarité numérique et catégorielle en fonction des caractéristiques du jeu de données d'entrée. Ce système permet à la fois à des utilisateurs experts et non experts de choisir de façon efficace des mesures de similarité adaptées à leur cas d'usage, évitant ainsi les stratégies d'essais-erreurs qui sont souvent chronophages et coûteuses.

Title: Evaluation and Recommendation of Similarity Measures for Mixed Data Clustering

Key words: Mixed data clustering, Similarity measures, Recommendation system, Meta-Learning

Abstract: Clustering algorithms are essential in data mining, offering powerful tools to uncover hidden patterns and structures within datasets. aim to divide data points into coherent groups based on similarities or dissimilarities, making it easier to explore and understand complex data. A critical component of clustering algorithms is the similarity measure, which significantly affects their ability to identify meaningful patterns. Thus, selecting suitable similarity measures for clustering algorithms is a crucial challenge addressed in this thesis.

Our research focuses on clustering mixed data—datasets containing both numerical and categorical attributes—which are increasingly common in fields such as healthcare, finance, marketing, and social sciences. Traditional clustering algorithms, designed for homogeneous data, cannot be directly applied to mixed data due to the differing nature of numerical and categorical attribute types. This necessitates specialized approaches for mixed data clustering.

We categorize mixed data clustering methods into two groups: conversion-based approaches (referred to as homogenization methods) and non-conversion-based approaches (mixed methods). Through extensive experiments, we demonstrate that mixed methods are generally more effective, as they handle different data types directly without altering the dataset's inherent structure. In contrast, homogenization methods, which convert one data type into another, often lead to sub-optimal clustering results.

Focusing on mixed methods, we further investigate the impact of similarity measures on clustering performance. Unlike clustering algorithms for homogeneous data, mixed methods typically combine two similarity measures—one for numerical attributes and one for categorical attributes. Our experiments confirm that the choice of these similarity measures significantly influences clustering outcomes, underscoring the importance of selecting the appropriate measures for each dataset.

However, selecting the right similarity measures can be challenging, especially for non-experts, due to the wide range of available measures for each data type and their performance dependency on the dataset, clustering algorithm, and cluster validity index. To address this, we propose SIMREC, a similarity measure recommendation system for mixed data clustering. SIMREC leverages meta-learning to identify relationships between dataset characteristics and the performance of similarity measures for different mixed data clustering algorithms and cluster validity indices. Given a mixed dataset, clustering algorithm, and validity index, the system recommends optimal pairs of numerical and categorical similarity measures based on the dataset characteristics. This system aims to assist both expert and non-expert users in efficiently selecting similarity measures, avoiding time-consuming trial-and-error and search-based strategies.