Clustering complex datasets
algorithms and applications

Zhong, Haodi

Awarding institution:
King's College London

Clustering complex datasets: algorithms and applications

Haodi Zhong

The Department of Informatics
King's College London

This dissertation is submitted for the degree of
Doctor of Philosophy

December 2022

To my family for their unfailing support.

Acknowledgements

First and foremost, I would like to express my sincere thanks to Dr. Grigorios Loukides. Without your support and hard work, I could not complete my master's degree and PhD successfully. From 2016 to 2022, you dedicated a lot of time to guide me. I learned many valuable skills. These will be a great treasure for the rest of my life and guide me in my upcoming career. I am pretty sure these memorable six years will also be a big part of my life. Thank you again.

I would also like to thank Professor Solon P. Pissis and Dr. Robert Gwadera for their excellent collaboration.

Without my parents' constant care and encouragement, I would not achieve what I have. Thank you dad and mom; I will spend more time with you now.

I want to thank all of my friends at King's College London. Thank you for accompanying me through this wonderful time.

I would finally like to acknowledge Kings-China Scholarship Council PhD Scholarship Programme (K-CSC) for their financial support throughout my PhD.

List of Publications

During the course of my PhD studies, I have co-authored and published technical papers in traditional academic formats and venues. The works in Chapters 3, 4 and 5 of this thesis are published at the following journals:

- **Haodi Zhong**, Grigorios Loukides*, Robert Gwadera, *Clustering datasets with demographics and diagnosis codes*, Journal of Biomedical Informatics, 2020
- **Haodi Zhong**, Grigorios Loukides*, Solon P Pissis, *Clustering demographics and sequences of diagnosis codes*, IEEE Journal of Biomedical and Health Informatics, 2021
- **Haodi Zhong**, Grigorios Loukides*, Solon P Pissis, *Clustering sequence graphs*, Data & Knowledge Engineering, 2022

Abstract

Clustering is a fundamental data mining task aiming to partition a given dataset into groups, called clusters, which should be compact and well-separated. In this thesis, we focus on clustering three types of data which are difficult to be handled appropriately by existing clustering algorithms, due to their structural properties: Relational Transaction (RT) datasets, Relational Sequential (RS) datasets, and sequence graphs.

RT-datasets are comprised of relational (single-valued) and transaction (set-valued) attributes and are commonly used to model Electronic Health Record (EHR) data; a patient's demographics are modeled as relational attributes and the patient's set of diagnosis codes is modeled as a transaction attribute. We propose the first approach for clustering an RT dataset comprised of patient demographics and diagnosis codes. Our approach represents the dataset in a binary form in which the features are selected demographic values, as well as combinations of frequent and correlated diagnosis codes. This representation enables measuring similarity between records using cosine similarity and finding compact, well-separated clusters through hierarchical clustering. Our experiments demonstrate that our approach constructs clusters with correlated demographics and diagnosis codes, and that it is efficient and scalable.

RS-datasets are comprised of relational attributes, as well as a sequential attribute. These datasets are also commonly used to model EHR data; a patient's sequence of diagnosis codes is modeled as a sequential attribute. Clustering an RS-dataset is helpful for analyses ranging from pattern mining to classification. However, existing methods are not appropriate to perform this task. Thus, we initiate a study of how an RS-dataset can be clustered effectively and efficiently. First, we formalize the task of clustering an RS-dataset as an optimization problem. Second, we propose a distance measure to quantify the pairwise similarity between records of an RS-dataset. Third, we develop an algorithm which first identifies k representative records (centers), for a given k, and then constructs k clusters, each containing one center and the records that are closer to the center compared to other centers. Experiments using two EHR datasets demonstrate that our algorithm constructs compact and well-separated clusters and that it is efficient and scalable.

A sequence graph is a graph whose nodes are labeled with sequences of letters (i.e., strings). Sequence graphs are commonly encountered in social networks, where nodes represent users and edges represent user friendships, or in e-commerce, where nodes represent consumers and edges represent consumers' trust relationships. Clustering the nodes of a sequence graph allows detecting user communities in a social network, or identifying groups of consumers with bonds of trust among them in e-commerce. However, this problem has not been considered before, to our knowledge. We thus introduce the problem of clustering a sequence graph. We first propose two pairwise distance measures for sequence graphs, one based on edit distance and shortest path distance and another one based on SimRank. We then formalize the problem under each measure, showing also that it is NP-hard. In addition, we design a polynomial-time 2-approximation algorithm, as well as a heuristic for the problem. Experiments using real datasets and a case study demonstrate the effectiveness and efficiency of our methods.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

With the advancement of computer technologies, tremendous amounts of data are being generated e.g., from digital devices and social networks. For example, in 2020, each person generated on average about 1.7 megabytes every second [2]. Also, in 2021, the total amount of data consumed globally was 79 zettabytes [3].

Big data can be defined as a collection of data that is too large or complex for traditional data processing applications to handle [4]. With the rapid development of data storage and networking, Big Data are featured in many fields of engineering and science, such as biomedicine, physics, and sociology. Also, Big Data affect many aspects of daily life [5]. Yes, the era of Big Data has arrived.

Big Data have typically high volume, complex structure and are updated rapidly [4]. These characteristics make Big Data challenging to be processed. One of the fundamental challenges in Big Data processing is to extract useful information or knowledge from data with complex structure. For example, extracting knowledge from data modeled as sequences, time series, graphs, or images is needed to optimize medical processes or improve the quality of medical care [6].

Data mining methods [7] have been proven successful in extracting actionable knowledge. For example, pattern mining methods [8] are able to discover useful associations between data values. They can reveal combinations of products that are frequently purchased together, or statistically significant combinations of disorders that are diagnosed during a patient's hospital visit. As another example, clustering methods are able to "partition a set of data points into natural groups, called clusters, such that points within a group are very similar" (i.e., clusters are compact) and "points between different groups are as dissimilar as possible" (i.e., clusters are well-separated) [7]. Clustering allows us to (I) gain insight into data (e.g., detecting anomalies and identifying salient features), (II) identify the degree of similarity among organisms (e.g., phylogenetic relationships); and (III) summarize data by using cluster prototypes, which is important in anonymization [9].

Traditional data mining methods usually require data to be represented as vectors [7]. However, in real-world applications, much information is difficult to be modeled as a vector without loss of information that may harm the quality of data mining results. Take for example a genomic sequence. It is naturally modeled as a sequence of letters from $\{\texttt{A}, \texttt{C}, \texttt{T}, \texttt{G}\}$. The genomic sequence can be represented in a vector format by taking all length-k substrings. However, this inevitably loses information. To show this, we provide Example 1.

Example 1. *Let $T_1 = \texttt{ACA}$ and $T_2 = \texttt{CAC}$ be two sequences. The 2-grams [10] of both of these sequences are* AC *and* CA *and each of these 2-grams appears only once in T_1 or in T_2. Thus, T_1 and T_2 have the same vector representation $(1, 1)$ where the first 1 denotes the frequency of* AC *and the second 1 denotes the frequency of* CA*, assuming a lexicographic order of q-grams. Since T_1 and T_2 have the same vector representation, they are treated as equal by traditional data mining methods, although they are not.*

Similarly, there are values with: (I) hierarchical relationships, such as diagnosis codes in a diagnosis classification system [11], (II) temporal relationships, such as sensor readings in time-series, or (III) pairwise relationships, such as between users who are friends in a social network. In all these cases, a simple vector representation unavoidably incurs information loss. Many clustering methods were also designed for vector data [12]. Examples are the k-means algorithm and hierarchical clustering algorithms [12]. The complexity of Big Data makes the application of these clustering methods difficult and calls for new clustering approaches.

Motivated by the importance of complex data and the inability of clustering algorithms to deal with many types of complex data that are important in applications, in this thesis, we focus on the development of clustering algorithms for different types of complex data. Specifically, we consider clustering three different types of complex data that are often encountered in applications, RT-datasets, RS-datasets, and sequence graphs:

RT-datasets: A single-valued (or atomic) attribute is an attribute containing one value per record. A set-valued attribute is an attribute containing a set of values per record. Datasets containing both single-valued and set-valued attributes are referred to as RT-datasets (for Relational Transaction datasets) [9]. Consider the dataset in Table 1.1. Each record corresponds to a different patient and contains their demographics and a *set* of diagnosis codes. The dataset contains two types of attributes: (I) Single-valued (or atomic) attributes. The value in these attributes can be a number (respectively, category), in which case the attribute is numerical (respectively, categorical). For example, in Table 1.1, each

Table 1.1 A (toy) example of an RT-dataset. *Gender*, F is for Female and M for Male. The diagnosis codes are represented as ICD-9 codes [1]. The attribute *ID* is for reference.

ID	Gender	Age	Diagnosis codes
1	F	77	$250.00, 272.4, 278.01, 401.9$
2	M	71	$244.9, 285.1, 530.81$
3	F	46	$421.0, 427.31, 584.9$
4	F	78	$250.00, 272.4, 401.9, 414.8$
5	M	73	$244.9, 530.81, 648.01, 661.11$
6	F	48	$285.1, 427.31, 584.9$
7	F	80	$196.6, 250.00, 272.4, 401.9$
8	M	73	$244.9, 401.9, 530.81$
9	F	48	$427.31, 584.9, 693.0$
10	F	75	$250.00, 272.4, 401.9, 560.1$
11	M	73	$218.0, 244.9, 530.81$
12	F	49	$427.31, 584.9, 995.91$

patient's record contains one value in the numerical attribute *Age* and another value in the categorical attribute *Gender*. (II) A set-valued attribute. For example, in Table 1.1, a patient's record contains a set of diagnosis codes in the *Diagnosis codes* attribute.

Table 1.2 A (toy) example of an RS-dataset. *Age*, *Gender*, and *Ethnicity* are demographic attributes; the *Diagnosis codes sequence* attribute is comprised of ICD-9 codes.

Age	Gender	Ethnicity	Diagnosis codes sequence
69	M	Black	$(414.01, 250.00, 272.4, 401.9, 412, 696.1)$
1	F	White	$(765.18, 774.2, 765.27, 769)$
67	M	Black	$(414.01, 4111, 272.1, 250.00, 401.9)$
48	F	White	$(441.2, 401.9, 345.90, 414.01)$
50	F	White	$(414.01, 250.01, 401.9, 412, 2720)$
0	F	White	$(765.19, 769, 774.2, 779.3, 765.28, 771.7)$
61	F	White	$(414.01, 424.0, 440.21, 427.89, 250.00, 401.9)$
1	F	White	$(765.16, 775.6, 765.27, 769)$
68	M	Black	$(414.01, 411.1, 250.01, 401.9, 272.0)$

RS-datasets: A dataset containing single-valued attributes and a sequence is referred to as an RS-dataset (for Relational Sequential datasets). Note that an RS-dataset differs from an RT-dataset in that it contains a sequence instead of a set-valued attribute. Consider the dataset in Table 1.2. Each record corresponds to a different patient and contains their demographics and a *sequence* of diagnosis codes. The first record corresponds to a 69-year old black male patient who is associated with six diagnoses (ICD-9 codes) [13]: first with 414.01 (coronary atherosclerosis of native coronary artery), then with 250.00 (diabetes mellitus type II without complications), and next with $272.4, 401.9, 412$ and 696.1.

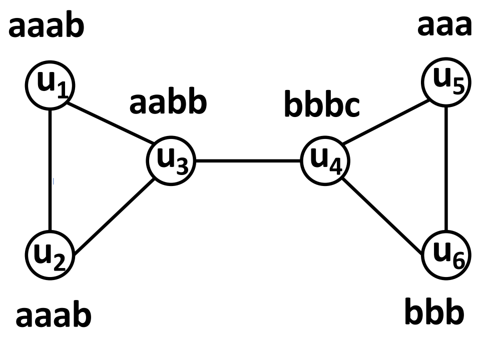


Fig. 1.1 A sequence graph.

Sequence graphs: A graph whose nodes are labeled with sequences of letters (i.e., strings) is referred to as a sequence graph. Consider the graph in Fig. 1.1. The set of nodes is $\{u_1, u_2, u_3, u_4, u_5, u_6\}$ and the node labels are the strings $\{\texttt{aaa}, \texttt{aaab}, \texttt{aabb}, \texttt{bbb}, \texttt{bbbc}\}$. For example, the label of node u_1 is the string aaab.

In this thesis, we develop and evaluate clustering algorithms for all the three aforementioned types of data. Meanwhile, we solve below three research questions.

Research questions:

1. How can we cluster a given RT-dataset so that each cluster (i.e., set of records comprised of a set of relational attributes and a set of diagnosis codes) represents patients that have similar demographics and set of diagnosis codes? We address this question in Chapter 3.

2. How can we cluster a given RS-dataset so that each cluster (i.e., set of records comprised of a set of relational attributes and a sequence of diagnosis codes) represents patients that have similar demographics and sequence of diagnosis codes? We address this question in Chapter 4.

3. How can we cluster a sequence graph so that each cluster (i.e., set of graph nodes each associated with a string) contains nodes that are structurally similar and have similar node labels (strings)? We address this question in Chapter 5.

The thesis makes the following specific contributions:

- **For clustering RT-datasets:** We propose a new approach for clustering an RT-dataset. Our approach represents the dataset in a binary form in which the features are selected demographic values, as well as combinations (patterns) of frequent and correlated diagnosis codes (comorbidities). This representation enables measuring similarity between records using cosine similarity [7], an effective measure for binary-represented data, and finding compact, well-separated clusters through hierarchical clustering [14]. For example, Table 1.3 is a clustered RT-dataset produced from the dataset in Table 1.1 by our method. Such clusters allow discovering relationships between the clinical profiles of patients and can be provided as input to classification and anonymization methods [15–19]. Our experiments demonstrate the effectiveness and efficiency of our approach. In particular, they show that our approach outperforms four baselines in terms of clustering quality, it can construct clusters with correlated demographics and diagnosis codes, and it is efficient and scalable.

Table 1.3 A clustered RT-dataset produced from the dataset in Table 1.1 by our proposed method. The attributes *Cluster ID* and *ID* are for reference.

Cluster ID	**ID**	**Gender**	**Age**	**Diagnosis codes**
1	1	F	77	250.00, 272.4, 278.01, 401.9
1	4	F	78	250.00, 272.4, 401.9, 414.8
1	7	F	80	196.6, 250.00, 272.4, 401.9
1	10	F	75	250.00, 272.4, 401.9, 560.1
2	2	M	71	244.9, 285.1, 530.81
2	5	M	73	244.9, 530.81, 648.01, 661.11
2	8	M	73	244.9, 401.9, 530.81
2	11	M	73	218.0, 244.9, 530.81
3	3	F	46	421.0, 427.31, 584.9
3	6	F	48	285.1, 427.31, 584.9
3	9	F	48	427.31, 584.9, 693.0
3	12	F	49	427.31, 584.9, 995.91

Table 1.4 A clustered RS-dataset produced from the dataset in Table 1.2 by our algorithm. *Cluster ID* is for reference.

Cluster ID	Age	Gender	Ethnicity	Diagnosis codes sequence
1	69	M	Black	(414.01, 250.00, 272.4, 401.9, 412, 696.1)
1	67	M	Black	(414.01, 411.1, 272.1, 250.00, 401.9)
1	68	M	Black	(414.01, 411.1, 250.00, 401.9, 272.0)
2	1	F	White	(765.18, 774.2, 765.27, 769)
2	0	F	White	(765.19, 774.6, 765.28, 779.3, 769, 771.7)
2	1	F	White	(765.16, 774.2, 765.27, 769)
3	48	F	White	(441.2, 414.01, 345.90, 440.32, 250.00, 401.9)
3	50	F	White	(414.01, 440.31, 250.01, 401.9, 412, 272.0)
3	61	F	White	(414.01, 424.0, 440.21, 427.89, 250.00, 401.9)

- **For clustering RS-datasets:** Motivated by the importance of the task of clustering an RS-dataset in applications such as clinical pathway mining [20], anonymization [15], causation inference [21], visualization [22, 23], and trend discovery [24, 25], we formalize this task as an optimization problem. We prove that the problem is computationally hard, and we propose an effective and efficient algorithm to address it. This algorithm uses a distance measure we develop and works by identifying k representative records (centers), for a given k, and then constructing k clusters, each containing one center and the records that are closer to the center compared to other centers. For example, Table 1.4 is a clustered RS-dataset produced from the dataset in Table 1.2 by our algorithm. Clearly, the patients in each cluster have similar values in demographics and also similar diagnosis codes that occur in similar order. Our experiments demonstrate that our algorithm can construct compact and well-separated clusters, which preserve meaningful relationships between demographics and sequences of diagnosis codes. In addition, they show that our algorithm is efficient and scalable.

- **For clustering sequence graph:** We introduce the problem of clustering sequence graphs and study variants of the problem based on the k-center [26, 27] and k-median [26, 28] problems. The task of clustering sequence graphs is important for several analyses (e.g., geo-social network [29], e-commerce [30], and medicine [31]). We first propose a product metric and a measure based on SimRank [32] to capture the distance between two nodes of a sequence graph, as well as a proxy for each measure. We then propose an approximation algorithm and a heuristic, which outperform attribute-based graph clustering methods, as shown experimentally. Last, we propose a methodology that applies our measures (and the corresponding clustering algorithms) to evaluate whether a given phylogenetic tree is in accordance with a given ground truth clustering. Experiments using real datasets and a case study demonstrate the effectiveness and efficiency of our methods.

The rest of the thesis is organized as follows. Chapter 2 reviews clustering methods for EHR data. Chapter 3 presents the proposed approach for clustering RT-datasets. Chapter 4 presents the proposed approach for clustering RS-datasets. Chapter 5 presents the proposed approaches for clustering sequence graphs. Last, Chapter 6 concludes the thesis.

Chapter 2

Electronic health records clustering review

Electronic Health Records (EHR) consist of a range of different patient attributes, such as demographics, medications, laboratory test results, diagnosis codes, and procedures. In practice, the EHR data of a patient can be represented as a set, a record in a relational dataset, or as a sequence when the ordering of values or temporal information is important. We briefly discuss such representations in Section 2.1. Subsequently, in Section 2.2, we review methods for EHR data clustering, based on the aforementioned data types; sets, relational data, and sequences.

2.1 Representing EHR data

2.1.1 Set representation

A set is a collection of unique elements. In the context of EHR data, the diagnosis codes assigned to a patient during one or more visits can be represented as a set (see the last column of Table 1.1) [33, 34]. A collection of sets, each corresponding to a different patient, can then be given as input to a clustering algorithm.

2.1.2 Relational data representation

A relational database is a collection of data items with pre-defined relationships between them. These items are organized as a set of tables, which are used to hold information about the objects to be represented in a database [35]. A relational table, also referred to as a relational dataset, has columns (attributes) and rows (records). Each record in the table has a fixed number of attributes and takes a single value in each of these attributes.

Furthermore, each attribute value is typically drawn from a small set of values. EHR data can be presented as a relational table. For example, in Table 1.1, *Gender* is an attribute, whose value in a record can be either F (for female) or M (for male). Alternatively, an attribute value can represent a patient's medication, laboratory test result, diagnosis code, or procedure.

2.1.3 Sequence representation

A sequence can be thought of as a collection of elements with a particular order, where the order implies the temporal information [36]. A sequential dataset is a collection of sequences, which do not necessarily have the same number of elements. In the context of EHR data, a sequence can represent information spanning a patient's entire lifetime of care. For example, in Table 1.2, the last attribute of each record is a sequence of diagnosis codes. A sequence can "capture genetic and lifestyle risks, signal the onset of diseases, show the advent of new morbidities and comorbidities, indicate the time and stage of diagnosis, and document the development of treatment plans and their efficacy" [37].

2.2 Clustering EHR data

We categorize existing algorithms for clustering EHR data into algorithms applied to sets, relational data, or sequences.

2.2.1 Clustering sets for EHR

A patient's diagnosis codes are commonly modeled as a set-valued attribute [33, 34]. Many methods first use pattern mining to extract patterns comprised of diagnosis codes from a collection of sets (also referred to as transaction dataset) and then perform clustering of the collection, based on the extracted patterns. In Section 2.2.1.1, we review such pattern-based clustering methods. Alternative approaches to pattern-based clustering have also been considered, although they have not been specifically applied to EHR data, to the best of our knowledge. We briefly review such approaches in Section 2.2.1.2.

2.2.1.1 Pattern-based clustering

Pattern mining is an important task aiming to discover associated attribute values, often in a dataset comprised of one set-valued attribute. The values in the set-valued attribute are referred to as *items*. There are different ways of modeling associations, leading to different representations of patterns [7], such as *frequent*, *maximal-frequent*, and *all-confident* patterns (see Section 3.3 for details).

As mentioned above, pattern mining is used as a first step for pattern-based clustering. Several existing works aim to mine frequent patterns and associations from EHR data [38–44]. For example, the work of [39] introduced a novel method to mine association rules, which was used to evaluate comorbidities (i.e. correlated frequent patterns). It also developed a comorbidity interestingness score to rank index morbidities. The work of [40] applied the Apriori algorithm [45] to mine associations and then analyzed the strengths of the associations among hypertension and other diseases. The work of [41] proposed a method for mining association rules, which was used to identify disease-related genes using MeSH terms. Last, the work of [42] focuses on predicting associations between diseases and miRNAs. In particular, it extended a recommendation algorithm by utilizing the network structure, information propagation, and several notions of similarity, such as miRNA functional similarity, disease semantic similarity, and Gaussian interaction profile kernel similarity for miRNAs and diseases.

Pattern-based clustering methods are effective for clustering high dimensional data [46], since the number of patterns used by these methods is typically smaller than that of the distinct values in the dataset (e.g., ICD-10-CM has $68,000$ codes, ICD-9-CM has $13,000$ codes). Also, by focusing on records containing patterns with certain properties, such as frequent patterns [47], pattern-based clustering methods construct clusters that are not “too” small and are thus easy to interpret based on the patterns. There are several pattern-based clustering methods [48, 49, 33, 34, 50], all based on frequent patterns. For example, the work of [34] proposed a relative risk (RR) measure [51] based on statistical co-occurrences of pairs of patient diagnoses, as well as a hierarchical agglomerative clustering algorithm to construct the clustering hierarchy, based on the relative risk of co-occurrences of patient diagnoses. The work of [33] proposed a clustering method that first uses a variant of RR to measure the strength of the relationship between two diagnoses and then constructs a multimorbidity network, by connecting all pairs of diagnoses that are related. Next, it applies the M-algorithm [52] to cluster the multimorbidity network. Each clustered subgraph of the network represents a set of patients that have similar diagnoses (i.e., a cluster). The work of [33] also performed a cluster analysis of diagnoses, using data from the Finnish Health Care Registers for primary and specialized health care visits and inpatient care.

2.2.1.2 Other approaches

Some works [53–55] can cluster a set-valued attribute without resorting to pattern mining. These works can be directly applied to cluster a collection of sets comprised of diagnosis codes, although they have not been applied in this context. For example, ROCK [55] is an algorithm that is based on the concept of links between sets. Two sets are neighbors, if their similarity, based on a distance measure for sets such as Jaccard distance [7], exceeds a threshold. The number of links between two sets is then simply the number of neighbors these sets have in common. Sets belonging to the same cluster will generally have many common neighbors and thus more links. Therefore, the ROCK algorithm merges clusters with the largest number of common neighbors first to create high-quality clusters. The CLOPE algorithm [53] measures the similarity between two sets based on a measure that takes into account the total number of items in the cluster, the number of distinct items in a cluster, and the cluster sizes. It is more efficient than ROCK and equally effective in practice. The *Weighted Coverage Density* (WCD) [54] measure is the crux of an algorithm proposed in [54]. WCD is similar in principle to the similarity criterion used in CLOPE and aims at creating clusters with as many frequent items as possible, while controlling the items overlap between clusters implicitly. WCD has been employed in a sampling-based framework [54], which was shown to be more scalable and effective than CLOPE.

2.2.2 Clustering relational datasets for EHR

Demographics, medications, and laboratory test results are typically transformed into feature vectors of a small number of dimensions (typically 20 or fewer). Then, existing shallow (non-deep) clustering algorithms cluster a collection of such feature vectors (i.e., a relational dataset). In the following, we review three main types of such clustering algorithms; hierarchical, partitional, and density-based. We also examine works that applied these algorithms, or their variations, on relational EHR data.

2.2.2.1 Hierarchical clustering

Hierarchical clustering [14] is an approach that seeks to build a hierarchy (tree) of clusters. Strategies for hierarchical clustering generally fall into two categories:

- **Agglomerative:** Each cluster is comprised of a single record initially, and two clusters are merged as one moves up the hierarchy.

- **Divisive:** A cluster comprised of all records is created initially, and it is split into smaller clusters recursively as one moves down the hierarchy.

The merge decisions in agglomerative clustering, as well as the split decisions in divisive clustering, are based on linkage functions (e.g., single-linkage, average-linkage, and complete linkage) [7]. For example, in single-linkage agglomerative clustering two clusters are merged if they have the smallest minimum pairwise distance (i.e., if they contain the closest pair of elements), while in complete linkage they are merged if they have the smallest maximum pairwise distance.

Several works aim to discover clusters from EHR based on hierarchical clustering [56–59]. For example, the work of [57] proposed a clustering method based on hierarchical-k-means [60], which first computes a hierarchical clustering, then cuts the hierarchy in k clusters, and next computes the centroids for each cluster, which are used to initialize k-means. This method was used to cluster a relational dataset whose records contain laboratory test results and vital signs. The work of [56] employed an agglomerative hierarchical clustering algorithm, based on average-linkage. The distance measure used in this algorithm considers the semantic similarity between attribute values. An agglomerative hierarchical clustering algorithm, based on average-linkage, was also employed in [58] to cluster 289 "high-burden" diseases and in [59] to create clusters corresponding to patient risk groups.

2.2.2.2 Partitional clustering

Partitional clustering algorithms [14] construct a desired number of clusters, k, by first partitioning a relational dataset into k parts, each representing a different cluster. Then, they assign each record of the dataset into one of the parts, in a way that aims to optimize an objective function (e.g., the sum of squared error criterion [14]). Typically, partitional clustering algorithms are iterative. That is, they assign the records into clusters, calculate the error after the clustering asssignment, and then refine the clusters multiple times, in an attempt to improve their objective function.

The most well-known partitional clustering algorithm is k-means [61]. Given a desired number of clusters, k, and a relational dataset, this algorithm selects k records as cluster representatives. Then, it assigns each other record of the dataset to the cluster containing its nearest representative, according to a distance function. After that, the representatives are updated and if they change, the records are assigned again to the clusters containing their nearest representatives. This process (update of representatives and record assignment) is performed, until none of the representatives change. The k-means algorithm uses the Euclidean distance as its distance function and the centroids (i.e., vectors that minimize the sum of squared Euclidean distances between themselves and each record in the input dataset) as representatives. It also selects the initial representatives uniformly at random.

There are numerous variants of k-means, aiming to improve its effectiveness and/or efficiency (see [62] for a survey). For example, k-means++ [63] is a variant of k-means that selects the first centroid randomly and each other centroid with a probability proportional to its contribution to the total within-cluster sum of squares measure (i.e., the sum of squared Euclidean distance between each record and its centroid). Importantly, k-means++ produces a discretization that is within $O(\log(k))$ from the optimal discretization [63]. This implies that the discretized values are comprised of very similar numerical values, in terms of the total within-cluster sum of squares measure.

Another partitional algorithm is k-medoids [64]. It is similar to k-means but has two major differences from it. First, the representatives selected by k-medoids must be records of the dataset, while this is not a requirement in k-means. This helps interpretability. Second, k-medoids minimizes a sum of pairwise dissimilarities instead of a sum of squared Euclidean distances. This makes it more robust than k-means to outliers and noise. There are numerous adaptations of k-medoids (see [65] and references therein). They can be categorized into those that produce a similar result with k-medoids, such as FastPAM and FastPAMI [66], and others that trade-off effectiveness for increased efficiency, such as CLARA [67] and CLARANS [68].

Several partitional clustering algorithms have been applied to relational EHR data. For example, the work of [69] applied k-means to a relational dataset in the context of antimicrobial resistance. The dataset was comprised of patient demographics and attributes about admission and treatments. In addition, the work of [70] used k-means++ on a relational dataset, constructed by embedding [71] clinical documents. The attribute values in the constructed dataset are numbers (weights) that capture semantic relationships between terms in the documents. The work of [72] applied several variations of k-means that are able to deal with numerical and categorical attributes, as well as k-medoids, to a Chronic Kidney Disease (CKD) dataset containing demographics and laboratory measurements of patients. Last, the work of [73] evaluated k-medoids using five popular distance measures on simulated EHR datasets of different types (numerical, categorical, or both numerical and categorical).

2.2.2.3 Density-based clustering

The notion of density is central in density-based clustering algorithms, and it can be defined in multiple ways [74], e.g., based on the number of records that lie within a region (neighborhood), or on data dependent similarity measures. The goal of density-based clustering algorithms is to create clusters as "areas of high point density that are separated by areas of low point density" [75]. This criterion allows the density-based clustering algorithms to be able to create arbitrarily shaped clusters. Another benefit of such algorithms is their ability to detect arbitrarily shaped clusters and to be robust to outliers and noise [74].

Two widely-used density-based clustering algorithms are DBSCAN [76] and OPTICS [77]. DBSCAN uses as density the number of records that lie within a region, which is defined based on a radius. The clusters are created based on a threshold for density and a radius, in two steps. First, each record is associated with its region. Second, the records are assigned into clusters based on the distance among them. Several works have applied DBSCAN or adaptations of it on EHR data [78–80]. For example, the work of [78] proposed an algorithm that is conceptually similar to DBSCAN but uses a gradual radius relaxation strategy and an index to improve the efficiency of clustering. The work of [79] applied DBSCAN on a relational dataset, created by applying dimensionality reduction using t-SNE [81], in order to detect longitudinal relationships between diseases. In addition, the work of [80] applied DBSCAN to cluster comorbidities based on the disease codes of an Autism Spectrum Disorder patient cohort.

OPTICS uses the same input parameters as DBSCAN (range and density threshold) but also considers records contained in clusters of dense clusters and uses a priority queue. It has been employed in [82] to cluster a relational dataset, created by applying t-SNE to a dataset containing more than one hundred attributes, including sociodemographics, comorbidities, vital signs, and laboratory test results.

2.2.3 Clustering sequences for EHR

The elements of a sequence in the context of EHR data can represent different types of information, including diagnosis codes [37, 83, 84]. Consequently, the clustering of a collection of sequences derived from EHR data may offer significant benefits [85, 86]. For example, it can help predicting when a patient will be diagnosed with a disease or identifying links between diagnostic events and clinically understandable phenotypes. In this section, we review clustering algorithms for EHR-derived sequential data. We categorize them into deep learning and non-deep learning based.

2.2.3.1 Deep learning for EHR sequential data

Deep clustering aims at creating meaningful groups of unstructured data, or high dimensional data, with deep neural networks [87]. For clustering sequential data, which are inherently high-dimensional, the following deep learning architectures have been used across many domains [88]: Auto-Encoder (AE) [89], Convolutional Neural Network

(CNN) [90], Recurrent Neural Network (RNN) [91], Attention Neural Network (ANN) [92], and Long-Short Term Memory (LSTM) [93].

In the following, we review recent works [86, 94–99] showing that the aforementioned architectures can learn useful information from EHR sequential data, which helps their accurate clustering. We refer to the survey of [87] for an in-depth discussion of these architectures and their use in clustering and to the survey of [100] for an in-depth review of different deep clustering approaches that do not focus on EHR data.

The work of [86] proposed a supervised deep learning model utilizing Auto-Encoders (AEs) to cluster EHR data, based on the identification of clinically understandable phenotypes with respect to both outcome prediction and patient trajectory. The data considered in this work is a collection of records, one per patient. Each of these records has two components. The first component is a sequence, comprised of numerical vectors (e.g., age, vital signs, haematological variables, and serum variables), and the second component is a one-hot encoded vector of four possible outcomes (class labels); hospital discharge, the first instance of unplanned entry to ICU, cardiac arrest, and death. The deep learning model is based on a novel loss function aiming to address the problems of class imbalance and cluster collapse. The model also includes a feature-time attention mechanism to identify cluster-based phenotype importance across time and feature dimensions.

The work of [94] proposed a supervised deep learning model to cluster a dataset representing patients with Parkinson's disease (PD). The patients are associated with sequences, which are comprised of input features and target features. The input features are demographics, clinical features, biospecimen, and imaging data, and the target features are variables that were shown to be related to PD progression. The features (i.e., elements of the sequence) are either binary or numerical. The objective is to create clusters that represent PD progression subtypes and contain patients that are similar with respect to temporal trends in their records. To achieve this, the sequences are given as input to LSTM [93], which standardizes and densifies them. Then, the Dynamic Time Warping (DTW) [101] algorithm is employed in order to calculate pairwise sequence similarities, from which PD subtypes are derived. Last, the k-means clustering algorithm (see Section 2.2.2.2) is applied to identify distinct subtypes in the dataset.

The work of [95] proposed another supervised deep learning model. It is based on RNNs, and it was proposed for predictive clustering of EHR time-series. Predictive clustering aims at finding cluster assignments and centroids by learning discrete representations of time-series that best describe the future outcome distribution. The data considered in [95] is a collection of records, one per patient. Each record is a sequence of pairs and each pair is comprised of a number (representing a demographic value, genetic mutation, bacterial infection, lung function score, therapeutic management, or diagnosis on comorbidities) and an outcome (representing a class label). The proposed model uses novel loss functions to favor clusters having homogeneous future outcomes (e.g., adverse events, comorbidities, etc.). The work also uses optimization procedures for avoiding trivial cluster assignments and centroids.

The work of [96] proposed an unsupervised deep learning model, which is based on word embedding, CNNs, and AEs to transform patient trajectories into low-dimensional latent vectors. In this work, each patient trajectory is a sequence of medical concepts of fixed length, and each medical concept is extracted from free text using specialized tools [102]. Then, the representations learned by the model are used to enable patient stratification by applying hierarchical clustering to different multi-disease and disease-specific patient cohorts.

The work of [97] proposed an unsupervised clustering method, which is combined with AEs. The method can be applied to discover distinct movement patterns that can identify individuals' risk of adverse acute events, but it was also applied to an EHR dataset. In the EHR dataset, each record has two parts, one with demographics (e.g., age, gender, and body weight) and another with several time-series (e.g., systolic blood pressure and diastolic blood pressure). The time-series part of the data is fed into an AE to construct a low-dimension representation of the signal, and this representation is then used for the time-series clustering. In addition, k-means is applied to cluster the demographics part. Then, a technique, termed coordinated clustering, is used to align the timer-series and static clustering outcomes.

The work of [98] proposed an unsupervised adaptive clustering model. The model is explainable and aims to help psychologists identify the most important aspects of emotions of mentally ill people. The input data are patient-authored text and terms that are answers to a PHQ-9 questionnaire [103]. The data are first embedded by using natural language processing, and then a deep learning model with ANN is adopted to train and create clusters.

The work of [99] proposed an unsupervised deep learning method, based on AEs, to identify clusters of chronic cough patients. The method gets as input a collection of sequences, each comprised of diagnosis codes. Then, it uses the Bag-of-word method [104] to transform each sequence into a vector, which is subsequently fed into the deep model. The deep model iteratively optimizes the learning step and clustering step. Next, descriptive statistics are computed to determine patient characteristics associated with different clusters.

2.2.3.2 Non-deep learning for EHR sequential data

In the following, we review non-deep learning methods [85, 105, 106] for clustering EHR sequential data.

The work of [85] proposed a new clustering method, which extends the conventional Latent Dirichlet Allocation (LDA) [107] and uses a Poisson distribution to model a patient's sequence of disease diagnoses. The method gets as input a collection of sequences, each comprised of diagnosis codes. The method discovers latent diseases of clusters and their posterior probabilities for each patient. These probabilities are then used as features to obtain the clusters of patient records.

The work of [105] proposed a supervised contrastive learning framework [108], for the clustering and retrieval of cardiac signals. A signal is modeled as a time series, based on multiple patient attributes (e.g., disease class and age). The resulting clusters could help one to reliably search for and retrieve relevant instances from clinical databases.

The work of [106] proposed a method that gets as input a collection of time-series, each comprised of laboratory test results (e.g., the series of blood pressure measurements of a patient). The method first converts the time-series into strings (sequences of letters). Then, it gives these strings as input to a partitional clustering algorithm [109], which produces the clusters. The clustering algorithm uses edit distance to capture similarity between strings.

Chapter 3

Clustering RT-datasets

Clustering data derived from Electronic Health Record (EHR) systems is important to discover relationships between the clinical profiles of patients and as a preprocessing step for analysis tasks, such as classification. However, the heterogeneity of these data makes the application of existing clustering methods difficult and calls for new clustering approaches. In this chapter, we propose the first approach for clustering a dataset in which each record contains a patient's values in demographic attributes and their *set* of diagnosis codes. Our approach represents the dataset in a binary form in which the features are selected demographic values, as well as combinations (patterns) of frequent and correlated diagnosis codes. This representation enables measuring similarity between records using cosine similarity, an effective measure for binary-represented data, and finding compact, well-separated clusters through hierarchical clustering. Our experiments using two publicly available EHR datasets, comprised of over 26,000 and 52,000 records, demonstrate that our approach is able to construct clusters with correlated demographics and diagnosis codes, and that it is efficient and scalable.

3.1 Overview

An Electronic Health Record (EHR) can be defined as an electronic record of the medical and treatment history of a patient [110], which contains (among others) a patient's demographics, diagnoses, medications, and laboratory results. EHRs can benefit healthcare delivery, by reducing documentation time [111] and facilitating the sharing of patient information [112].

In addition, EHRs can improve clinical research and data-driven quality measures through the application of data mining technologies [113, 114]. Such technologies can be used, for example, to guide the treatment of patients [115], by partitioning the data into meaningful groups through clustering, or by identifying co-occurring diagnoses (*comorbidities*) that help prognosis and quality of care assessment, through pattern mining. However, the heterogeneity of EHR data makes several existing data mining methods inapplicable to EHR data, calling for new methods [114].

3.1.1 Motivation

Table 3.1 A (toy) example of an RT-dataset (copy of Table 1.1). *Gender*, F is for Female and M for Male. The diagnosis codes are represented as ICD-9 codes [1]. The attribute *ID* is for reference.

ID	Gender	Age	Diagnosis codes
1	F	77	250.00, 272.4, 278.01, 401.9
2	M	71	244.9, 285.1, 530.81
3	F	46	421.0, 427.31, 584.9
4	F	78	250.00, 272.4, 401.9, 414.8
5	M	73	244.9, 530.81, 648.01, 661.11
6	F	48	285.1, 427.31, 584.9
7	F	80	196.6, 250.00, 272.4, 401.9
8	M	73	244.9, 401.9, 530.81
9	F	48	427.31, 584.9, 693.0
10	F	75	250.00, 272.4, 401.9, 560.1
11	M	73	218.0, 244.9, 530.81
12	F	49	427.31, 584.9, 995.91

As discussed in Chapter 1, each record in an RT-dataset contains single-valued attributes and a set-valued attribute (see Table 3.1). We consider the task of clustering an RT-dataset comprised of demographics and diagnosis codes. This task aims to create meaningful groups of records (*clusters*) that share similar demographics and diagnosis codes. In other words, the task aims to find natural, hidden structures of the data [14]. For example, our method may produce a cluster with male patients under 60 associated with diseases of the respiratory system, another cluster with male patients over 60 associated with mental disorders, and a third cluster of female patients under 40 associated with complications of pregnancy. Furthermore, the records in each cluster contain correlated diagnosis codes affecting many patients, which helps the interpretability of clusters [47]. The created clusters are useful for several analytic tasks, including: (I) visualization (e.g., to obtain insights on patient subpopulations by examining the visualized clusters), (II) query answering (e.g., to derive aggregate statistics about patient subpopulations in different clusters and use them to compare the subpopulations), (III) anonymization [15] (e.g., to use the clusters as input to algorithms that transform the values in each cluster to protect patient privacy), and (IV) classification (e.g., to preprocess a dataset in order to derive classes of records, which can subsequently be used for performing classification more efficiently and effectively [116, 117]). Thus, one can cluster an RT-dataset and then use the clustering result in one or more of these tasks.

However, existing clustering algorithms are not designed to cluster an RT-dataset comprised of demographics and diagnosis codes. This is because, as explained in [15, 9]:

(I) Most clustering algorithms use a single similarity measure, and it is difficult to design a measure that captures the similarity of records with both single-valued and set-valued attributes. The reason is that single-valued attributes, such as demographics, and set-valued attributes, such as the attribute comprised of diagnosis codes, have different semantics. That is, there is one value per record in a demographic attribute, among a relatively small number of possible values, while there is a large number of diagnosis codes per record, among a very large number of possible diagnosis codes. This makes it difficult to find a single function (similarity measure) to capture how similar the demographics and diagnosis codes of two or more records are. For instance, Euclidean distance, which is applicable to numerical demographics, is not suitable for measuring distance between sets of diagnosis codes, and Jaccard distance, which is applicable to sets of diagnosis codes, is not suitable for measuring distance between numerical demographics.

(II) Multi-objective clustering algorithms that aim to optimize several measures simultaneously are not suitable for RT-datasets. For example, using two-level (hybrid) optimization strategies, which first try to cluster demographics and then diagnosis codes (e.g., the strategy used in [15]), are not able to find high-quality clusters, as shown in our experiments.

Furthermore, bi-clustering (also referred to as co-clustering) methods (e.g., [118, 119]) are not designed to cluster an RT-dataset comprised of demographics and diagnosis codes. This is because they may produce clusters containing parts of records, while in our clustering clusters must contain entire records.

3.1.2 Contributions

We propose the first clustering approach that is designed for an RT-dataset comprised of demographics and diagnosis codes. The main idea of our approach is to construct a record representation that allows measuring similarity between records, based on both demographics and diagnosis codes.

To construct such a representation, we *discretize* [120] numerical demographics (i.e., replace their values with aggregate values) and select subsets of the diagnosis codes contained in the dataset, which are referred to as *patterns*. Then, we represent each record of the RT-dataset using one-hot encoding, producing a *binary representation* of the dataset (see Table 3.2). The features (columns) in the binary representation are: (I) the values in each discretized numerical demographic attribute, (II) the values in each categorical demographic attribute, and (III) the selected patterns. A value of 1 (respectively, 0) in a feature of the binary representation implies that the record contains (respectively, does not contain) the feature. Based on the binary representation, we construct clusters comprised of similar records, by applying a clustering algorithm that is suitable for binary-represented data.

Table 3.2 Binary representation of the RT-dataset in Table 3.1. The features in the binary representation are: $\{F\}$ and $\{M\}$, the values in the categorical demographic attribute *Gender*; $\{45,\ldots,49\}$, $\{70,\ldots,74\}$, and $\{75,\ldots,80\}$, the values of the discretized numerical attribute *Age*; and $\{250.00, 272.4, 401.9\}$, $\{244.9, 530.81\}$, and $\{427.31, 584.9\}$, the patterns comprised of diagnosis codes. The binary representation is clustered into three clusters, with *Cluster ID*s 1, 2, and 3. The attributes *Cluster ID* and *ID* are for reference.

Cluster ID	ID	{F}	{M}	{45,...,49}	{70,...,74}	{75,...,80}	{250.00,272.4,401.9}	{244.9,530.81}	{427.31,584.9}
1	1	1	0	0	0	1	1	0	0
1	4	1	0	0	0	1	1	0	0
1	7	1	0	0	0	1	1	0	0
1	10	1	0	0	0	1	1	0	0
2	2	0	1	0	1	0	0	1	0
2	5	0	1	0	1	0	0	1	0
2	8	0	1	0	1	0	0	1	0
2	11	0	1	0	1	0	0	1	0
3	3	1	0	1	0	0	0	0	1
3	6	1	0	1	0	0	0	0	1
3	9	1	0	1	0	0	0	0	1
3	12	1	0	1	0	0	0	0	1

Yet, there are two challenges that need to be tackled to realize our approach. First, we need a way to select patterns that help constructing a high-quality clustering. Second, we need a way to construct high-quality clusters efficiently. We address these challenges by proposing two methods; Maximal-frequent All-confident pattern Selection (MAS) and Pattern-based Clustering (PC):

1. The MAS method selects patterns that:

(I) occur in a large number of records,

(II) are comprised of correlated diagnosis codes (i.e., any codes in the pattern imply the other codes in the pattern with high probability), and

(III) co-occur in a large number of records, when the patterns share diagnosis codes.

These three properties affect the clustering result as follows: Property I favors patterns of diagnoses that many patients have, thus very small clusters, which are difficult to interpret, are avoided. Property II favors clusters with correlated diagnosis codes, thus meaningless clusters with diagnoses that co-occur by chance are avoided. Property III favors compact and well-separated clusters [47]. Example 2 illustrates the MAS method.

Example 2. MAS *is applied to the diagnosis code attribute of the* RT*-dataset in Table 3.1, and it finds the following three patterns (see Table 3.2):* $\{250.00, 272.4, 401.9\}$, $\{244.9, 530.81\}$*, and* $\{427.31, 584.9\}$. *Each pattern is comprised of a set of diagnosis codes that appears in at least* 4 *records, and it does not share diagnosis codes with other patterns. Also, the diagnosis codes in each pattern are correlated, because the presence of any subset of these diagnosis in Table 3.1 implies the presence of the remaining diagnosis codes. For example, the first pattern* $\{250.00, 272.4, 401.9\}$ *is comprised of the diagnosis codes* 250.00*,* 272.4*, and* 401.9 *(corresponding to "*Diabetes mellitus without mention of complication*", "*other and unspecified hyperlipidemia*", and "*essential hypertension*", respectively) and appears in records with IDs* 1*,* 4*,* 7*, and* 10*. Whenever* 250.00 *is contained in a record,* 272.4 *and* 401.9 *are contained in the same record as well.*

2. The PC method takes as input an RT-dataset, as well as the patterns selected by MAS, and it outputs a clustering of the dataset. PC performs the following tasks: (I) It creates the binary representation of the RT dataset. In this representation, the records containing the diagnosis codes in at least one of the selected patterns by MAS comprise the dataset to be clustered. This allows our method to focus on records which contain frequent and correlated diagnosis codes and lead to interpretable clusters. The remaining records (*unclustered* records) can be dealt with separately, with different strategies depending on the application requirements (see Chapter 6). (II) It clusters the binary-represented dataset, to create groups of similar records with respect to the features and hence with respect to both demographics and diagnosis codes. (III) It constructs the final clustered RT-dataset. Clustering is performed with the hierarchical average-linkage agglomerative clustering algorithm [14] based on cosine similarity [7]. The reason we employ hierarchical average-linkage agglomerative clustering is that, unlike partitional, density-based, as well as single and complete linkage hierarchical algorithms, it does not use the notion of clustering representative, which is inappropriate for binary-represented data [55, 121]. The reason we employ the cosine similarity is that, unlike Euclidean or Jaccard distance, it is effective for clustering binary-represented data that contain many features [122]. The benefit of our approach is that it creates clusters comprised of correlated diagnosis codes (due to the selected patterns) that are also correlated with demographics (due to clustering). Example 3 illustrates the PC method.

Example 3 (Cont'd from Example 2)**.** *Table 3.3 shows the output of applying our* PC *method to the* RT*-dataset in Table 3.1, using the patterns* $\{250.00, 272.4, 401.9\}$, $\{244.9, 530.81\}$*, and* $\{427.31, 584.9\}$ *that were selected by* MAS. PC *first constructs the binary representation of the dataset, shown in Table 3.2. The features in Table 3.2 are the values* F *and* M *in the categorical demographic attribute* Gender*, the discretized values* $\{45, \dots, 49\}$*,* $\{70, \dots, 74\}$*, and* $\{75, \dots, 80\}$ *in the numerical demographic attribute* Age*, and the input patterns. Each record in Table 3.2 has an* 1 *(respectively,* 0*) in a feature, if it contains (respectively, does not contain) the feature. For example, the record with ID 1 in Table 3.1, which represents a* 77*-year old female patient with diagnosis codes* 250.00*,* 272.4*, and* 401.9 *(among others), corresponds to the record with ID 1 in Table 3.2, which contains* 1 *in the feature* $\{F\}$ *for* Gender*, the discretized value* $\{75, \dots, 80\}$ *for* Age*, and the pattern* $\{250.00, 272.4, 401.9\}$ *for* Diagnosis codes*. The binary-represented data in Table 3.2 are then grouped into three clusters, so that records in the same cluster share similar features (which implies that they have similar demographics and diagnosis codes). For example, the cluster with Cluster ID 1 in Table 3.2 is comprised of the records with IDs* 1*,* 4*,* 7 *and* 10*. These records correspond to female patients between* 75 *and* 80 *and contain all diagnosis codes in the pattern* $\{250.00, 272.4, 401.9\}$*. Next, the clustered* RT*-dataset in Table 3.3 is contructed, by simply adding into each cluster the records from the* RT*-dataset in Table 3.1 which were clustered together in the binary representation in Table 3.2.*

Table 3.3 A clustered RT-dataset produced from the binary representation in Table 3.2. The attributes *Cluster ID* and *ID* are for reference.

Cluster ID	ID	Gender	Age	Diagnosis codes
1	1	F	77	250.00, 272.4, 278.01, 401.9
1	4	F	78	250.00, 272.4, 401.9, 414.8
1	7	F	80	196.6, 250.00, 272.4, 401.9
1	10	F	75	250.00, 272.4, 401.9, 560.1
2	2	M	71	244.9, 285.1, 530.81
2	5	M	73	244.9, 530.81, 648.01, 661.11
2	8	M	73	244.9, 401.9, 530.81
2	11	M	73	218.0, 244.9, 530.81
3	3	F	46	421.0, 427.31, 584.9
3	6	F	48	285.1, 427.31, 584.9
3	9	F	48	427.31, 584.9, 693.0
3	12	F	49	427.31, 584.9, 995.91

We implemented our approach, which applies the MAS and then the PC method. Our approach is referred to as MASPC. We evaluated the effectiveness and efficiency of

MASPC using two publicly available EHR datasets that contain approximately 26,000 and 53,000 records, respectively. Our results show that MASPC finds clusters that: (I) are compact and well-separated, outperforming three baselines that are founded upon other types of frequent patterns, as well as a baseline following the two-level (hybrid) optimization strategy [15], and (II) preserve correlations between demographics and diagnosis codes and among diagnosis codes. The results also show that MASPC takes less than 6 minutes and scales well with the number of records and number of diagnosis codes in the input RT-dataset.

Thus, our specific contributions can be summarized as follows:

1. We propose the first approach for clustering an RT-dataset comprised of demographics and diagnosis codes.
2. We develop MAS, an algorithm for selecting patterns that help constructing a high-quality clustering of an RT-dataset.
3. We develop an algorithm that clusters an RT-dataset based on the patterns selected by MAS.
4. We evaluated the effectiveness and efficiency of our approach using to EHR datasets.

3.1.3 Chapter organization

The rest of the chapter is organized as follows. Section 3.2 work discusses related work of clustering RT-datasets. Section 3.3 provides the necessary background. Section 3.4 provides a definition of the problem addressed in this chapter. Sections 3.5 and 3.6 present our approach for clustering RT-datasets, as well as baselines for the same task, respectively. Last, Section 3.7 presents our experimental evaluation.

3.2 Related work

In this section, we discuss the methods that are closer to EHR clustering and the problem we study. For extensive surveys on mining EHR data, the reader is referred to [123, 114]. Section 3.2.1 discusses clustering for EHR data, Section 3.2.2 provides a brief overview of pattern-based clustering, and Section 3.2.3 discusses pattern mining on EHR data.

3.2.1 EHR data clustering

We categorize existing methods for clustering EHR data, based on the type of data they are applied to.

3.2.1.1 Demographics

Different from RT-datasets, datasets comprised of a set of demographic attributes are inherently low-dimensional (e.g., they typically include fewer than 20 demographics). Thus, they can be clustered using many existing clustering algorithms, which were discussed in Chapter 2. These include hierarchical (e.g., single-linkage, average-linkage, and complete linkage) [14], partitional (e.g., k-means [61], k-means++ [63], and k-medoids [64]), and density-based (e.g., DBSCAN [7] and OPTICS [77]) algorithms. For example, an interesting recent work [78] applies density-based clustering on patient data. The reason that the aforementioned algorithms are not suitable for clustering an RT-dataset is that their similarity measures cannot capture similarity effectively for set-valued attributes. This is because set-valued attributes (such as the diagnosis codes attribute) are inherently high dimensional and the similarity between high dimensional data records cannot be captured effectively by distance measures used in many existing algorithms, as explained in [55, 15]. This is known as the curse of high dimensionality [7].

3.2.1.2 Diagnosis codes

Similar to RT-datasets, datasets in which each record is comprised of a set of diagnosis codes are inherently high-dimensional (i.e., the domain of a set-valued attribute typically contains thousands of diagnosis codes). Such datasets can be clustered using algorithms developed for set-valued (also referred to as transaction) data, which were discussed in Chapter 2. Examples of such algorithms are CLOPE [53], SCALE [54], ROCK [55], and SV-k-modes [124]. For example, SV-k-modes works similar to k-means, but it uses a set of values in a set-valued attribute as cluster representative, instead of the centroid used in k-means. This algorithm can be applied to datasets that also have single-valued attributes (i.e., RT-datasets), by using centroids as representatives in single-valued attributes. However, SV-k-modes is applicable only to set-valued attributes with a very small domain size (i.e., attributes with 10 to 50 distinct values), due to its exponential time complexity with respect to the domain size of set-valued attributes. Consequently, unlike our approach, SV-k-modes is not suitable to cluster a set-valued attribute comprised of diagnosis codes, whose domain size is in the order of thousands. Examples of interesting works on clustering EHR datasets comprised of diagnosis codes are [125] and [126]. These algorithms [53–55, 125, 126] cannot be used to cluster an RT-dataset, because their similarity measures cannot be applied to single-valued attributes, such as the demographic attributes in an RT-dataset.

3.2.1.3 Other high-dimensional data

There are clustering methods that are applied to a dataset comprised of trajectories, genomic sequences, or text. For example, [127] and [128] focus on clustering trajectory data, in which trajectories represent sequences of diseases, while [129] and [130] study clustering genomic data. The works in [131] and [132] study the topic of clustering medical text. Clearly, the methods proposed in [127–132] cannot be considered as alternatives to our approach, because the data they are applied to have very different semantics compared to those of the attributes in an RT-dataset.

3.2.2 Pattern-based clustering

As discussed in Chapter 2, pattern-based clustering methods employ pattern mining to help subsequent clustering. In the following, we review pattern-based clustering methods, based on the type of data they are applied to.

3.2.2.1 Documents

There are methods aiming to cluster a dataset of documents. Each record of the dataset corresponds to a document which is represented as a bag of words (i.e., multi-set comprised of the words in the document) [133]. These methods generally produce low-quality clusterings, due to the high-dimensionality of the data (i.e., the large number of words) in the bag-of-words representation. Motivated by this limitation, the works in [134–136] use frequent itemsets as representative patterns for documents. The goal of these works is to reduce the dimensionality of the document representation, so as to improve clustering accuracy and efficiency. Our MAS algorithm is similar to the works in [134–136] in that it also uses patterns to deal with the high dimensionality of data (diagnosis codes in our case). The difference is that our algorithm uses maximal-frequent all-confident itemsets (MFAs), in order to capture correlations between diagnosis codes. The use of MFAs leads to more accurate clustering than using frequent itemsets (or even maximal-frequent itemsets [7] that are more effective for clustering than frequent itemsets), as shown in our experiments.

3.2.2.2 Gene expression data

Gene expression data are typically represented as a real-valued matrix, where each row corresponds to a gene and each column to a condition [119]. However, there is a significant difficulty to cluster such a matrix [137] because, only under specific experimental conditions, a group of genes can show the same activation patterns. For handling this difficulty, bi-clustering methods, which simultaneously cluster the rows and columns of the matrix, have been proposed [138]. Bi-clustering methods (e.g., [119, 137, 139, 140, 118]) produce, as clusters, submatrices in which subgroups of genes exhibit highly correlated activities for subgroups of conditions. Some of these methods [118] also employ patterns for bi-clustering, such as *frequent patterns* and *association rules* (see Section 3.3 for details). All bi-clustering methods aim to simultaneously cluster the rows and columns of the matrix. Thus, contrary to our approach, they cannot be used to group records into clusters. Specifically, a row corresponding to a record in an RT-dataset could participate in multiple clusters, if we applied bi-clustering to the binary representation in PC.

3.2.3 Pattern mining on EHR data

As discussed in Chapter 2, several works aim to discover frequent itemsets from EHR data [38–40], while others aim to discover associations on EHR and genomic data [41–44]. Different from these works, our objective is not to discover frequent itemsets or generally associations, but to cluster RT-datasets comprised of demographics and diagnosis codes. Therefore, as part of clustering, we discover maximal-frequent all-confident patterns, referred to as MFAs. MFAs help us construct clusters that contain correlated diagnosis codes, which in turn helps clustering quality.

3.3 Background

In this section, we introduce some preliminary concepts. In particular, Section 3.3.1 discusses the concept of RT-dataset, while Section 3.3.2 discusses the concept of itemset mining. Section 3.3.3 discusses the all-confidence measure and the concept of maximal-frequent all-confident itemsets. Section 3.3.4 discusses the type of clustering algorithm we employ, and Section 3.3.5 discusses clustering quality indices.

Table 3.4 summarizes the acronyms used in the chapter.

3.3.1 RT-datasets

We consider an RT-dataset $\mathcal{D}$, in which every record corresponds to a distinct patient. Each record r in $\mathcal{D}$ is comprised of one or more demographic attributes that can be numerical or categorical, and of a set-valued attribute containing diagnosis codes. Without loss of generality, we assume that the first l attributes in $\mathcal{D}$, denoted with $\mathcal{A}^1, \ldots, \mathcal{A}^l$, are demographic attributes, and the last attribute, $\mathcal{A}^{l+1}$, is a set-valued attribute. The diagnosis codes can be represented in different formats. For example, they can be ICD-9 codes or ICD-10 codes. It is also easy to convert ICD-10 codes into ICD-9 codes, using General

Table 3.4 Acronyms and their full names.

Acronym	Full name
EHR	Electronic Health Record
RT-dataset	Relational Transaction dataset
MAS	Maximal-frequent All-confident pattern Selection
PC	Pattern-based Clusteringx
MASPC	MAS and PC is MASPC
MFA	Maximal-Frequent All-confident itemset
MFI	Maximal-Frequent Itemset
SI	Silhouette Index
CI	Calinski-Harabasz Index
MSPC	Maximal-frequent pattern Selection PC
MSPC$^+$	Maximal-frequent pattern Selection PC with 1-length patterns
MASPC$^+$	Maximal-frequent All-confident pattern Selection PC with 1-length patterns
LOINC	Logical Observation Identifiers Names and Codes

Equivalence Mappings [141]. Extensions to RT-datasets comprised of more than one set-valued attributes are left for future work (see Chapter 6).

The domain size (i.e., the number of distinct values) of an attribute $\mathcal{A}^i$, $i \in [1, l+1]$, is denoted with $|\mathcal{A}^i|$, and the projection of a record r in $\mathcal{D}$ on $\mathcal{A}^i$ is denoted with $r[\mathcal{A}^i]$. For example, $r[\mathcal{A}^{l+1}]$ is the set comprised of all diagnosis codes in record r. For brevity, we will refer to the dataset comprised of the projection of each record of $\mathcal{D}$ on $\mathcal{A}^{l+1}$ as $\tilde{\mathcal{D}}$.

3.3.2 Itemset mining

In the following, we introduce some basic concepts related to itemset mining [7] which are used in our approach.

3.3.2.1 Frequent itemsets and their mining

A subset $I \subseteq \mathcal{A}^{l+1}$ is called an *itemset*. An itemset I may be represented as a set of items $I = \{i_1, i_2, \ldots, i_{|I|}\}$. The number of items in I is denoted with $|I|$ and referred to as the *length* of I. An itemset I' that contains all items of I and potentially other items is a *superitemset* of I, and I is a *subitemset* of I'. We may write $I \subseteq I'$ to denote that I is a subitemset of I'. In our case, each record r of the dataset $\tilde{\mathcal{D}}$ contains an itemset that is comprised of the diagnosis codes contained in r.

The *support* (relative frequency) of an itemset I in the dataset $\tilde{\mathcal{D}}$ is denoted by $sup_{\tilde{\mathcal{D}}}(I)$ and defined as the fraction of records in $\tilde{\mathcal{D}}$ that contain I as a subitemset. Given a support threshold $minSup$, an itemset I is called a *frequent itemset* in $\tilde{\mathcal{D}}$ if $sup_{\tilde{\mathcal{D}}}(I) \geq minSup$. If $sup_{\tilde{\mathcal{D}}}(I) < minSup$, I is called an *infrequent itemset*. The support satisfies the following *downward-closure* property: $sup_{\tilde{\mathcal{D}}}(I) \geq sup_{\tilde{\mathcal{D}}}(I')$, if and only if I is a subitemset of I'. Thus, all subitemsets of a frequent itemset are frequent, and all superitemsets of an infrequent itemset are infrequent. Example 4 illustrates the concepts of a frequent itemset and support, and the downward-closure property.

Example 4. *Consider the dataset in Table 3.5 as $\tilde{\mathcal{D}}$ and the itemsets $I_1 = \{250.00, 272.4, 401.9\}$, $I_2 = \{244.9, 272.4\}$, $I_3 = \{401.9\}$, and $I_4 = \{250.00, 272.4, 401.9\}$. Each of these itemsets corresponds to a different patient. Table 3.6 is a binary representation of Table 3.5, where the diagnosis codes $\{250.00\}$, $\{244.9\}$, $\{272.4\}$, and $\{401.9\}$ are used as features. If $minSup = 0.5$, then $I_1 = \{250.00, 272.4, 401.9\}$ is a frequent itemset, because $sup_{\tilde{\mathcal{D}}}(I_1) = \frac{2}{4} \geq minSup$ (i.e., I_1 appears in at least two out of four records). Due to the downward-closure property, any subitemset of I_1 is also frequent. For instance, the subitemset I_3 of I_1 has $sup_{\tilde{\mathcal{D}}}(I_3) = \frac{3}{4} \geq minSup$ and hence I_3 is also frequent.*

Table 3.5 An example of a dataset $\tilde{\mathcal{D}}$. The attribute *ID* is for reference.

ID	**Diagnosis Codes**
1	250.00, 272.4, 401.9
2	244.9, 272.4
3	401.9
4	250.00, 272.4, 401.9

Table 3.6 The dataset in Table 3.5 represented in a binary format, where the features are the items (diagnosis codes) in the dataset. The attribute *ID* is for reference.

	Items (Diagnosis Codes)			
ID	250.00	244.9	272.4	401.9
1	1	0	1	1
2	0	1	1	0
3	0	0	0	1
4	1	0	1	1

The problem of frequent itemset mining is to find all frequent itemsets in $\tilde{\mathcal{D}}$, for a given threshold $minSup$. There are many algorithms for solving the problem (see [7]), one of which is *FP-growth* [7]. *FP-growth* is actually one of the most efficient algorithms for the problem. It uses an extended prefix-tree (FP-tree) structure to store the collection of itemsets (e.g., the data in Table 3.5) in a compressed form and adopts a divide-and-conquer approach to decompose both the mining tasks and the datasets. This allows pruning the space of possible itemsets, to efficiently find the frequent ones.

3.3.2.2 Maximal frequent itemset and their mining

Although algorithms such as FP-growth can be used to obtain the complete set of frequent itemsets, this set may contain too many patterns, which degrades the quality of clustering (see Section 3.5). Therefore, we instead use the set of *maximal-frequent* itemsets. An itemset I is maximal-frequent, if no superitemset of I is itself frequent. Consider again Example 4. The itemset $\{250.00, 272.4, 401.9\}$ is a maximal-frequent itemset, because its superitemset $\{250.00, 244.9, 272.4, 401.9\}$ is itself not frequent, for $minSup = 0.5$.

3.3.2.3 Association rule and confidence

Given an itemset I, an *association rule* is defined as an implication of the form $X \Rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \emptyset$. The itemsets X and Y are called the *antecedent* and *consequent* of the rule, respectively. The confidence of a rule $X \Rightarrow Y$ in the dataset $\tilde{\mathcal{D}}$ is defined as follows:

$$conf_{\tilde{\mathcal{D}}}(X \Rightarrow Y) = \frac{sup_{\tilde{\mathcal{D}}}(X \cup Y)}{sup_{\tilde{\mathcal{D}}}(X)},$$

where $X \cup Y$ means that both X and Y are present.

Example 5 (Cont'd from Example 4)**.** *Given an itemset $I = \{250.00, 272.4, 401.9\}$, the association rule $\{250.00, 272.4\} \Rightarrow 401.9$ has confidence $\frac{0.5}{0.5} = 1$ in Table 3.5. This implies that all (i.e., 100%) of the records containing $\{250.00, 272.4\}$ also contain 401.9.*

Confidence can be interpreted as an estimate of the probability $P(Y|X)$ (i.e., the probability of finding the consequent of the rule in records containing the antecedent).

3.3.3 All-confidence and Maximal-frequent all-confident itemsets

While confidence is an important measure, it is not suitable for mining itemsets comprised of correlated items. Thus, if we used patterns with high confidence as features in the binary representation constructed PC, we could obtain clusters with diagnosis codes that have no dependence relationships (e.g., diagnosis codes that co-occur by chance). To address this issue, we use patterns that have high *all-confidence* [142] instead.

Before introducing all-confidence, we discuss why confidence cannot be used to capture item correlations. Two items i_1 and i_2 are correlated if they co-occur in a sufficiently large fraction of records. However, we cannot use the confidence (i.e., the probability $P(i_1|i_2)$ or $P(i_2|i_1)$) alone to decide if i_1 and i_2 are correlated. This is because $P(i_2|i_1)$ may be quite different from $P(i_1|i_2)$, as shown in Example 6.

Example 6. *Consider Table 3.7, in which the items (diagnosis codes) $i_1 = O14.90$ and $i_2 = 332.1$ represent "pre-eclampsia" and "secondary parkinsonism", respectively, and* *that two items are correlated when they co-occur in at least $\frac{1}{2}$ of the records of the table. $P(i_2|i_1) = 1 > \frac{1}{2}$ and $P(i_1|i_2) = \frac{1}{3} < \frac{1}{2}$, so it is difficult to use one of the two probabilities to decide if i_1 and i_2 are correlated.*

Table 3.7 Dataset represented in a binary format, in which the features are the items i_1 and i_2. The attribute ID is for reference.

	Items (Diagnosis Codes)	
ID	$i_1 = O14.90$	$i_2 = 332.1$
1	0	1
2	1	1
3	0	1
4	0	0

To address this issue, we can make a worst-case assumption, according to which i_1 and i_2 are correlated when the minimum of the two probabilities $P(i_1|i_2)$ and $P(i_2|i_1)$ is sufficiently large (e.g., at least equal to $\frac{1}{2}$ in Example 3.7). This leads to the *all-confidence* measure [142] defined as follows, for a dataset $\tilde{\mathcal{D}}$ and two itemsets $X = \{i_1\}$ and $Y = \{i_2\}$ (i.e., $|X \cup Y| = 2$):

$$allConf_{\tilde{\mathcal{D}}}(X \cup Y) = \min\{P(Y|X), P(X|Y)\}. \tag{3.1}$$

Eq. (3.1) favors a rule $X \Rightarrow Y$ in which X implies Y, and Y implies X, with sufficiently high probability. In the general case of an itemset $I = X \cup Y$ of length $|X \cup Y| \geq 2$ and a dataset $\tilde{\mathcal{D}}$, all-confidence can be defined as follows:

$$allConf_{\tilde{\mathcal{D}}}(I) = \min_{a_i \in I} \left\{ \frac{sup_{\tilde{\mathcal{D}}}(I)}{sup_{\tilde{\mathcal{D}}}(a_i)} \right\}. \tag{3.2}$$

The right hand side of Eq. (3.2) computes the confidence of the least favorable rule (i.e., the rule $X \Rightarrow I \setminus X$, where $|X| = 1$). Therefore, given a threshold $minAc$, an itemset with $allConf_{\tilde{\mathcal{D}}}(I) \geq minAc$, referred to as *all-confident* itemset, ensures that any itemset $X \subseteq I$ implies any other itemset $I \setminus X$ with probability at least $minAc$ [142]. In other words, the items within I are correlated.

When an itemset is maximal-frequent and all-confident, we refer to it as a *maximal-frequent all-confident itemset* (MFA). Example 7 illustrates an MFA and the fact that its subitemsets are correlated.

Example 7. *Consider the MFA $I = \{250.00, 272.4, 401.9\}$ and a dataset $\tilde{\mathcal{D}}$ corresponding to the* RT*-dataset in Table 3.1 (i.e., the projection of the dataset in Table 3.1 on the* *diagnosis codes attribute). The support of I is* $\frac{4}{12}$*, because it appears in four records out of* 12 *records of the dataset (namely, those with IDs* $1, 4, 7$ *and* 10*). The support of each of the items (diagnosis codes)* $\{250.00\}$*,* $\{272.4\}$ *and* $\{401.9\}$ *in I is also* $\frac{4}{12}$*. Thus, according to Eq. (3.2),* $allConf_{\tilde{\mathcal{D}}}(I) = allConf_{\tilde{\mathcal{D}}}(\{250.00, 272.4, 401.9\}) = \frac{4}{4} = 1$*. Therefore, when we know that the diagnosis code* $X = \{250.00\}$ *appears in a record, we can infer that the diagnosis codes in* $I \setminus X = \{274.2, 401.9\}$ *will appear in the record with probability at least* $minAc = 1$ *(i.e., we are certain that* 250.00 *implies the diagnosis codes* 274.2 *and* 401.9 *with probability* 1*), and the same holds for any other subitemset* X *and* $I \setminus X$*.*

In summary, given a dataset $\tilde{\mathcal{D}}$, a support threshold $minSup$, and an all-confidence threshold $minAc$, an itemset I is called:

- *Frequent itemset*, if $sup_{\tilde{\mathcal{D}}}(I) \geq minSup$.
- *Maximal-frequent itemset* (MFI), if I is frequent and no superitemset of I is itself frequent.
- *Maximal-frequent all-confident itemset* (MFA), if I is maximal-frequent and $allConf_{\tilde{\mathcal{D}}}(I) \geq minAc$.

Note, there are several algorithms for mining MFIs [143], [144], and [145]. In our work, we employ the FPMAX algorithm [145], because it is more efficient than the algorithms in [143] and [144], as explained in [145]. We then construct the set of MFAs by keeping each MFI I with $allConf_{\tilde{\mathcal{D}}}(I) \geq minAc$.

3.3.4 Agglomerative average-linkage hierarchical clustering

The algorithm [14] starts by creating clusters, each containing a different record of the dataset, and it outputs a hierarchy (tree) of clusters. The leaves in the hierarchy correspond to the initial clusters, the root to a single cluster comprised of all records, and each other node corresponds to a cluster of the records in its subtree. The hierarchy is built bottom-up, by merging the two clusters with the smallest average pairwise distance between the records in these clusters. The algorithm can use many different distance measures for measuring the distance between two clusters. The final clusters are produced by selecting a subset of nodes in the hierarchy. The selected nodes correspond to a set of clusters that is a partition of the dataset.

3.3.5 Quality indices

There are various criteria to measure the quality of a clustering. Two popular criteria that have been shown to perform well [146] are the Silhouette Index (SI) and the Calinski-Harabasz Index (CI).

SI [147] measures how well records fit into their clusters and is defined in Eq. (3.3):

$$SI(C) = \frac{1}{N} \cdot \sum_{i=1}^{k} \sum_{r \in c_i} \frac{b(r, c_i) - a(r, c_i)}{\max(a(r, c_i), b(r, c_i))}, \tag{3.3}$$

where $C = \{c_1, \dots, c_k\}$ is a clustering comprised of k clusters, N is the number of records in C, r is a record in a cluster c_i, $a()$ is the average cosine distance of r to all other records in c_i, and $b()$ is the average cosine distance of r to the records of the *neighboring* cluster c_j, $j \neq i$ (i.e., the cluster whose records have the smallest average cosine distance to r). Note that in SI other measures may be used instead of cosine distance [147]. SI takes values in $[-1, 1]$, and values larger than 0 imply a good clustering. Intuitively, a large SI value indicates that clusters are compact and well-separated (i.e., the records in a cluster are closer together compared to records in their neighboring cluster).

CI is a ratio of the between-cluster distance to the within-cluster distance. For a clustering C, CI is defined in Eq. (3.4):

$$CI(C) = \frac{\sum_{c_i \in C} |c_i| \cdot E\left(\hat{c}_i, \hat{\mathcal{D}}\right) / (k-1)}{\sum_{c_i \in C} \sum_{r \in C_i} E\left(r, \hat{c}_i\right) / (N-k)}, \tag{3.4}$$

where $|c_i|$ is the number of records in cluster c_i, $\hat{c}_i$ is the centroid of c_i, $\hat{\mathcal{D}}$ is the centroid of the dataset, and E is the squared Euclidean distance. Large CI values imply a good clustering. That is a clustering with compact and well-separated clusters (i.e., the centroid of a cluster is far from the average centroid and close to the records in the cluster).

3.4 Problem Definition

We now formally define the clustering problem that we aim to solve.

Problem 1. *Given an* RT*-dataset $\mathcal{D}$, a binary representation $\mathcal{B} = \{B_1, ..., B_{|\mathcal{D}|}\}$ of $\mathcal{D}$, and a parameter k, construct a partition $C = \{c_1, \dots, c_k\}$ of $\mathcal{D}$ with maximum $\sum_{i \in [1,k]} \sum_{r, r' \in c_i} cos(B_r, B_{r'})$, where c_i is a cluster and $cos(B_r, B_{r'}) = \frac{B_r \cdot B_{r'}}{||B_r||||B_{r'}||}$ is the cosine* *similarity measure between the records B_r and $B_{r'}$ in $\mathcal{B}$, which correspond to the records r and r', respectively, in c_i.*

The problem takes as input an RT-dataset $\mathcal{D}$, the number of clusters k, and the binary representation of $\mathcal{D}$, and it requires finding a clustering of k clusters for $\mathcal{D}$ such that the records in each cluster are similar. Specifically, it seeks to maximize the total cosine similarity, which is measured on the records of the binary representation of $\mathcal{D}$. The problem is NP-complete (this follows easily from [148, 149]), which justifies the development of heuristics.

3.5 Clustering RT-datasets using MASPC

This section presents our MASPC approach for clustering an RT-dataset, as specified in Problem 1. MASPC applies: (I) the *MAS* algorithm, which discovers maximal-frequent all-confident patterns (MFAs), and (II) the *PC* algorithm, which constructs and clusters the binary representation of the RT-dataset, and produces the clustered RT-dataset.

In the following, we explain the operation of the MAS and PC algorithms.

3.5.1 The MAS algorithm

MAS works in two phases:

(I) **MFA mining**: In this phase, all MFAs are mined from the input RT-dataset.

(II) **Pattern selection**: In this phase, a subset of MFAs that help the subsequent clustering by the PC algorithm to construct clusters of high quality are selected.

We now discuss in detail each phase (see the pseudocode of MAS in Algorithm 1).

MFA mining. In this phase, MAS projects each record of the input RT-dataset on the diagnosis codes attribute (line 1) and then finds all MFAs, using the FPMAX algorithm [145] (line 2). Then, in line 3, the algorithm selects MFAs (maximal-frequent itemsets with confidence at least $minAc$) that are comprised of at least 2 diagnosis codes (i.e., MFAs with length at least 2).

Pattern selection. In this phase, MAS iterates over each MFA I, starting from those with the largest support (lines 5 to 11). If there is no I that has been selected, then I is added into the set of selected MFAs $\mathcal{I}$ and is not considered again (lines 6 to 8). Otherwise, the algorithm checks that each MFA I' in $\mathcal{I}$ does not share diagnosis codes with I and that there are at least $minOv$ records that contain both I and I', where $minOv$ is a user-specified

Algorithm 1 MAS(Maximal-frequent All-confident pattern Selection)

```
Input: Dataset $\mathcal{D}$, Minimum support threshold $minSup$, Minimum all-confidence threshold $minAc$, Minimum pattern overlap threshold $minOv$
Output: Set of MFAs $\mathcal{I}$
    // MFA mining phase
 1: $\tilde{\mathcal{D}} \leftarrow \{r \mid r[\mathcal{A}^{l+1}] \in \mathcal{D}\}$
 2: $MFI \leftarrow \{I \mid sup_{\tilde{\mathcal{D}}}(I) \geq minSup \land \nexists I' \supseteq I : sup_{\tilde{\mathcal{D}}}(I') \geq minSup\}$
 3: $MFA \leftarrow \{I \mid I \in MFI \land allConf_{\tilde{\mathcal{D}}}(I) \geq minAc \land |I| \geq 2\}$
    // Pattern selection phase
 4: $\mathcal{I} \leftarrow \{\}$
 5: for each pattern $I$ in MFA in descending order of support in $\tilde{\mathcal{D}}$ do
 6:     if $\mathcal{I} = \{\}$ then
 7:         $\mathcal{I} \leftarrow \mathcal{I} \cup \{I\}$
 8:         $MFA \leftarrow MFA \setminus \{I\}$
 9:     else if $I \cap I' = \{\}$ OR ($sup_{\tilde{\mathcal{D}}}(I \cup I') \geq minOv$), for each $I'$ in $\mathcal{I}$ then
10:         $\mathcal{I} \leftarrow \mathcal{I} \cup \{I\}$
11:         $MFA \leftarrow MFA \setminus \{I\}$
12:     end if
13: end for
14: return $\mathcal{I}$
```

threshold (line 9). In this case, I is added into $\mathcal{I}$ and not considered again (lines 10 to 11). Last, MAS returns the MFAs in $\mathcal{I}$ (line 14).

A critical step in MAS that affects the quality of the subsequent clustering by PC is the selection of patterns that are added into $\mathcal{I}$. We select patterns of length at least 2 that: (I) occur frequently (i.e., have support at least $minSup$) and thus can be used to measure the similarity of many records, and (II) are comprised of correlated diagnosis codes (i.e., have all-confidence at least $minAc$). We exclude MFAs of length 1 because they are not comprised of correlated diagnosis codes and do not help the quality of clustering, as shown in Section 3.7. We also control the overlap between the selected patterns, by selecting patterns that either do not share diagnosis codes, or patterns that co-occur in at least $minOv$ records. The selection of thresholds $minSup$, $minAc$, and $minOv$ affects the quality and efficiency of clustering, as shown in Section 3.7.

We now provide the motivation for our strategy that controls the overlap between patterns. First, we note that using many patterns that share diagnosis codes would bias the similarity measurement towards diagnosis codes that are contained in multiple patterns, leading to poor clusters. For example, consider two patterns I and I' which have length 10 and share 9 diagnosis codes, and that there are many records containing I or I' but few records which contain all 11 diagnosis codes in $I \cup I'$ and no other diagnosis code. If we include both patterns in the binary representation of the dataset, the records containing I or

I' will be represented by one feature and those containing $I \cup I'$ by two. Clearly, the latter records are not similar with those supporting I or I', based on the binary representation, and are only a few. Thus, they will be clustered with other records dissimilar to them, which leads to a poor clustering. Therefore, we do not select the pattern I' when I' contains diagnosis codes that are also contained in I. On the other hand, when there are many records containing I' and I, then we select I' because these records can be added into a separate cluster that contains no other record. This cluster would be of high quality, because the records it contains are similar, since they share 9 out of their 11 diagnosis codes.

The worst-case time complexity of the MAS algorithm is exponential in $|\mathcal{A}^{l+1}|$, the domain size of the diagnosis codes attribute. The bottleneck is the mining of maximal frequent itemsets, which in the worst case takes exponential time in $|\mathcal{A}^{l+1}|$. This is because, in the worst case, the number of maximal frequent itemsets is exponential in $|\mathcal{A}^{l+1}|$ [150]. All other operations of MAS take polynomial time. Despite the exponential worst-case complexity of mining maximal frequent itemsets, the task is performed in reasonable time (within seconds or minutes) in practice, based on algorithms such as FPMAX [145]. Thus, MAS is scalable in practice (see Section 3.7).

3.5.2 The PC algorithm

In the following, we present the PC algorithm. PC gets as input an RT-dataset, the patterns selected by MAS, as well as the number of clusters k, and it works in two phases:

(I) **Binary representation construction**: In this phase, the binary representation of the input RT-dataset is constructed, based on the MFAs that were previously selected by MAS.

(II) **Hierarchical clustering**: In this phase, the binary representation is clustered by hierarchical clustering and the clustered RT-dataset is constructed based on the result.

We now discuss each phase in detail.

Binary representation construction. In this phase, the algorithm constructs the binary representation $\mathcal{M}$ of the input RT-dataset $\mathcal{D}$. Recall from Section 3.1 that there is a one-to-one correspondence between the records in $\mathcal{M}$ and $\mathcal{D}$, and that the features (columns) of $\mathcal{M}$ correspond to the discretized values of the numerical demographic attributes, the values of the categorical demographic attributes, and the input MFAs. The discretization of each numerical demographic attribute is performed by applying the k-means++ algorithm [63]; see Section 2.2.2.2. The number of discretized values in the attribute is selected automatically by the Elbow method [151]. The Elbow method is an effective heuristic [151], which, in our method, applies k-means++ with different values of k and automatically selects the k value that leads to the best result with respect to the total within-cluster sum of squares measure. We use the Elbow method, because it allows discretization with no parameters. Alternatively, the users can discretize the numerical demographic attributes using other methods [152] and provide the discretized values as input to PC.

Hierarchical clustering. In this phase, the algorithm applies agglomerative average-linkage hierarchical clustering (see Section 3.3.4) with cosine similarity to the binary representation $\mathcal{M}$. This partitions $\mathcal{M}$ into k clusters, where k is the input parameter of PC. We used hierarchical clustering, because it was shown to be effective for clustering binary-represented and EHR data [127], while it does not need to construct cluster representatives, which is problematic for set-valued data. We used cosine similarity, because it was shown to be effective for clustering binary-represented data with many features such as $\mathcal{M}$ [122]. Note, there may be records in $\mathcal{D}$ that do not contain the diagnosis codes in any MFA in $\mathcal{M}$. These records are referred to as unclustered records, and intuitively they are dissimilar to all other records which are represented in $\mathcal{M}$. Thus, they are removed from $\mathcal{M}$ (i.e., these records are not added into any clusters), because this helps the quality of clustering as explained in [47]. There are, however, several alternative strategies for dealing with unclustered records, which we discuss in Section 3.7 and Chapter 6. Last, PC constructs the clustered RT-dataset by clustering the records in the dataset according to the way they were clustered in the binary representation $\mathcal{M}$ (i.e., constructing a cluster with all records of the RT-dataset that are in the same cluster after agglomerative hierarchical-clustering has been applied to $\mathcal{M}$).

The time complexity of PC is $O(|\mathcal{D}| \cdot |\mathcal{I}| + |\mathcal{D}|^2)$, with the first and second term corresponding to step 1 and 2, respectively. Assuming $|\mathcal{D}| \geq |\mathcal{I}|$, the bottleneck is the execution of the agglomerative average-linkage hierarchical clustering algorithm which has a complexity of $O(|\mathcal{D}|^2)$, when implemented based on the efficient nearest-neighbor-chain algorithm [153].

3.6 Baselines for clustering RT-datasets

This section presents four baselines for clustering an RT-dataset. The first baseline is referred to as HYBRID. HYBRID is inspired by the two-level (hybrid) optimization strategy of Poulis et al. [15]. It works in two phases:

1. **Binary representation construction**: This phase is similar to the binary representation construction phase of PC; the difference is that each diagnosis code in $\mathcal{D}$ is

considered as a feature (column in the binary representation), whereas PC considers each MFA as a feature instead. The resultant binary-represented dataset is denoted with $\mathcal{M}_{Hybrid}$.

2. **k-Medoids clustering**: In this phase, $\mathcal{M}_{Hybrid}$ is partitioned into p clusters, by applying an efficient implementation of the k-medoids algorithm [64] with cosine similarity multiple times. First, we apply the algorithm of Park et al. [64] to the projection of $\mathcal{M}_{Hybrid}$ on the features (columns) corresponding to demographics, setting the number of clusters to a threshold a. Then, we apply the algorithm to each of the resultant a clusters separately. Specifically, we consider the projection of $\mathcal{M}_{Hybrid}$ on the features corresponding to diagnosis codes of the cluster and partition it into b smaller clusters. Therefore, at the end of this phase, $a \cdot b = p$ clusters are created.

Besides HYBRID, we considered as baselines three variations of MASPC, whose differences from MASPC are as follows:

- MSPC (for Maximal-frequent pattern Selection PC): It considers maximal-frequent patterns of length at least 2 instead of MFAs of length at least 2. That is, in line 3 of MAS, it uses $MFA \leftarrow \{I \mid I \in MFI \wedge |I| \geq 2\}$.

- MSPC$^+$ (for Maximal-frequent pattern Selection PC with 1-length patterns): It includes all maximal-frequent patterns. That is, in line 3 of MAS, it uses $MFA \leftarrow \{I \mid I \in MFI\}$.

- MASPC$^+$ (for Maximal-frequent All-confident pattern Selection PC with 1-length patterns): In line 3 of MAS, it also includes MFAs of length 1 (i.e., comprised of a single diagnosis code).

The reason we used these variations of MASPC was to examine the impact of our choice to use MFAs of length at least 2 vs.: (I) maximal-frequent patterns of length at least 2, (II) maximal-frequent patterns of any length, and (III) MFAs of any length.

3.7 Experimental evaluation

In the following section, we evaluate our approach in terms of compactness and separation of clustering, as well as in terms of runtime.

3.7.1 Datasets

We used two publicly available RT-datasets, comprised of demographics and diagnosis codes:

- VERMONT [154]. The dataset contains de-identified inpatient discharge data in Vermont during 2015. Specifically, each row in the dataset describes an individual emergency department encounter and is comprised of the demographic attributes *Age* and *Gender*, and of the set-valued attribute *Diagnosis-codes*. This dataset was used in [155, 156].
- INFORMS [157]. The dataset contains de-identified patient data and was used in the Informs Data Mining Contest 2008. Specifically, each row in the dataset is the join of the Demographics and Conditions tables from the training section during 2004 and contains the demographics {*Year of birth, Gender, Race, Poverty*} and the set-valued attribute *Diagnosis-codes*. This dataset was used in [15, 158, 159].

The characteristics of the datasets we used are shown in Table 3.8.

Table 3.8 Description of the datasets used in our experiments.

Dataset	**$\#$ of records $\|D\|$**	**$\#$ of demographics**	**$\#$ of distinct diagnosis codes $\|\mathcal{A}^{l+1}\|$**	**Max $\#$ of diagnosis codes/record**	**Avg $\#$ of diagnosis codes/record**
VERMONT	$52,789$	2	$13,521$	20	10.44
INFORMS	$26,304$	4	558	32	3.54

3.7.2 Experimental setup

We compared our approach against the four baseline methods in Section 3.6, because no existing clustering algorithms can be applied to an RT-dataset comprised of demographics and diagnosis codes (see Section 3.2).

The quality of clusters is measured using two popular quality indices; SI and CI (see Section 3.3.5). This is because the datasets we use do not have information about a ground truth clustering, and we are not aware of RT-datasets comprised of demographics and diagnosis codes that come with such a clustering. To demonstrate the efficiency of MASPC, we compared it against HYBRID. We excluded the other baselines from the efficiency comparison because their runtime was similar to that of MASPC.

The default parameter values that are used in our experiments are shown in Table 3.9. They were selected heuristically with the intention to cover different cases regarding the effectiveness of our approach. The only exception was k, which was selected using the Elbow method for our approach. For each other parameter, we tried a large range of values and then selected different values among the range, with the intention not to favor our method or any other tested method. For example, we used a default value for $minSup$ that leads to the best result in VERMONT and the worst result in INFORMS, and a default parameter for $minOv$ and $minAc$ that leads to a middling result in both VERMONT and INFORMS. This heuristic procedure is clearly inefficient, and error-prone (many values need to be tested and good values may be missed). However, it has been employed by other pattern-based clustering methods [47]. This is because there does not exist an automated methodology for selecting these data-dependent parameters in an optimal way, to the best of our knowledge. We further discuss parameter selection in Section 6.2.1.3 of Chapter 6. We emphasize that, in each of our experiments, our approach outperformed all others across all tested values. Thus, it was not possible to tune the other approaches, by choosing their favorable parameters, so that they outperform our MASPC algorithm.

Unless otherwise stated, each parameter was configured using its default value. Also, MASPC and the baselines use the same parameters where possible. That is, MASPC uses the same: (I) k with MSPC, MSPC⁺, MASPC⁺, and HYBRID, (II) $minSup$ and $minOv$ with MSPC, MSPC⁺, and MASPC⁺, and (III) $minAc$ with MASPC⁺. Furthermore, following [47], we applied HYBRID and the variations of MASPC to the datasets produced by MASPC. These datasets do not contain unclustered records (i.e., they only contain records with the diagnosis codes of at least one MFA). By construction, the variations of MASPC cluster all records of these datasets, since they select a superset of the patterns selected by MASPC (i.e., they do not lead to new unclustered records). We have also evaluated all tested methods following a different strategy of dealing with unclustered records. In this strategy, the methods are applied to the entire dataset and the unclustered records, if any, form a single cluster (see Section 6.2.1 of Chapter 6 for details).

We have implemented all algorithms in Python 3, and we ran all experiments on an Intel i7 with 1.90 GHz and 16 GB of RAM. The source code is available at https://bitbucket.org/EHR_Clustering/maspc/src/master/.

Table 3.9 The default values for parameters for each dataset. $minSup$ is the minimum support threshold, $minAc$ is the minimum all-confidence threshold, $minOv$ is the minimum pattern overlap threshold, and k is the desired number of clusters. a and b are the parameters of HYBRID, selected as the best pair of values in terms of quality, among the pairs whose product equals k.

Dataset	minSup %	minAc	minOv	k	a	b
VERMONT	1	0.1	40	15	3	5
INFORMS	0.7	0.11	10	4	2	2

3.7.3 Clustering quality measurement

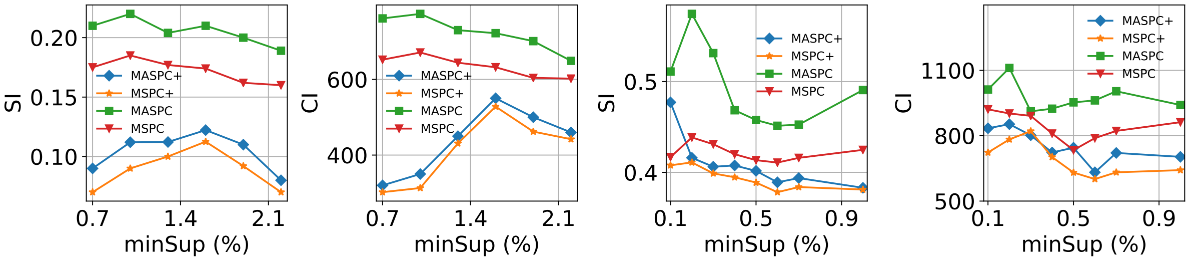
In this section, we show the superiority of MASPC over its 3 variations and HYBRID in terms of being able to construct a high-quality clustering, comprised of compact and well-separated clusters. We consider the impact of parameters $minSup$, $minAc$, $minOv$, and k. We omit HYBRID from the results of all experiments in which it performed much worse than all other methods.

3.7.3.1 Impact of minimum support threshold $minSup$

Figs. 3.1a, 3.1b, 3.1c and 3.1d show that MASPC outperforms the baselines, for all tested $minSup$ values, with respect to SI and CI, being able to find a more meaningful clustering than them. Specifically, the SI scores of MASPC were on average at least 17% and up to 130% better than those of the baselines and similar trends were observed for the CI scores. The reason is that MASPC does not consider a large number of maximal-frequent patterns that have length 1 and/or $allConf < minAc$ (see Table A.1 and Table A.2 of Appendix A for VERMONT and INFORMS, respectively). Thus, it creates a binary representation with fewer columns, compared to the binary representations created by baselines, which leads to a better clustering. Since MASPC uses fewer patterns to construct clusters with correlated and frequent diagnosis codes, which is a novel and practically important feature of our approach, it led to unclustered records. This is because: (I) It uses only all-confident frequent patterns (e.g., see in Tables A.1 and A.2 of Appendix A that MASPC uses on average 11% fewer patterns than MSPC). (II) It uses only MFAs of length larger than 1 (e.g., see in Tables A.1 and A.2 of Appendix A that MASPC uses on average 51% fewer patterns than MASPC$^+$). The large number of length-1 patterns in both datasets compared to those of length at least 2 is attributed to the sparsity of the datasets. Due to I and II, a large number of records that do not support any MFA are unclustered (see Table A.3 and Table A.4 of Appendix A for VERMONT and INFORMS, respectively). Note, a larger $minSup$ means that patterns need to be appear more frequently, which results in fewer selected patterns, but also the patterns are shorter so they have less chance to overlap, which results in more selected patterns. Thus, the number of unclustered records may increase or decrease as $minSup$ gets larger.

3.7.3.2 Impact of minimum all-confidence threshold $minAc$

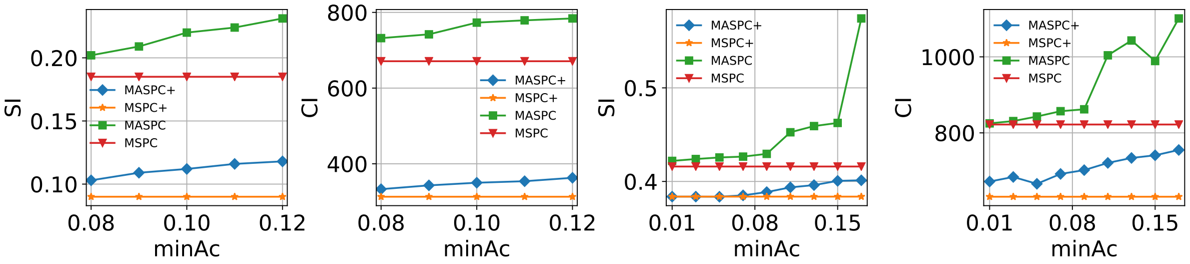
Figs. 3.2a, 3.2b, 3.2c and 3.2d show that MASPC outperforms MASPC$^+$, for all tested $minAc$ values, with respect to SI and CI. This suggests that excluding MFAs of length 1 as MASPC does helps clustering. Specifically, the SI scores of MASPC were on



(a) VERMONT (b) VERMONT (c) INFORMS (d) INFORMS

Fig. 3.1 SI and CI scores vs. minimum support threshold $minSup$.

average at least 95% and up to 119% better than those of MASPC⁺ and similar trends were observed for the CI scores. The baselines MSPC and MSPC⁺, whose scores are not affected by $minAc$, were also much worse than our approach. Note that increasing $minAc$ improves SI and CI, since some patterns comprised of less correlated diagnosis codes are not selected for MASPC and MASPC⁺ that are based on MFAs. Since MASPC and MASPC⁺ select fewer patterns as $minAc$ increases (see Tables A.5 and A.6 of Appendix A), they create a binary representation with fewer columns that leads to a clustering of higher quality. A larger $minAc$ leads to more unclustered records (see Tables A.7 and A.8 of Appendix A). This is because there are fewer MFAs, so there are more records that do not support any MFA.

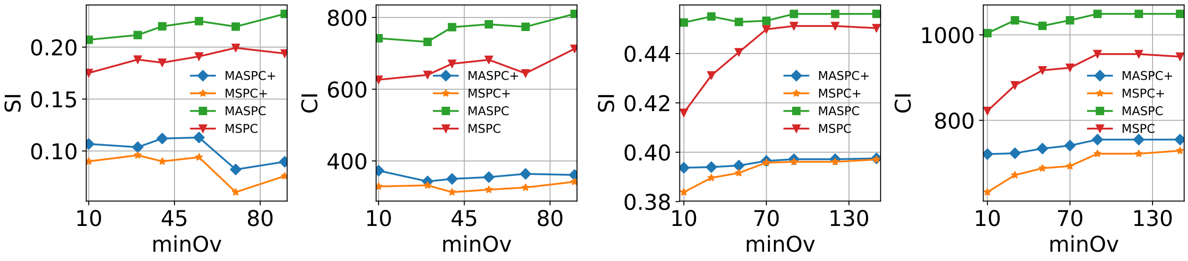


(a) VERMONT (b) VERMONT (c) INFORMS (d) INFORMS

Fig. 3.2 SI and CI scores vs. minimum all-confidence threshold $minAc$.

3.7.3.3 Impact of minimum pattern overlap threshold $minOv$

Figs. 3.3a, 3.3b, 3.3c and 3.3d show that MASPC outperforms the baselines, for all tested $minOv$ values, with respect to SI and CI. Specifically, the SI scores of MASPC were on average at least 16% and up to 117% better than those of the baselines and similar trends were observed for the CI scores. Note that increasing $minOv$ improved the clustering result by avoiding to consider patterns that share diagnosis codes and appear



(a) VERMONT (b) VERMONT (c) INFORMS (d) INFORMS

Fig. 3.3 SI and CI scores vs. minimum pattern overlap threshold $minOv$.

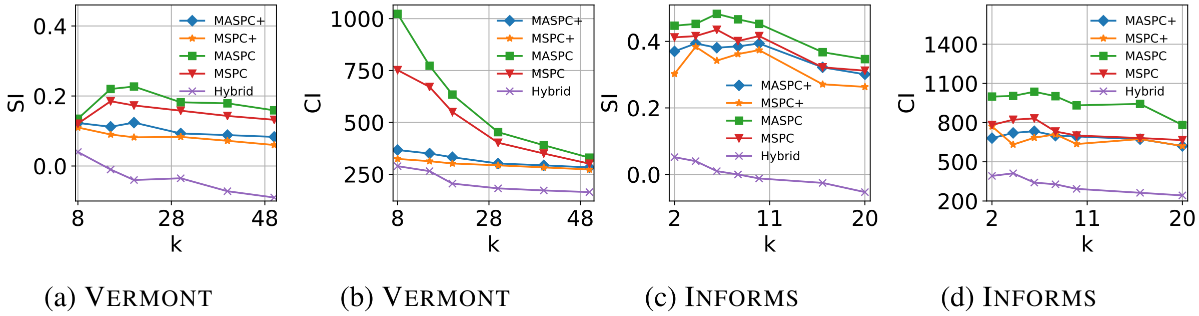
in few ($< minOv$) number of records. This supports our design choice to control the presence of such patterns with the parameter $minOv$ (see Section 3.5). Note that a larger $minOv$ results in fewer patterns (see Tables A.9 and A.10 of Appendix A), because it is more difficult for a pattern to co-occur in a large number (at least $minOv$) of records. Since MASPC uses a small number of patterns, to be able to construct clusters with correlated diagnosis codes, it results in unclustered records (see Tables A.11 and A.12 of Appendix A). The number of unclustered records increases with $minOv$, because there are fewer selected patterns and hence there are more records that do not support any of them.

3.7.3.4 Impact of number of clusters k

Figs. 3.4a, 3.4b, 3.4c and 3.4d show that MASPC outperforms all four baselines (recall that we included HYBRID in these experiments, because it uses the parameter k), for all tested k values, with respect to SI and CI. Specifically, the SI scores of MASPC were on average at least 23% and up to 843% better than those of the baselines, and similar trends were observed for CI. Note that HYBRID was the worst among the baselines. This is attributed to two facts: (I) There are many diagnosis codes (i.e, 13521), which are used as features (columns) in the binary representation. This in turn makes clustering difficult due to the curse of dimensionality. (II) The HYBRID algorithm does not take into account correlations between demographics and diagnosis. This is because it clusters first demographics, and then diagnosis codes within each cluster, and records that are similar with respect to demographics are often dissimilar with respect to diagnosis codes.

3.7.3.5 Clustering quality when unclustered records are clustered together

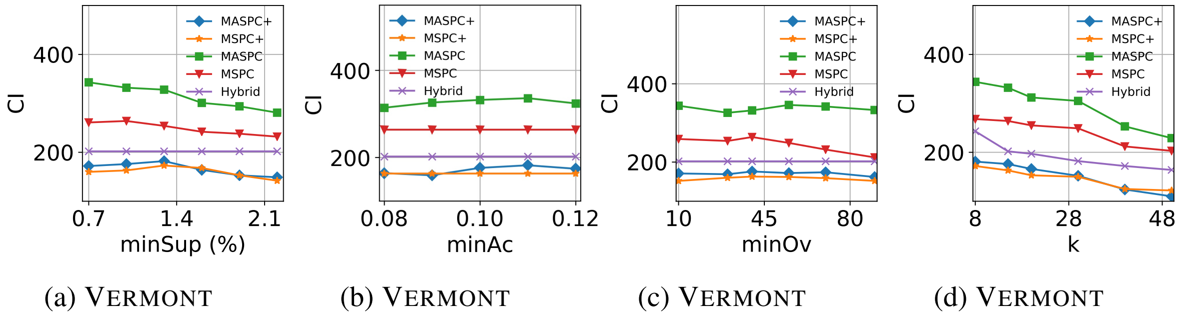
We report results for a different strategy of dealing with unclustered records, in which all unclustered records are clustered together (see Chapter 6 for further discussion of the strategy). We report a small subset of our experiments using VERMONT and INFORMS for CI. Other experiments were qualitatively similar to those we present and therefore



(a) VERMONT (b) VERMONT (c) INFORMS (d) INFORMS

Fig. 3.4 SI and CI scores vs. number of clusters k

they have been omitted. As can be seen in Figs. 3.5 and 3.6, MASPC outperformed all other methods, for all tested values of the parameters $minSup$, $minAc$, $minOv$, and k. For example, in Fig. 3.5a, the CI scores for MASPC were on average at least 26% and up to 96% better than those of the other methods. Similarly, the CI scores for MASPC in Fig. 3.5c were on average at least 37% and up to 113% better than those of the other methods. This is attributed to: (I) the pattern selection strategy of MASPC, which selects patterns that lead to higher quality clusters than those selected by its variations, and (II) the clustering strategy of MASPC, which outperforms the strategy of HYBRID that treats demographics and diagnosis codes separately.

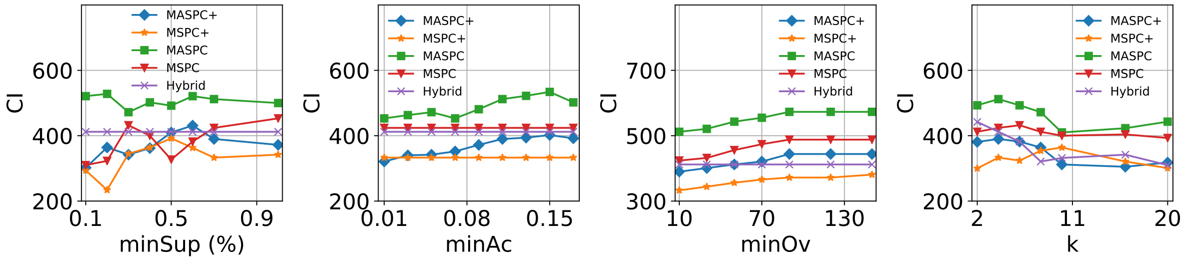


(a) VERMONT (b) VERMONT (c) VERMONT (d) VERMONT

Fig. 3.5 CI scores vs. $minSup$, $minAc$, $minOv$, and k for VERMONT when unclustered records are clustered together.

3.7.4 Efficiency of computation

We evaluate the runtime of MASPC and Hybrid, for varying $minSup$, $minAc$, and $minOv$. We do not report the impact of k because increasing k did not substantially affect the runtime of MASPC (since hierarchical clustering builds the entire dendrogram for any k). However, the runtime of MASPC was always lower than that of HYBRID. We also omit the evaluation of the variations of MASPC because their running time is very similar to that of MASPC.



(a) INFORMS (b) INFORMS (c) INFORMS (d) INFORMS

Fig. 3.6 CI scores vs. $minSup$, $minAc$, $minOv$, and k for INFORMS when unclustered records are clustered together.

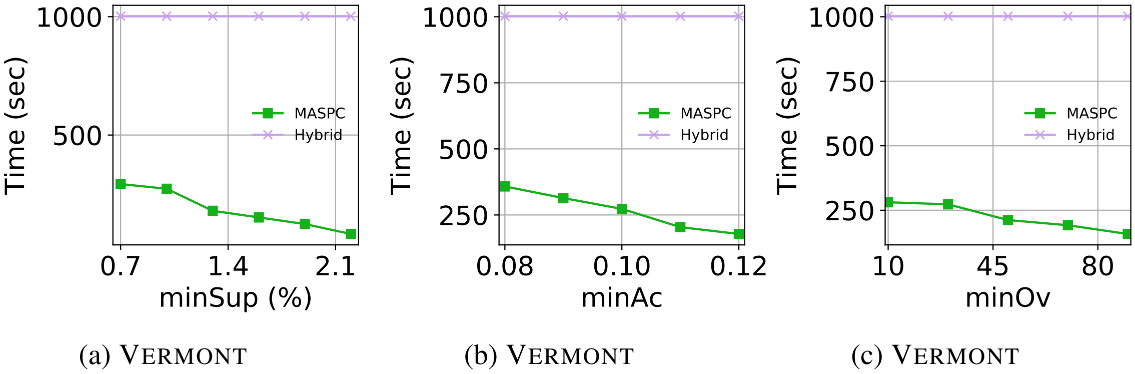
3.7.4.1 Impact of thresholds $minSup$, $minAc$, and $minOv$

Fig. 3.7a shows that MASPC required significantly less time than HYBRID, for all tested values of $minSup$, being up to 12.22 times faster. This is because HYBRID executes k-medoids multiple times, once for each cluster. On the other hand MASPC is applied once and to a binary representation of the RT-dataset that contains a relatively small number of features. Note also that MASPC takes substantially less time as $minSup$ increases. This is expected because fewer MFAs are selected, which leads to a smaller binary representation and hence faster clustering. For the same reason, MASPC is faster than HYBRID and takes less time as $minAc$ or $minOv$ increases (see Figs. 3.7b and 3.7c). Specifically, MASPC was up to 5.63 and 6.34 times faster than HYBRID as $minAc$ and $minOv$ increases respectively. The results for INFORMS were quantitatively similar (omitted).

Recall that MASPC applies the MAS algorithm, to select MFAs, and then the PC algorithm, to perform clustering. Clearly, the runtime of each of these two algorithms is also affected by the thresholds $minSup$, $minAc$ and $minOv$. On average, 44% of the time required by MASPC is spent by MAS and the remaining by PC (see Appendix A.7 for details).

3.7.4.2 Impact of dataset size $|\mathcal{D}|$

Fig. 3.8a shows the runtime of MASPC and HYBRID for increasingly larger random subsets of the VERMONT dataset. MASPC scales better with $|\mathcal{D}|$ compared to HYBRID, being on average 3.18 and up to 5.8 times faster. As can be seen in Appendix A.8, 43% of the time of MASPC is spent by MAS and the remaining by PC. Note also that MASPC scales linearly with $|\mathcal{D}|$, which makes it suitable for clustering datasets with a large number of records. On the other hand, HYBRID is not scalable, as it needs to apply the k-medoids



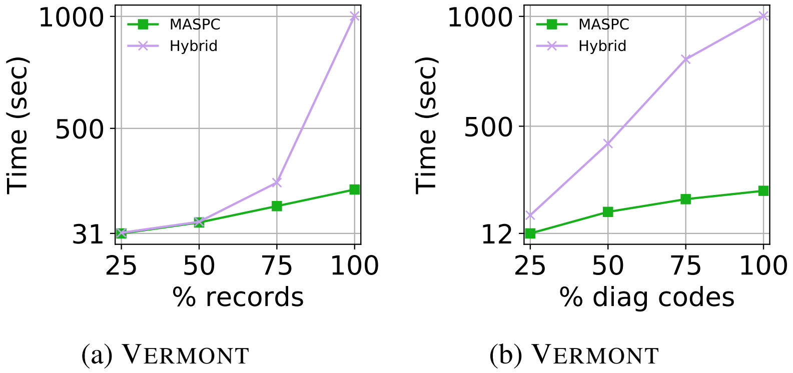
(a) VERMONT (b) VERMONT (c) VERMONT

Fig. 3.7 Runtime vs. (a) $minSup$, (b) $minAc$, and (c) $minOv$.

algorithm, whose time complexity is quadratic with respect to the dataset size, multiple times.

3.7.4.3 Impact of domain size $|\mathcal{A}^{l+1}|$

Fig. 3.8b shows the runtime of MASPC and HYBRID for an increasingly larger percentage of diagnosis codes of VERMONT. As can be seen in Appendix A.9, 44% of the time of MASPC is spent by MAS and the remaining by PC. MASPC scales better than HYBRID, being on average 4.6 and up to 5.8 times faster. Note also that MASPC scales linearly with the percentage of diagnosis codes. On the other hand, HYBRID is not scalable because of the larger datasets (clusters) to which k-medoids is applied.

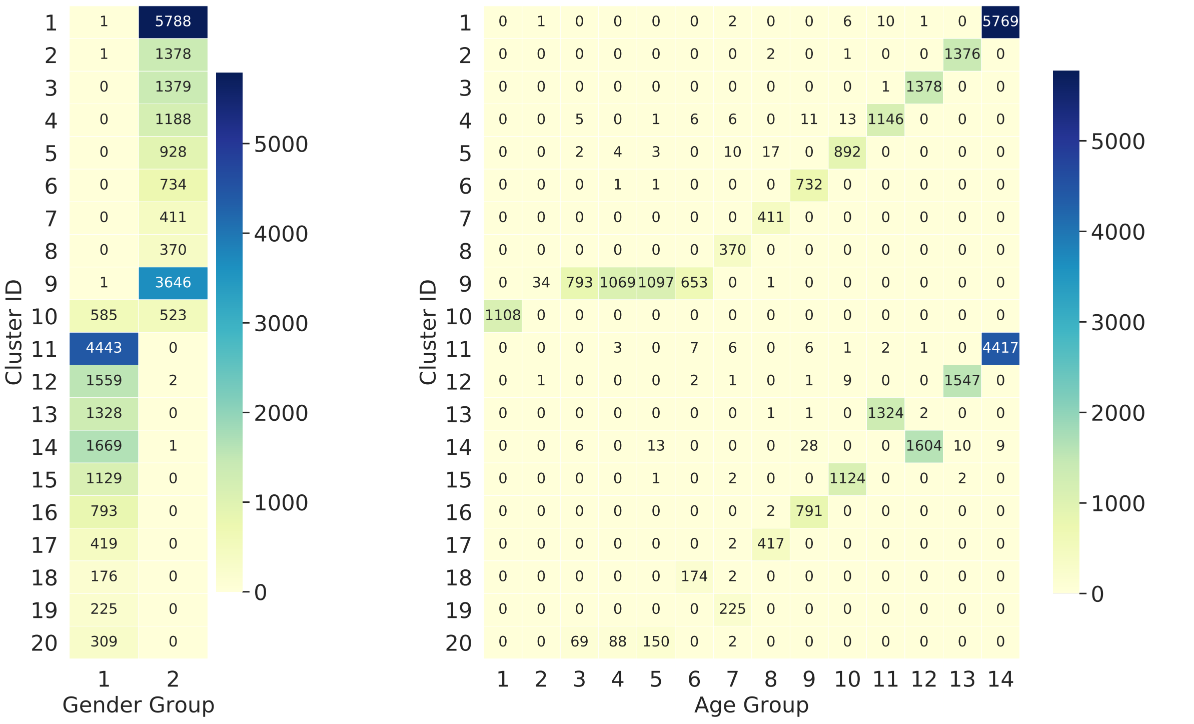


(a) VERMONT (b) VERMONT

Fig. 3.8 Runtime vs. (a) dataset size, and (b) number of diagnosis codes.

3.7.5 Overview of demographics and patterns in clusters

In this section, we examine the clusters constructed by MASPC when applied to the VERMONT dataset with $k = 20$. We demonstrate that the created clusters are compact and well-separated with respect to demographics. We also show that the clusters allow finding correlations between diagnosis codes and between diagnosis codes and demographics that have been documented in the medical literature. We omit the results for the smaller INFORMS dataset, because they were qualitatively similar.



Cluster ID	1	2
1	1	5788
2	1	1378
3	0	1379
4	0	1188
5	0	928
6	0	734
7	0	411
8	0	370
9	1	3646
10	585	523
11	4443	0
12	1559	2
13	1328	0
14	1669	1
15	1129	0
16	793	0
17	419	0
18	176	0
19	225	0
20	309	0

(a)

Cluster ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	1	0	0	0	0	2	0	0	6	10	1	0	5769
2	0	0	0	0	0	0	0	2	0	1	0	0	1376	0
3	0	0	0	0	0	0	0	0	0	0	1	1378	0	0
4	0	0	5	0	1	6	6	0	11	13	1146	0	0	0
5	0	0	2	4	3	0	10	17	0	892	0	0	0	0
6	0	0	0	1	1	0	0	0	732	0	0	0	0	0
7	0	0	0	0	0	0	0	411	0	0	0	0	0	0
8	0	0	0	0	0	0	370	0	0	0	0	0	0	0
9	0	34	793	1069	1097	653	0	1	0	0	0	0	0	0
10	1108	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	3	0	7	6	0	6	1	2	1	0	4417
12	0	1	0	0	0	2	1	0	1	9	0	0	1547	0
13	0	0	0	0	0	0	0	1	1	0	1324	2	0	0
14	0	0	6	0	13	0	0	0	28	0	0	1604	10	9
15	0	0	0	0	1	0	2	0	0	1124	0	0	2	0
16	0	0	0	0	0	0	0	2	791	0	0	0	0	0
17	0	0	0	0	0	0	2	417	0	0	0	0	0	0
18	0	0	0	0	0	174	2	0	0	0	0	0	0	0
19	0	0	0	0	0	0	225	0	0	0	0	0	0	0
20	0	0	69	88	150	0	2	0	0	0	0	0	0	0

(b)

Gender Group	Value
1	Male
2	Female

(c)

	Age Group													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Value	< 1	1-17	18-24	25-29	30-34	35-39	40-44	45-49	50-54	55-59	60-64	65-69	70-74	≥ 75

(d)

Fig. 3.9 Heat Map for the clustering of the VERMONT dataset for: (a) *Gender*, and (b) *Age*. The x axis shows different groups for *Gender* or *Age*. The correspondence between groups for *Gender* and *Age* and their values is shown in (c) and (d), respectively. The number of records in each cluster that belong to a certain Gender or Age Group are shown as numbers in each cell of a heat map.

3.7.5.1 Demographics in the clusters

Fig. 3.9 shows detailed information about the demographics of the patients in each cluster. For example, in cluster with *Cluster ID* 1, most patients ($\frac{5788}{5799}$) are female (see the heat map in Fig. 3.9a and Table 3.9c) and have age over 75 (see the heat map in Fig. 3.9b and Table 3.9d). Note that the clusters are compact and well-separated with respect to demographics, which allows meaningful analyses based on demographics. This is encouraging because MASPC does not specifically optimize the similarity of records with respect to demographics, as it aims to create clusters with records that are similar with respect to both demographics and diagnosis codes. Note that each of the clusters with *Cluster ID*s 1 to 9 in Fig. 3.9 contains mostly female patients of a different age group. Similarly, each of the clusters with *Cluster ID*s 11 to 20 contains mostly male patients of a different age group. Last, the cluster with *Cluster ID* 10 contains both male and female patients of age < 1.

3.7.5.2 Top-3 most frequent MFAs in each cluster

Table 3.10a shows the top-3 most frequent MFAs in each cluster, and Table 3.10b shows the clusters that each of these MFAs is contained in. Note, 27 out of the 41 distinct MFAs appear in one cluster, while 9 (respectively, 5) MFAs appear in 2 (respectively, 3) clusters. Thus, the clusters are relatively well separated with respect to these patterns. Furthermore, the patterns that appear in more than one cluster are generally comprised of diseases that affect a large number of patients. Moreover, the MFAs are comprised of correlated diseases, as we discuss later in this section.

3.7.5.3 Correlations among diagnosis codes in MFAs

Table 3.11 shows 9 representative MFAs that are contained in the clusters produced by MASPC. It can be seen that the diagnosis codes in each MFA are indeed correlated, and this is supported by one or more indicative papers from the medical literature. For example, the first pattern $\{491.21, V46.2\}$ contains a diagnosis code for a type of bronchitis for which supplementing oxygen is a common treatment. Similarly, the fourth pattern $\{327.23, 278.01\}$ contains the diagnosis codes of “obstructive sleep apnea (adult) (pediatric)” and “morbid obesity”, which are correlated, since obstructive sleep apnea is a common comorbidity in obese patients. These results suggest that using all-confidence is an effective way of finding patterns with correlated diagnosis codes.

3.7.5.4 Correlations between diagnosis codes in MFAs and demographics

Table 3.12 shows 6 representative MFAs that are contained in the clusters produced by MASPC. It can be seen that the diagnosis codes in each MFA are indeed correlated, and that this is supported by one or more indicative papers from the medical literature. In addition, the MFAs are correlated with demographics, and these correlations are also

Table 3.10 (a) Top-3 most frequent MFAs in each cluster constructed by MASPC (see Table A.16 in Appendix A for the full name of each ICD code). (b) The MFAs in Table 3.10a and the clusters they are contained in.

Cluster	MFAs		
ID	top-1 frequent	top-2 frequent	top-3 frequent
1	{E78.5,I25.10}	{V49.86,V58.66}	{V49.86,V66.7}
2	{272.4,401.9,530.81}	{491.21,V46.2}	{E78.5,K21.9}
3	{250.00,V58.67}	{278.01,327.23}	{038.9,995.91}
4	{244.9,530.81}	{E78.5,I25.10}	{275.2,276.8}
5	{I10,K21.9}	{491.21,V46.2}	{I10,Z87.891}
6	{305.1,311}	{300.00,V58.69}	{E78.5,I10}
7	{250.00,278.00}	{278.00,530.81}	{305.1,496}
8	{275.2,276.8}	{278.00,311}	{428.0,496}
9	{645.11,V27.0}	{664.11,V27.0}	{659.71,V27.0}
10	{V05.3,V30.00}	{041.49,599.0}	{V49.86,V66.7}
11	{I25.10,Z79.82}	{038.9,995.91}	{I12.9,N18.3}
12	{E78.5,I25.10}	{414.01,496}	{428.0,584.9}
13	{250.60,357.2}	{E78.5,I10}	{272.4,401.9,530.81}
14	{I10,Z87.891}	{250.00,V58.67}	{I10,K21.9}
15	{412,V58.66}	{305.1,530.81}	{250.00,V58.67}
16	{I10,K21.9}	{300.00,305.1}	{250.00,584.9}
17	{E78.5,I10}	{278.01,327.23}	{403.90,414.01}
18	{272.4,V58.67}	{305.1,311}	{305.1,V58.69}
19	{311,338.29}	{305.1,V58.69}	{272.4,V58.67}
20	{311,V58.69}	{305.1,338.29}	{038.9,995.91}

(a)

MFAs	Cluster ID	MFAs	Cluster ID
{038.9,995.91}	3,11,20	{311,V58.69}	20
{041.49,599.0}	10	{403.90,414.01}	17
{244.9,530.81}	4	{412,V58.66}	15
{250.00,278.00}	7	{414.01,496}	12
{250.60,357.2}	13	{428.0,496}	8
{250.00,584.9}	16	{428.0,584.9}	12
{250.00,V58.67}	3,14,15	{491.21,V46.2}	2,5
{275.2,276.8}	4,8	{645.11,V27.0}	9
{272.4,401.9,530.81}	2,13	{659.71,V27.0}	9
{272.4,V58.67}	18,19	{664.11,V27.0}	9
{278.00,311}	8	{E78.5,I10}	6,13,17
{278.00,530.81}	7	{E78.5,I25.10}	1,4,12
{278.01,327.23}	3,17	{E78.5,K21.9}	2
{300.00,305.1}	16	{I10,K21.9}	5,15,16
{300.00,V58.69}	6	{I10,Z87.891}	5,14
{305.1,311}	6,18	{I12.9,N18.3}	11
{305.1,338.29}	20	{I25.10,Z79.82}	11
{305.1,496}	7	{V05.3,V30.00}	10
{305.1,530.81}	15	{V49.86,V58.66}	1
{305.1,V58.69}	18,19	{V49.86,V66.7}	1,10
{311,338.29}	19		

(b)

supported by papers from the medical literature. The correlation is of the form "an MFA appears more often for patient groups with certain demographics" (e.g., male). We show that the same conclusion can be made from the clusters constructed by MASPC (e.g., when each cluster is in an aggregate form containing the range or set of values in each demographic that are contained in the cluster and the selected MFAs by MAS), by

Table 3.11 MFAs in the clusters constructed by MASPC. For each pattern, we refer to one or more indicative papers from the medical literature supporting that the diagnosis codes in the MFA are correlated.

MFA	Reference
491.21 (Obstructive chronic bronchitis with (acute) exacerbation) V46.2 (Dependence on supplemental oxygen)	[160, 161]
I25.10 (Atherosclerotic heart disease of native coronary artery without angina pectoris) Z79.82 (Long term (current) use of aspirin)	[162, 163]
E78.5 (Hyperlipidemia, unspecified) K21.9 (Gastro-esophageal reflux disease without esophagitis)	[164]
327.23 (Obstructive sleep apnea (adult)(pediatric)) 278.01 (Morbid obesity)	[165, 166]
Z87.891 (Personal history of nicotine dependence) E78.5 (Hyperlipidemia, unspecified)	[167]
I10 (Essential (primary) hypertension) Z87.891 (Personal history of nicotine dependence)	[168]
I10 (Essential (primary) hypertension) K21.9 (Gastro-esophageal reflux disease without esophagitis)	[169]
428.0 (Congestive heart failure, unspecified) 584.9 (Acute kidney failure, unspecified)	[170]
276.8 (Hypopotassemia) 275.2 (Disorders of magnesium metabolism)	[171]

computing the relative frequency for the clusters containing patients of the group of interest (e.g., male) and comparing it with that for the other clusters (e.g., female).

In the following, we discuss each MFA in Table 3.12.

Table 3.12 MFAs in the clusters constructed by MASPC. For each pattern, we refer to one or more indicative papers from the medical literature supporting that the diagnosis codes in the MFA are correlated and also that they are correlated with demographics.

MFA	Reference
357.2 (Polyneuropathy in diabetes) 250.60 (Diabetes with neurological manifestations, type II or unspecified type, not stated as uncontrolled)	[172]
038.9 (Unspecified septicemia) 995.91 (Sepsis)	[173]
I12.9 (Hypertensive chronic kidney disease with stage 1 through stage 4 chronic kidney disease, or unspecified chronic kidney disease) N18.3 (Chronic kidney disease, stage 3 (moderate))	[174]
E78.5 (Hyperlipidemia, unspecified) I25.10 (Atherosclerotic heart disease of native coronary artery without angina pectoris)	[175, 176]
I10 (Essential (primary) hypertension) E78.5 (Hyperlipidemia, unspecified)	[177]
V30.00 (Single liveborn, born in hospital, delivered without mention of cesarean section) V05.3 (Need for prophylactic vaccination and inoculation against viral hepatitis)	[178]

- $\{357.2, 250.60\}$: This MFA corresponds to a known comorbidity [160, 161] that appears more often in old and in male patients. In line with this, we found that the relative frequency of this MFA in clusters comprised almost entirely ($> 96\%$) of male patients is larger by 146% compared to the relative frequency of the MFA in clusters comprised almost entirely ($> 96\%$) of female patients. We also found that the relative frequency of the MFA in clusters comprised entirely of patients aged ≥ 60 (old according to [179]) is larger by 35.3% compared to the relative frequency of the pattern in clusters comprised entirely of patients aged < 60.

- $\{038.0, 995.91\}$: This MFA corresponds to a known comorbidity that appears more often in old patients [173]. In line with this, we found that this observation can also be made from the clusters constructed by MASPC. Specifically, the relative frequency of the MFA in clusters comprised entirely of patients of age ≥ 60 (old) is larger by 27% compared to the relative frequency of the pattern in clusters comprised entirely of patients aged < 60.

- $\{I12.9, N18.3\}$: This MFA corresponds to a known comorbidity that appears more often in patients who are at least 75 years old, compared to patients who are between at least 65 and less than 75 years old [174]. In line with this, we found that this observation can also be made from the clusters constructed by MASPC. Specifically, the relative frequency of this MFA in clusters comprised entirely of patients at least 75 years old is larger by 70% compared to the relative frequency of the pattern in clusters comprised entirely of patients who are at least 65 and less than 75 years old.

- $\{E78.5, I25.10\}$: This MFA corresponds to a known comorbidity (atherosclerosis is the result of hyperlipidemia and hyperlipidemia is a well-known risk factor for atherosclerotic heart disease) which appears more often in old patients [175, 176]. In line with this, we found that the relative frequency of this MFA in clusters comprised almost entirely of patients of age ≥ 60 is larger by 274% compared to the relative frequency of the pattern in clusters comprised entirely of patients aged < 60.

- $\{I10, E78.5\}$: This MFA corresponds to a known comorbidity that appears more often in patients over 45, and it also appears more often in male compared to female patients [174]. In line with this, we found that the relative frequency of this pattern in clusters comprised of patients aged ≥ 45 is 18.6 times more compared to that in clusters comprised of patient aged < 45. Also, we found that the relative frequency of this MFA in clusters comprised almost entirely ($> 96\%$) of male patients is 15.2% larger compared to that in clusters comprised almost entirely ($> 96\%$) of female patients.

- $\{V30.00, V05.3\}$: There is a strong correlation of this MFA with age, since $V30.00$ corresponds to the birth of a baby and $V05.3$ to a vaccine that all babies should get short after birth [177]. In line with this, we found that the relative frequency of this MFA in the cluster of patients with age < 1 is 99.2% and that this MFA did not occur in any other cluster.

3.7.5.5 Correlations between combinations of diagnosis codes (patterns), which are not part of MFAs, and demographics

Table 3.13 shows 6 representative patterns that are contained in the clusters produced by MASPC and contain a diagnosis code associated with only male or female patients. Clearly, if the clusters preserved correlations between demographics and diagnosis codes, such patterns would be found in clusters containing mostly male or female patients. This was indeed the case for the clusters produced by MASPC; the first three patterns in Table 3.13 appeared in clusters containing only male patients, while the remaining two patterns appeared in clusters containing only female patients.

Table 3.13 Patterns in the clusters contained by MASPC. The codes $N52.9$ and 185 are associated only with male patients, while the codes 174.9 and 182.0 only with female patients.

Patterns
N52.9 (Male erectile dysfunction, unspecified) I10 (Essential (primary) hypertension)
185 (Malignant neoplasm of prostate) 188.9 (Malignant neoplasm of bladder, part unspecified)
185 (Malignant neoplasm of prostate) 250.00 (Diabetes mellitus without mention of complication, type II or unspecified type, not stated as uncontrolled)
174.9 (Malignant neoplasm of breast (female), unspecified) 311 (Depressive disorder, not elsewhere classified)
182.0 (Malignant neoplasm of corpus uteri, except isthmus) 401.9 (Unspecified essential hypertension)

Chapter 4

Clustering RS-datasets

A Relational-Sequential dataset (or RS-dataset for short) contains records comprised of a patient's values in demographic attributes and their sequence of diagnosis codes. The task of clustering an RS-dataset is helpful for analyses ranging from pattern mining to classification. However, existing methods are not appropriate to perform this task. Thus, we initiate a study of how an RS-dataset can be clustered effectively and efficiently. We formalize the task of clustering an RS-dataset as an optimization problem. At the heart of the problem is a distance measure we design to quantify the pairwise similarity between records of an RS-dataset. Our measure uses a tree structure that encodes hierarchical relationships between records, based on their demographics, as well as an edit-distance-like measure that captures both the sequentiality and the semantic similarity of diagnosis codes. We also develop an algorithm which first identifies *k* representative records (centers), for a given *k*, and then constructs *k* clusters, each containing one center and the records that are closer to the center compared to other centers. Experiments using two Electronic Health Record datasets demonstrate that our algorithm constructs compact and well-separated clusters, which preserve meaningful relationships between demographics and sequences of diagnosis codes, while being efficient and scalable.

4.1 Overview

Electronic Health Records (EHRs) contain a wealth of information (e.g., demographics, diagnoses, medications, and laboratory results) about many patients over long time periods. Data mining techniques are increasingly being employed on EHR data to derive actionable knowledge from such information [113, 114, 38, 180], which can guide clinical decision support [181], public health monitoring [182], and patient treatment [115].

4.1.1 Motivation

Table 4.1 A (toy) example of an RS-dataset (copy of Table 1.2). *Age*, *Gender*, and *Ethnicity* are demographic attributes; the *Diagnosis codes sequence* attribute is comprised of ICD-9 codes.

Age	**Gender**	**Ethnicity**	**Diagnosis codes sequence**
69	M	Black	(414.01, 250.00, 272.4, 401.9, 412, 696.1)
1	F	White	(765.18, 774.2, 765.27, 769)
67	M	Black	(414.01, 4111, 272.1, 250.00, 401.9)
48	F	White	(441.2, 401.9, 345.90, 414.01)
50	F	White	(414.01, 250.01, 401.9, 412, 2720)
0	F	White	(765.19, 769, 774.2, 779.3, 765.28, 771.7)
61	F	White	(414.01, 424.0, 440.21, 427.89, 250.00, 401.9)
1	F	White	(765.16, 775.6, 765.27, 769)
68	M	Black	(414.01, 411.1, 250.01, 401.9, 272.0)

Table 4.2 A clustered RS-dataset produced from the dataset in Table 4.1 by our algorithm. *Cluster ID* is for reference.

Cluster ID	Age	Gender	Ethnicity	Diagnosis codes sequence
1	69	M	Black	(414.01, 250.00, 272.4, 401.9, 412, 696.1)
1	67	M	Black	(414.01, 411.1, 272.1, 250.00, 401.9)
1	68	M	Black	(414.01, 411.1, 250.00, 401.9, 272.0)
2	1	F	White	(765.18, 774.2, 765.27, 769)
2	0	F	White	(765.19, 774.6, 765.28, 779.3, 769, 771.7)
2	1	F	White	(765.16, 774.2, 765.27, 769)
3	48	F	White	(441.2, 414.01, 345.90, 440.32, 250.00, 401.9)
3	50	F	White	(414.01, 440.31, 250.01, 401.9, 412, 272.0)
3	61	F	White	(414.01, 424.0, 440.21, 427.89, 250.00, 401.9)

Recall from Chapter 1 that a Relational Sequential (or RS-dataset for short) contains single-valued attributes and a sequence. In this chapter, we consider the task of clustering an RS-dataset in which every record contains a patient's demographics and a sequence of the patient's diagnosis codes. An example of such dataset is in Table 4.1. The first record corresponds to a 69-year old black male patient who is associated with six diagnoses (ICD-9 codes) [13]: first with 414.01 (coronary atherosclerosis of native coronary artery), then with 250.00 (diabetes mellitus type II without complications), and next with 272.4, 401.9, 412 and 696.1. We aim to construct clusters with similar values in demographics and also similar diagnosis codes that occur in similar order. An example of a clustering of the dataset in Table 4.1 is in Table 4.2. The first cluster (records with *Cluster ID* 1) represents black males over 60 sharing the sequence (414.01, 250.00, 401.9).

After clustering an RS-dataset, one can: (I) discover trends from each cluster, such as disease progression for patients with similar demographics, (II) compare trends across clusters, which could improve diagnosis and treatment decision making, as well as epidemiological analysis and research [24, 25], or (III) visualize the clusters [22, 23]. Furthermore, one can discover frequently occurring temporal condition patterns [21] from each cluster, which may inform research into causation or other associations [21]. Moreover, clustering an RS-dataset can be applied before: (I) classification to improve the accuracy of a classification model [183], (II) anonymization to enhance data utility [15], or (III) clinical pathway mining [20] to extract pathways for distinct types of patients.

An RS-dataset contains two fundamentally different types of data; demographics that are modeled as relational attributes and a sequence of diagnosis codes. Thus, its clustering is particularly challenging. In fact, as it will be explained in Section 4.2, existing clustering algorithms (e.g., that of [181] which clusters a relational dataset comprised of demographics and clinical information) are inappropriate to address this task. Also, it is inappropriate to first convert an RS-dataset into a dataset that can be clustered with an existing clustering algorithm and then clustering it using that algorithm.

4.1.2 Contributions

Motivated by the usefulness of the task of clustering RS-datasets and the ineffectiveness of existing methods to address it, we propose the first approach for this task. The clusters constructed by our approach can be used in analytic tasks, including visualization, classification, clinical pathway mining, and trend discovery. These tasks could help practice review and support decisions and potentially improve patient treatment and care. Also, our work implies the need for developing new methods for clustering complex EHR data.

Our work makes the following specific contributions:

1. The task of clustering an RS-dataset is formalized as a k-center [184, 185] optimization problem.

2. A new distance measure is proposed. The distance measure captures the pairwise similarity between records of an RS-dataset, based on their demographics and sequences of diagnosis codes, in a unified manner. The similarity with respect to demographics is captured using a tree structure that encodes hierarchical relationships between records based on their demographics. The similarity with respect to sequences of diagnosis codes is captured by an edit-distance-like measure that we develop to account for the semantic similarity of diagnosis codes, as specified by well-defined taxonomies.

3. A new clustering algorithm is designed. The algorithm first identifies k records (centers) representing clusters, for a given k, and then constructs k clusters, each containing a center and the records that are closer to this center, with respect to our distance measure. Our algorithm finds centers that are no more than two times worse than the best possible k centers.

4. Experiments, using two EHR datasets, are conducted to show that our algorithm constructs clusters which are: (1) compact (two times more compact on average compared to clusters constructed by state-of-the-art algorithms [186, 187] that we adapt to cluster RS-datasets); (2) well-separated (different clusters contain different values in Age, Gender, and Ethnicity, as well as different frequent sequences of diagnosis codes); and (3) able to preserve meaningful patterns that are documented in the medical literature. Our algorithm is also shown to be efficient and scalable with respect to the number of records in the RS-dataset and the number of clusters.

4.1.3 Chapter organization

The rest of the chapter is organized as follows. Section 4.2 work discusses related work of clustering RS-datasets. Section 4.3 provides the necessary background. Section 4.4 presents our distance measures for RS-datasets. Section 4.5 provides a definition of the problem addressed in this chapter. Section 4.6 presents our approach for clustering RS-datasets. Section 4.7 presents our experimental evaluation.

4.2 Related Work

Data mining techniques that are being applied to EHR data include sequential pattern mining [8], classification [188], and clustering [189]. For example, sequential pattern mining and clustering have been employed in the task of clinical pathway mining (see e.g., [20]). In the following, we focus on clustering and refer readers to [114] for a survey on EHR data mining.

As discussed in Chapter 2, there is a very large number of clustering algorithms [189] such as k-means [61], hierarchical clustering [190], k-medoids [64], and DBSCAN [76]. There is also much work on EHR data clustering [180, 127, 186, 191], including works for clustering clinical and/or demographic attributes [181]. However, these algorithms are inappropriate to cluster RS-datasets, as: (I) they assume an input dataset that contains a single attribute type (e.g., atomic or set-valued [186]); and (II) their similarity measures cannot be directly applied to records with attributes of multiple types [7].

One may naturally wonder whether an RS-dataset can be first transformed so that it contains a single attribute type and then clustered using an existing algorithm. For example, a sequence of diagnosis codes in an RS-dataset can be transformed into a set of atomic attributes, each containing the frequency of a q-gram (i.e., a substring of q letters) of the sequence. Considering all distinct q-grams of all sequences and assuming a fixed order for them allows us to find a set of atomic attributes that is common to all sequences of diagnosis codes in an RS-dataset. These attributes can then replace the *sequence of diagnosis codes* attribute in the RS-dataset to transform it into a relational dataset that can be clustered with existing algorithms such as k-medoids [64]. However, this transformation inevitably incurs information loss which harms the quality of clustering (see Section 5.7). The same holds for transformations which represent demographics as a sequence. For example, applying one-hot encoding to the demographic values in a record of an RS-dataset, as in [186], creates a (binary) sequence to which we can append the sequence of diagnosis codes. This leads to a sequential dataset which can be clustered with existing algorithms (e.g., hierarchical clustering [190]). Yet, this transformation incurs fake ordering information which affects our ability to meaningfully measure similarity between records of the transformed dataset. Specifically, the values of demographics become ordered and precede diagnosis codes. However, such an artificial ordering has no semantic meaning as there is no natural ordering among attributes; only diagnosis codes can be ordered. Similar issues arise when one transforms an RS-dataset record into a vector, as required by the unsupervised deep learning algorithms reviewed in Chapter 2.

One may also wonder whether algorithms designed for RT-datasets (see Chapter 3) are appropriate for clustering RS-datasets. For instance, recall that the MASPC algorithm works by: (I) projecting an RT-dataset on a number of carefully selected patterns (sets of demographic values and diagnosis codes), and (II) clustering the projected dataset based on a hierarchical clustering algorithm [190]. The patterns are selected so that they are frequent and comprised of correlated diagnosis codes. It is easy to cluster an RS-dataset using MASPC by: (I) keeping only one occurrence of every diagnosis code in every record of the RS-dataset (this converts an RS-dataset into an RT-dataset), (II) applying MASPC on the RT-dataset, and (III) replacing the set of diagnosis codes in every record of the clustered RT-dataset with the sequence of diagnosis codes of its corresponding record in the RS-dataset. However, the resultant clusters contain dissimilar records with respect to their diagnosis codes, as shown in Section 5.7. This is because MASPC does not consider multiple occurrences of diagnosis codes in a record nor the order in which these diagnosis codes occur in the record.

4.3 Background

In the following, we summarize the notation used in this chapter (see Table 4.3) and introduce some preliminary concepts.

Table 4.3 Table of notation.

Notation	Definition
$\mathbf{D_{RS}}$	RS-dataset
$A^i, i \in [1, l]$	i-th demographic
s	Sequence of diagnosis codes
r^{dem}	Projection of record r on demographics
r^{seq}	Projection of record r on sequence
d_{JC}, d_{WE}, d_{JCE}, d_J, d_{LCS}	Distance functions
$\mathbf{T_{RS}}$	RS-Tree
w_{dem}, w_{diag}	Weights

4.3.1 RS-dataset

An RS-dataset is denoted by $\mathbf{D_{RS}}$ and contains $l \geq 1$ demographic attributes, $A^1, \ldots, A^l$. Each demographic attribute can be numerical or categorical. Each record in $\mathbf{D_{RS}}$ is a vector containing l values, one in each demographic, and a sequence $s = (s[1]s[2]\ldots s[n])$ comprised of $|s| = n$ diagnosis codes. The projection of a record r on the set of demographic attributes (respectively, sequence s) is denoted by r^{dem} (respectively, r^{seq}). The diagnosis codes in a sequence are drawn from an alphabet $\sum$ and can be represented in different formats, e.g., as ICD-9 codes. In the latter case, $\sum$ is the set of all ICD-9 codes.

4.3.2 Semantic Distance

Jiang-Conrath (JC) distance [192] is a well-established measure to calculate the semantic distance between two concepts, represented as nodes in a tree. JC distance is based on information content (IC) [193], a measure of specificity which we define first. The information content of a node u in a tree with L leaves is defined as $IC(u) = -\log_2((\frac{\text{leafdesc}(u)}{\text{asc}(u)+1} + 1)/(L + 1))$ [193], where asc(u) (respectively, leafdesc(u)) is the number of ascendants (respectively, leaf-level descendants) of u. Thus, tree nodes close to leaves, which correspond to more specific concepts, get lower information content values.

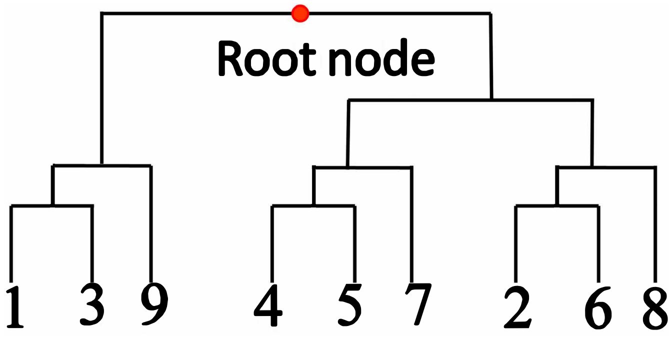


Fig. 4.1 An example tree.

The JC distance for two nodes u_i, u_j is denoted by $d_{JC}(u_i, u_j)$ and computed using (4.1):

$$d_{JC}(u_i, u_j) = (IC(u_i) + IC(u_j)) - 2\times\ IC(LCA(u_i, u_j)), \tag{4.1}$$

where $LCA(u_i, u_j)$ denotes the least common ancestor of u_i, u_j in the tree. For example, using the tree of Fig. 4.1, we have: $L = 9$, $\text{leafdesc}(1) = \text{leafdesc}(4) = 0$, $\text{asc}(1) = 3$, $\text{asc}(4) = 4$, $IC(1) = IC(4) = -\log_2(\frac{1}{10})$, and $IC(LCA(1, 4)) = IC(root) = 0$. Thus, $d_{JC}(1, 4) = -2\log_2(\frac{1}{10}) \approx 6.64$.

4.3.3 Weighted Edit Distance

Edit distance [194] is commonly used to capture the distance between two sequences s_i, s_j and is expressed as the minimum number of element insertions, deletions, and substitutions needed to transform s_i to s_j. There are several unweighted and weighted measures based on edit distance; see [195] for a recent reference. We employ a weighted version of edit distance from [196], which considers semantic similarity. It is denoted by $d_{WE}(s_j, s_i)$ and can be computed using (4.2):

$$d_{WE}(s_i, s_j) = \begin{cases} |s_i| & \text{if } |s_j| = 0 \\ |s_j| & \text{if } |s_i| = 0 \\ d_{WE}(\text{tail}(s_i), \text{tail}(s_j)) & \text{if } s_i[1] = s_j[1] \\ \min \begin{cases} d_{WE}(\text{tail}(s_i), s_j) + 1 \\ d_{WE}(s_i, \text{tail}(s_j)) + 1 \\ d_{WE}(\text{tail}(s_i), \text{tail}(s_j)) + \text{sub}(s_i[1], s_j[1]) \end{cases} & \text{otherwise.} \end{cases} \tag{4.2}$$

where $\text{tail}(s_i) = (s_i[2] \ldots s_i[|s_i|])$ and $\text{sub}(x, y) \in [0, 1]$ is the cost of substituting element x with y. d_{WE} differs from edit distance in that the substitution cost is given by sub()

instead of being 1. Let $s_i = (a, b)$, $s_j = (a, c)$, sub$(a, a) = 0$, and sub$(b, c) = 0.5$. Then, $d_{WE}(s_i, s_j) = 0.5$ as $s_i[1] = s_j[1] = a$, and $d_{WE}(\text{tail}(s_i), \text{tail}(s_j)) = d_{WE}(b, c) = 0.5$.

4.4 Distance Measure for RS-datasets

In this section, we first define the concept of RS-Tree, which is used by our distance measure, and then define our distance measure.

4.4.1 RS-Tree $\mathbf{T_{RS}}$

An RS-Tree, denoted by $\mathbf{T_{RS}}$, encodes hierarchical relationships between records of an RS-dataset $\mathbf{D_{RS}}$ regarding demographics. $\mathbf{T_{RS}}$ is constructed in two steps: (I) The records in $\mathbf{D_{RS}}$ are partitioned into groups, each containing all records with the same values in all demographics. This is to quickly identify pairs of records that have distance zero based on demographics. (II) Agglomerative average-linkage hierarchical clustering [190] is applied to the groups obtained from step I. The distance between two groups is measured using the Hamming distance [7], computed over demographics. To apply Hamming distance, we discretize numerical demographic attributes (if any) [7].

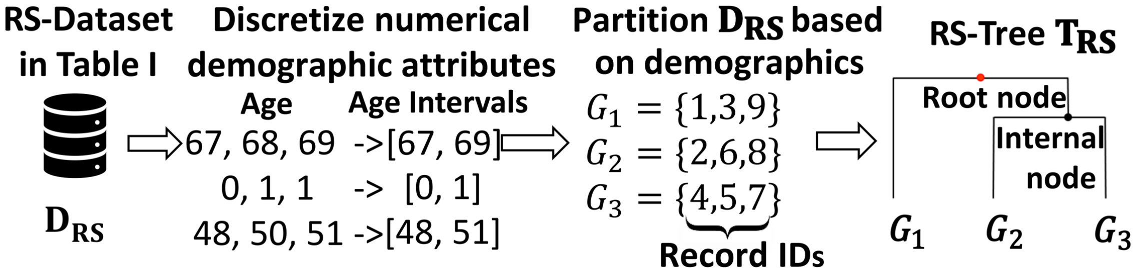


Fig. 4.2 RS-Tree Construction. Record ID x corresponds to the x-th record in Table 4.1.

The leaves of an RS-Tree represent groups of records with the same demographics and the root a group comprised of all records. Fig. 4.2 illustrates an RS-Tree constructed from the RS-dataset in Table 4.1, after discretizing *Age*. The three groups of records (leaves in the RS-Tree) are created in step I.

An RS-Tree can be used to measure the distance between two records based on their demographics without requiring user input. This is an important benefit over existing distance measures [15], which need users to set several data-dependent parameters. Also, RS-Tree can consider the hierarchical information of all demographic attributes, which can provide better clustering results as shown in Section 4.7.

4.4.2 The d_{JCE} Measure

Our measure, d_{JCE} (for Jiang-Conrath Edit distance), captures the distance between two records in an RS-Dataset based on both their demographics and sequences of diagnosis codes, in a unified manner. d_{JCE} is computed for a pair of records, r_i, r_j, based on the JC-distance for demographics and the weighted edit distance for sequences of diagnosis codes, as shown in (4.3):

$$d_{JCE}(r_i, r_j) = \sqrt{w_{dem} \cdot d_{JC}(r_i^{dem}, r_j^{dem}) + w_{diag} \cdot d_{WE}(r_i^{seq}, r_j^{seq})}, \qquad (4.3)$$

where w_{dem}, w_{diag} are weights trading off the importance of d_{JC} and d_{WE} in the computation of d_{JCE}. The distance d_{JC} is computed using the RS-Tree $\mathbf{T_{RS}}$ that is constructed from $\mathbf{D_{RS}}$, as explained in Section 4.4.1.

To effectively use d_{JCE} in our context, we modify it in two ways. First, we define the substitution cost function sub() in d_{WE} to reflect the semantic distance between diagnosis codes. Specifically, for any two ICD-9 codes i, j, we set:

$$\text{sub}(i, j) = d_{JC}(u_i, u_j) / \max_{u_q, u_p \in \mathbf{H}} d_{JC}(u_q, u_p), \qquad (4.4)$$

where u_i and u_j are the nodes in the standard ICD-9 code hierarchy $\mathbf{H}$ [197] that correspond to i and j, respectively. This gives a zero substitution cost when $i = j$ and a smaller cost for semantically similar diagnosis codes. For example, sub$(401.9, 414.01) <$ sub$(401.9, 250.00)$ as the first two codes are closer with respect to the ICD-9 code hierarchy, since they both represent diseases of the circulatory system. The substitution cost function in d_{WE} captures that two sequences with semantically similar diagnosis codes are more similar. This is not captured by edit distance, which penalizes every pair of different ICD-9 codes equally. Second, we modify d_{JCE} to avoid bias in favor of d_{JC} or d_{WE} by: (I) normalizing d_{JC} (respectively, d_{WE}) by dividing with its maximum possible value $\max_{r_i, r_j \in \mathbf{D_{RS}}} d_{JC}(r_i^{dem}, r_j^{dem})$ (respectively, $\max(|r_i^{seq}|, |r_j^{seq}|)$), and (II) selecting weights in Eq. 4.3, so that $w_{dem}, w_{diag} \in [0, 1]$ and $w_{dem} + w_{diag} = 1$. Normalization ensures that the values of d_{JCE} are in $[0, 1]$ and that $d_{JCE}(r_i, r_j) = 0$, if and only if $r_i^{dem} = r_j^{dem}$ and $r_i^{seq} = r_j^{seq}$. Of note, d_{JCE} is a metric, since d_{JC} and d_{WE} are metrics [198], which makes it generic enough for other uses (e.g., it can be incorporated into information retrieval algorithms used for clinical reasoning [199]).

4.5 Problem definition

We define the following clustering problem:

Problem 2 (RS-Dataset k-Center (RSDC))**.** *Given an RS-dataset* $\mathbf{D_{RS}}$ *and an integer* $k > 0$*, find a set of* C *of* k *records in* $\mathbf{D_{RS}}$*, referred to as centers, such that the distance* $\max_{r \in \mathbf{D_{RS}}} \min_{c \in C} d_{JCE}\,(r, c)$ *is minimized.*

RSDC aims to find a set C of k centers such that the largest distance of any record that is not in C to its closest center is minimum. RSDC can be formulated as a k-Center problem with d_{JCE} as its objective function (see Section 4.5.1). After finding C, the final clustering of $\mathbf{D_{RS}}$ is obtained by adding each center to a different cluster and then adding each record that is not a center to the cluster where its closest center is (breaking ties arbitrarily).

4.5.1 Mathematical formulation of RSDC problem

Define a binary variable y_j such that $y_j = 1$, if record $r_j \in \mathbf{D_{RS}}$ is selected as a center and 0 otherwise. Also, define binary variables x_{ij} such that $x_{ij} = 1$, if $r_i \in \mathbf{D_{RS}}$ is closest to center $r_j \in \mathbf{D_{RS}}$ and 0 otherwise. Then, RSDC can be formulated as follows (the formulation is similar to the k-center formulation in [200]):

$$\begin{aligned}
\text{minimize} \quad & z && && \text{(4.5a)}\\
\text{subject to} \quad & \sum_{j \in [1, |\mathbf{D_{RS}}|]} d_{JCE}(r_i, r_j) \cdot x_{ij} \leq z && \forall i \in [1, |\mathbf{D_{RS}}|] && \text{(4.5b)}\\
& \sum_{j \in [1, |\mathbf{D_{RS}}|]} x_{ij} = 1 && \forall i \in [1, |\mathbf{D_{RS}}|] && \text{(4.5c)}\\
& x_{ij} \leq y_j && \forall i, j \in [1, |\mathbf{D_{RS}}|] && \text{(4.5d)}\\
& \sum_{j \in [1, |\mathbf{D_{RS}}|]} y_j = k && && \text{(4.5e)}\\
& y_j \in \{0, 1\} && \forall j \in [1, |\mathbf{D_{RS}}|] && \text{(4.5f)}\\
& x_{ij} \in \{0, 1\} && \forall i, j \in [1, |\mathbf{D_{RS}}|] && \text{(4.5g)}\\
& z \in \mathbb{R} && && \text{(4.5h)}
\end{aligned}$$

Eqs. 4.5a and 4.5b ensure that the objective value is no less than the maximum record-to-center distance. Eq. 4.5c assigns each record to exactly one center. Eq. 4.5d ensures that no record is assigned to j unless there is a center at j. Eq. 4.5e ensures there are k centers selected and Eqs. 4.5f and 4.5g are the binary restrictions.

There are several other formulations of clustering problems; see [201] and references therein. For example, Vinod [202] considered a general partitional clustering problem where the objective was to create k clusters that minimize the total assignment cost (i.e., sum of costs of assigning the i-th data point to the j-th cluster). We also minimize an assignment cost which is calculated specifically using d_{JCE}. Furthermore, the meaning of the binary variables and the output in our work is different from [202] . Other examples of clustering problem formulations appeared in [203], which considers the 1-norm, and in [204], which identifies the best k cluster planes, instead of the best k centers.

4.5.2 Hardness of RSDC

We show that it is hard to solve RSDC optimally, or even to obtain a solution that is less than two times worse than the optimal solution.

First, we prove that the RSDS problem is NP-hard by reducing the decision version of RSDSC from the decision version of the well-known metric k-Center problem [26]. The latter is NP-complete, as shown in [26] using a reduction from the decision version of the dominating set problem [205]. The decision version of the dominating set problem asks if there is in an undirected graph $G = (V, E)$ a subset of nodes $S \subseteq V$ of size at most k such that each node in $V \setminus S$ has at least one neighbor in S.

Theorem 1. *The RSDC problem is NP-hard.*

Proof. The decision version of RSDC asks whether there exists a set C of k centers such that $\max_{r \in \mathbf{D_{RS}}} \min_{c \in C} d_{JCE}(r, c) \leq B$ for a given real number B. The decision version of RSDC is clearly in NP. Thus, it suffices to show that the decision version of RSDC can be reduced from the decision version of the NP-complete metric k-Center problem [26]. A metric space (P, d) is an ordered pair, where P is a set and $d : P \times P \to R^+$ is a metric (also referred to as a distance function) on P. Given a metric space (P, d), where P is a set of n points, an integer $k \in [1, n]$, and a real number B, the decision version of the metric k-Center problem asks to decide whether there exists a subset $S \subseteq P$ of k points such that $\max_{p \in P} \min_{s \in S} d(p, s) \leq B$. For the reduction, we first construct an instance I_{RSDC} of the decision version of RSDC in polynomial time from any instance I_{kC} of the decision version of the metric k-Center by: (I) adding a record r into an RS-dataset $\mathbf{D_{RS}}$ for each point $p \in P$, (II) setting $d_{JCE}(r, r') = d(p, p')$ for every pair $p, p' \in P$ that corresponds to a pair of records $r, r' \in \mathbf{D_{RS}}$, and (III) setting the parameters k (respectively, B) in I_{RSDC} to the value of the parameter k (respectively, B) in I_{kC}. We then observe that if I_{RSDC} has a positive answer, then there is a set C of k records such that $\max_{r\in\mathbf{D_{RS}}}\min_{c\in C} d_{JCE}\left(r,c\right)\leq B$, which correspond to a subset S of k points in P for which $\max_{p\in P}\min_{s\in S} d(p,s)\leq B$. Thus, I_{kC} has a positive answer. It is easy to see that the converse also holds. □

Theorem 2. *The RSDSC problem can be approximated within a factor of* 2.

Proof. We reduce RSDC to the metric k-Center problem, which can be approximated within a factor of 2 [185]. We first construct an instance I_{kC} of the metric k-Center from any given instance I_{RSDC} of RSDC in polynomial time by: (I) adding a point p into the set of points P, for each record r in the RS-dataset $\mathbf{D_{RS}}$, (II) setting $d\left(p,p'\right)=d_{JCE}\left(r,r'\right)$ for every pair of records $r,r'\in\mathbf{D_{RS}}$ that corresponds to a pair of points $p,p'\in P$, and (III) setting the parameters k (respectively, B) in I_{kC} to the value of the parameter k (respectively, B) in I_{RSDC}. We then prove the correspondence between a solution S_{kC} to I_{kC} and a solution S_{RSDC} to I_{RSDC}. If S_{kC} is a solution to I_{kC}, then there is a subset S of k points in P for which $\max_{p\in P}\min_{s\in S} d(p,s)$ is minimum. These points correspond to a set C of k records s.t. $\max_{r\in\mathbf{D_{RS}}}\min_{c\in C} d_{JCE}\left(r,c\right)$ is minimum. Thus, S_{RSDC} is a solution to RSDC. Clearly, the converse also holds. □

Theorem 3. *The RSDSC problem cannot be approximated within a factor of* $2-\epsilon$*, for any* $\epsilon>0$*, in polynomial time unless P=NP.*

Proof. We reduce the metric k-Center problem, which cannot be approximated within a factor of $2-\varepsilon$, for any $\varepsilon>0$ [185], to RSDC. Given any instance I_{kC} of the metric k-Center, we construct an instance I_{RSDC} of RSDC in polynomial time in the size of I_{kC} by: (I) adding a record r into an RS-dataset $\mathbf{D_{RS}}$ for each point $p\in P$, (II) setting $d_{JCE}\left(r,r'\right)=d\left(p,p'\right)$ for every pair $p,p'\in P$ that corresponds to a pair of records $r,r'\in\mathbf{D_{RS}}$, and (III) setting the parameters k (respectively, B) in I_{RSDC} to the value of the parameter k (respectively, B) in I_{kC}. Since we have proved the correspondence between a solution S_{kC} to I_{kC} and a solution S_{RSDC} to I_{RSDC} above, we have a reduction from the metric k-Center problem to RSDC. □

4.6 Clustering RS-datasets using DDSCA

Our Demographics and Diagnosis Sequences Clustering Algorithm (DDSCA) clusters an RS-dataset based on the RSDC problem. DDSCA works in two phases (see Fig. 4.3): (I) RS-Tree Construction; and (II) RS-Dataset Clustering, which is based on the algorithm of [185]. We now explain each phase (see Algorithm 2 for the pseudocode):

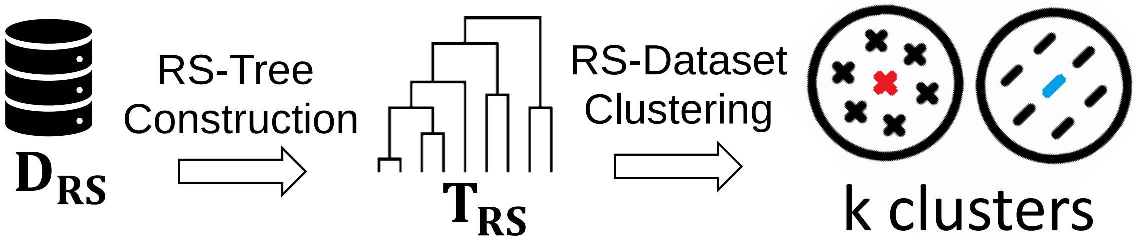


Fig. 4.3 DDSCA workflow. After constructing the RS-tree, we are able to calculate the JC-distance d_{JC} for any two records. This is combined with the weighted edit distance for sequences of diagnosis codes to obtain d_{JCE}. The latter is used by the DDSCA algorithm to obtain the final clusters.

Algorithm 2 DDSCA ($\mathbf{D_{RS}}$,w_{dem}, w_{diag}, k)

```
Input: Dataset $\mathbf{D_{RS}}$, $w_{dem}$, $w_{diag}$, the number of clusters $k$
Output: a set of clusters $U$
     // RS-Tree Construction Phase
 1:  $\mathbf{T_{RS}} \leftarrow$ Agglomerative-Average-Linkage($\mathbf{D_{RS}}$)
     // RS-Dataset Clustering Phase
 2:  $c \leftarrow$ arbitrary record from $\mathbf{D_{RS}}$
 3:  $U \leftarrow \{c\}$   // Set of clusters
 4:  $C \leftarrow c$   // Set of centers
 5:  while $|C| < k$ do
 6:      Select a record $c$ from $\mathbf{D_{RS}}$ such that $c \notin C$ and $\min_{c' \in C} d_{JCE}(c, c')$ is maximized
 7:      $U \leftarrow U \cup \{c\}$
 8:      $C \leftarrow C \cup c$
 9:  end while
10:  for $r \in \mathbf{D_{RS}}$ do
11:      if $r \notin C$ then
12:          Assign $r$ to the cluster in $U$ whose center $c$ has minimum $d_{JCE}(c, r)$
13:      end if
14:  end for
15:  return $U$
```

RS-Tree Construction Phase: In this phase (line 1), the RS-Tree is constructed from the input dataset $\mathbf{D_{RS}}$.

RS-Dataset Clustering Phase: The set C of k centers is constructed iteratively (lines 2-8), as follows. An arbitrarily selected record of $\mathbf{D_{RS}}$ becomes the first cluster in the set of clusters U and added into C (lines 2-4). Then, in each iteration, another record whose distance from its closest center is as large as possible is found and added into C and into U (lines 5-8). The process continues until k records are added into C. After that, k clusters are built (lines 9-11). Then, each record that is not in C is added into the cluster whose center is closest to it (line 11). Last, U is returned (line 12).

Despite its simplicity, DDSCA computes the best possible centers that can be computed in polynomial time. Specifically, we prove that it computes a set C that is at most 2 times worse with respect to d_{JCE} than the optimal solution to RSDC (see Section 4.6.1)

4.6.1 Approximation guarantee of DDSCA

The fact that DDSCA finds a set C of k centers that is at most 2 times worse with respect to d_{JCE} than the best possible set of k centers follows from the following two facts: (1) DDSCA selects centers with a strategy following that of the algorithm of Gonzalez et al. [185]. (2) The algorithm of [185] has an approximation ratio of 2. The fact that DDSCA finds the best possible k centers in polynomial time follows from the fact that the RSDC problem cannot be approximated within a factor smaller than 2 unless $P = NP$ (see Theorem 3).

4.6.2 The time complexity of DDSA

DDSA takes $O(|\mathbf{D_{RS}}|^2 \cdot l + [(\max_{r^{seq} \in \mathbf{D_{RS}}} |r^{seq}|)^2 + |\mathbf{T_{RS}}|]k|\mathbf{D_{RS}}|)$ time, where $|\mathbf{T_{RS}}|$ is the number of nodes in $\mathbf{T_{RS}}$, k is the number of clusters, and $|\mathbf{D_{RS}}|$ is the number of records in $\mathbf{D_{RS}}$. Line 1 takes $O(|\mathbf{D_{RS}}|^2 \cdot l)$ time due to the use of agglomerative average-linkage hierarchical clustering with Hamming distance. Lines 5-9 (and also Lines 10-14) take $O([(\max_{r^{seq} \in \mathbf{D_{RS}}} |r^{seq}|)^2 + |\mathbf{T_{RS}}|]k|\mathbf{D_{RS}}|)$ time, where the terms in square brackets correspond to the computation of d_{JCE}.

4.7 Experimental evaluation

4.7.1 Data

Two RS-datasets, MIMIC [206] and INFORMS [207], were used. Each record in these datasets contains the demographics *Age*, *Gender*, and *Ethnicity* of a patient and a sequence with the patients' ICD-9 codes. A similar dataset to INFORMS but with sets instead of sequences of diagnosis codes was used in [186, 15]. Age in MIMIC and INFORMS was discretized using a standard hierarchy [208], in different ways (see Section B.1 in Appendix B). Table 4.4 summarizes the characteristics of MIMIC and INFORMS.

4.7.2 Competitors

We compared DDSCA, in terms of effectiveness and efficiency, against two state-of-the-art clustering methods that we adapted to cluster RS-datasets: AGC (Adaptive Graph

Table 4.4 Datasets characteristics.

Dataset	$\lvert\mathbf{D_{RS}}\rvert$	# of demographics	$\lvert\Sigma\rvert$	Max # of diag. codes/record	Avg # of diag. codes/record
MIMIC	$37,730$	3	$5,558$	39	9.21
INFORMS	$26,630$	3	545	46	3.63

Convolution) [187] and MASPC (Maximal-frequent All-confident pattern Selection with Pattern-based Clustering) from Chapter 3. AGC and MASPC were chosen, as they outperformed deep-learning-based methods (e.g., [209]) and a hybrid algorithm along the lines of [15], respectively.

Details on AGC and MASPC are provided below.

AGC [187]: AGC gets as input a graph whose nodes have vectors as attributes, and it outputs a partition of the nodes of the graph into k groups. In our context, the graph is a tree, $\mathbf{T_{AGC}}$, that is similar to the RS-tree $\mathbf{T_{RS}}$ of DDSCA. The difference between $\mathbf{T_{AGC}}$ and $\mathbf{T_{RS}}$ and is that: (1) Each record r in any leaf-level node of $\mathbf{T_{AGC}}$ does not contain the sequence of diagnosis r^{seq} but a vector with the frequencies of all q-grams (i.e., substrings of length q) of r^{seq}, and (2) each record r in a non-leaf node of $\mathbf{T_{AGC}}$ contains a dummy vector of zeros. The differences are because each node in an input graph of AGC must contain a vector of values. AGC uses graph convolution and spectral clustering [210] to construct the groups of $\mathbf{T_{AGC}}$. However, some groups may contain non-leaf nodes, which contain no useful information about sequences. Thus, we remove such nodes from the groups. This does not affect the quality of clustering. Next, we obtain the clusters by going over each group and adding into a cluster all the records in $\mathbf{D_{RS}}$ corresponding to the nodes in the group.

MASPC: Recall from Chapter 3 that MASPC gets as input a dataset in which each record contains values in demographics, as well as a set of diagnosis codes, and it outputs k clusters of the dataset. This dataset is produced by an RS-dataset, and the final clusters are produced as described in Section 4.2. If there are records that are unclustered, MASPC adds them into a single cluster, without considering how similar they are to other records.

4.7.3 Evaluation measures

For evaluating the quality of clusters based on demographics, we used the ASPJ (Average Sum of Pairwise Jaccard distance) measure, defined as follows:

$$\text{ASPJ} = \frac{1}{k}\sum_{c\in C}\sum_{r_i,r_j\in c} d_J(r_i^{dem}, r_j^{dem}), \tag{4.6}$$

where C is a set of k clusters and $d_J()$ is Jaccard distance [7]. For evaluating the quality of clusters based on diagnosis codes, we used the ASPWE (Average Sum of Pairwise Weighted Edit distance) measure, defined as follows:

$$\text{ASPWE} = \frac{1}{k} \sum_{c \in C} \sum_{r_i, r_j \in c} d_{WE}(r_i^{seq}, r_j^{seq}). \tag{4.7}$$

ASPJ and ASPWE are well-known internal clustering indices [189] measuring how similar are records in clusters with respect to demographics and sequences of diagnosis codes, respectively. We also used ASPLCS, which is similar to ASPWE but instead of d_{WE} it is based on the Longest Common Subsequence (LCS) distance measure [20]. The latter is defined as $d_{LCS}(r_i^{seq}, r_j^{seq}) = |r_i^{seq}| + |r_j^{seq}| - 2 \cdot \ell(r_i^{seq}, r_j^{seq})$, where $\ell(r_i^{seq}, r_j^{seq})$ is the longest common subsequence of the diagnosis code sequences r_i^{seq} and r_j^{seq}. Small values in ASPJ, ASPWE and ASPLCS are preferred. External clustering indices were not used, since our datasets do not have ground-truth clusters.

For evaluating the compactness of clusters, we defined a simple measure, called Average Cluster Compactness (ACC). Let $\mathbf{D_{RS}}$ be an RS-dataset that is clustered into a set C of k clusters and $\{u_1^i, \ldots, u_m^i\}$ be the set of values of an attribute A^i, $i \in [1, l]$, in these clusters. ACC is defined as follows:

$$\text{ACC}(\{A^1, \ldots, A^l\}) = \frac{1}{k \cdot l} \sum_{i \in [1,l]} \sum_{c \in C} \frac{\max_{j \in [1,m]} RF(u_j^i, c)}{\sum_{j \in [1,m]} RF(u_j^i, c)}, \tag{4.8}$$

where c is a cluster and $RF(u_j^i, c)$ is the relative frequency of a value in A^i that is contained in c. Large values in ACC are preferred. For example, for a single attribute A^i, ACC$(\{A^i\})$ takes values in $(0, 1]$ and large values indicate that the most frequent value in clusters appears in many records of the clusters. Similarly, ACC$(\{A^i\}) = 1$ means that every cluster contains only one value in A^i and thus the clusters are as compact as possible with respect to this attribute.

4.7.4 Environment and code

We implemented DDSCA in Python 3 and used the Python implementations of AGC[1] and MASPC[2]. All experiments were performed on an Intel i9 at 3.70GHz with 64GB RAM. Our source code is available at https://bitbucket.org/EHR_Clustering/ddsca/src/master/. Unless stated otherwise, we used $k = 50$ (selected by the Elbow Method [211]) for all methods and $w_{dem} = w_{diag} = 0.5$ for our method. This weight configuration was chosen

[1] https://github.com/karenlatong/AGC-master

[2] https://bitbucket.org/EHR_Clustering/maspc/src/master/

to treat demographics and diagnosis codes equally, as the competitors do. Also, it resulted in much more compact clusters compared to alternative configurations (see Section B.2 in Appendix B). All parameters for AGC and MASPC were set to their default values from [187, 186].

4.7.5 Clustering effectiveness

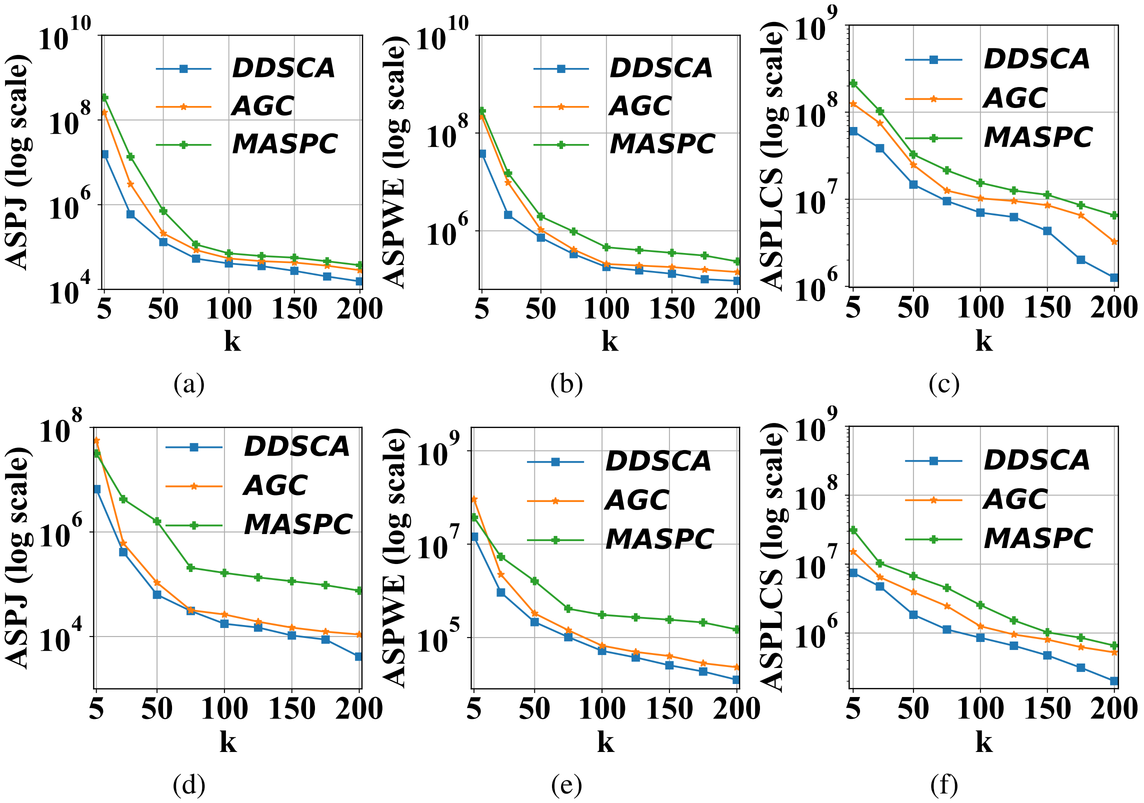
Fig. 4.4 shows that DDSCA outperformed both competitors in all tested cases, according to all measures. For example, it created clusters with 184% (respectively, 167%) lower ASPWE (respectively, ASPLCS) and 178% lower ASPJ on average compared to AGC in MIMIC. This demonstrates that DDSCA created more compact clusters, which allow meaningful analyses based on demographics and diagnosis codes. Similar measures to ASPLCS but with other string distance functions were also used and analogous results were obtained (see Section B.3 in Appendix B). AGC did not perform well because it has to transform sequences of diagnosis codes into vectors of q-gram frequencies (see Section 4.2), which inevitably affects sequence similarity measurement and does not consider the semantic distance between diagnosis codes. For example, the sequences $(414.01, 250.00, 414.01)$ and $(250.00, 414.01, 250.00)$ are considered maximally similar by AGC, since they have the same 2-gram frequencies and hence the same vector representation. Yet, they are dissimilar when the ordering of diagnosis codes and their semantic similarity is considered. MASPC did not perform well because it operates on sets of diagnosis codes.

4.7.6 Impact of q for AGC

We also examined the impact of q for AGC. Fig. 4.5 shows that cluster quality in MIMIC and INFORMS became worse as q increased. This is because the number of distinct q-grams increases, and thus ASPWE and ASPJ increased as well. Therefore, $q = 2$ that we selected as the default value is a suitable choice.

4.7.7 Compactness of clusters

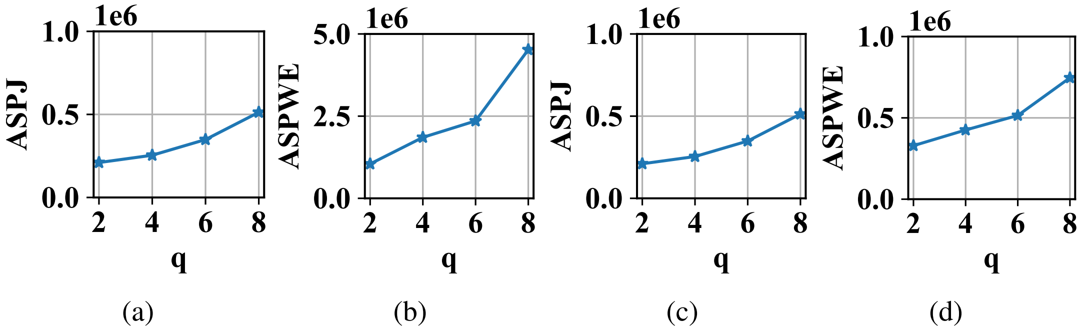
Table 4.5 shows that, when applied to either MIMIC or INFORMS, DDSCA creates much more compact clusters than the competitors with respect to all attributes and also with respect to each of the attributes separately. For example, the ACC of DDSCA over all attributes was 1.94 (respectively, 1.6) times better than that of AGC, the best competitor, in MIMIC and (respectively, INFORMS). The reasons that AGC and MASPC do not perform well are the same as those discussed in the experiment above.



(a) (b) (c)

(d) (e) (f)

Fig. 4.4 (a) ASPJ, (b) ASPWE, and (c) ASPLCS vs. *k* for MIMIC. (d) ASPJ, (e) ASPWE, and (f) ASPLCS vs. *k* for INFORMS.



(a) (b) (c) (d)

Fig. 4.5 (a) ASPJ and (b) ASPWE vs. q for MIMIC. (c) ASPJ and (d) ASPWE vs. q for INFORMS.

Fig. 4.6 shows the distribution of values in each demographic and the distribution of ICD Chapters [3] [212] in ten clusters constructed by applying DDSCA with $k = 50$ on MIMIC. The reported clusters were selected randomly among the best 25 clusters with respect to ACC, computed over all attributes. For the results of all clusters, see Section B.4 in Appendix B. The corresponding results of AGC and MASPC are not reported as they

[3] We used first 17 Chapters, as the data does not have E and V codes.

Table 4.5 ACC for: (a) MIMIC and (b) INFORMS. ICD Chapter sequences are produced by replacing every ICD-9 code with its corresponding ICD Chapter.

Methods	ACC	ACC	ACC	ACC	ACC
	(*Age*)	(*Gender*)	(*Ethnicity*)	(*ICD Chapter seq.*)	(all attributes)
DDSCA	0.725	0.924	0.749	0.893	0.823
AGC	0.412	0.432	0.514	0.341	0.425
MASPC	0.315	0.354	0.415	0.385	0.367

(a)

Methods	ACC	ACC	ACC	ACC	ACC
	(*Age*)	(*Gender*)	(*Ethnicity*)	(*ICD Chapter seq.*)	(all attributes)
DDSCA	0.595	0.847	0.805	0.831	0.769
AGC	0.334	0.563	0.601	0.423	0.480
MASPC	0.336	0.412	0.445	0.407	0.400

(b)

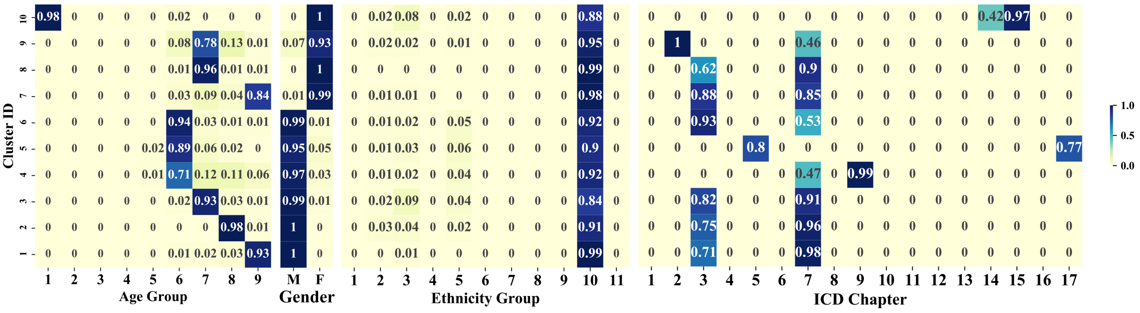


Fig. 4.6 Heat map of MIMIC for *Age*, *Gender*, *Ethnicity*, and ICD Chapter. The description of values in Age and Ethnicity groups are in Section B.1 of Chapter B. The values in the cells are ratios of records in a cluster (e.g., 0.98 of records in cluster with ID 10 have their Age values in Age Group 1 corresponding to Newborns). The sum of ratios for a cluster over the ICD Chapters is not 1, since a record can contain diagnosis codes belonging into multiple ICD Chapters.

were much worse, as expected by the ACC measures for these algorithms. Most records in each cluster have the same value in each demographic attribute, and their top-2 frequent diagnosis codes belong to two ICD Chapters. This implies that clusters are compact. For example, in Cluster 1, 93% of patients are over 80 years old, 100% are male, and 99% are white. Meanwhile, 98% of patients in Cluster 1 have at least one diagnosis code in ICD Chapter 7 (Diseases of the Circulatory System), and 71% have at least one diagnosis code in ICD Chapter 3 (Endocrine, Nutritional and Metabolic Diseases, and Immunity Disorders). Similar results were obtained for INFORMS (see Section B.5 in Appendix B).

4.7.8 Separability of clusters

Table 4.6 The top-1 (i.e., most) frequent value in each demographic, top-2 frequent ICD Chapters, and top-3 frequent sequential patterns of each cluster. A top-3 frequent sequential pattern in a cluster c is a sequence of ICD-9 codes appearing in the first, second, or third largest number of records in c.

Cluster ID	Gender	Age	Ethnicity	ICD Chapters	Top-3 Frequent Sequential Patterns		
1	M	Over 80	White	$\{3,7\}$	$(427.31, 428.0)$	$(401.9, 427.31)$	$(414.01, 427.31)$
2	M	Aged	White	$\{3,7\}$	$(401.9, 414.01)$	$(414.01, 427.31)$	$(401.9, 427.31)$
3	M	Middle aged	White	$\{3,7\}$	$(401.9, 414.01)$	$(272.4, 401.9)$	$(272.0, 401.9)$
4	M	Adult	White	$\{7,9\}$	$(401.9, 530.81)$	$(414.01, 530.81)$	$(272.4, 530.81)$
5	M	Adult	White	$\{5,17\}$	$(305.1, 401.9)$	$(305.00, 305.1)$	$(305.1, 518.81)$
6	M	Adult	White	$\{3,7\}$	$(276.2, 518.81)$	$(276.2, 584.9)$	$(518.81, 584.9)$
7	F	Over 80	White	$\{3,7\}$	$(427.31, 428.0)$	$(428.0, 584.9)$	$(401.9, 427.31)$
8	F	Middle aged	White	$\{3,7\}$	$(401.9, 414.01)$	$(272.4, 401.9)$	$(401.9, 427.31)$
9	F	Middle aged	White	$\{2,7\}$	$(198.3, 198.5)$	$(197.7, 198.5)$	$(197.0, 198.5)$
10	F	Newborn	White	$\{14,15\}$	$(769, 774.2)$	$(774.2, 770.81)$	$(774.2, 779.3)$

Table 4.6 shows that, when applied to MIMIC, DDSCA created well-separated clusters (i.e., clusters which differ with respect to their most frequent values in demographics and ICD Chapters, and with respect to frequent sequential patterns mined from them). For example, Clusters 4, 5, and 6 have the same most frequent values in Age, Gender, and Ethnicity but are different with respect to their top-3 frequent sequential patterns. Well-separated clusters are more interpretable (e.g., they can be concisely described based on frequent patterns [186]). Note that some patterns appear in more than one clusters. This is expected because they are comprised of diagnosis codes associated with patients that have different demographics. For example, the pattern (401.9, 427.31) which appears in four clusters implies that many middle-aged, or older patients, in each cluster were associated first with hypertension and then with atrial fibrillation. This is expected because hypertension is a risk factor of atrial fibrillation, and these diagnoses are common among middle-aged and aged patients [213]. Similar results to those of Table 4.6 for INFORMS are in Section B.6 in Appendix B. The competitors created significantly less separable clusters compared to our method (e.g., their top-3 frequent sequential patterns have very low frequency), so we do not report detailed results for them.

4.7.9 The medical relevance of top-3 frequent sequential patterns in clusters

Having shown that our algorithm outperforms the competitors, we now show that it creates clusters in which frequent sequential patterns capture relationships between diagnosis codes that are documented in the medical literature. Specifically, we discuss the top-3 frequent sequential patterns in the clusters of Table 4.6 that were created by our method when applied to MIMIC (see Section B.7 in Appendix B for results on INFORMS). These patterns were discovered using the algorithm in [214]. Since the clusters are also compact and well-separated, they allow discovering potentially useful patterns associated with patients with similar demographics.

Cluster 1: Atrial fibrillation (427.31) that appears in all patterns frequently co-exists with congestive heart failure (428.0) [215], or with hypertension (401.9) [213], or with coronary artery disease (such as "coronary atherosclerosis of native coronary artery" denoted by 414.01) [216].

Cluster 2: Hypertension (401.9) frequently co-exists with coronary artery disease (414.01) [217], or with atrial fibrillation (427.31) [213], while coronary artery disease (414.01) frequently co-exists with atrial fibrillation (427.31) [216]. Interestingly, the number of patients with atrial fibrillation in Cluster 1 (containing patients aged 80 or over) is larger than that of Cluster 2 (containing patients aged 65 to less than 80). This is explained by the fact that the prevalence of atrial fibrillation increases with age [218].

Cluster 3: Hypertension (401.9) that appears in all patterns frequently co-exists with coronary artery disease (414.01) [217], or with hyperlipidemia (272.4) [219], or with pure hypercholesterolemia (272.0) [220].

Cluster 4: Esophageal reflux or Gastroesophageal reflux disease (530.81) that appears in all patterns frequently co-exists with hypertension (401.9) [221], or with coronary artery disease (414.01) [222], or with hyperlipidemia (272.4) [223].

Cluster 5: Tobacco use disorder (305.1) that appears in all patterns frequently co-occurs with hypertension (401.9) [224], or with non-dependent alcohol abuse (305.00) [225], or with diseases of the lung (such as acute respiratory failure denoted by 518.81) [226].

Cluster 6: Acidosis (276.2) frequently co-occurs with acute respiratory failure (518.81) [227], or with acute kidney failure (584.9) [228].

Cluster 7: The patients in this cluster are very similar to those in Cluster 1 regarding Age and Ethnicity but differ in Gender. Thus, two of the three patterns in Cluster 7 also appear in Cluster 1. Furthermore, all patients are very old (the most frequent Age category is "Aged, 80 and over"), thus it is expected to have frequent patterns with coronary artery disease (414.01) (see Cluster 1), congestive heart failure (428.0) (see Cluster 7), and hypertension (401.9) (see Clusters 1 and 7). The reason is that ageing predisposes to a high incidence and prevalence of coronary artery disease, congestive heart failure, and hypertension (401.9) in both males and females [229, 230].

Cluster 8: The patients in this cluster are very similar to those in Cluster 3 regarding Age, Ethnicity, ICD Chapters but differ in Gender. Thus, two of the three patterns in Cluster 8 also appear in Cluster 3. In both of these clusters, many patients have hypertension

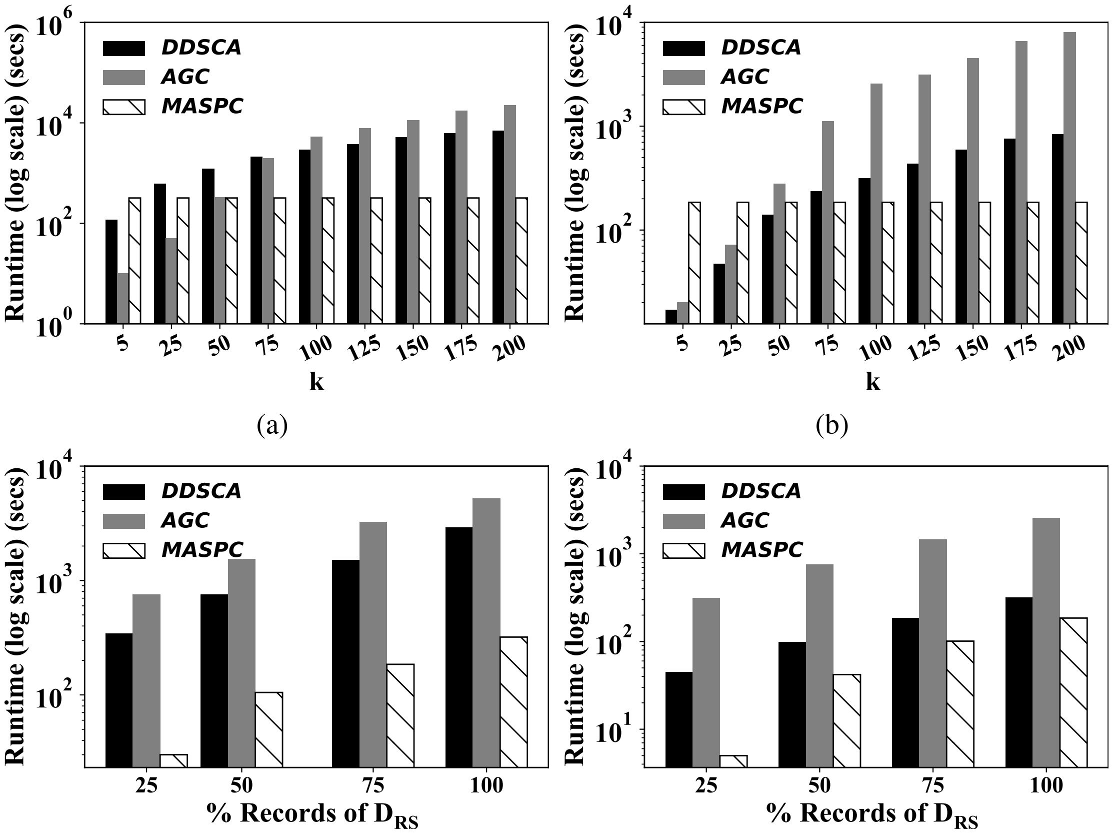
(401.9), but the number of patients with hypertension in Cluster 3 is three times larger than that in Cluster 8. This is expected because hypertension is more prevalent among males (99% of patients in Cluster 3 are male) than females (all patients in Cluster 8 are female) [231].

Cluster 9: The patterns in this cluster all contain secondary malignant neoplasm of bone and bone marrow (198.5). This diagnosis code frequently co-exists with secondary malignant neoplasm of brain and spinal cord (198.3), or with secondary malignant neoplasm of liver (197.7), or with secondary malignant neoplasm of lung (197.0). Since secondary bone cancer occurs when cancers that develop elsewhere spread, or metastasize, to the bones [232], it is expected that 198.5 frequently co-occurs with secondary cancers in other organs (198.3, or 197.7, or 197.0).

Cluster 10: Neonatal jaundice (774.2) that appears in all patterns frequently co-occurs with respiratory distress syndrome (769) [233], or with primary apnea of newborn (770.81)[234], or with feeding problems (779.3)[235].

4.7.10 Runtime

Fig. 4.7 shows the runtime of all methods for varying number of clusters k and varying number of records. DDSCA scaled better than AGC with respect to both criteria and was faster in most cases. MASPC was the fastest (except for $k < 50$), mainly because it directly added a large percentage $(32\%$ to $45\%)$ of records into a single cluster without considering their similarity to other records. This helped runtime but harmed effectiveness. As discussed above, MASPC was the least effective method. The runtime of MASPC is not affected by k, as it uses hierarchical clustering, which is insensitive to k.



(a)

(b)

(c)

(d)

Fig. 4.7 Runtime vs. k: (a) MIMIC and (b) INFORMS. Runtime vs. % of records in the input dataset for: (c) MIMIC and (d) INFORMS.

Chapter 5

Clustering sequence graphs

In application domains ranging from social networks to e-commerce, it is important to cluster users with respect to both their relationships (e.g., friendship or trust) and their actions (e.g., visited locations or rated products). Motivated by these applications, we introduce here the task of clustering the nodes of a *sequence graph*, i.e., a graph whose nodes are labeled with strings, which model, for example, sequences of users' visited locations or rated products. Both string clustering algorithms and graph clustering algorithms are inappropriate to deal with this task, as they do not consider the structure of strings and graph simultaneously. Moreover, attributed graph clustering algorithms generally construct poor solutions because they need to represent a string as a vector of attributes, which inevitably loses information and may harm clustering quality. We thus introduce the problem of clustering a sequence graph. We first propose two pairwise distance measures for sequence graphs, one based on edit distance and shortest path distance and another one based on SimRank. We then formalize the problem under each measure, showing also that it is NP-hard. In addition, we design a polynomial-time 2-approximation algorithm, as well as a heuristic for the problem. Experiments using real datasets and a case study demonstrate the effectiveness and efficiency of our methods.

5.1 Overview

Graph clustering [236] is a fundamental data mining task, which seeks to partition the nodes of an input graph, so that similar nodes form a group, referred to as cluster. The task is important in application domains such as social networks, where nodes represent users, edges represent friendship relationships between users and clustering aims to detect user communities [237]; or e-commerce, where nodes represent consumers, edges represent trust relationships between consumers and clustering aims to identify groups of consumers with bonds of trust among them [238]. The task is also important in medicine, as clustering patient similarity networks (i.e., graphs whose nodes represent patients and edges connect similar patients according to demographics or genetic mutations) allows identifying clinically homogeneous patient groups [239].

5.1.1 Motivation

However, often in these application domains, the similarity among users does not depend only on user relationships but also on actions or events associated to users. Take for example a geo-social network, such as Foursquare, where users are associated with their history of visited locations. Two connected users may naturally be regarded as more similar in the network, if they have a similar history of visited locations [240, 241]. For example, similarity with respect to both the sequence of visited locations and the social network structure is considered in recommendation [240] and location prediction [241]. Likewise, in e-commerce, two connected users may be regarded as more similar in a trust network, if they have a similar history of rated products [242]. Also, in medicine, two connected users may be regarded as more similar (e.g., with respect to disease progression), if they have a similar history of diagnoses [243].

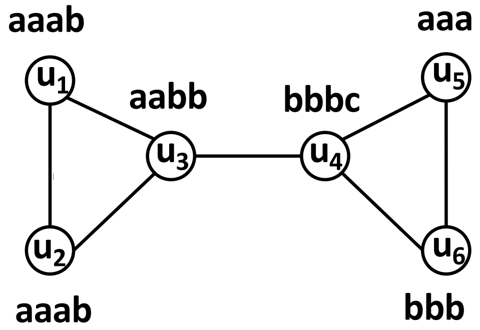


Fig. 5.1 A sequence graph (copy of Fig. 1.1).

Thus, motivated by these application domains, we introduce the problem of clustering a graph whose nodes are labeled with sequences of letters (i.e., strings). We refer to such graphs as *sequence graphs* (see Fig. 5.1). In the aforementioned application domains, an edge models a relationship between users. Alternatively, an edge in a sequence graph can model a relationship between strings (e.g., two nodes are connected, if their associated genomic sequences share sufficiently many substrings [244, 245]).

While natural, the problem of clustering a sequence graph has not been considered before, to the best of our knowledge, although graphs with sequence-labeled nodes are being used extensively in bioinformatics [244]. In addition, existing clustering methods are not appropriate to address the problem. For example, algorithms for clustering an attributed graph [246–252, 187], in which nodes are labeled with a vector of attribute values, are not appropriate for clustering a sequence graph. This is because to apply these algorithms one needs to convert a string into a vector of attribute values, which inevitably loses information and severely harms clustering quality, as we discuss in more detail in Section 5.2.

5.1.2 Contributions

Our work makes the following specific contributions:

1. We propose two measures to quantify the distance between a pair of nodes in a sequence graph. Both our measures take into account the similarity between node labels (strings) and the structural similarity of nodes; however, in different ways. Our first measure is a product metric [198], based on edit distance and shortest path distance, while our second measure is based on edit distance and SimRank [32]. Shortest path distance is efficient to compute but is based on a single path between two nodes. SimRank is less efficient to compute but takes into account all tours (paths that may have cycles) between two nodes u and v, and it aggregates similarities of multi-hop neighbors of u and v, which helps producing high-quality clustering results. We propose a fixed-point iteration algorithm for computing our SimRank-based measure. We also consider a proxy measure for each of our measures, which approximates edit distance between strings by efficiently embedding them into a Hamming distance space. This allows distance computation in linear time in the length of the two strings.

2. We formalize the problem of clustering a sequence graph under either of our measures, in the context of prototype-based clustering [236], where clusters are built around representative nodes. We consider two versions of our problem. The first is based on the k-center problem [26, 27], in which the goal is to minimize the maximum distance between any node and its closest cluster representative. The second is based on the k-median problem [26, 28], in which the goal is to minimize the sum of distances between each node and its closest representative. We show that both versions of our problem are NP-hard. For the first version, we design a 2-approximation algorithm and show that no polynomial-time algorithm achieves a better approximation guarantee. For the second version, we design a heuristic.

3. We compare our algorithms against four state-of-the-art attributed graph clustering algorithms [247, 251, 252, 187], using two datasets from trust-aware e-commerce websites. Our results show that, unlike the competitors, our algorithms create high-quality clusters containing structurally similar nodes with similar strings (e.g., users with trust bonds who are similar with respect to their history of reviewed products). Our results also show that the proxy measures we introduce do not substantially affect quality, while greatly improving runtime efficiency. For instance, our algorithm that utilizes the proxy of the product metric was up to 8 times faster than the most efficient competitor.

4. We present a case study on phylogenetic trees [253]. A phylogenetic tree is often constructed from a set of strings, each representing the genomic sequence of an organism. It is a hierarchical representation of all clusterings of the genomic sequences of the organisms and is often modeled as a binary tree whose leaf nodes are labeled with the strings. Such a tree can thus be seen as a special type of a sequence graph. Given a phylogenetic tree and a positive integer k, the computational task we consider is to cluster the leaf nodes of the tree into k clusters, by taking into account both the tree topology and the sequences corresponding to leaf nodes. We can then evaluate how accurate this clustering is by comparing it to a ground truth clustering. If these two clusterings are similar, then the phylogenetic tree is meaningful. Our results indeed show that the measures we introduce (and the corresponding clustering algorithms) can reliably evaluate whether a given phylogenetic tree is in accordance with a given ground truth clustering consisting of k clusters. On the other hand, we show that four state-of-the-art attributed graph clustering algorithms [247, 251, 252, 187] are not suitable for this task. The results of the case study also indicate that our methods could potentially be useful in other bioinformatics applications, such as evaluating phylogenetic networks [254].

5.1.3 Chapter organization

The remaining of the chapter is organized as follows. In Section 5.2, we review related work of clustering sequence graphs. In Section 5.3, we define some preliminary concepts. In Section 5.4, we present our distance measures for sequence graph clustering. In Section 5.5, we define four sequence clustering problems that we aim to solve and study their hardness. In Section 5.6, we present our algorithms for addressing these problems. In Section 5.7, we present an experimental evaluation of our algorithms. In Section 5.8, we present a case study to showcase the applicability of our algorithms.

5.2 Related Work

Our work is related to sequence clustering and graph clustering. Therefore, in the following, we briefly review algorithms for clustering a collection of sequences (strings), as well as algorithms for clustering a (non-attributed) graph (see [255, 12] for surveys). Our work is also related to attributed graph clustering. As mentioned in Section 5.1, in an attributed graph, each node is labeled with a vector of attribute values. We, therefore, review some recent works on attribute graph clustering and refer the reader to [256] for a survey. Last, we discuss the use of string-labeled graphs in bioinformatics.

5.2.1 Sequence clustering and graph clustering

Algorithms for clustering a collection of sequences (strings) measure distance between sequences directly [257], or first project the sequences into a set of patterns (e.g., q-grams) and then measure distances on the projected space [47, 245]. Alternatively, they employ generative models for the input collection of sequences, which are used to obtain likelihood-based distances between sequences [258]. In any case, the distance measures are given as input to a clustering algorithm for vector data [12] to obtain clusters.

Algorithms for clustering a graph employ graph partitioning (e.g., they solve a minimum cut problem [259]), spectral clustering [260, 261], or cohesive subgraph detection techniques [262]. Alternatively, they [263, 264] learn a node embedding into a vector space, which is then fed into a clustering algorithm for vector data.

Algorithms for clustering a collection of sequences [47, 245, 258], or a graph [262, 263, 236] are not appropriate for clustering sequence graphs, as observed in our experiments. This is because they utilize either only the strings (i.e., they cluster the collection of labels in a sequence graph while ignoring the graph structure) or only the graph structure (i.e., they cluster a sequence graph while ignoring its labels), although both the strings and graph structure determine clustering quality in our setting. Also, we cannot convert a sequence graph to a graph with string distances as edge weights and then cluster the weighted graph. This is because our clustering problem needs distances between strings of nodes that are not connected, and it is not possible to compute these distances by combining edge weights.

5.2.2 Attributed graph clustering

Algorithms for clustering an attributed graph utilize both the graph structure and the attribute values of nodes [246–252, 187]. They employ, for example, graph convolution [187], matrix factorization [248, 247], or attributed graph embedding into a vector space [251, 252].

An example of a graph convolution based algorithm is Adaptive Graph Convolution (AGC) [187], which was also mentioned in Section 4.7.2. The AGC algorithm uses graph convolution to obtain smooth feature representations of node attributes and spectral clustering. The underlying assumption of AGC is that nodes that are close in the graph will be clustered together [187]. An example of a matrix factorization based algorithm is Text-Associated DeepWalk (TADW) [247]. This algorithm is based on DeepWalk [263], which uses textual attributes to supervise random walks on graphs. Examples of embedding-based algorithms are Text Enhanced Network Embedding (TENE) [251] and Binarized Attributed Network Embedding (BANE) [252]. Both of these algorithms aim at learning a low-dimensional vector representation for each node and its associated attributes in the attributed graph. BANE uses Weisfeiler-Lehman graph kernels [265] to encode dependencies between node edges and attributes into a binary code representation. This representation encodes first-order proximities [266] between nodes. TENE aims to jointly learn vector representations based on both first-order and second-order proximities [266], as well as on the text cluster membership matrix.

Although one can represent string labels as attribute vectors (e.g., by representing a string as a vector of q-grams and their frequencies [245] or their tf-idf scores [251]) and then apply an attributed graph clustering algorithm to our problem, such a representation inevitably loses information and thus severely degrades the quality of clustering, as shown in our experiments. The reason it loses information is because one needs to assume a single order for the q-grams of all sequences to construct the vector representations of all sequences. However, the q-grams do not appear in the same order in all sequences. To illustrate this point, we provide the following example.

Example 8. *Let $T_1 = \texttt{aba}$ and $T_2 = \texttt{bab}$ be two sequences. The 2-grams of both of these sequences are* ab *and* ba *and each of these 2-grams appears only once in T_1 or in T_2. Thus, T_1 and T_2 have the same vector representation $(1, 1)$ where the first 1 denotes the frequency of* ab *and the second 1 denotes the frequency of* ba*, assuming a lexicographic order of q-grams. Since T_1 and T_2 have the same vector representation, they are treated as equal by attributed graph clustering algorithms, although they are not. Similarly, consider a dataset $\{T_1, T_2, T_3\}$ with T_1 and T_2 as before and $T_3 = \texttt{ccc}$. The vector representation of both T_1 and T_2 when using td-idf scores instead of frequencies is $(\log_2(3/2), \log_2(3/2))$, so again T_1 and T_2 are treated as equal in similarity computation. Thus, the similarity information of sequences is not captured well after they are represented based on q-grams.*

5.2.3 Sequence-labeled graphs in bioinformatics

In bioinformatics, graphs with sequence-labeled nodes have been used extensively in the following context: the nodes represent (short) DNA fragments read by sequencing technologies; and a weighted (directed) edge (u, v) represents the length of the suffix-prefix overlap between sequence u and sequence v. The goal is then to assemble these fragments into a candidate genome represented by some trail in the graph [244]. Let us stress that this task is not related to clustering.

5.3 Background

In the following, we summarize the notation used in this chapter (see Table 5.1) and introduce some preliminary concepts.

Table 5.1 Table of notation.

Notation	Definition
Σ	An alphabet
$T[i..j]$	A substring
ε	A empty string
G	A simple graph
$\mathcal{G}$	A sequence graph
V	A set of nodes
E	A set of edges
$\mathcal{S}$	A set of strings
d_E, d_H, d_{SP}, d_{ESP}, d_{ESR}	Distance functions

5.3.1 Strings

An *alphabet* Σ is a finite non-empty set of elements called *letters*; we denote its size by $|\Sigma|$. A *string* $T = T[0]T[1]\dots T[n-1]$ is a sequence of letters of *length* $|T| = n$ over Σ. For two positions i and j on T, we denote by $T[i..j] = T[i]\dots T[j]$ the *substring* of T that starts at position i and ends at position j of T. By ε we denote the *empty string* of length 0. We refer to a length-q substring of a string T as a q-gram (e.g., in Fig. 5.1, `aab` is a 3-gram of string `aaab`).

5.3.2 Graphs

Let $G = (V, E)$ be a simple graph[1], where V is a set of nodes and $E \subseteq V \times V$ is a set of edges. The set of neighbors of a node $u \in V$ is $n(u) = \{v \in V \mid (u, v) \in E\}$. The size $|n(u)|$ of $n(u)$ is the *degree* of u. A path p between two nodes u and v in G is a sequence of edges $e_1, \dots, e_{|p|}$ such that u is the start node of e_1 and v is the end node of $e_{|p|}$ and all nodes are distinct (i.e., p has no cycles). A tour is a path that may have cycles.

A *sequence graph* $\mathcal{G} = (V, E, \mathcal{S}, \mathcal{F})$ is a tuple, where V is a set of nodes, $E \subseteq V \times V$ is a set of edges, $\mathcal{S}$ is a set of strings drawn from an alphabet $\Sigma_{\mathcal{S}}$, and $\mathcal{F} : V \rightarrow \mathcal{S} \cup \varepsilon$ is a function that outputs a string from $\mathcal{S}$ or the empty string. In particular,

[1] A simple graph is an unweighted, undirected graph with no loops or multiple edges.

each node $u \in V$ is associated with a string $\mathcal{F}(u) \in \mathcal{S} \cup \varepsilon$. For example, in Fig. 5.1, $\mathcal{S} = \{\texttt{aaa}, \texttt{aaab}, \texttt{aabb}, \texttt{bbb}, \texttt{bbbc}\}$, $\Sigma_{\mathcal{S}} = \{\texttt{a}, \texttt{b}, \texttt{c}\}$, and $\mathcal{F}(u_1) = \texttt{aaab}$.

A *k-clustering* of $\mathcal{G}$ for $k \in [1, |V|]$ is a partition of V into k subsets, called *clusters*. We may omit k from k-clustering when it is clear from the context.

5.3.3 Distance Measures

The *edit distance* $d_E(U, V)$ between two strings U and V is defined as the minimum number of elementary edit operations (letter insertion, deletion, or substitution) to transform U to V. For U and V of equal length, the *Hamming distance* $d_H(U, V)$ is defined as the minimum number of substitutions to transform U to V. For example, $d_E(\texttt{aaab}, \texttt{aabb}) = d_H(\texttt{aaab}, \texttt{aabb}) = 1$. If U and V are not of the same length, we set $d_H(U, V) = \infty$, for completeness.

The *shortest path distance* $d_{SP}(u, v)$ for two nodes u and v of a graph is defined as the length of the shortest path between u and v. For completeness, we set $d_{SP}(u, v) = \infty$, if there is no path between u and v. For example, in Fig. 5.1, $d_{SP}(u_1, u_3) = 1$.

Given a constant $c \in (0, 1)$, referred to as *decay factor*, the *SimRank score* $SR(u, v)$ between u and v is defined as follows [32]:

$$SR(u,v) = \begin{cases} 1, & \text{if } u = v \\ 0, & \text{if } n(u) = \emptyset \text{ or } n(v) = \emptyset \\ \frac{c}{|n(u)||n(v)|} \sum\limits_{u' \in n(u)} \sum\limits_{v' \in n(v)} SR\left(u', v'\right), & \text{otherwise.} \end{cases} \tag{5.1}$$

The intuition behind SimRank is that two nodes are similar if they are connected to similar nodes. SimRank aggregates similarities based on paths.

A *metric space* (X, d) is an ordered pair, where X is a set and d is a metric on X. For example, the set of strings $\mathcal{S}$ of a sequence graph $\mathcal{G}$ together with the edit distance d_E, which is a metric, define the metric space $(\mathcal{S}, d_E)$.

5.4 Distance Measures for Sequence Graphs

In the following, we discuss our distance measures for sequence graphs.

5.4.1 The d_{ESP} Measure

Given a sequence graph $\mathcal{G} = (V, E, \mathcal{S}, \mathcal{F})$, metric spaces $(\mathcal{S}, d_E)$ and (V, d_{SP}), nodes $u, v \in V$ and strings $\mathcal{F}(u), \mathcal{F}(v)$, the d_{ESP} (E is for Edit distance and SP for Shortest Path distance) measure is defined as:

$$d_{ESP}((\mathcal{F}(u), \mathcal{F}(v)), (u, v)) = \sqrt{(d_E(\mathcal{F}(u), \mathcal{F}(v)))^2 + (d_{SP}(u, v))^2}.$$

The d_{ESP} measure considers the distance between two nodes based on the edit distance between the strings of the nodes and the shortest path distance between the nodes. For example, in Fig. 5.1, $d_{ESP}(u_1, u_3) = \sqrt{2}$ because $d_E(\texttt{aaab}, \texttt{aabb}) = 1$ and $d_{SP}(u_1, u_3) = 1$. That is, d_{ESP} combines the two metric spaces $(\mathcal{S}, d_E)$ and (V, d_{SP}) into a metric space which measures similarity among string-labeled nodes, as if they were points in a 2D space. Note that d_{ESP} is a metric. This is because both edit distance and shortest path distance are metrics [267, 268] and d_{ESP} is a 2-product metric [198] on the Cartesian product of the set of strings $\mathcal{S}$ and the set of nodes V in a sequence graph $\mathcal{G} = (V, E, \mathcal{S}, \mathcal{F})$.

5.4.2 The d_{ESR} Measure

Our d_{ESR} (E is for Edit distance and SR is for SimRank) measure captures the intuition of SimRank, according to which u and v are similar when they are reachable by similar nodes. However, d_{ESR} differs from SimRank in that it also considers the similarity of the strings of u and v and the strings of their reachable nodes. For example, among two node pairs with the same SimRank score (e.g., u_1, u_5 and u_1, u_6 in Fig. 5.1), d_{ESR} treats the node pair having more similar strings with respect to edit distance as more similar (e.g., in Fig. 5.1, $d_{ESR}(u_1, u_5) < d_{ESR}(u_1, u_6)$).

d_{ESR} is based on ESR, a similarity score over a pair of nodes u and v, defined in Eq. 5.2. Specifically, we define $d_{ESR}(u, v) = 1 - ESR(u, v)$.

$$ESR(u, v) = \begin{cases} 1, & \text{if } u = v \\ 0, & \text{if } n(u) = \emptyset \text{ or } n(v) = \emptyset \\ \frac{\sigma(u,v)}{|n(u)||n(v)|} \sum\limits_{u' \in n(u)} \sum\limits_{v' \in n(v)} ESR\left(u', v'\right), & \text{otherwise} \end{cases} \tag{5.2}$$

where $\sigma(u, v) = (1 - \frac{d_E(\mathcal{F}(u), \mathcal{F}(v))}{\max(|\mathcal{F}(u)|, |\mathcal{F}(v)|)}) \cdot (1 - \gamma)$ is a similarity score between strings and d_E is the edit distance. Note that $\sigma(u, v)$ is computed by subtracting the normalized edit distance from 1 and multiplying by $(1-\gamma)$ for a small real number $\gamma > 0$. This ensures that $\sigma(u, v) < 1$ for any pair of nodes (u, v), which is required for the iterative computation of ESR (see Theorem 4). Importantly, it also ensures that the multiplication does not substantially change similarity (i.e., the values of $\sigma(u, v)$ and of $(1 - \frac{d_E(\mathcal{F}(u),\mathcal{F}(v))}{\max(|\mathcal{F}(u)|,|\mathcal{F}(v)|)})$ are nearly equal).

Note that ESR cannot be derived from SimRank by setting $c = \sigma(u, v)$ in Eq. 5.1, since in SimRank $c > 0$ [32] while $\sigma(u, v)$ may be 0. Furthermore, ESR is not a simple weighted version of SimRank with $c = 1$ and weight $\sigma(u, v)$, since $c < 1$ in Eq. 5.1. ESR is also different from SemSim [269], which was developed for attributed graphs and assumes a large value of normalized semantic similarity between any pair of values [2]. Also, it is easy to see that ESR differs from d_{ESP} in that it considers all reachable nodes from u and v as well as their strings, while d_{ESP} considers only the strings of u and v.

In the following, we show that a fixed-point iteration method, similar to that developed for SimRank [32], can compute our ESR measure. Specifically, in Theorem 4, we define a function $R_\ell(u, v)$, for a node pair u, v and an integer $\ell \geq 0$, which quantifies the similarity of u and v. Next we prove that $R_\ell(u, v)$ can be computed iteratively and that the values of $R_\ell(u, v)$ converge to $ESR(u, v)$. Clearly, our measure can also benefit from efficiency optimizations of SimRank (e.g., [270]) and be extended to address the "zero-similarity" problem [271].

Theorem 4. *Let $u, v \in V$ be a pair of nodes in $\mathcal{G}$ and $R_\ell(u, v)$ be a recursive function, defined as follows for some integer $\ell \geq 0$:*

$$R_{\ell+1}(u,v) = \begin{cases} 1, & u = v \\ 0, & n(u) = \emptyset \textit{ or } n(v) = \emptyset \\ \frac{\sigma(u,v)}{|n(u)||n(v)|} \sum\limits_{u' \in n(u)} \sum\limits_{v' \in n(v)} R_\ell\left(u', v'\right), & otherwise \end{cases} \tag{5.3}$$

with $R_0(u, v) = \begin{cases} 1, & \textit{for } u = v \\ 0, & \textit{otherwise} \end{cases}$. Then, for each $u, v \in V$, there exists a unique solution to Eq. 5.3 such that $\lim_{\ell \to \infty} R_\ell(u, v) = ESR(u, v)$.

Proof.

We first show the following three properties for R_ℓ:

I Symmetry: $R_\ell(u, v) = R_\ell(v, u)$.

II Maximum self similarity: $R_\ell(u, u) = 1$.

[2] Specifically, SemSim uses a decay factor (alike c in SimRank) that must be smaller than the semantic similarity between the attribute vectors of any two nodes. The latter is zero, or near-zero, in sequence graphs, as they contain empty strings, or strings that are largely different, which makes SemSim inapplicable to our setting.

III Monotonicity: $0 \leq R_\ell(u,v) \leq R_{\ell+1}(u,v) \leq 1$.

Then, we will use these properties for proving that the solution of R_ℓ always *exists* and is *unique* to complete the proof.

Symmetry and Maximum self similarity: Properties I and II hold due to the definition of R_ℓ.

Monotonicity: For $u = v$, $R_0(u,v) = R_1(u,v) = \ldots = 1$, so clearly Property III holds. For $u \neq v$ with $n(u) = \emptyset$ or $n(v) = \emptyset$, Property III holds due to the definition of R_ℓ. In any other case, we show that Property III also holds by induction. (Induction base) For $\ell = 0$, the definition of R_ℓ implies:

$$0 \leq R_0(u,v) \leq R_1(u,v) \leq 1. \tag{5.4}$$

(Induction hypothesis) For an integer $\ell > 0$, $0 \leq R_{\ell-1}(u,v) \leq R_\ell(u,v) \leq 1$ holds.

From the induction hypothesis, it holds that:
$R_{\ell+1}(u,v) - R_\ell(u,v) = \frac{\sigma(u,v)}{|n(u)||n(v)|} \cdot \sum_{u' \in n(u)} \sum_{v' \in n(v)} \left(R_\ell(u',v') - R_{\ell-1}(u',v') \right).$

$R_\ell(u',v') \geq R_{\ell-1}(u',v')$ holds for any integer $\ell > 0$ and any $u', v' \in V$. Specifically, for $u' = v'$, as well as for u', v' such that $n(u') = \emptyset$ or $n(v') = \emptyset$, $R_\ell(u',v') \geq R_{\ell-1}(u',v')$ holds from the definition of R_ℓ. In any other case, $R_\ell(u',v') \geq R_{\ell-1}(u',v')$ holds from the inductive hypothesis. Also, $\frac{\sigma(u,v)}{|n(u)||n(v)|} \geq 0$ holds by definition. Thus, $R_{\ell+1}(u,v) - R_\ell(u,v) \geq 0 \Rightarrow R_{\ell+1}(u,v) \geq R_\ell(u,v)$ holds for $\ell > 0$. Therefore, it suffices to show that $0 \leq R_\ell(u,v)$ and $R_{\ell+1}(u,v) \leq 1$ hold, for $\ell > 0$. The first inequality holds, due to the induction hypothesis. For the second inequality, we have:

$$\begin{aligned} R_{\ell+1}(u,v) &= \frac{\sigma(u,v)}{|n(u)||n(v)|} \sum_{u' \in n(u)} \sum_{v' \in n(v)} R_\ell\left(u',v'\right) && (5.5) \\ &\leq \frac{\sigma(a,b)}{|n(a)||n(b)|} \cdot |n(a)||n(b)| \cdot 1 && (5.6) \\ &\leq 1. && (5.7) \end{aligned}$$

Eq. 5.5 is due to Eq. 5.3. Eq. 5.6 is due to the induction hypothesis. Eq 5.7 is due to the definition of $\sigma(\cdot,\cdot)$. Therefore, $0 \leq R_\ell(u,v) \leq R_{\ell+1}(u,v) \leq 1$ holds for $\ell > 0$, and thus we have shown by induction that $0 \leq R_\ell(u,v) \leq R_\ell(u,v) \leq 1$ holds for any $\ell \geq 0$.

Existence: For $u = v$, or for $u \neq v$ such that $n(u) = \emptyset$ or $n(v) = \emptyset$, it is clear that $\lim_{\ell \to \infty} R_{\ell+1}(u,v) = ESR(u,v)$. In any other case, by the Monotone Convergence Theorem [272, Section 2.5.1, Theorem 1], we know that the series $\{R_\ell(u,v)\}$ for any u, v converges, since it is non-decreasing and bounded due to Property III. Furthermore, by

Corollary 1 in [272, Section 2.5.1], we know that $\{R_\ell(u,v)\}$ tends to its least upper bound. Thus, it holds $\lim_{\ell\to\infty} R_{\ell+1}(u,v) = \lim_{\ell\to\infty} R_\ell(u,v)$. Therefore,

$$\lim_{\ell\to\infty} R_{\ell+1}(u,v) = \frac{\sigma(u,v)}{|n(u)||n(v)|} \cdot \lim_{\ell\to\infty} \sum_{u'\in n(u)} \sum_{v'\in n(v)} R_\ell(u',v') \tag{5.8}$$

$$= \frac{\sigma(u,v)}{|n(u)||n(v)|} \cdot \sum_{u'\in n(u)} \sum_{v'\in n(v)} \lim_{\ell\to\infty} R_\ell(u',v'). \tag{5.9}$$

Eq. 5.8 is due to the Sum Rule of limits [272, Section 5.1.3, Theorem 3]. Let $R(u,v) = \lim_{\ell\to\infty} R_{\ell+1}(u,v)$. Then, from Eq. 5.9, we have

$$R(u,v) = \frac{\sigma(u,v)}{|n(u)||n(v)|} \cdot \sum_{u'\in n(u)} \sum_{v'\in n(v)} R(u',v').$$

This shows that the limit of $R_\ell(\cdot,\cdot)$ with respect to ℓ satisfies the ESR equation for $u \neq v$ with $n(u) \neq \emptyset$ and $n(v) \neq \emptyset$. Thus, we have shown existence.

Uniqueness: let $M = \max_{(u,v)} |s(u,v) - s'(u,v)|$, where $s(u,v)$ and $s'(u,v)$ are two solutions to the ESR, for some pair $u, v \in V$. It suffices to show $M = 0$, as this means $s(u,v)$ and $s'(u,v)$ are the same (i.e., the solution is unique). Let $M = |s(a,b) - s'(a,b)|$ for some $a, b \in V$. For $a = b$, $M = |1 - 1| = 0$ due to Property II. For $a \neq b$ such that $n(u) = \emptyset$ or $n(v) = \emptyset$, $M = 0$. For a, b such that $n(a) \neq \emptyset$ and $n(b) \neq \emptyset$, we distinguish two cases for $\sigma(a,b)$. For $\sigma(a,b) = 0$, clearly $M = 0$. For $\sigma(a,b) > 0$, we have:

$$M = |s(a,b) - s'(a,b)| \tag{5.10}$$

$$= |\frac{\sigma(a,b)}{|n(a)||n(b)|} \cdot \sum_{u'\in n(a)} \sum_{v'\in n(b)} (s(u',v') - s'(u',v'))| \tag{5.11}$$

$$= |\frac{\sigma(a,b)}{|n(a)||n(b)|}| \cdot | \sum_{u'\in n(a)} \sum_{v'\in n(b)} (s(u',v') - s'(u',v'))| \tag{5.12}$$

$$\leq \frac{\sigma(a,b)}{|n(a)||n(b)|} \cdot \sum_{u'\in n(a)} \sum_{v'\in n(b)} |s(u',v') - s'(u',v')| \tag{5.13}$$

$$\leq \frac{\sigma(a,b)}{|n(a)||n(b)|} \cdot |n(a)||n(b)| M \tag{5.14}$$

$$\leq \sigma(a,b) \cdot M. \tag{5.15}$$

Eq. 5.10 is by the definition of M. Eq 5.11 is by Eq. 5.2. Eq 5.12 is by the multiplicativity of absolute value. Eq 5.13 is by the triangle inequality of absolute value. Eq 5.14 is because each $|s(u',v') - s'(u',v')| \leq M$ by the definition of M. Since $\sigma(a,b) > 0$

in the case we examine, $\sigma(a, b) < 1$ by definition, and $M \geq 0$ by definition, Eq. 5.15 (i.e., $M \leq \sigma(a, b) \cdot M$) holds if and only if $M = 0$. To see this, let $M > 0$. Then, $\frac{M}{M} = 1 \leq \sigma(a, b)$, which does not hold by the definition of $\sigma(\cdot, \cdot)$. □

5.4.3 Proxy Measures for d_{ESP} and d_{ESR}

Since exact edit distance computation requires quadratic time (assuming the Strong Exponential Time Hypothesis (SETH) is true [273]), d_{ESP} and d_{ESR} are expensive to compute for long strings. Therefore, we propose a proxy $d_{\widehat{ESP}}$ and $d_{\widehat{ESR}}$ for d_{ESP} and d_{ESR}, respectively. Instead of considering the strings of a pair of nodes, the proxies consider embeddings of these strings into Hamming distance space. An embedding from a metric space M_1 to a metric space M_2 is a mapping of points from M_1 to M_2 such that distances are preserved up to some factor D known as *the distortion*. We employ the CGK algorithm [274], which provides a probabilistic embedding with *linear* distortion. The CGK algorithm runs in *linear* time, and if the edit distance between two input strings is K, then the Hamming distance between their embeddings is between $K/2$ and $O(K^2)$ with *good* probability [274]. By incorporating CGK instead of edit distance in our algorithms, we substantially improve their efficiency without substantially degrading effectiveness, as it will be shown in Section 5.7.

5.5 Sequence Graph k-Center and k-Median

We present two clustering problems for sequence graphs inspired by the k-Center and the k-Median problems [26–28].

5.5.1 Sequence Graph k-Center (SGC)

The SGC problem, defined in Problem 3 below, requires finding k nodes, referred to as *centers*, such that the maximum distance of any node to a closest center with respect to d_{ESP} is minimized.

Problem 3 (Sequence Graph k-Center (SGC)). *Given a sequence graph $\mathcal{G} = (V, E, \mathcal{S}, \mathcal{F})$ and an integer $k \in [1, |V|]$, find a subset $\mathcal{C} \subseteq V$ of k nodes such that $\max_{u \in V} \min_{c \in \mathcal{C}} d_{ESP}(u, c)$ is minimized.*

A clustering $\{V_1, \ldots, V_k\}$ of $\mathcal{G}$ is obtained from a solution $\mathcal{C}$ to SGC, by assigning into each cluster V_i, $i \in [1, k]$, a center $c_i \in \mathcal{C}$ and also every node $u \notin \mathcal{C}$ that is closer to this center compared to other centers (i.e., every u such that $d_{ESP}(u, c_i) < d_{ESP}(u, c_j)$, for each $j \neq i$). If a node is in equal distance from multiple clusters, it is assigned into an arbitrarily selected cluster containing one of these centers.

The *decision version* of SGC asks whether there exists a subset $\mathcal{C} \subseteq V$ of k nodes such that

$$\max_{u \in V} \min_{c \in \mathcal{C}} d_{ESP}(u, c) \leq B,$$

for a given real number B.

We show that SGC is NP-hard by showing that its decision version is NP-complete. We prove this result via a reduction from the decision version of metric k-Center [27], which is known to be NP-complete [27].

Problem 4 (Metric k-Center (decision version) [27])**.** *Given a metric space (P, d), where P is a set of n points and $d : P \times P \rightarrow \mathbb{R}^+$ is a distance function, an integer $k \in [1, n]$, and a real number B, decide whether there exists a subset $C \subseteq P$ of k points such that $\max_{p \in P} \min_{c \in C} d(p, c) \leq B$.*

Lemma 1. *The decision version of SGC is NP-complete.*

Proof. The decision version of SGC is clearly in NP. In the following, we show that it can be reduced from the decision version of metric k-Center. Given any instance $\mathcal{I}_{kC}$ of the decision version of metric k-Center, we construct an instance $\mathcal{I}_{SGC}$ of the decision version of SGC in polynomial time, by creating a sequence graph $\mathcal{G} = (V, E, \mathcal{S}, \mathcal{F})$ as follows: (I) We add a node u into V, for each point $p \in P$, where P is the set of points in $\mathcal{I}_{kC}$. (II) We set $E = \emptyset$. (III) We set $\mathcal{S} = \{\varepsilon\}$. (IV) We define $\mathcal{F}$ such that $\mathcal{F}(u) = \varepsilon$, for each $u \in V$. We also set k and B in $\mathcal{I}_{SGC}$ to k and B in $\mathcal{I}_{kC}$, respectively, and $d_{ESP}((\mathcal{F}(u), \mathcal{F}(v)), (u, v)) = d(p, p')$, for every pair $(p, p') \in P$ corresponding to pair $(u, v) \in V$. This completes the construction of $\mathcal{I}_{SGC}$.

In the following, we prove that $\mathcal{I}_{SGC}$ has a positive answer if and only if $\mathcal{I}_{kC}$ has a positive answer.

($\Rightarrow$) If $\mathcal{I}_{SGC}$ has a positive answer, there is a subset $\mathcal{C} \subseteq V$ of k nodes such that $\max_{u \in V} \min_{c \in \mathcal{C}} d_{ESP}(u, c) \leq B$. These nodes correspond to a subset $C \subseteq P$ of k points such that $\max_{p \in P} \min_{c \in C} d(p, c) \leq B$. Thus, $\mathcal{I}_{kC}$ has a positive answer.

($\Leftarrow$) If $\mathcal{I}_{kC}$ has a positive answer, then there is a subset $C \subseteq P$ of k points such that $\max_{p \in P} \min_{c \in C} d(p, c) \leq B$. These points correspond to a subset $\mathcal{C} \subseteq V$ of k nodes such that $\max_{u \in V} \min_{c \in \mathcal{C}} d_{ESP}(u, c) \leq B$. Thus, $\mathcal{I}_{SGC}$ has a positive answer. □

Due to Lemma 1, we obtain Theorem 5 directly.

Theorem 5. *SGC is NP-hard.*

Proof. The statement follows from Lemma 1. □

We also consider the following variant of SGC which uses d_{ESR} instead of d_{ESP}:

Problem 5 (Sequence Graph k-Center (SGC) with d_{ESR})**.** *Given a sequence graph $\mathcal{G} = (V, E, \mathcal{S}, \mathcal{F})$ and an integer $k \in [1, |V|]$, find a subset $\mathcal{C} \subseteq V$ of k nodes such that $\max_{u \in V} \min_{c \in \mathcal{C}} d_{ESR}(u, c)$ is minimized.*

The NP-hardness of the variant follows from a similar reduction to that of Lemma 1 (omitted). The main change is that we reduce from the decision version of the k-center problem in which d is a dissimilarity function (not necessarily a metric) [26] and that we use d_{ESR} instead of d_{ESP}.

5.5.2 Sequence Graph k-Median (SGM)

The Sequence Graph k-Median (SGM) problem requires finding k nodes, referred to as *representatives*, such that the sum of distances between nodes and their closest representative with respect to d_{ESP} is minimized.

Problem 6 (Sequence Graph k-Median (SGM))**.** *Given a sequence graph $\mathcal{G} = (V, E, \mathcal{S}, \mathcal{F})$ and an integer $k \in [1, |V|]$, find a set $\mathcal{C} \subseteq V$ of k nodes such that $\sum_{u \in V} \min_{c \in \mathcal{C}} d_{ESP}(u, c)$ is minimized.*

Then, a clustering $\{V_1, \ldots, V_k\}$ is obtained from $\mathcal{C}$ as in the SGC problem.

We prove that SGM is NP-hard by reducing the decision version of SGM from the decision version of the metric k-Median problem, which is NP-complete [28]. The decision version of SGM asks whether there exists a subset $\mathcal{C} \subseteq V$ of k nodes such that $\sum_{u \in V} \min_{c \in \mathcal{C}} d_{ESP}(u, c) \leq B$, for a given real number B.

Problem 7 (Metric k-Median (decision version) [28])**.** *Given a metric space (P, d), where P is a set of n points and $d : P \times P \to \mathbb{R}^+$ is a distance function, an integer $k \in [1, n]$, and a real number T, decide whether there exists a subset $C \subseteq P$ such that $\sum_{p \in P} \min_{c \in C} d(p, c) \leq T$.*

Lemma 2. *The decision version of SGM is NP-complete.*

Proof. Given any instance $\mathcal{I}_{kM}$ of the decision version of metric k-Median, we construct an instance $\mathcal{I}_{SGM}$ of the decision version of SGM in polynomial time, by creating a sequence graph $\mathcal{G} = (V, E, \mathcal{S}, \mathcal{F})$ as follows: (I) We add a node u into V, for each point $p \in P$, where P is the set of points in $\mathcal{I}_{kM}$. (II) We set $E = \emptyset$. (III) We set

$\mathcal{S} = \{\varepsilon\}$. (IV) We define $\mathcal{F}$ such that $\mathcal{F}(u) = \varepsilon$ for each $u \in V$. We also set k in $\mathcal{I}_{SGM}$ to k in $\mathcal{I}_{kM}$, and $d_{ESP}(u, v) = d(p, p')$, for every pair $(p, p') \in P$ corresponding to pair $(u, v) \in V$. Last, we set B in $\mathcal{I}_{SGM}$ to T. This completes the construction.

In the following, we prove that $\mathcal{I}_{SGM}$ has a positive answer if and only if $\mathcal{I}_{kM}$ has a positive answer.

($\Rightarrow$) If $\mathcal{I}_{SGM}$ has a positive answer, then there is a subset $\mathcal{C} \subseteq V$ of k nodes such that $\sum_{u \in V} \min_{c \in \mathcal{C}} d_{ESP}(u, c) \leq B$. These nodes correspond to a subset $C \subseteq P$ of k points such that $\sum_{p \in P} \min_{c \in C} d(p, c) \leq T$. Thus, $\mathcal{I}_{kM}$ has a positive answer.

($\Leftarrow$) If $\mathcal{I}_{kM}$ has a positive answer, then there is a subset $C \subseteq P$ of k points such that $\sum_{p \in P} \min_{c \in C} d(p, c) \leq T$. These points correspond to a subset $\mathcal{C} \subseteq V$ of k nodes such that $\sum_{u \in V} \min_{c \in \mathcal{C}} d_{ESP}(u, c) \leq B$. Thus, $\mathcal{I}_{SGM}$ has a positive answer. □

Due to Lemma 2, we obtain Theorem 6 directly.

Theorem 6. *SGM is NP-hard.*

Proof. The statement follows from Lemma 2. □

We also consider a variant of SGM which uses d_{ESR} instead of d_{ESP}:

Problem 8 (Sequence Graph k-Median (SGM) with d_{ESR})**.** *Given a sequence graph $\mathcal{G} = (V, E, \mathcal{S}, \mathcal{F})$ and an integer $k \in [1, |V|]$, find a set $\mathcal{C} \subseteq V$ of k nodes such that $\sum_{u \in V} \min_{c \in \mathcal{C}} d_{ESR}(u, c)$ is minimized.*

The decision version of Problem 8 asks whether there exists a subset $\mathcal{C} \subseteq V$ of k nodes such that $\sum_{u \in V} \min_{c \in \mathcal{C}} d_{ESR}(u, c) \leq B$, for a given real number B. Below, we provide a reduction from the decision version of k-Median [26], which implies that Problem 8 is NP-hard.

Lemma 3. *The decision version of Problem 8 is NP-complete.*

Proof. Given any instance $\mathcal{I}_{kM}$ of the decision version of k-Median, we construct an instance $\mathcal{I}_{P2}$ of the decision version of Problem 8 in polynomial time, by creating a sequence graph $\mathcal{G} = (V, E, \mathcal{S}, \mathcal{F})$ as follows: (I) We add a node u into V, for each point $p \in P$, where P is the set of points in the decision version of k-Median. (II) We set $E = \emptyset$. (III) We set $\mathcal{S} = \{\varepsilon\}$. (IV) We define $\mathcal{F}$ such that $\mathcal{F}(u) = \varepsilon$ for each $u \in V$. We also set k in $\mathcal{I}_{P2}$ to k in $\mathcal{I}_{kM}$, and set $d_{ESR}(u, v) = d(p, p')/(\max_{(p,p') \in P} d(p, p') + 1)$, for every pair $(p, p') \in P$ corresponding to pair $(u, v) \in V$, where $d : P \times P \to \mathbb{R}^+$ is the dissimilarity function in the decision version of k-Median. For brevity, we denote $\max_{(p,p') \in P} d(p, p') + 1$ by μ. The division with μ ensures that a value in $[0, 1]$ is assigned to d_{ESR} and is needed because d_{ESR} takes values in $[0, 1]$, whereas d takes values in $\mathbb{R}^+$.

Clearly, μ can be computed in polynomial time. Last, we set B in $\mathcal{I}_{P2}$ to $\frac{T}{\mu}$. This completes the construction.

In the following, we prove that $\mathcal{I}_{P2}$ has a positive answer if and only if $\mathcal{I}_{kM}$ has a positive answer.

($\Rightarrow$) If $\mathcal{I}_{P2}$ has a positive answer, then there is a subset $\mathcal{C} \subseteq V$ of k nodes such that $\sum_{u \in V} \min_{c \in \mathcal{C}} d_{ESR}(u, c) \leq B$. These nodes correspond to a subset $C \subseteq P$ of k points such that $\sum_{p \in P} \min_{c' \in C} d(p, c') \leq T$, since $d_{ESR}(u, c) = \frac{d(p,c')}{\mu}$, for every pair of nodes $(u, c) \in \mathcal{C}$ corresponding to a pair of points $(p, c') \in C$, and $B = \frac{T}{\mu}$. Thus, $\mathcal{I}_{kM}$ has a positive answer.

($\Leftarrow$) If $\mathcal{I}_{kM}$ has a positive answer, then there is a subset $C \subseteq P$ of k points such that $\sum_{p \in P} \min_{c' \in C} d(p, c') \leq T$. These points correspond to a subset $\mathcal{C} \subseteq V$ of k nodes such that $\sum_{u \in V} \min_{c \in \mathcal{C}} d_{ESR}(u, c) \leq B$, since $d(p, c') = d_{ESR}(u, c) \cdot \mu$, for every pair of pair of points $(p, c') \in C$ corresponding to a pair of nodes $(u, c) \in \mathcal{C}$, and $T = B \cdot \mu$. Thus, $\mathcal{I}_{P2}$ has a positive answer. □

5.6 Algorithms for Clustering Sequence Graphs

We present algorithms for the SGC and SGM problems. These algorithms are presented with d_{ESP} but they can also use d_{ESR}, or the proxies of these measures, instead (see Section 5.7).

5.6.1 Approximation Algorithm for SGC

We begin by showing a polynomial-time approximation-preserving reduction from SGC to (the optimization version of) metric k-Center [27] defined below. This allows approximating SGC within a factor of 2.

Problem 9 (Metric k-Center [27]). *Given a metric space (P, d), where P is a set of n points and $d : P \times P \to \mathbb{R}^+$ is a distance function, and an integer $k \in [1, n]$, find a subset $C \subseteq P$ of k points such that* $\max_{p \in P} \min_{c \in C} d(p, c)$ *is minimized.*

Lemma 4. *SGC can be reduced to metric k-Center.*

Proof. Given any instance $\mathcal{I}_{SGC}$ of SGC, we construct an instance $\mathcal{I}_{kC}$ of metric k-Center in polynomial time, as follows: (I) We construct a set P of points by adding into an initially empty set P a point p, for each node $u \in V$ in the graph of $\mathcal{I}_{SGC}$. (II) We set k and B in $\mathcal{I}_{kC}$ to k and B in $\mathcal{I}_{SGC}$, respectively. (III) We set $d(p, p') =$

$d_{ESP}((\mathcal{F}(u), \mathcal{F}(v)), (u, v))$, for every pair of nodes $(u, v) \in V$ corresponding to a pair of points $(p, p') \in P$. This completes the construction of $\mathcal{I}_{kC}$.

In the following, we prove the correspondence between a solution $\mathcal{S}_{kC}$ to $\mathcal{I}_{kC}$ and a solution $\mathcal{S}_{SGC}$ to $\mathcal{I}_{SGC}$.

($\Rightarrow$) If $\mathcal{S}_{kC}$ is a solution to $\mathcal{I}_{kC}$, then there is a subset $C \subseteq P$ of k points such that $\max_{p \in P} \min_{c \in C} d(p, c)$ is minimum. These points correspond to a subset $\mathcal{C} \subseteq V$ of k nodes such that $\max_{u \in V} \min_{c \in \mathcal{C}} d_{ESP}(u, c)$ is minimum. Thus, $\mathcal{S}_{SGC}$ is a solution to $\mathcal{I}_{SGC}$.

($\Leftarrow$) If $\mathcal{S}_{SGC}$ is a solution to $\mathcal{I}_{SGC}$, then there is a subset $\mathcal{C} \subseteq V$ of k nodes such that $\max_{u \in V} \min_{c \in \mathcal{C}} d_{ESP}(u, c)$ is minimum. These nodes correspond a subset $C \subseteq P$ of k points such that $\max_{p \in P} \min_{c \in C} d(p, c)$ is minimum. Thus, $\mathcal{S}_{kC}$ is a solution to $\mathcal{I}_{kC}$. □

Theorem 7. *SGC can be approximated within a factor of* 2*, for any* $\epsilon > 0$.

Proof. The statement follows from Lemma 4. □

We also show that SGC cannot be approximated within a $2 - \epsilon$ factor, for any $\epsilon > 0$, by reducing metric k-Center to SGC.

Theorem 8. *SGC cannot be approximated within a* $2 - \epsilon$ *factor, for any* $\epsilon > 0$.

Proof. Metric k-Center cannot be approximated within a $2-\epsilon$ factor for any $\epsilon > 0$ [27]. Thus, it suffices to reduce k-Center to SGC.

Given any instance $\mathcal{I}_{kC}$ of metric k-Center, we construct an instance $\mathcal{I}_{SGC}$ of SGC in polynomial time, as follows. First, we create a sequence graph $\mathcal{G} = (V, E, \mathcal{S}, \mathcal{F})$ as follows: (I) We add a node u into V, for each point $p \in P$, where P is the set of points in $\mathcal{I}_{kC}$. (II) We set $E = \emptyset$. (III) We set $\mathcal{S} = \{\varepsilon\}$. (IV) We define $\mathcal{F}$ such that $\mathcal{F}(u) = \varepsilon$, for each $u \in V$. We also set k in $\mathcal{I}_{SGC}$ to k in $\mathcal{I}_{kC}$, and $d_{ESP}((\mathcal{F}(u), \mathcal{F}(v))) = d(p, p')$, for every pair $(p, p') \in P$ corresponding to pair $(u, v) \in V$. This completes the construction of $\mathcal{I}_{SGC}$.

The correspondence between a solution $\mathcal{S}_{SGC}$ to $\mathcal{I}_{SGC}$ and a solution $\mathcal{S}_{kC}$ to $\mathcal{I}_{kC}$ holds due to Lemma 4. □

Therefore, we develop SGC-APPROX, a 2-approximation algorithm for SGC which is based on the algorithm of Gonzalez [185] for metric k-Center. Note that, by Theorem 8, it is not possible to design a polynomial-time approximation algorithm for SGC with better approximation ratio than that of SGC-APPROX.

Our algorithm works as follows (see Algorithm 3 for the pseudocode). It adds an arbitrary node c into an initially empty set of clusters $\mathcal{C}$ and into an auxiliary set C that will contain the selected centers (lines 1 to 3). Then, it performs $k - 1$ iterations (lines

Algorithm 3 SGC-APPROX($\mathcal{G}$, k)

```
Input: Sequence graph $\mathcal{G}(V, E, \mathcal{S}, \mathcal{F})$ and the number of clusters $k$
Output: a set of clusters $\mathcal{C}$
 1: Select an arbitrary node $c$ from $V$
 2: Add cluster $\{c\}$ into an empty set of clusters $\mathcal{C}$
 3: Add node $c$ into an empty set $C$
 4: while $|C| < k$ do
 5:     Select node $u$ such that $\min_{u \in V \setminus C, c \in C} d_{ESP}(u, c)$ is maximized
 6:     Add cluster $\{u\}$ into set of clusters $\mathcal{C}$
 7:     Add node $u$ into set $C$
 8: end while
 9: for each node $u \in V \setminus C$ do
10:     Add $u$ into the cluster in $\mathcal{C}$ whose center $c$ has minimum $d_{ESP}(u, c)$
11: end for
12: return $\mathcal{C}$
```

4 to 8). In each iteration, it finds a node that is as far as possible from its closest node in C with respect to d_{ESP}. This node is selected as a center and is added into $\mathcal{C}$ and into C. After that, SGC-APPROX constructs and returns a clustering comprised of a center and its closest nodes with respect to d_{ESP}, breaking ties arbitrarily (lines 9 to 12).

The approximation guarantee of SGC-APPROX is given below:

Theorem 9. *SGC-APPROX finds a solution $\mathcal{C}$ to SGC with*

$$\max_{u \in V} \min_{c \in \mathcal{C}} d_{ESP}(u, c) \leq 2 \cdot \max_{u \in V} \min_{c \in \mathcal{C}^*} d_{ESP}(u, c),$$

where $\mathcal{C}^*$ is an optimal solution to SGC, in $O(((\max_{s \in \mathcal{S}} |s|)^2 + |E|)k|V|)$ time.

Proof. The approximation guarantee of SGC-APPROX follows from that the algorithm of Gonzalez [185] has an approximation factor of 2 for the metric k-Center problem and from Theorem 7.

The time complexity of SGC-APPROX is $O(((\max_{s \in \mathcal{S}} |s|)^2 + |E|)k|V|)$. This is because, in each of the $O(k)$ iterations, the algorithm computes d_{ESP} between u and v, where u is one of the $O(k)$ nodes in $\mathcal{C}$ and v is one of the $O(|V|)$ nodes that are not contained in $\mathcal{C}$, and each such computation takes $O((\max_{s \in \mathcal{S}} |s|)^2 + |E|)$ time: $O((\max_{s \in \mathcal{S}} |s|)^2)$ for computing d_E [275]; and $O(|E|)$ for computing d_{SP} using the algorithm of Thorup [276]. □

5.6.2 Heuristic for SGM

We propose SGM-HEUR (see Algorithm 4 for the pseudocode), a heuristic based on an efficient and effective version of the k-medoids algorithm [64]. Our heuristic selects k

nodes with a smallest score $\sum_{v\in V} \frac{d_{ESP}(u,v)}{\sum_{v'\in V} d_{ESP}(v,v')}$, treats each such node as a cluster representative (medoid), and adds each other node into the cluster of its closest representative (lines 1 to 8). After that, it updates the representatives and the clusters, as k-medoids does, until the clusters do not change or M iterations are performed; whichever happens first (lines 9 to 21). Last, it returns the set of clusters (line 22). SGM-HEUR requires $O([|V|^2 \cdot ((\max_{s\in\mathcal{S}} |s|)^2 + |E|)] + [|V|kM])$ time. The term in the first (respectively, second) pair of square brackets is the time for computing d_{ESP} for all node pairs (respectively, the time for deriving the clustering).

Algorithm 4 SGM-HEUR($\mathcal{G}$, M, k)

```
Input: Sequence graph $\mathcal{G}(V, E, \mathcal{S}, \mathcal{F})$, maximum number of iterations $M$, and the number
    of clusters $k$
Output: a set $\mathcal{C}$ of $k$ clusters
    /* Select initial medoids */
 1: Create empty set $\mathcal{C}$
 2: $C \leftarrow$ set of $k$ nodes in $V$ with a smallest $\sum_{v\in V'} \frac{d_{ESP}(v,v')}{\sum_{v'\in V} d_{ESP}(v,v')}$
    /* Construct initial clusters */
 3: while $C \neq \emptyset$ do
 4:     Select a node $u \in C$ arbitrarily
 5:     Create a cluster $c = \{u\}$ with medoid $\mu(c) = u$
 6:     Add $c$ into $\mathcal{C}$
 7:     Remove $c$ from $C$
 8: end while
 9: Add each $u \in V$ into a cluster $c \in \mathcal{C}$ such that $d_{ESP}(u, \mu(c))$ is minimum
10: $S_0 \leftarrow \sum_{u\in c: c\in\mathcal{C}} d_{ESP}(u, \mu(c))$
11: $S_m \leftarrow 0$, for each $m \in [1, M]$
12: for each $m \in [1, M]$ do
    /* Update medoids */
13:     for each $c \in \mathcal{C}$ do
14:         $\mu(c) \leftarrow$ node $u \in c$ with a minimum $\sum_{u'\in c} d_{ESP}(u, u')$
15:     end for
    /* Assign nodes into clusters */
16:     Add each $u \in V$ into a cluster $c \in \mathcal{C}$ such that $d_{ESP}(u, \mu(c))$ is minimum
17:     $S_m \leftarrow \sum_{u\in c: c\in\mathcal{C}} d_{ESP}(u, \mu(c))$
18:     if $S_m = S_{m-1}$ then
19:         break
20:     end if
21: end for
22: return $\mathcal{C}$
```

5.7 Experimental Evaluation

In this section, we evaluate our algorithms with respect to effectiveness and efficiency. We also demonstrate that ESR and $\widehat{ESR}$ and hence d_{ESR} and $d_{\widehat{ESR}}$ converge fast.

5.7.1 Evaluated Methods

We tested SGC-APPROX and SGM-HEUR with d_{ESP}, $d_{\widehat{ESP}}$, d_{ESR}, or $d_{\widehat{ESR}}$. We report results for SGC-APPROX with d_{ESP} (referred to as CA) and with $d_{\widehat{ESP}}$ (referred to as CA$_E$), as well as for SGM-HEUR with d_{ESR} (referred to as MH) and with $d_{\widehat{ESR}}$ (referred to as MH$_E$). The use of d_{ESR} and $d_{\widehat{ESR}}$ in SGC-APPROX did not substantially help quality but reduced efficiency, while the use of d_{ESP} and $d_{\widehat{ESP}}$ in SGM-HEUR did not substantially help efficiency but reduced quality; thus we omit these results. In all our algorithms, we used a normalized version of d_{ESP}, where d_E and d_{SP} is divided by its maximum value, and a similarly normalized version of $d_{\widehat{ESP}}$. We compared our algorithms against four state-of-the-art attributed graph clustering methods which employ different techniques (see Section 5.2): (I) Text-Associated DeepWalk (TADW) [247], (II) Text Enhanced Network Embedding (TENE) [251], (III) Binarized Attributed Network Embedding (BANE) [252], and (IV) Adaptive Graph Convolution (AGC) [187].

To use these methods, we first constructed the set $\mathcal{Q} = \cup_{s \in \mathcal{S}} Q_s$, where Q_s is the set of q-grams of a string $s \in \mathcal{S}$, and then embedded the string $\mathcal{F}(u)$ of each node $u \in V$ into an attribute vector $\mathcal{A}_u$ such that $\mathcal{A}_u[i]$ is equal to: (I) 1, if $\mathcal{F}(u)$ contains the q-gram with lexicographic rank[3] i in $\mathcal{Q}$, and 0 otherwise, (II) the frequency in $\mathcal{F}(u)$ of the q-gram with lexicographic rank i in $\mathcal{Q}$, or (III) the tf-idf score in $\mathcal{S}$ of the q-gram with lexicographic rank i in $\mathcal{Q}$. Note that such embeddings have already been used in the literature, for instance, in [47, 245, 251]. We report results for the best embedding method for each competitor and q. The real-valued embedding constructed by TADW or TENE was fed into k-means, following [277], while the binary embedding of BANE was fed into k-medoids [64] (with Hamming distance). We also implemented variants of CA and MH that represent a string using a vector of q-grams, as TADW and TENE do, reporting results for the best embedding method and q. We refer to the variant of CA (respectively, MH) as CA$^{\text{vec}}$ (respectively, MH$^{\text{vec}}$). Methods for clustering strings (k-medoids [64], k-means [236]) or graphs (spectral clustering [236]) performed worse than [247, 251, 252]; thus we omit their results. We summarize all tested methods in Table 5.2, for ease of reference.

[3] The *lexicographic rank* of a string in a set of strings is the rank of the string in the lexicographically sorted list of all the strings in the set.

Table 5.2 Summary of the methods used in experiments.

Acronym	Description
CA	SGC-APPROX with d_{ESP}
CA_E	SGC-APPROX with $d_{\widehat{ESP}}$
MH	SGM-HEUR with d_{ESR}
MH_E	SGM-HEUR with $d_{\widehat{ESR}}$
$\mathrm{CA}^{\mathrm{vec}}$	Variant of CA with vector of q-grams
$\mathrm{MH}^{\mathrm{vec}}$	Variant of MH with vector of q-grams
TADW	Text-Associated DeepWalk [247]
TENE	Text Enhanced Network Embedding [251]
BANE	Binarized Attributed Network Embedding [252]
AGC	Adaptive Graph Convolution [187]

5.7.2 Datasets and Setup

We used the Ciao (CIAO) and Epinions (EPIN) datasets from https://www.cse.msu.edu/~tangjili/datasetcode/truststudy.htm.

In these datasets, each user (node) is associated with a (potentially empty) sequence of reviewed products (string), and an edge connects users who trust each other. Table 5.3 summarizes the characteristics of the datasets we used.

Table 5.3 Datasets characteristics.

Dataset	# nodes	# edges	avg., max. degree	alphabet size $\vert\Sigma_S\vert$	avg., max. string length
CIAO	2,375	43,690	36.792, 453	6	15.185, 868
EPIN	22,165	286,822	25.881, 2,032	27	41.609, 5,357

Clustering quality was quantified using: (I) the Average Sum of Pairwise Edit distances (ASPE), defined as $\frac{1}{k}\sum_{c\in\mathcal{C}}\sum_{u,v\in c} d_E(\mathcal{F}(u),\mathcal{F}(v))$; and (II) Modularity [237], a well-established measure of network (graph) clustering quality, expressed as the fraction of the edges that fall within the given clusters minus the expected such fraction if edges were distributed at random. Small ASPE and large Modularity values are preferred. The speed of convergence of ESR (see Theorem 4) was measured using Average Relative Difference (ARD) [269], defined as $\frac{1}{|V|^2}\sum_{u,v\in V}\frac{R_{\ell+1}(u,v)-R_\ell(u,v)}{R_\ell(u,v)}$, where ℓ and $\ell+1$ are two consecutive iterations of the fixed-point iteration algorithm. ARD was also used with $\widehat{ESR}$ (defined as $\widehat{ESR}(u,v) = 1 - d_{\widehat{ESR}}(u,v)$) and SimRank.

By default, we used $k = 15$, $q = 2$, $\gamma = 10^{-9}$, and $\ell = 5$. We also used the default value $M = 300$ [64] and default parameter values for competitors. The proxy measures were implemented following [278]. All results are averaged over 10 runs. All algorithms were implemented in Python 3 and executed on an Intel i9 at 3.70GHz with 64 GB RAM. Our source code is available at https://rebrand.ly/SGcode.

5.7.3 String vs. Vector Representation

We show that representing strings as attribute vectors has a negative impact on clustering quality via comparing our algorithms to the variants CA$^{\text{vec}}$ and MH$^{\text{vec}}$. As can be seen in Fig. 5.2, CA$^{\text{vec}}$ resulted in higher (worse) ASPE than both CA and CA$_E$. For example, the average ASPE for CA$^{\text{vec}}$ was higher than that of CA (respectively, CA$_E$) by 104% (respectively, 56%) on average (over the two datasets). Also, CA$^{\text{vec}}$ resulted in lower (worse) Modularity than both CA and CA$_E$. For example, the average Modularity for CA$^{\text{vec}}$ was lower than that of CA (respectively, CA$_E$) by 18% (respectively, 33%) (over the two datasets). As can be seen in Fig. 5.3, MH$^{\text{vec}}$ performed worse than our algorithms in terms of both ASPE and Modularity. These results show the benefit of measuring similarity by using strings directly as our algorithms do. Since CA$^{\text{vec}}$ and MH$^{\text{vec}}$ performed worse than CA and MH, we do not report results for them in the remaining of this section.

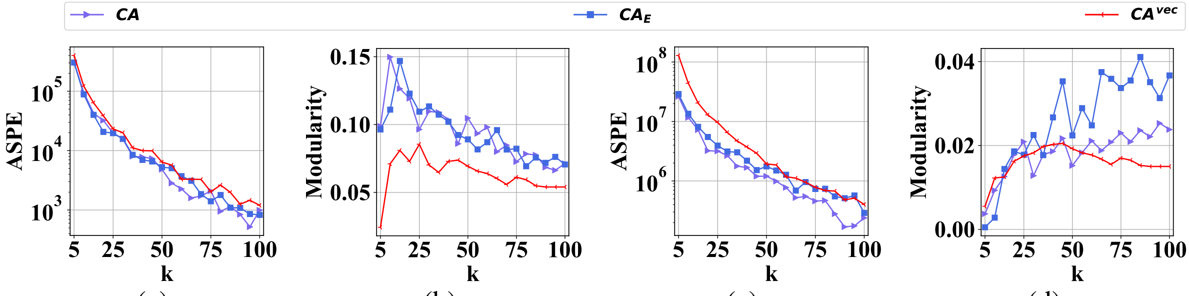


Fig. 5.2 Comparison of our algorithms against the CA$^{\text{vec}}$ variant that represents strings as attribute vectors: (a) ASPE vs. k for CIAO. (b) Modularity vs. k for CIAO. (c) ASPE vs. k for EPIN. (d) Modularity vs. k for EPIN.

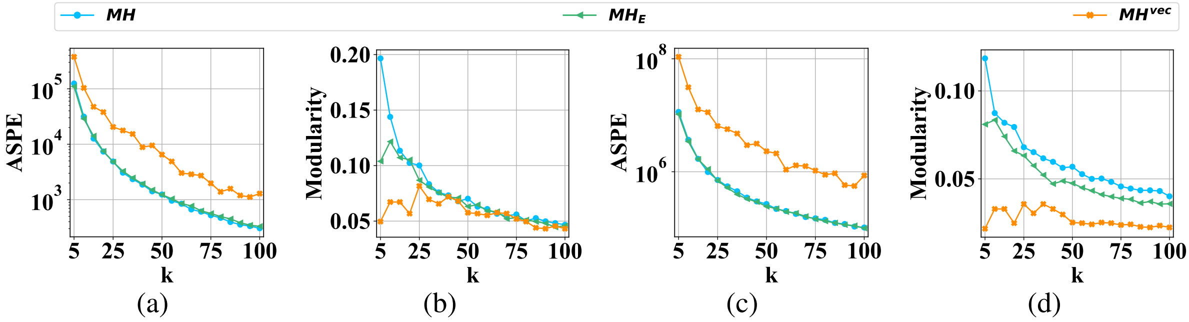


Fig. 5.3 Comparison of our algorithms against the MH$^{\text{vec}}$ variant that represents strings as attribute vectors: (a) ASPE vs. k for CIAO. (b) Modularity vs. k for CIAO. (c) ASPE vs. k for EPIN. (d) Modularity vs. k for EPIN.

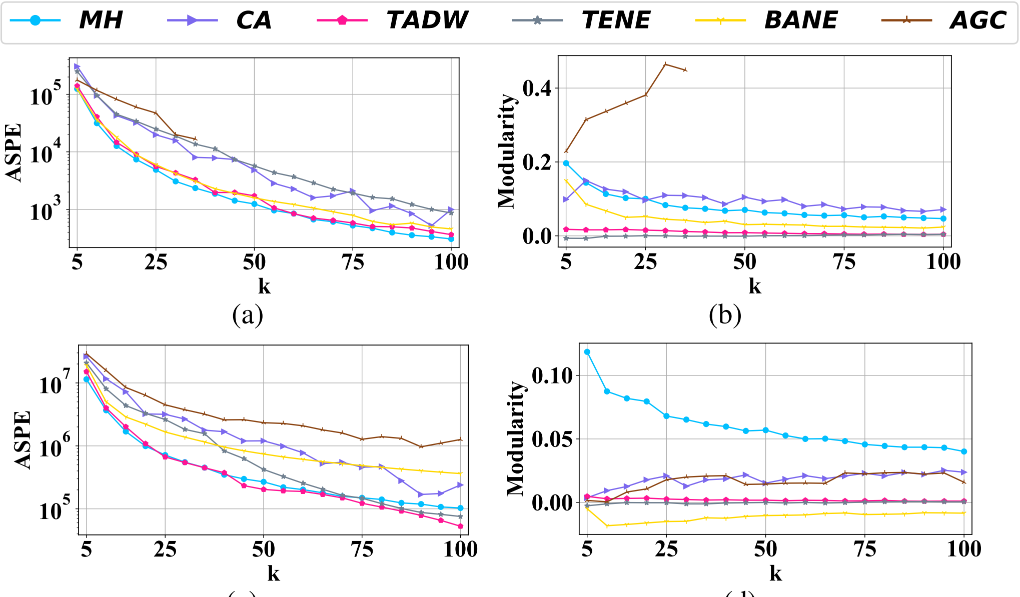


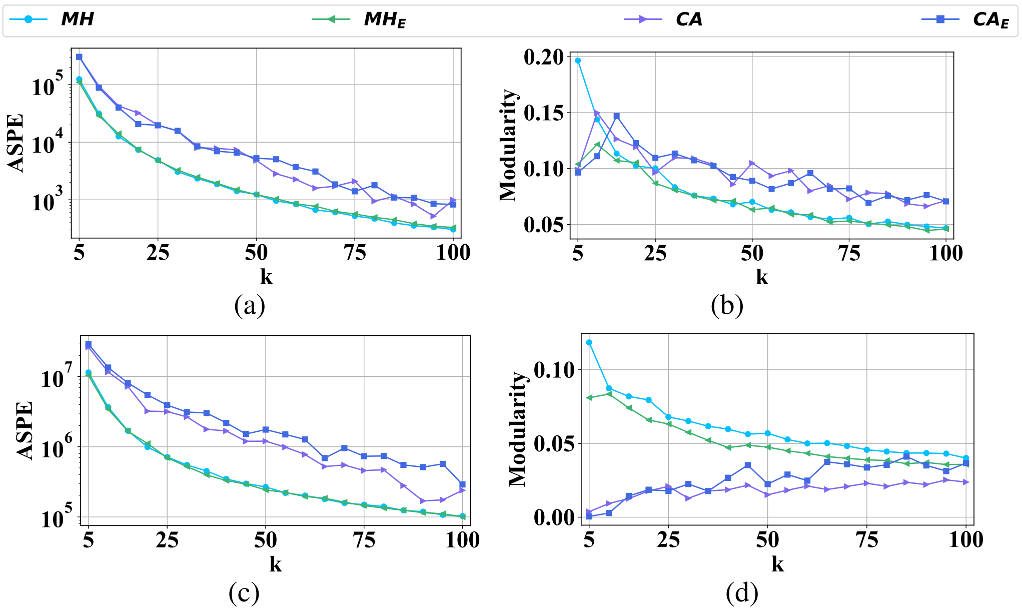
Fig. 5.4 Comparison of our algorithms against state-of-the-art algorithms for attributed graph clustering: (a) ASPE and (b) Modularity vs. k for CIAO. (c) ASPE and (d) Modularity vs. k for EPIN.

5.7.4 Clustering Quality

We show that our algorithms created clusters containing nodes with similar strings, as ASPE is lower than that of most competitors (see Fig. 5.4(a)), which are also structurally similar, as Modularity is higher than that of most competitors (see Fig. 5.4(b)). In addition, the use of proxy measures by our algorithms did not substantially affect clustering quality, as CA$_E$ and MH$_E$ performed very similarly to CA and MH, respectively (see Fig. 5.5).

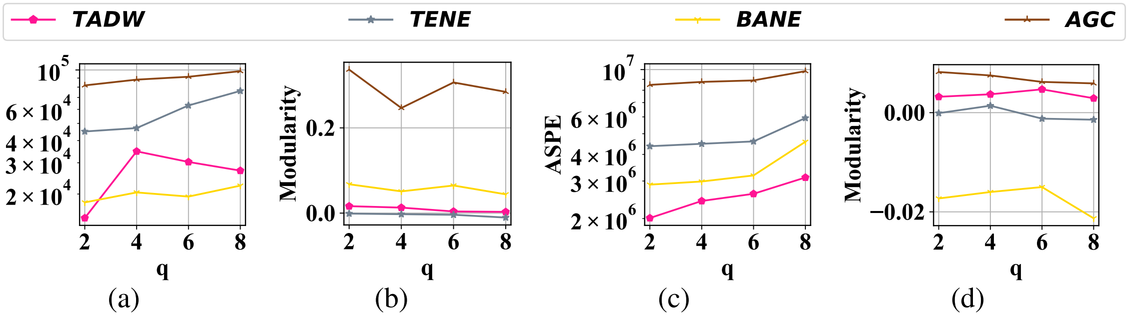
On the other hand, the competitors achieved a worse clustering than our algorithms, when considering both ASPE and Modularity together, and some were worse in both of these measures. For example, TENE performed poorly in terms of both ASPE and Modularity, while BANE and TADW performed the worst in terms of Modularity and worse than MH and MH$_E$ in terms of ASPE. AGC performed the worst in terms of ASPE but the best in terms of Modularity for $k < 36$. For $k \geq 36$, AGC did not terminate because the eigenvalue decomposition it employs to find the convolution failed. Similar results were obtained for the EPIN dataset (see Figs. 5.4(c) and 5.4(d)).

The reason that AGC performed poorly with respect to ASPE is that it favors Modularity by design, since it assumes that nodes that are close in the graph will likely be clustered together, as mentioned in Section 5.2. This assumption is not necessarily true. In fact, in our setting, AGC created clusters comprised of nodes that are close in the graph but have quite dissimilar strings and this led to poor clusters in terms of ASPE. The reason



(a) (b) (c) (d)

Fig. 5.5 Comparison of MH and CA against MH_E and CA_E: (a) ASPE and (b) Modularity vs. k for CIAO. (c) ASPE and (d) Modularity vs. k for EPIN.



(a) (b) (c) (d)

Fig. 5.6 Impact of q on clustering quality for competitors: (a) ASPE and (b) Modularity vs. q for CIAO. (c) ASPE and (d) Modularity vs. q for EPIN.

that TADW performed poorly with respect to Modularity is that it supervises random walks based on attribute vectors, which resulted in clusters with nodes that are far apart in the graph. The reason that BANE (respectively, TENE) performed poorly in terms of Modularity is that it does not use higher than first (respectively, second) order proximities to capture the distance of nodes in the graph. However, such proximities are important to consider [266] because good clusters may be constructed based on higher than second order proximities.

The good performance of our methods is due to three factors: (I) d_{ESR} and d_{ESP} can capture graph distance and string distance in a unified manner. (II) Unlike the competitors, our methods use the strings directly in similarity measurements instead of representing strings as attribute vectors, which may lose similarity information. (III) Unlike TENE and

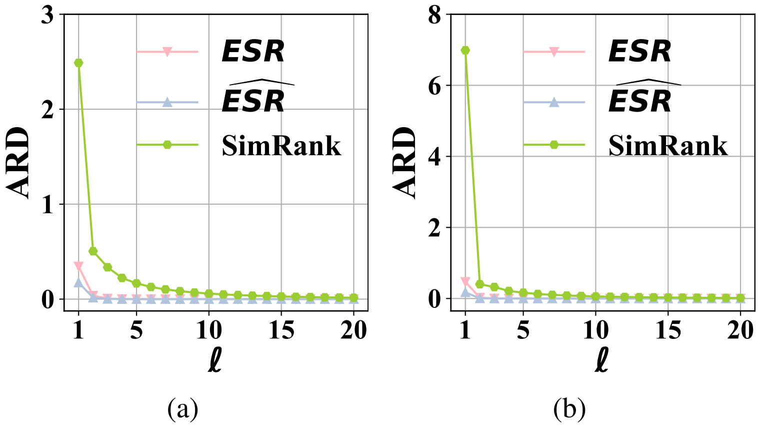
BANE, which only use first or first and second order proximities, our methods employ measures that capture distance between two nodes based also on longer paths.

Note that the values of ASPE and Modularity depend on the similarity between strings and the similarity between nodes, respectively. Thus, it may not be possible to create a clustering with both low ASPE and high Modularity, when close nodes have different strings and vice versa.

Besides, we also examined the impact of q (length of q-gram used in the vector representation of a string by the competitors). Figs. 5.6 (a)-(b) show that cluster quality in CIAO became worse as q increased. This is because the number of distinct q-grams (i.e., $|\mathcal{Q}|$) increases and thus ASPE increased as well. Thus, the default value $q = 2$ we used is a fair choice. The results for EPIN were similar (see Figs. 5.6 (c)-(d)).

5.7.5 Convergence

We show that ESR and $\widehat{ESR}$ converge faster than SimRank, which helps efficiency (see Fig. 5.7) This is attributed to the impact of string similarity (e.g., $\sigma(u, v) = 0$ implies $R_{\ell+1}(u, v) = 0$ in Eq. 5.3). In fact, the ARD scores for ESR and $\widehat{ESR}$ were smaller than 10^{-3} after 5 iterations, while those for SimRank were an order of magnitude larger even after 20 iterations.



(a)

(b)

Fig. 5.7 Convergence speed for our measures and SimRank: ARD vs. ℓ (number of iterations of fixed-point iteration algorithm) for: (a) CIAO and (b) EPIN.

5.7.6 Runtime

We examined the runtime of all methods for varying number of nodes (see Fig. 5.8). CA$_E$ and MH$_E$ were much more efficient than all competitors. For instance, CA$_E$ was up to 8 and 3 times faster than TENE, the fastest competitor, in the experiment of Fig. 5.8(a) and 5.8(b), respectively. As expected, CA$_E$ and MH$_E$ were faster than CA and MH, since the last two algorithms need to compute edit distance instead of the more efficient to compute proxy measures. For example, MH$_E$ was two orders of magnitude faster than MH in the case of clustering CIAO and even faster in the case of clustering EPIN. In addition, CA$_E$ was approximately two times faster than CA. The impact of using the proxy measure in the algorithms for SGC was less significant compared to the algorithms for SGM. This is because in the former there are fewer edit distance computations, as in SGC there is no need to compute all pairwise distances between strings. CA was faster than MH, as expected by the complexity analysis (see Section 5.6). For example, CA was more than 50 times faster than MH in the case of the CIAO dataset and more than two order of magnitudes faster in the case of the EPIN dataset.

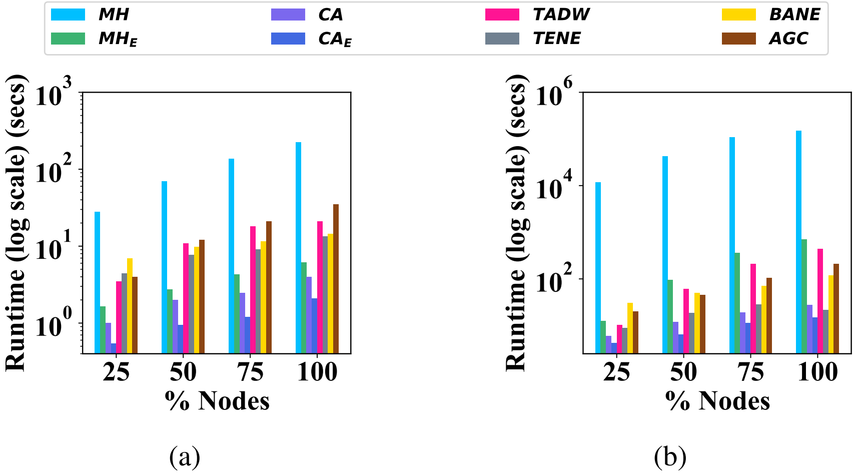


Fig. 5.8 Efficiency for our methods and the competitors: Runtime vs. % of nodes for: (a) CIAO and (b) EPIN.

5.8 Case Study: Clustering Phylogenetic Trees

As discussed in Chapter 1, an edge in a sequence graph can model a relationship between users or a relationship between strings. In Section 5.7, we demonstrated the effectiveness of our approach in applications where edges represent relationships between users and the input data are modeled as a graph. We now proceed to presenting a case study that highlights the effectiveness of our approach when edges represent relationships between strings and the input data are modeled as a *phylogenetic tree* [253].

In particular, we consider the domain of bioinformatics and the application of evaluating the quality of a phylogenetic tree [253]. A phylogenetic tree is a rooted or unrooted leaf-labeled bifurcating (binary) tree that represents evolutionary relationships among biological organisms [253]. A phylogenetic tree can be inferred from a set of strings, each representing the genomic sequence of an organism. Each leaf of the tree corresponds to a different organism and is labeled with a string representing the genomic sequence of the organism, while each non-leaf node u corresponds to a cluster comprised of all strings associated with the leaves of the subtree rooted at u. Thus, a phylogenetic tree is a hierarchical representation of all clusterings of the strings of its leaves.

A phylogenetic tree T whose leaves correspond to a set of strings S can be constructed by different methods (e.g., by agglomerative hierarchical clustering methods [253]). To evaluate the quality of T, one can compare it with a ground truth clustering $\mathcal{C}$ of S. Let k be the number of clusters in $\mathcal{C}$. Clearly, T cannot be compared with $\mathcal{C}$ directly, since the former is a binary tree (i.e., a 2D structure), whereas the latter is a partition of S into k clusters (i.e., a 1D structure). Therefore, one needs to first "flatten" T, by creating a clustering $\mathcal{C}'$ of its leaves that has k clusters, and then compare $\mathcal{C}'$ with $\mathcal{C}$. If these two clusterings are similar, T is of high quality, as it accurately reflects the evolutionary relationships between the organisms corresponding to the strings of S according to the ground truth clustering.

It is easy to see that T can be modeled as a sequence graph $\mathcal{G} = (V, E, \mathcal{S}, \mathcal{F})$ with V (respectively, E) being the set of nodes (respectively, edges) of T, $\mathcal{S}$ being the set of strings S corresponding to the leaves of T (i.e., the leaf labels), and $\mathcal{F}$ being a function that associates each leaf of T with is corresponding string in S and each non-leaf node in T with the empty string. Thus, we can construct $\mathcal{C}'$ by first clustering the sequence graph corresponding to T with k equal to the number of clusters in the ground truth clustering, and then creating, for each resultant cluster c, a cluster c' in $\mathcal{C}'$ that is comprised of the non-empty strings corresponding to the nodes in c. After that, we can compare $\mathcal{C}'$ with the ground truth clustering $\mathcal{C}$ using measures that compare clusterings (e.g., the measures in [279–281]).

In what follows, we first discuss the data and setup we used in our case study and then the results of the case study.

5.8.1 Data and Setup of case study

We used three datasets, referred to as Ebolavirus (EBOL), Influenza (INFL), and Coronavirus (COR). The characteristics of these datasets are summarized in Table 5.4. In these datasets, each record is a genomic sequence of a different virus type (e.g., in EBOL, there are 59 records and each record corresponds to a different type of Ebolavirus). All genomic sequences were downloaded from the NCBI GenBank [282] based on their accession numbers provided in [283].

Table 5.4 Datasets characteristics

Dataset	# of leaves	# of edges	avg., max. degree	alphabet size $\lvert\Sigma_{\mathcal{S}}\rvert$	avg., max. string length	ground truth clusters
EBOL	59	116	1.98, 3	4	18,932, 18,961	5
INFL	38	74	1.97, 3	4	1,406, 1,467	5
COR	34	66	1.97, 3	4	27,567, 31,357	9

For each dataset, we obtained the phylogenetic tree from [283], and the ground truth clustering from the NCBI GenBank [282] using the BioPython library [284]. Specifically, each cluster in the ground truth clustering is comprised of all sequences with the same value in the *Organism* field, for EBOL and INFL, or all sequences with the same value in the last element of the *Taxonomy* field for COR (since all sequences in this dataset had the same value in *Organism*). It can be readily verified from Figs. C.1, C.2, and C.3 in Appendix C that the phylogenetic trees we used are in accordance with the ground truth clustering. That is, each ground truth cluster contains leaves that are close together in the phylogenetic tree.

We constructed a clustering $\mathcal{C}'$ from the sequence graph corresponding to a phylogenetic tree T by applying one of our methods (CA, CA$_E$, MH, and MH$_E$) or a competitor (TADW, TENE, BANE, AGC), configured as in Section 5.7.

To measure similarity between $\mathcal{C}'$ and the ground truth clustering, we used three well-established measures that compare similarity between two clusterings based on their labels: Clustering Accuracy (ACC) [279], Normalized Mutual Information (NMI) [280], and macro-F_1 score [281]. These measures take values in $[0, 1]$ with larger values indicating a more accurate (i.e., closer to the ground truth) clustering.

ACC is computed based on Eq. 5.16:

$$\text{ACC}(\mathcal{C}', \mathcal{C}) = \frac{\sum_{i=1}^{|S|} \mathbf{1}\left(l_i = m\left(c_i\right)\right)}{|S|}, \tag{5.16}$$

where S is the set of strings in the input sequence graph, l_i is the ground truth label of the i-th string in the input sequence graph, c_i is the id of the cluster where this string belongs in $\mathcal{C}'$ that is used as clustering label, $m()$ is the optimal mapping function that permutes clustering labels to match the ground truth labels[4], and $\mathbf{1}()$ outputs 1 if its argument is true and 0 otherwise.

NMI is computed based on Eq. 5.17:

[4] The optimal mapping function can be computed based on the Hungarian algorithm [285].

$$\text{NMI}(\mathcal{C}, \mathcal{C}') = \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} n_{ij} \log_2 \frac{n_{ij}}{n_i \hat{n}_j}}{\sqrt{\left(\sum_{i=1}^{k} n_i \log_2 \frac{n_i}{n}\right) \left(\sum_{j=1}^{k} \hat{n}_j \log_2 \frac{\hat{n}_j}{n}\right)}}, \tag{5.17}$$

where n_i denotes the number of strings in the i-th cluster in $\mathcal{C}'$, $\hat{n}_j$ denotes the number of strings belonging to the j-th cluster in the ground truth clustering $\mathcal{C}$, n_{ij} is the number of strings belonging both in the i-th cluster in $\mathcal{C}'$ and in the j-th ground truth cluster, and k is the number of clusters.

The macro-F_1 measure is based on the F_1 measure. F_1 assumes a setting where there are only two different labels, namely 0 and 1, in the ground truth clustering, and it is computed based on Eq. 5.18:

$$F_1 = 2 \cdot \frac{\text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}, \tag{5.18}$$

where $\text{Prec} = \frac{TP}{TP+FP}$ and $\text{Rec} = \frac{TP}{TP+FN}$. In turn, TP denotes the number of strings with label 1 in both the ground truth clustering and $\mathcal{C}'$, while FN (respectively, FP) denotes the number of strings with label 1 (respectively, 0) in the ground truth clustering and 0 (respectively, 1) in $\mathcal{C}'$. When the ground truth clustering contains L labels, the macro-F_1 measure is defined based on Eq. 5.19:

$$\sum_{c=1}^{L} F_1(c)/L, \tag{5.19}$$

where $F_1(c)$ is the F_1 score obtained for a two-label setting, in which 1 is the label of cluster c and 0 is the label of any other cluster.

We used the default parameters of Section 5.7 for all methods. All experiments ran on the PC mentioned in Section 5.7.

5.8.2 Clustering Quality

Since the phylogenetic trees we used are in accordance with the ground truth (see Figs. C.1, C.2, and C.3 in Appendix C), we expect that a good clustering method for evaluating a phylogenetic tree would have a high value (close to 1) in ACC, NMI, and macro-F_1. The higher the value, the better the clustering method, as the clustering it constructs is more similar to the ground truth clustering.

Tables 5.5, 5.6, and 5.7 show that the clusterings created by our methods are substantially more similar to the ground truth clustering compared to those created by the competitors. CA was the best performing method, outperforming MH in all tested cases,

Table 5.5 ACC for different methods applied with $k = 5$ on EBOL and INFL, and with $k = 9$ on COR. A × denotes that a method did not produce a score, because it did not produce k clusters. The values for the best performing method are in bold.

Methods	EBOL	INFL	COR
CA	**0.904**	**0.947**	**0.941**
CA$_E$	0.805	0.894	0.882
MH	0.814	0.845	0.756
MH$_E$	0.712	0.753	0.729
TADW	0.627	0.447	0.382
TENE	0.576	0.394	0.352
BANE	×	0.368	×
AGC	0.423	0.317	0.352

Table 5.6 NMI for different methods applied with $k = 5$ on EBOL and INFL, and with $k = 9$ on COR. A × denotes that a method did not produce a score, because it did not produce k clusters. The values for the best performing method are in bold.

Methods	EBOL	INFL	COR
CA	**0.894**	**0.962**	**0.957**
CA$_E$	0.880	0.857	0.918
MH	0.837	0.884	0.720
MH$_E$	0.701	0.835	0.701
TADW	0.426	0.269	0.389
TENE	0.317	0.255	0.434
BANE	×	0.313	×
AGC	0.435	0.322	0.376

Table 5.7 macro-F_1 for different methods applied with $k = 5$ on EBOL and INFL, and with $k = 9$ on COR. A × denotes that a method did not produce a score, because it did not produce k clusters. The values for the best performing method are in bold.

Methods	EBOL	INFL	COR
CA	**0.901**	**0.943**	**0.886**
CA$_E$	0.717	0.889	0.848
MH	0.751	0.840	0.762
MH$_E$	0.693	0.708	0.709
TADW	0.483	0.376	0.375
TENE	0.371	0.401	0.317
BANE	×	0.321	×
AGC	0.401	0.345	0.286

due to its objective function. Specifically, its ACC, NMI, and macro-F_1 scores were 15.6%, 15.2%, and 16% larger on average (over all datasets) than those of MH, respectively. In addition, the use of proxy measures in our methods did not substantially affect clustering quality. This is encouraging as our methods using proxy measures, namely CA_E and MH_E, are more efficient, as discussed in Section 5.7.

On the other hand, the competitors did not perform well. For example, the ACC, NMI, and macro-F_1 scores for CA were 91.8%, 159.5%, and 121.2% larger on average (over all datasets) than the best competitor TADW. The main reason for the poor performance of competitors is that, in the application we consider, only the leaves of a phylogenetic tree have non-empty strings associated with them, while a large number of non-leaf nodes are associated with the empty string. This leads the competitors to construct clusters with leaf nodes associated with different strings, as their assumptions (close nodes in the tree should be clustered together for AGC, and low order proximities are sufficient to cluster nodes in the tree for BANE and TENE) are invalidated. Last, note that in the case of EBOL and COR, BANE did not produce k clusters and thus a meaningful clustering for the purpose of our case study, since it learned fewer than k binary code representations.

Chapter 6

Conclusion

6.1 Summary of Contributions

In this thesis, we considered clustering three different types of complex data that are often encountered in applications: RT-datasets, RS-datasets, and sequence graphs. We formalized the task of clustering each type of data as an optimization problem, examined the hardness of the problem, and developed new clustering algorithms for solving it. We also demonstrated the effectiveness and efficiency of the algorithms experimentally.

At the same time, there are three research questions that have been answered in this thesis:

1. How can we cluster a given RT-dataset so that each cluster (i.e., set of records comprised of a set of relational attributes and a set of diagnosis codes) represents patients that have similar demographics and set of diagnosis codes? We have addressed this question in Chapter 3.

2. How can we cluster a given RS-dataset so that each cluster (i.e., set of records comprised of a set of relational attributes and a sequence of diagnosis codes) represents patients that have similar demographics and sequence of diagnosis codes? We have addressed this question in Chapter 4.

3. How can we cluster a sequence graph so that each cluster (i.e., set of graph nodes each associated with a string) contains nodes that are structurally similar and have similar node labels (strings)? We have addressed this question in Chapter 5.

The main contributions of the thesis can be summarized as follows:

(I) *For clustering RT-datasets*, a new approach was proposed. The approach represents the dataset in a binary form in which the features are selected demographic values, as well as combinations (patterns) of frequent and correlated diagnosis codes (comorbidities). This representation enables measuring similarity between records using cosine similarity, an effective measure for binary-represented data, and allows finding compact, well-separated clusters through hierarchical clustering. The main clinical implication of this approach is that it was able to find correlations between diagnosis codes and between diagnosis codes and demographics that have been documented in the medical literature and correspond to to known comorbidities. This shows the effectiveness of our clustering algorithm and in particular of the all-confidence measure that aims specifically to capture such correlations. The practical value of this research is two-fold. First, our methodology (data modeling, measure, approach) can lead to the developments of new clustering algorithms. Second, our publicly available implementation may be used to cluster an EHR RT-dataset.

(II) *For clustering RS-datasets*, the task of clustering an RS-dataset was formalized as an optimization problem which was shown to be computationally hard. Also, an effective and efficient algorithm that addresses the problem was proposed. This algorithm uses a distance measure we developed and works by first identifying k representative records (centers), for a given k, and then constructing k clusters, each containing one center and the records that are closer to the center compared to other centers. The main clinical implication of this approach is that it was able to create clusters in which frequent sequential patterns capture relationships between diagnosis codes that are documented in the medical literature. Furthermore, the created clusters allowed discovering potentially clinically useful patterns associated with patients that have similar demographics. The practical value of this research is similar to that of the last paragraph.

(III) *For clustering sequence graphs*, the task of clustering a sequence graph was formalized as an optimization problem and variants of the problem based on the k-center [26, 27] and k-median [26, 28] problems were studied. Specifically, we first proposed a product metric and a measure based on SimRank [32] to capture the distance between two nodes of a sequence graph, as well as a proxy for each measure. We then proposed an approximation algorithm and a heuristic, which outperform attribute-based graph clustering methods, as shown experimentally. Last, we proposed a methodology that applies our measures (and the corresponding clustering algorithms) to evaluate whether a given phylogenetic tree is in accordance with a given ground truth clustering. The practical value of the research is two-fold. First, our publicly available implementation may be used to cluster the nodes of a graph, which is important in the domain of social networks and of e-commerce. Second, and more specifically, our implementation allows a bioinformatician to evaluate the accuracy of a given clustering of a phylogenetic tree by comparing it to a ground truth clustering. If these two clusterings are similar, then the phylogenetic tree is meaningful.

6.2 Future Work

In the following, we identify some future directions for extending the work in this thesis.

6.2.1 Future work for clustering RT-datasets

In the following, we explain how MASPC can be extended to handle different types of healthcare data. Moreover, limitations of MASPC which provide opportunities for further research are discussed.

6.2.1.1 Dealing with multiple set-valued attributes

MASPC was developed for clustering RT-datasets with demographics and diagnosis codes, which are useful in several medical analysis applications [15–19]. However, EHR data may also contain laboratory data (e.g., Logical Observation Identifiers Names and Codes (LOINC)) and medications. In this case, we can preprocess the input dataset to create a set-valued attribute comprised of all diagnosis codes, values in laboratory results, and medications[1] and then apply our approach to the pre-processed dataset with no modification. To see why this is the case, note that: (I) the MAS algorithm mines and selects patterns of any type (i.e., the support and all-confidence measures are applied to any itemset, irrespectively of whether its values are for example diagnosis codes or medications), (II) the binary representation in phase 1 of PC can be constructed for MFAs of any type (i.e., an MFA is a set of values of any type, such as diagnosis codes and laboratory results, which corresponds to a feature in the binary representation), and (III) the agglomerative clustering algorithm in phase 2 of PC can be applied to a binary representation whose features are MFAs of any type (i.e., the cosine similarity is measured between binary vectors, whose elements correspond to MFAs of any type).

6.2.1.2 Dealing with temporally-annotated sequential data

Also, MASPC was developed for data in which the diagnosis codes in a record are represented as a set. However, it is possible that a dataset contains temporal information, such as the date on which a diagnosis code was assigned to a patient. While temporal information is useful (e.g., in the context of longitudinal studies), it makes clustering challenging [286], calling for a new approach. Developing an approach for clustering an RT-dataset in which there is a temporally-annotated sequence of diagnosis codes, instead

[1] Formally, let $\mathcal{A}_1^{l+1}, \mathcal{A}_2^{l+1}, \mathcal{A}_3^{l+1}$ be set-valued attributes representing diagnosis codes, laboratory results, and medications respectively. The dataset $\mathcal{D}$ will contain a single set-valued attribute $\mathcal{A}^{l+1}$ and each record r in $\mathcal{D}$ will have a set of values that is the union of the values it has in $\mathcal{A}_1^{l+1}, \mathcal{A}_2^{l+1}$, and $\mathcal{A}_3^{l+1}$.

of a set of diagnosis codes, is an interesting and challenging avenue for future work. The challenges come from the fact that: (I) more than one occurrence of the same diagnosis code may be contained in a patient's record, if the patient has been diagnosed with the code at different times, and (II) there is a temporal ordering of diagnosis codes. Due to I, the data are generally of higher dimensionality than the data comprised of demographics and sets of diagnosis codes that we considered in Chapter 3. Due to II, itemset mining techniques are no longer applicable to construct a binary representation, since the patterns should capture the temporal ordering of values. Also, standard similarity measures, such as cosine similarity or Euclidean distance, do not apply and specialized measures which are developed for temporally-annotated data (e.g., Dynamic Time Warping [287]) should be considered instead.

6.2.1.3 Configuring parameters

The performance of MASPC, in terms of clustering quality, depends on the quality of the input dataset and on the configuration of the thresholds $minSup$, $minAc$, and $minOv$. This is because the values of these thresholds may affect the number of MFAs selected by the MAS algorithm, which in turn affects the number of unclustered records and the quality of the clustering performed by PC. Specifically, small $minSup$, $minAc$ and $minOv$ values lead to a large number of MFAs and few unclustered records. On the other hand, large $minSup$, $minAc$, and $minOv$ values lead to a small number of MFAs and many unclustered records. When $minSup$, $minAc$, and $minOv$ are too small, the quality of clustering is low due to the curse of high dimensionality (i.e., the binary representation has too many features which makes similarity difficult to capture [55, 15]). Furthermore, small $minsup$ and $minAc$ values imply that clusters are comprised of diagnosis codes that are associated with a few patients only and are not correlated. So, fewer unclustered records do not necessarily imply a higher quality result. When $minSup$, $minAc$, and $minOv$ are too large, the quality of clustering is low, because the patterns are too few to capture the similarity between records. For example, when MAS selects a single MFA due to the use of very large thresholds, then the records that support this MFA are deemed similar, even though they contain many different diagnosis codes. Thus, it is important for the thresholds $minSup$, $minAc$, and $minOv$ to be configured appropriately. This is challenging, because there are multiple inter-dependent and data-dependent thresholds that need to be configured appropriately and because the thresholds for mining ($minSup$, $minAc$, and $minOv$) need to be determined before the clustering part of our method is performed. Towards this end, a tool with graphical user interface that applies heuristics for configuring the thresholds $minSup$ and $minAc$ would be useful. The main idea of the heuristics is to try different parameters (e.g., a few equidistant values between the minimum and maximum values of $minSup$ ($\frac{1}{|\mathcal{D}|}$ and 1) and $minAc$ (0 and 1)), plot the number of unclustered records and the quality of the resultant clustering, and either ask the user to select the threshold values they prefer, or automatically choose the threshold values that lead to a good trade-off between the number of unclustered records and the clustering quality. Furthermore, it may be possible to improve the efficiency of the tool by: (I) sharing the mining results among runs (since the patterns mined with low $minSup$ or $minAc$ are a superset of those mined with higher $minSup$ or $minAc$), and (II) exploiting the anti-monotonicity property of the support and all-confidence measure [7, 142] to reduce the number of the values of $minSup$ and $minAc$ that should be checked. The tool could be a starting point for addressing the issue of selecting good thresholds. Yet, it would be interesting to develop a general principled methodology for selecting thresholds. This is an important avenue for future work.

6.2.1.4 Dealing with unclustered records

After configuring the thresholds and applying MASPC, one needs to decide how to deal with the *unclustered* records that may exist. This can be performed as a post-processing task. One strategy that is followed by other pattern-based clustering methods [47] is to exclude the unclustered records from the clustering result. The justification behind this strategy is that these records are not similar to any other record (based on the patterns), and thus they should not be considered, to avoid degrading the quality of clusters and harming their interpretability (based on the patterns). This strategy may be suitable for applications such as visualization, classification, and anonymization, where it is important to have coherent clusters. For example, in anonymization, it is better to remove the unclustered records rather than creating a cluster containing all of them [15]. This is because the anonymized values of dissimilar records in a cluster are too coarse to be useful [15]. In other applications, the clustered dataset may need to contain each record of the input dataset, because each record is important for analysis. Examples are clinical and epidemiology applications, as well as statistical query answering. In clinical applications, for example, records representing patients that are dissimilar to all others because they have rare diagnosis codes should be contained in the clustered dataset. To ensure this, each unclustered record can be treated as a separate cluster, or the unclustered records may be put into a single cluster, which is released as is. We have performed experiments using the strategy that puts all unclustered records into a single cluster (see Section 3.7.3 in Chapter 3) and found that our algorithm outperforms all other baselines. Last, it is also possible to set a threshold representing the largest allowable number of unclustered records, based on user application requirements, and re-executing MASPC with lower $minSup$, $minAc$, and $minOv$ until the number of unclustered records is at most equal to the threshold. For example, setting $minSup = \frac{1}{|\mathcal{D}|}$, $minAc = 0$, and $minOv = 1$ would treat almost every record as an MFA, resulting in a very small number of unclustered records. The result of our algorithm when the number of unclustered records does not exceed the threshold will then be output.

6.2.1.5 Semantic similarity and importance of diagnosis codes in pattern mining

Our approach uses entire patterns as features to perform clustering. We have shown how to mine patterns that lead to accurate clusters in practice in a reasonable amount of time, which is encouraging, given that the task of pattern mining is far from straightforward and has been the focus of research for decades. The mining of patterns in our approach is performed by exploiting properties, such as the frequency of patterns and correlations between diagnosis codes, using a popular FP-tree based framework. However, there are other properties of patterns that may influence clustering accuracy and have not been considered. One such property is the semantic similarity of diagnosis codes, as captured by the ICD-code hierarchy. Another is the "importance" of diagnosis codes (e.g., to consider the difference between primary and secondary diagnosis codes). The idea would be to attempt to mine patterns subject to some constraints according to these properties and then use them as features for clustering.

However, further study is needed to perform the mining of such patterns efficiently. The difficulty in doing so comes from: (I) the exponentially large space of possible patterns (e.g., some constraints may not have sufficient pruning power to allow the mining of patterns in reasonable time), (II) the monotonicity that properties must satisfy to work with the FP-tree based framework, and (III) the fact that the aforementioned properties should be enforced together with others such as frequency, which have been shown to be necessary for a high-quality clustering.

6.2.2 Future work for clustering RS-datasets

DDSCA is limited in 5 aspects which provide opportunities for future work.

1. DDSCA does not deal with datasets in which diagnosis codes are associated with dates of visits, which are helpful in longitudinal studies [286]. One way to deal with such datasets is to treat the date of visit corresponding to the diagnosis codes in a record in the same way as a demographic. However, this is not appropriate when there are multiple dates, each corresponding to some of the diagnosis codes in a

record, due to the curse of dimensionality [7]. In this case, a fundamentally new approach is required.

2. DDSCA was evaluated using data containing ICD-9 codes. Although our distance measure can easily be modified to deal with other types of diagnosis codes such as ICD-10 codes [288] (or, in general, any sequence of events), further work is needed to evaluate its effectiveness in such settings. Similarly, it would be interesting to extend DDSCA to consider sequences comprised of other patient information, such as medications and lab results. This requires further work, as capturing the similarity of multiple inter-related sequences is challenging.

3. It is useful to evaluate the effectiveness of DDSCA in tasks beyond frequent sequential pattern mining. An example of such a task is causal relationship discovery [289], for which we have obtained some preliminary results (see Section B.8 in Appendix B). Other examples of such tasks are classification [183] and anonymization [15], which we also leave for future work.

4. It is useful to examine whether long frequent sequential patterns [60] (e.g., patterns comprised more than two diagnosis codes) can be mined from the clusters created by our methods. This is possible by configuring the sequential pattern mining method (e.g., the algorithm in [214] in the experiment of Table 4.6) to mine more of the top frequent sequential patterns. It would be interesting to examine meaningful such relationships and the accuracy of such a clustering approach on real EHR data.

5. There may be constraints related to which diagnosis codes may or may not appear in the same cluster. The concept is similar to the must-link and cannot-link constraints [290]. Our approach is not dealing with such constraints and incorporating them is not straightforward. Intuitively, this is because our method operates on entire records comprised of demographics and sequences of diagnosis codes and does not offer fine-grained control when grouping certain diagnosis codes together in clusters.

6.2.3 Future work for clustering sequence graphs

Our methods for clustering a sequence graph are limited in 3 aspects which provide opportunities for future work.

1. The proposed methods do not deal with graphs containing temporal information. Specifically, each node and/or edge in a graph can have a timestamp, showing when the node or edge appeared. In addition, the proposed methods do not deal with

dynamic graphs. For example, a node may persist, but its neighbours may be changed as time goes on (e.g., a node u can have v and w as neighbors at time point t_1 and only v as neighbor at time point t_2). Also, the string label of a node may change over time. Clustering sequence graphs containing temporal information, as well as clustering dynamic sequence graphs, requires new approaches.

2. The distance measure $d_{ESP}(u, v)$ we proposed does not consider the strings of the nodes that lie in the path (or paths) between two nodes u and v. For example, consider that the string of both u and v is $\mathtt{aaa}$ and that the shortest path between u and v that has length 3. In this case, $d_{ESP}(u, v)$ will consider u and v as similar because: (I) they have the same labels and (II) their shortest path distance is 3, so d_{ESP} will be very low. The nodes u and v will be considered similar even when the strings of the nodes that lie in the shortest path between u to v do not contain any $\mathtt{a}$. There may be cases, however, in which the strings of such nodes should also affect the similarity between u and v. For example, consider a social network, in which a string represents the locations where a user checked in. Two users in this social network may be regarded are more similar when their friends visited similar locations with them in the same order (i.e., they have similar strings). It may be interesting to see how d_{ESP} can be modified to capture similarity in such applications.

3. It is helpful to examine whether one can cluster a string graph, constructed based on EHR data, to get meaningful insights about patients. In this string graph, a node corresponds to a patient and its label is a string that may represent diagnosis codes, procedures, or medications, or alternatively a combination of these. Furthermore, each edge represents a relationship [291] between two patients.

References

[1] National Center for Health Statistics, "International Classification of Diseases - Ninth Revision," https://www.cdc.gov/nchs/icd/icd9cm.htm, 2015.

[2] "Data takes a quantum leap," 2020, https://quantium.com/wp-content/uploads/2016/08/BFM_Quantium.pdf.

[3] "Data never sleeps 9.0," 2020, https://www.domo.com/learn/infographic/data-never-sleeps-9.

[4] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 1, pp. 97–107, 2013.

[5] M. Herland, T. M. Khoshgoftaar, and R. Wald, "A review of data mining using big data in health informatics," *Journal of Big data*, vol. 1, no. 1, pp. 1–35, 2014.

[6] W.-T. Wu, Y.-J. Li, A.-Z. Feng, L. Li, T. Huang, A.-D. Xu, and J. Lyu, "Data mining in clinical big data: the frequently used databases, steps, and methodological models," *Military Medical Research*, vol. 8, no. 1, pp. 1–12, 2021.

[7] M. J. Zaki, W. M. Jr, and W. Meira, *Data mining and analysis: fundamental concepts and algorithms*, 2014.

[8] P. Fournier-Viger, J. C.-W. Lin, R.-U. Kiran, Y.-S. Koh, and R. Thomas, "A survey of sequential pattern mining," *Data Science and Pattern Recognition*, vol. 1, no. 1, p. 54–77, 2017.

[9] G. Poulis, G. Loukides, A. Gkoulalas-Divanis, and S. Skiadopoulos, "Anonymizing data with relational and transaction attributes," in *ECML-PKDD*, 2013, pp. 353–369.

[10] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig, "Syntactic clustering of the web," *Computer networks and ISDN systems*, vol. 29, no. 8-13, pp. 1157–1166, 1997.

[11] M. L. Wolraich and D. D. Drotar, "Chapter 6 - diagnostic classification systems," in *Developmental-Behavioral Pediatrics*. Mosby, 2008, pp. 109–122.

[12] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[13] International Classification of Diseases - Ninth Revision, https://www.cdc.gov/nchs/icd/icd9cm.htm, Last accessed 2021-06-01.

[14] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on neural networks*, vol. 16, no. 3, pp. 645–678, 2005.

[15] G. Poulis, G. Loukides, S. Skiadopoulos, and A. Gkoulalas-Divanis, “Anonymizing datasets with demographics and diagnosis codes in the presence of utility constraints,” *Journal of Biomedical Informatics*, vol. 65, pp. 76–96, 2017.

[16] Centers for Medicare & Medicaid Services, “Proposed changes to the CMS-HCC risk adjustment model for payment year 2017,” 2015.

[17] A. Kemp, D. B. Preen, C. Saunders, C. D. J. Holman, M. Bulsara, K. Rogers, and E. E. Roughead, “Ascertaining invasive breast cancer cases; the validity of administrative and self-reported data sources in australia,” *BMC Medical Research Methodology*, vol. 13, no. 1, p. 17, 2013.

[18] G. Tsoumakas and I. Katakis, “Multi-label classification: An overview,” *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[19] N. Mohammed, X. Jiang, R. Chen, B. C. Fung, and L. Ohno-Machado, “Privacy-preserving heterogeneous health data sharing,” *Journal of the American Medical Informatics Association*, vol. 20, no. 3, pp. 462–469, 2012.

[20] Y. Zhang, R. Padman, and N. Patel, “Paving the COWpath: Learning and visualizing clinical pathways from electronic health record data,” *Journal of Biomedical Informatics*, vol. 58, pp. 186–197, 2015.

[21] E. A. Campbell, E. J. Bass, and A. J. Masino, “Temporal condition pattern mining in large, sparse electronic health record data: A case study in characterizing pediatric asthma,” *Journal of the American Medical Informatics Association*, vol. 27, no. 4, pp. 558–566, 2020.

[22] D. Chushig-Muzo, C. Soguero-Ruiz, A. P. Engelbrecht, P. D. M. Bohoyo, and I. Mora-Jiménez, “Data-driven visual characterization of patient health-status using electronic health records and self-organizing maps,” *IEEE Access*, vol. 8, pp. 137 019–137 031, 2020.

[23] R. Guo, T. Fujiwara, Y. Li, K. M. Lima, S. Sen, N. K. Tran, and K.-L. Ma, “Comparative visual analytics for assessing medical records with sequence embedding,” *Visual Informatics*, vol. 4, no. 2, pp. 72–85, 2020.

[24] Y. Wang, Y. Zhao, T. M. Therneau, E. J. Atkinson, A. P. Tafti, N. Zhang, S. Amin, A. H. Limper, S. Khosla, and H. Liu, “Unsupervised machine learning for the discovery of latent disease clusters and patient subgroups using electronic health records,” *Journal of Biomedical Informatics*, vol. 102, p. 103364, 2020.

[25] P. M. Schnell, Q. Tang, W. W. Offen, and B. P. Carlin, “A bayesian credible subgroups approach to identifying patient subgroups with positive treatment effects,” *Biometrics*, vol. 72, no. 4, pp. 1026–1036, 2016.

[26] D. S. Hochbaum, “When are NP-hard location problems easy?” *Annals of Operations Research*, vol. 1, p. 201–214, 1984.

[27] V. V. Vazirani, *k-Center*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 47–53.

[28] O. Kariv and S. L. Hakimi, “An algorithmic approach to network location problems. ii: The p-medians,” *SIAM Journal on Applied Mathematics*, vol. 37, no. 3, pp. 539–560, 1979.

[29] J. Shi, N. Mamoulis, D. Wu, and D. W. Cheung, “Density-based place clustering in geo-social networks,” in *SIGMOD*, 2014, p. 99–110.

[30] P. Massa and P. Avesani, “Trust-aware recommender systems,” in *RecSys*, 2007, p. 17–24.

[31] S. Pai and G. D. Bader, “Patient similarity networks for precision medicine,” *Journal of Molecular Biology*, vol. 430, no. 18, Part A, pp. 2924 – 2938, 2018.

[32] G. Jeh and J. Widom, “SimRank: a measure of structural-context similarity,” in *KDD*, 2002, pp. 538–543.

[33] P. Fränti, S. Sieranoja, K. Wikström, T. Laatikainen *et al.*, “Clustering diagnoses from 58 million patient visits in finland between 2015 and 2018,” *JMIR Medical Informatics*, vol. 10, no. 5, p. e35422, 2022.

[34] A. Wartelle, F. Mourad-Chehade, F. Yalaoui, J. Chrusciel, D. Laplanche, and S. Sanchez, “Clustering of a health dataset using diagnosis co-occurrences,” *Applied Sciences*, vol. 11, no. 5, p. 2373, 2021.

[35] E. F. Codd, “A relational model of data for large shared data banks,” *Communications of the ACM*, vol. 13, no. 6, pp. 377–387, 1970.

[36] M. J. Zaki, W. Meira Jr, and W. Meira, *Data mining and analysis: fundamental concepts and algorithms*, 2014.

[37] B. Jin, C. Che, Z. Liu, S. Zhang, X. Yin, and X. Wei, “Predicting the risk of heart failure with ehr sequential data modeling,” *Ieee Access*, vol. 6, pp. 9256–9261, 2018.

[38] A. Wright, E. S.Chen, and F. L. Maloney, “An automated technique for identifying associations between medications, laboratory results and problems,” *Journal of Biomedical Informatics*, vol. 43, no. 6, pp. 891–901, 2010.

[39] F. P. Held, F. Blyth, D. Gnjidic, V. Hirani, V. Naganathan, L. M. Waite, M. J. Seibel, J. Rollo, D. J. Handelsman, R. G. Cumming, and D. G. L. Couteur, “Association rules analysis of comorbidity and multimorbidity: The concord health and aging in men project,” *The Journals of Gerontology: Series A*, vol. 71, no. 5, pp. 625–631, 2015.

[40] A. M. Shin, I. H. Lee, G. H. Lee, H. J. Park, H. S. Park, K. I. Yoon, J. J. Lee, and Y. N. Kim, “Diagnostic analysis of patients with essential hypertension using association rule mining,” *Healthcare informatics research*, vol. 16, no. 2, pp. 77–81, 2010.

[41] J. Kim, C. Bang, H. Hwang, D. Kim, C. Park, and S. Park, “IMA: Identifying disease-related genes using MeSH terms and association rules,” *Journal of biomedical informatics*, vol. 76, pp. 110–123, 2017.

[42] X. Chen, Y.-W. Niu, G.-H. Wang, and G.-Y. Yan, “Hamda: hybrid approach for mirna-disease association prediction,” *Journal of biomedical informatics*, vol. 76, pp. 50–58, 2017.

[43] M. hyung Kim, S. Banerjee, Y. Zhao, F. Wang, Y. Zhang, Y. Zhu, J. DeFerio, L. Evans, S. M. Park, and J. Pathak, "Association networks in a matched case-control design - Co-occurrence patterns of preexisting chronic medical conditions in patients with major depression versus their matched controls," *Journal of Biomedical Informatics*, vol. 87, pp. 88–95, 2018.

[44] V. Dinu, H. Zhao, and P. L. Miller, "Integrating domain knowledge with statistical and data mining methods for high-density genomic snp disease association analysis," *Journal of biomedical informatics*, vol. 40, no. 6, pp. 750–760, 2007.

[45] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, ser. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, p. 487–499.

[46] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *Transactions on Knowledge Discovery from Data*, vol. 3, no. 1, p. 1, 2009.

[47] V. Guralnik and G. Karypis, "A scalable algorithm for clustering sequential data," in *ICDM*, 2001, pp. 179–186.

[48] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *KDD*, vol. 6, no. 1, pp. 90–105, 2004.

[49] R. Gwadera, "Pattern-based solution risk model for strategic it outsourcing," in *ICDM*, vol. 7987, 2013, pp. 55–69.

[50] O. Niemenoja *et al.*, "Clustering and prediction of electronic health record data from mental health patients in a finnish healthcare environment," 2019.

[51] F. Folino, C. Pizzuti, and M. Ventura, "A comorbidity network approach to predict disease risk," in *International Conference on Information Technology in Bio-and Medical Informatics*. Springer, 2010, pp. 102–109.

[52] S. Sieranoja and P. Fränti, "Adapting k-means for graph clustering," *Knowledge and Information Systems*, vol. 64, no. 1, pp. 115–142, 2022.

[53] Y. Yang, X. Guan, and J. You, "Clope: a fast and effective clustering algorithm for transactional data," in *KDD*, 2002, pp. 682–687.

[54] H. Yan, K. Chen, and L. Liu, "Efficiently clustering transactional data with weighted coverage density," in *CIKM*, 2006, pp. 367–376.

[55] S. Guha, R. Rastogi, and K. Shim, "ROCK: A robust clustering algorithm for categorical attributes," *Information systems*, vol. 25, no. 5, pp. 345–366, 2000.

[56] K. R. Gøeg, R. Cornet, and S. K. Andersen, "Clustering clinical models from local electronic health records based on semantic similarity," *Journal of biomedical informatics*, vol. 54, pp. 294–304, 2015.

[57] H. Estiri, J. G. Klann, and S. N. Murphy, "A clustering approach for detecting implausible observation values in electronic health records data," *BMC medical informatics and decision making*, vol. 19, no. 1, pp. 1–16, 2019.

[58] V. Kuan, H. C. Fraser, M. Hingorani, S. Denaxas, A. Gonzalez-Izquierdo, K. Direk, D. Nitsch, R. Mathur, C. A. Parisinos, R. T. Lumbers *et al.*, “Data-driven identification of ageing-related diseases from electronic health records,” *Scientific reports*, vol. 11, no. 1, pp. 1–17, 2021.

[59] C. Sideris, B. Shahbazi, M. Pourhomayoun, N. Alshurafa, and M. Sarrafzadeh, “Using electronic health records to predict severity of condition for congestive heart failure patients,” in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, ser. UbiComp ’14 Adjunct. New York, NY, USA: Association for Computing Machinery, 2014, p. 1187–1192. [Online]. Available: https://doi.org/10.1145/2638728.2638815

[60] B. Chen, P. C. Tai, R. Harrison, and Y. Pan, “Novel hybrid hierarchical-k-means clustering method (hk-means) for microarray analysis,” in *2005 IEEE Computational Systems Bioinformatics Conference-Workshops (CSBW’05)*. IEEE, 2005, pp. 105–108.

[61] J. A. Hartigan and M. A. Wong, “A k-means clustering algorithm,” *Journal of the Royal Statistical Society*, vol. 28, no. 1, pp. 100–108, 1979.

[62] J. Blömer, C. Lammersen, M. Schmidt, and C. Sohler, *Theoretical Analysis of the k-Means Algorithm – A Survey*. Cham: Springer International Publishing, 2016, pp. 81–116. [Online]. Available: https://doi.org/10.1007/978-3-319-49487-6_3

[63] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” in *SODA*, 2007, pp. 1027–1035.

[64] H.-S. Park and C.-H. Jun, “A simple and fast algorithm for k-medoids clustering,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[65] M. Tiwari, M. J. Zhang, J. Mayclin, S. Thrun, C. Piech, and I. Shomorony, “Banditpam: Almost linear time k-medoids clustering via multi-armed bandits,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 10 211–10 222. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/73b817090081cef1bca77232f4532c5d-Paper.pdf

[66] E. Schubert and A. Zimek, “ELKI: A large open-source library for data analysis - ELKI release 0.7.5 "heidelberg",” *CoRR*, vol. abs/1902.03616, 2019. [Online]. Available: http://arxiv.org/abs/1902.03616

[67] *Partitioning Around Medoids (Program PAM)*. John Wiley Sons, Ltd, 1990, ch. 2, pp. 68–125. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470316801.ch2

[68] R. T. Ng and J. Han, “Clarans: A method for clustering objects for spatial data mining,” *IEEE Transactions on Knowledge amp; Data Engineering*, vol. 14, no. 05, pp. 1003–1016, sep 2002.

[69] A. Lopez-Martinez-Carrasco, J. M. Juarez, M. Campos, and B. Canovas-Segura, “A methodology based on trace-based clustering for patient phenotyping,” *Knowledge-Based Systems*, vol. 232, p. 107469, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705121007310

[70] R. Stewart, R. Jackson, R. Patel, S. Velupillai, G. Gkotsis, and D. Hoyle, "Knowledge discovery for deep phenotyping serious mental illness from electronic mental health records [version 2; referees: 2 approved]," *F1000Research*, vol. 7, p. 210, 2018.

[71] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[72] P. A. Popoola, J.-R. Tapamo, and A. G. Assounga, "Cluster analysis of mixed and missing chronic kidney disease data in kwazulu-natal province, south africa," *IEEE Access*, vol. 9, pp. 52 125–52 143, 2021.

[73] C. E. Coombes, X. Liu, Z. B. Abrams, K. R. Coombes, and G. Brock, "Simulation-derived best practices for clustering clinical data," *Journal of Biomedical Informatics*, vol. 118, p. 103788, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1532046421001179

[74] P. Bhattacharjee and P. Mitra, "A survey of density based clustering algorithms," *Frontiers of Computer Science*, vol. 15, no. 1, pp. 1–27, 2021.

[75] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 1, no. 3, pp. 231–240, 2011.

[76] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *KDD*, vol. 96, no. 34, 1996, pp. 226–231.

[77] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," in *SIGMOD*, vol. 28, no. 2, 1999, pp. 49–60.

[78] B. Andreopoulos, A. An, X. Wang, and D. Labudde, "Efficient layered density-based clustering of categorical data," *Journal of biomedical informatics*, vol. 42, no. 2, pp. 365–376, 2009.

[79] M. Maurits, T. Huizingal, S. Raychaudhuri, M. Reinders, E. Karlson, E. van den Akker, and R. Knevel, "A big-data approach to electronic health record data – using dimensionality reduction and clustering techniques to study longitudinal relationships between diseases," 2019.

[80] T. Lingren, P. Chen, J. Bochenek, F. Doshi-Velez, P. Manning-Courtney, J. Bickel, L. Wildenger Welchons, J. Reinhold, N. Bing, Y. Ni *et al.*, "Electronic health record based algorithm to identify patients with autism spectrum disorder," *PloS one*, vol. 11, no. 7, p. e0159621, 2016.

[81] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html

[82] R. B. Parikh, K. A. Linn, J. Yan, M. L. Maciejewski, A.-M. Rosland, K. G. Volpp, P. W. Groeneveld, and A. S. Navathe, “A machine learning approach to identify distinct subgroups of veterans at risk for hospitalization or death using administrative and electronic health record data,” *PloS one*, vol. 16, no. 2, p. e0247203, 2021.

[83] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, “Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis,” *IEEE journal of biomedical and health informatics*, vol. 22, no. 5, pp. 1589–1604, 2017.

[84] S. Yang, X. Zheng, C. Ji, and X. Chen, “Multi-layer representation learning and its application to electronic health records,” *Neural Processing Letters*, vol. 53, no. 2, pp. 1417–1433, 2021.

[85] Y. Wang, Y. Zhao, T. M. Therneau, E. J. Atkinson, A. P. Tafti, N. Zhang, S. Amin, A. H. Limper, S. Khosla, and H. Liu, “Unsupervised machine learning for the discovery of latent disease clusters and patient subgroups using electronic health records,” *Journal of biomedical informatics*, vol. 102, p. 103364, 2020.

[86] H. Aguiar, M. Santos, P. Watkinson, and T. Zhu, “Learning of cluster-based feature importance for electronic health record time-series,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 161–179.

[87] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, “A survey of clustering with deep learning: From the perspective of network architecture,” *IEEE Access*, vol. 6, pp. 39 501–39 514, 2018.

[88] C. Xiao, E. Choi, and J. Sun, “Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review,” *Journal of the American Medical Informatics Association*, vol. 25, no. 10, pp. 1419–1428, 2018.

[89] Q. Ma, J. Zheng, S. Li, and G. W. Cottrell, “Learning representations for time series clustering,” *Advances in neural information processing systems*, vol. 32, 2019.

[90] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, “Deepant: A deep learning approach for unsupervised anomaly detection in time series,” *Ieee Access*, vol. 7, pp. 1991–2005, 2018.

[91] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, “State-of-the-art in artificial neural network applications: A survey,” *Heliyon*, vol. 4, no. 11, p. e00938, 2018.

[92] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[93] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[94] X. Zhang, J. Chou, J. Liang, C. Xiao, Y. Zhao, H. Sarva, C. Henchcliffe, and F. Wang, “Data-driven subtyping of parkinson’s disease using longitudinal clinical records: a cohort study,” *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[95] C. Lee and M. Van Der Schaar, “Temporal phenotyping using deep predictive clustering of disease progression,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 5767–5777.

[96] I. Landi, B. S. Glicksberg, H.-C. Lee, S. Cherng, G. Landi, M. Danieletto, J. T. Dudley, C. Furlanello, and R. Miotto, “Deep representation learning of electronic health records to unlock patient stratification at scale,” *NPJ digital medicine*, vol. 3, no. 1, pp. 1–11, 2020.

[97] R. Ramazi, M. E. Bowen, and R. Beheshti, “Predicting acute events using the movement patterns of older adults: an unsupervised clustering method,” in *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 2022, pp. 1–9.

[98] U. Ahmed, J. C.-W. Lin, and G. Srivastava, “Mental health treatments using an explainable adaptive clustering model,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2022, pp. 212–222.

[99] W. Shao, X. Luo, Z. Zhang, Z. Han, V. Chandrasekaran, V. Turzhitsky, V. Bali, A. R. Roberts, M. Metzger, J. Baker *et al.*, “Application of unsupervised deep learning algorithms for identification of specific clusters of chronic cough patients from emr data,” *BMC bioinformatics*, vol. 23, no. 3, pp. 1–14, 2022.

[100] S. Zhou, H. Xu, Z. Zheng, J. Chen, J. Bu, J. Wu, X. Wang, W. Zhu, M. Ester *et al.*, “A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions,” *arXiv preprint arXiv:2206.07579*, 2022.

[101] M. Müller, “Dynamic time warping,” *Information retrieval for music and motion*, pp. 69–84, 2007.

[102] P. LePendu, S. V. Iyer, C. Fairon, and N. H. Shah, “Annotation analysis for testing drug safety signals using unstructured clinical notes,” in *Journal of biomedical semantics*, vol. 3, no. 1. Springer, 2012, pp. 1–12.

[103] K. Kroenke, R. L. Spitzer, and J. B. Williams, “The phq-9: validity of a brief depression severity measure,” *Journal of general internal medicine*, vol. 16, no. 9, pp. 606–613, 2001.

[104] K. Blondeau, L. Dupont, V. Mertens, J. Tack, and D. Sifrim, “Improved diagnosis of gastro-oesophageal reflux in patients with unexplained chronic cough,” *Alimentary pharmacology & therapeutics*, vol. 25, no. 6, pp. 723–732, 2007.

[105] D. Kiyasseh, T. Zhu, and D. Clifton, “Crocs: Clustering and retrieval of cardiac signals based on patient disease class, sex, and age,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 557–15 569, 2021.

[106] J. Zhao, P. Papapetrou, L. Asker, and H. Boström, “Learning from heterogeneous temporal data in electronic health records,” *Journal of biomedical informatics*, vol. 65, pp. 105–119, 2017.

[107] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[108] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 661–18 673, 2020.

[109] L. Kaufman and P. J. Rousseeuw, "Partitioning around medoids (program pam)," *Finding groups in data: an introduction to cluster analysis*, vol. 344, pp. 68–125, 1990.

[110] Healthcare Information and Management Systems Society (HIMSS), https://www.himss.org/library/ehr, 2016.

[111] P. Campanella, E. Lovato, C. Marone, L. Fallacara, A. Mancuso, W. Ricciardi, and M. L. Specchia, "The impact of electronic health records on healthcare quality: a systematic review and meta-analysis," *The European Journal of Public Health*, vol. 26, no. 1, pp. 60–64, 2015.

[112] C. Rinner, S. K. Sauter, G. Endel, G. Heinze, S. Thurner, P. Klimek, and G. Duftschmid, "Improving the informational continuity of care in diabetes mellitus treatment with a nationwide shared EHR system: Estimates from austrian claims data," *International journal of medical informatics*, vol. 92, pp. 44–53, 2016.

[113] D. Gotz, J. Sun, N. Cao, and S. Ebadollahi, "Visual cluster analysis in support of clinical decision intelligence," in *AMIA*, vol. 2011, 2011, p. 481.

[114] P. Yadav, M. Steinbach, V. Kumar, and G. Simon, "Mining electronic health records (EHRs): a survey," *ACM Computing Surveys*, vol. 50, no. 6, p. 85, 2018.

[115] R. J. Carroll, A. E. Eyler, and J. C. Denny, "Intelligent use and clinical benefits of electronic health records in rheumatoid arthritis," *Expert Review of Clinical Immunology*, vol. 11, no. 3, pp. 329–337, 2015.

[116] N. Sokolovska, O. Cappé, and F. Yvon, "The asymptotics of semi-supervised learning in discriminative probabilistic models," in *ICML*, 2008, pp. 984–991.

[117] V. Nouri, M.-R. Akbarzadeh-T, and A. Rowhanimanesh, "A hybrid type-2 fuzzy clustering technique for input data preprocessing of classification algorithms," in *FUZZ-IEEE*, 2014, pp. 1131–1138.

[118] R. Henriques, F. L. Ferreira, and S. C. Madeira, "BicPAMS: software for biological data analysis with pattern-based biclustering," *BMC bioinformatics*, vol. 18, no. 1, p. 82, 2017.

[119] A. Zhang, C. Tang, and D. Jiang, "Cluster analysis for gene expression data: A survey," *IEEE Transactions on Knowledge & Data Engineering*, no. 11, pp. 1370–1386, 2004.

[120] J. Lustgarten, V. Gopalakrishnan, H. Grover, and S. V. S, "Improving classification performance with discretization on biomedical datasets," in *AMIA*, 2008, pp. 445–449.

[121] F. Giannotti, C. Gozzi, and G. Manco, "Clustering transactional data," in *ECML-PKDD*, 2002.

[122] A. S. Shirkhorshidi, S. Aghabozorgi, and T. Y. Wah, “A comparison study on similarity and dissimilarity measures in clustering continuous data,” *PLOS ONE*, vol. 10, 12 2015.

[123] P. B. Jensen, L. J. Jensen, and S. Brunak, “Mining electronic health records: towards better research applications and clinical care,” *Nature Reviews Genetics*, vol. 13, no. 6, p. 395, 2012.

[124] F. Cao, J. Z. Huang, J. Liang, X. Zhao, Y. Meng, K. Feng, and Y. Qian, “An algorithm for clustering categorical data with set-valued features,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4593–4606, 2018.

[125] L. Kalankesh, J. Weatherall, T. Ba-Dhfari, I. E. Buchan, and A. Brass, “Taming EHR data: Using semantic similarity to reduce dimensionality,” *Studies in health technology and informatics*, vol. 192, pp. 52–56, 2013.

[126] F. S. Roque, P. B. Jensen, H. Schmock, M. Dalgaard, M. Andreatta, T. Hansen, K. Søeby, S. Bredkjær, A. Juul, T. Werge, L. J. Jensen, and S. Brunak, “Using electronic patient records to discover disease correlations and stratify patient cohorts,” *PLOS Computational Biology*, vol. 7, no. 8, pp. 1–10, 2011.

[127] F. Doshi-Velez, Y. Ge, and I. Kohane, “Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis,” *Pediatrics*, vol. 133, no. 1, pp. e54–e63, 2014.

[128] S. Ghassempour, F. Girosi, and A. Maeder, “Clustering multivariate time series using hidden markov models,” *International journal of environmental research and public health*, vol. 11, no. 3, pp. 2741–2763, 2014.

[129] C. E. Lopez, S. Tucker, T. Salameh, and C. S. Tucker, “An unsupervised machine learning method for discovering patient clusters based on genetic signaturess,” *Journal of Biomedical Informatics*, vol. 85, pp. 30–39, 2018.

[130] A. Ultsch and J. Loetsch, “Machine-learned cluster identification in high-dimensional data,” *Journal of biomedical informatics*, vol. 66, pp. 95–104, 2017.

[131] H. Xu, Y. Wu, N. Elhadad, P. D. Stetson, and C. Friedman, “A new clustering method for detecting rare senses of abbreviations in clinical notes,” *Journal of Biomedical Informatics*, vol. 45, no. 6, pp. 1075–1083, 2012.

[132] M. Moradi, “CIBS: A biomedical text summarizer using topic-based sentence clustering,” *Journal of biomedical informatics*, vol. 88, pp. 53–61, 2018.

[133] C. C. Aggarwal and C. Zhai, “A survey of text clustering algorithms,” in *Mining Text Data*, 2012, pp. 77–128.

[134] B. C. Fung, K. Wang, and M. Ester, “Hierarchical document clustering using frequent itemsets,” in *SDM*, 2003, pp. 59–70.

[135] C. Su, Q. Chen, X. Wang, and X. Meng, “Text clustering approach based on maximal frequent term sets,” in *IEEE SMC*, 2009, pp. 1551–1556.

[136] G. Kiran, R. Shankar, and V. Pudi, “Frequent itemset based hierarchical document clustering using wikipedia as external knowledge,” in *International Conference on Knowledge-based and Intelligent Information and Engineering Systems*, 2010, pp. 11–20.

[137] S. C. Madeira and A. L. Oliveira, “Biclustering algorithms for biological data analysis: a survey,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, no. 1, pp. 24–45, 2004.

[138] Y. Cheng and G. M. Church, “Biclustering of expression data,” in *ISMB*, vol. 8, no. 2000, 2000, pp. 93–103.

[139] I. V. Mechelen, H.-H. Bock, and P. D. Boeck, “Two-mode clustering methods: a structured overview,” *Statistical methods in medical research*, vol. 13, no. 5, pp. 363–394, 2004.

[140] A. Tanay, R. Sharan, and R. Shamir, *Handbook of computational molecular biology*, vol. 9, no. 1-20, pp. 122–124, 2005.

[141] D. Cartwright, “ICD-9-CM to ICD-10-CM Codes: What? Why? How?” *Advances in wound care*, vol. 2, no. 10, pp. 588–592, 2013.

[142] E. R. Omiecinski, “Alternative interest measures for mining associations in databases,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 1, pp. 57–69, 2003.

[143] K. Gouda and M. J. Zaki, “Efficiently mining maximal frequent itemsets,” in *ICDM*, 2001, pp. 163–170.

[144] D. Burdick, M. Calimlim, and J. Gehrke, “MAFIA: a maximal frequent itemset algorithm for transactional databases,” in *ICDE*, vol. 1, 2001, pp. 443–452.

[145] G. Grahne and J. Zhu, “High performance mining of maximal frequent itemsets,” in *6th International Workshop on High Performance Data Mining*, vol. 16, 2003, p. 34.

[146] R. C. de Amorim and C. Hennig, “Recovering the number of clusters in data sets with noise features using feature rescaling factors,” *Information Sciences*, vol. 324, pp. 126–145, 2015.

[147] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[148] S. Sahni and T. Gonzalez, “P-complete approximation problems,” *Journal of the ACM*, vol. 23, no. 3, pp. 555–565, 1976.

[149] A. Czumaj and C. Sohler, “Small space representations for metric min-sum k-clustering and their applications,” in *STACS*, 2007, pp. 536–548.

[150] G. Yang, “The complexity of mining maximal frequent itemsets and maximal frequent patterns,” in *KDD*, 2004, pp. 344–353.

[151] T. M. Kodinariya and P. R. Makwana, “Review on determining number of cluster in K-means clustering,” *International Journal*, vol. 1, no. 6, pp. 90–95, 2013.

[152] L. Peng, W. Qing, and G. Yujia, “Study on comparison of discretization methods,” in *AICI*, vol. 4, 2009, pp. 380–384.

[153] D. Müllner, “Modern hierarchical, agglomerative clustering algorithms,” *CoRR*, vol. abs/1109.2378, 2011.

[154] Vermont Department of Health, “Vermont uniform hospital discharge data sets,” http://www.healthvermont.gov/health-statistics-vital-records/health-care-systems-reporting/hospital-discharge-data, 2017.

[155] K. Finison, M. Mohlman, C. Jones, M. Pinette, D. Jorgenson, A. Kinner, T. Tremblay, and D. Gottlieb, “Risk-adjustment methods for all-payer comparative performance reporting in vermont,” *BMC Health Services Research*, vol. 17, no. 1, p. 58, 2017.

[156] A. Johnson, L. Shulman, J. Kachajian, B. L. Sprague, F. Khan, T. James, D. Cranmer, P. Young, and R. Heimann, “Access to care in Vermont: factors linked with time to chemotherapy for women with breast cancer-a retrospective cohort study,” *Journal of oncology practice*, vol. 12, no. 9, pp. e848–e857, 2016.

[157] Informs Data Mining Contest, “Informs data mining contest data sets,” https://sites.google.com/site/informsdataminingcontest/data/, 2008.

[158] G. Loukides, J. Liagouris, A. Gkoulalas-Divanis, and M. Terrovitis, “Disassociation for electronic health record privacy,” *Journal of biomedical informatics*, vol. 50, pp. 46–61, 2014.

[159] S. Rosset, C. Perlich, G. Świrszcz, P. Melville, and Y. Liu, “Medical data mining: insights from winning two competitions,” *Data Mining and Knowledge Discovery*, vol. 20, no. 3, pp. 439–468, 2010.

[160] J. K. Stoller, R. J. Panos, S. Krachman, D. E. Doherty, B. Make, and Long-term Oxygen Treatment Trial Research Group, “Oxygen therapy for patients with COPD: current evidence and the long-term oxygen treatment trial,” *Chest*, vol. 138, no. 1, pp. 179–187, 2010.

[161] F. Zaidi, R. S. Lee, B. A. Buchcic, N. E. Bracken, H. A. Jaffe, M. Joo, V. Prieto-Centurion, A.-Y. Tan, and J. A. Krishnan, “Evaluation and documentation of supplemental oxygen requirements is rarely performed in patients hospitalized with COPD,” *Chronic Obstructive Pulmonary Diseases: Journal of the COPD Foundation*, vol. 4, no. 4, p. 287, 2017.

[162] S. Mora and J. E. Manson, “Aspirin for primary prevention of atherosclerotic cardiovascular disease: advances in diagnosis and treatment,” *JAMA Internal Medicine*, vol. 176, no. 8, pp. 1195–1204, 2016.

[163] J.-J. Sheu, J.-H. Kang, H.-Y. Lou, and H.-C. Lin, “Reflux esophagitis and the risk of stroke in young adults: a 1-year population-based follow-up study,” *Stroke*, vol. 41, no. 9, pp. 2033–2037, 2010.

[164] C.-H. Chen, C.-L. Lin, and C.-H. Kao, “Association between gastroesophageal reflux disease and coronary heart disease: A nationwide population-based analysis,” *Medicine*, vol. 95, no. 27, 2016.

[165] A. Romero-Corral, S. M. Caples, F. Lopez-Jimenez, and V. K. Somers, “Interactions between obesity and obstructive sleep apnea: implications for treatment,” *Chest*, vol. 137, no. 3, pp. 711–719, 2010.

[166] S. Jehan, F. Zizi, S. R. Pandi-Perumal, S. Wall, E. Auguste, A. K. Myers, G. Jean-Louis, and S. I. McFarlane, “Obstructive sleep apnea and obesity: implications for public health,” *Sleep medicine and disorders: international journal*, vol. 1, no. 4, 2017.

[167] M. Szkup, A. Jurczak, B. Karakiewicz, A. Kotwas, J. Kopeć, and E. Grochans, “Influence of cigarette smoking on hormone and lipid metabolism in women in late reproductive stage,” *Clinical interventions in aging*, vol. 13, p. 109, 2018.

[168] N. L. Benowitz, “Safety of nicotine in smokers with hypertension,” 2001.

[169] Z. tong Li, F. Ji, X. wei Han, L. Wang, Y. qiang Yue, and Z. gao Wang, “The role of gastroesophageal reflux in provoking high blood pressure episodes in patients with hypertension,” *Journal of clinical gastroenterology*, vol. 52, no. 8, p. 685, 2018.

[170] C. Tuegel and N. Bansal, “Heart failure in patients with kidney disease,” *Heart*, vol. 103, no. 23, pp. 1848–1853, 2017.

[171] C.-L. Huang and E. Kuo, “Mechanism of hypokalemia in magnesium deficiency,” *Journal of the American Society of Nephrology*, vol. 18, no. 10, pp. 2649–2652, 2007.

[172] L. M. Román-Pintos, G. Villegas-Rivera, A. D. Rodríguez-Carrizalez, A. G. Miranda-Díaz, and E. G. Cardona-Muñoz, “Diabetic polyneuropathy in type 2 diabetes mellitus: inflammation, oxidative stress, and mitochondrial function,” *Journal of diabetes research*, vol. 2016, 2016.

[173] P. Nasa, D. Juneja, and O. Singh, “Severe sepsis and septic shock in the elderly: an overview,” *World journal of critical care medicine*, vol. 1, no. 1, p. 23, 2012.

[174] M. Mallappallil, E. A. Friedman, B. G. Delano, S. I. McFarlane, and M. O. Salifu, “Chronic kidney disease in the elderly: evaluation and management,” *Clinical practice (London, England)*, vol. 11, no. 5, p. 525, 2014.

[175] M. Rafieian-Kopaei, M. Setorki, M. Doudi, A. Baradaran, and H. Nasri, “Atherosclerosis: process, indicators, risk factors and new hopes,” *International journal of preventive medicine*, vol. 5, no. 8, p. 927, 2014.

[176] Expert Panel on Detection and Evaluation and Treatment of High Blood Cholesterol in Adults, “Executive summary of the third report of the national cholesterol education program (NCEP) expert panel on detection, evaluation, and treatment of high blood cholesterol in adults (Adult Treatment Panel III).” *JAMA*, vol. 285, no. 19, p. 2486, 2001.

[177] J. Wang, J. J. Ma, J. Liu, D. D. Zeng, C. Song, and Z. Cao, “Prevalence and risk factors of comorbidities among hypertensive patients in china,” *International journal of medical sciences*, vol. 14, no. 3, pp. 201–212, 2017.

[178] R. P. Beasley, C.-Y. L. George, C.-H. Roan, L.-Y. Hwang, C.-C. Lan, F.-Y. Huang, and C.-L. Chen, "Prevention of perinatally transmitted hepatitis B virus infections with hepatitis B immune globulin and hepatitis B vaccine," *The Lancet*, vol. 322, no. 8359, pp. 1099–1102, 1983.

[179] World Health Organization in South-East Asia, "Health situation and trend assessment," http://www.searo.who.int/entity/health_situation_trends/data/chi/elderly-population/en/, 2019.

[180] L. Ohno-Machado, "Mining electronic health record data: finding the gold nuggets," *Journal of the American Medical Informatics Association*, vol. 22, no. 5, pp. 937–937, 2015.

[181] D. Gartner and R. Padman, "Mathematical modelling and cluster analysis in healthcare analytics-the case of length of stay management," in *ICIS*, 2016.

[182] J. M. Sanderson, D. C. Proops, L. Trieu, E. Santos, B. Polsky, and S. D. Ahuja, "Increasing the efficiency and yield of a tuberculosis contact investigation through electronic data systems matching," *Journal of the American Medical Informatics Association*, vol. 22, no. 5, pp. 1089–1093, 2015.

[183] N. Sokolovska, O. Cappé, and F. Yvon, "The asymptotics of semi-supervised learning in discriminative probabilistic models," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 984–991.

[184] H. Calik, M. Labbé, and H. Yaman, *p-Center Problems*, 2015, pp. 79–92.

[185] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, pp. 293–306, 1985.

[186] H. Zhong, G. Loukides, and R. Gwadera, "Clustering datasets with demographics and diagnosis codes," *Journal of Biomedical Informatics*, vol. 102, p. 103360, 2020.

[187] X. Zhang, H. Liu, Q. Li, and X. Wu, "Attributed graph clustering via adaptive graph convolution," in *IJCAI*, 2019, pp. 4327–4333.

[188] C. C. Aggarwal, Ed., *Data Classification: Algorithms and Applications*. CRC Press, 2014.

[189] G. Gan, C. Ma, and J. Wu, *Data clustering: theory, algorithms, and applications*. SIAM, 2020.

[190] Z. Bar-Joseph, D. K. Gifford, and T. S. Jaakkola, "Fast optimal leaf ordering for hierarchical clustering," *Bioinformatics*, vol. 17, pp. S22–S29, 2001.

[191] R. A. Hubbard, J. Xu, R. Siegel, Y. Chen, and I. Eneli, "Studying pediatric health outcomes with electronic health records using bayesian clustering and trajectory analysis," *Journal of Biomedical Informatics*, vol. 113, p. 103654, 2021.

[192] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in *COLING*, 1997.

[193] D. Sánchez, M. Batet, and D. Isern, "Ontology-based information content computation," *Knowledge-based systems*, vol. 24, no. 2, pp. 297–303, 2011.

[194] G. Navarro, “A guided tour to approximate string matching,” *ACM Computing Surveys*, vol. 33, no. 1, pp. 31–88, 2001.

[195] E. Aspland, P. R. Harper, D. Gartner, P. Webb, and P. Barrett-Lee, “Modified needleman–wunsch algorithm for clinical pathway clustering,” *Journal of Biomedical Informatics*, vol. 115, p. 103668, 2021.

[196] N. Miura and T. Takagi, “Wsl: sentence similarity using semantic distance between words,” in *SemEval)*, 2015, pp. 128–131.

[197] ICD 9 Code Hierarchy, https://en.wikipedia.org/wiki/List_of_ICD-9_codes, Last accessed 2021-06-01.

[198] S. Avgustinovich and D. Fon-Der-Flaass, “Cartesian products of graphs and metric spaces,” *European Journal of Combinatorics*, vol. 21, no. 7, pp. 847 – 851, 2000.

[199] L. Wirbka, W. E. Haefeli, and A. D. Meid, “A framework to build similarity-based cohorts for personalized treatment advice–a standardized, but flexible workflow with the r package simbaco,” *PloS one*, vol. 15, no. 5, p. e0233686, 2020.

[200] *Center Problems.* John Wiley & Sons, Ltd, 1995, ch. 5, pp. 154–197.

[201] S. Olafsson, X. Li, and S. Wu, “Operations research and data mining,” *European Journal of Operational Research*, vol. 187, no. 3, pp. 1429–1448, 2008.

[202] H. D. Vinod, “Integer programming and the theory of grouping,” *Journal of the American Statistical association*, vol. 64, no. 326, pp. 506–519, 1969.

[203] P. Bradley, O. Mangasarian, and W. Street, “Clustering via concave minimization,” *Advances in neural information processing systems*, vol. 9, 1996.

[204] P. S. Bradley and O. L. Mangasarian, “K-plane clustering,” *Journal of Global optimization*, vol. 16, no. 1, pp. 23–32, 2000.

[205] M. R. Garey and D. S. Johnson, “Computers and intractability,” *A Guide to the*, 1979.

[206] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, “Mimic-iii, a freely accessible critical care database,” *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[207] INFORMS, “Informs data mining contest,” https://sites.google.com/site/informsdataminingcontest/data/, Last accessed 2021-06-01.

[208] M. Kastner, N. L. Wilczynski, C. Walker-Dilks, K. A. McKibbon, and B. Haynes, “Age-specific search strategies for medline,” *Journal of Medical Internet Research*, vol. 8, no. 4, p. e25, 2006.

[209] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, “Adversarially regularized graph autoencoder for graph embedding,” in *IJCAI*, 2018, p. 2609–2615.

[210] U. Von Luxburg, “A tutorial on spectral clustering,” *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[211] D. J. Ketchen and C. L. Shook, “The application of cluster analysis in strategic management research: an analysis and critique,” *Strategic Management Journal*, vol. 17, no. 6, pp. 441–458, 1996.

[212] ICD-9-CM Chapters, https://icd.codes/icd9cm, Last accessed 2021-06-01.

[213] M. S. Dzeshka, A. Shantsila, E. Shantsila, and G. Y. Lip, “Atrial fibrillation and hypertension,” *Hypertension*, vol. 70, no. 5, pp. 854–861, 2017.

[214] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, “Prefixspan,: mining sequential patterns efficiently by prefix-projected pattern growth,” in *ICDE*, 2001, pp. 215–224.

[215] S. A. Lubitz, E. J. Benjamin, and P. T. Ellinor, “Atrial fibrillation in congestive heart failure,” *Heart Failure Clinics*, vol. 6, no. 2, pp. 187–200, 2010.

[216] E. Michniewicz, E. Mlodawska, P. Lopatowska, A. Tomaszuk-Kazberuk, and J. Malyszko, “Patients with atrial fibrillation and coronary artery disease–double trouble,” *Advances in Medical Sciences*, vol. 63, no. 1, pp. 30–35, 2018.

[217] T. Weber, I. Lang, R. Zweiker, S. Horn, R. R. Wenzel, B. Watschinger, J. Slany, B. Eber, F. X. Roithinger, and B. Metzler, “Hypertension and coronary artery disease: epidemiology, physiology, effects of treatment, and recommendations,” *Wiener klinische Wochenschrift*, vol. 128, no. 13, pp. 467–479, 2016.

[218] C. Wilkinson, O. Todd, A. Clegg, C. P. Gale, and M. Hall, “Management of atrial fibrillation for older people with frailty: a systematic review and meta-analysis,” *Age Ageing*, vol. 48, no. 2, pp. 196–203, 2019.

[219] R. P. Ames, “Hyperlipidemia in hypertension: causes and prevention,” *American heart journal*, vol. 122, no. 4, pp. 1219–1224, 1991.

[220] B. Ivanovic and M. Tadic, “Hypercholesterolemia and hypertension: two sides of the same coin,” *American Journal of Cardiovascular Drugs*, vol. 15, no. 6, pp. 403–414, 2015.

[221] Z.-t. Li, F. Ji, X.-w. Han, L. Wang, Y.-q. Yue, and Z.-g. Wang, “The role of gastroesophageal reflux in provoking high blood pressure episodes in patients with hypertension,” *Journal of Clinical Gastroenterology*, vol. 52, no. 8, p. 685, 2018.

[222] M. H. Mellow, A. G. Simpson, L. Watt, L. Schoolmeester, and O. L. Haye, “Esophageal acid perfusion in coronary artery disease: induction of myocardial ischemia,” *Gastroenterology*, vol. 85, no. 2, pp. 306–312, 1983.

[223] J. P. P. Moraes-Filho, T. Navarro-Rodriguez, J. N. Eisig, R. C. Barbuti, D. Chinzon, and E. M. Quigley, “Comorbidities are frequent in patients with gastroesophageal reflux disease in a tertiary health care hospital,” *Clinics*, vol. 64, no. 8, pp. 785–790, 2009.

[224] K. Gao, X. Shi, and W. Wang, “The life-course impact of smoking on hypertension, myocardial infarction and respiratory diseases,” *Scientific Reports*, vol. 7, no. 1, pp. 1–7, 2017.

[225] D. J. Drobes, "Concurrent alcohol and tobacco dependence: mechanisms and treatment," *Alcohol Res Health.*, vol. 26, no. 2, p. 136, 2002.

[226] B. Balbi, V. Cottin, S. Singh, W. De Wever, F. Herth, C. R. Cordeiro *et al.*, "Smoking-related lung diseases: a clinical perspective," *European Respiratory Journal*, vol. 35, no. 2, pp. 231–233, 2010.

[227] L. R. Engelking, *Textbook of Veterinary Physiological Chemistry, Updated 2/e*. Academic Press, 2010.

[228] P. K. Moore, R. K. Hsu, and K. D. Liu, "Management of acute kidney injury: core curriculum 2018," *American Journal of Kidney Diseases*, vol. 72, no. 1, pp. 136–148, 2018.

[229] P. Kazemian, G. Oudit, and B. I. Jugdutt, "Atrial fibrillation and heart failure in the elderly," *Heart Failure Reviews*, vol. 17, no. 4-5, pp. 597–613, 2012.

[230] M. V. Madhavan, B. J. Gersh, K. P. Alexander, C. B. Granger, and G. W. Stone, "Coronary artery disease in patients $\geq$ 80 years of age," *Journal of the American College of Cardiology*, vol. 71, no. 18, pp. 2015–2040, 2018.

[231] B. Everett and A. Zajacova, "Gender differences in hypertension and hypertension awareness among young adults," *Biodemography and Social Biology*, vol. 61, no. 1, pp. 1–17, 2015.

[232] C. Canal, R. Fontelo, I. Hamouda, J. Guillem-Marti, U. Cvelbar, and M.-P. Ginebra, "Plasma-induced selectivity in bone cancer cells death," *Free Radical Biology and Medicine*, vol. 110, pp. 72–80, 2017.

[233] C. Weir and W. S. Millar, "The effects of neonatal jaundice and respiratory complications on learning and habituation in 5-to 11-month-old infants," *Journal of Child Psychology and Psychiatry and Allied Disciplines*, vol. 38, no. 2, pp. 199–206, 1997.

[234] S. B. Amin and H. Wang, "Unbound unconjugated hyperbilirubinemia is associated with central apnea in premature infants," *The Journal of pediatrics*, vol. 166, no. 3, pp. 571–575, 2015.

[235] P. A. Dennery, D. S. Seidman, and D. K. Stevenson, "Neonatal hyperbilirubinemia," *The New England Journal of Medicine*, vol. 344, no. 8, pp. 581–590, 2001.

[236] P. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*, 2nd ed. Pearson, 2018.

[237] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[238] G. Guo, J. Zhang, D. Thalmann, A. Basu, and N. Yorke-Smith, "From ratings to trust: An empirical study of implicit trust in recommender systems," in *SAC*, 2014, p. 248–253.

[239] B. Wang, A. M. Mezlini, F. Demir, M. Fiume, Z. Tu, M. Brudno, B. Haibe-Kains, and A. Goldenberg, "Similarity network fusion for aggregating data types on a genomic scale," *Nature methods*, vol. 11, no. 3, pp. 333–337, 2014.

[240] H. Gao, J. Tang, and H. Liu, "Exploring social-historical ties on location-based social networks," in *AAAI*, 2012.

[241] X. Yu, A. Pan, L.-A. Tang, Z. Li, and J. Han, "Geo-friends recommendation in gps-based cyber-physical social network," in *ASONAM*, 2011, pp. 361–368.

[242] Y. Matsuo and H. Yamamoto, "Community gravity: measuring bidirectional effects by trust and rating on online social networks," in *WWW*, 2009, pp. 751–760.

[243] H. Zhong, G. Loukides, and S. P. Pissis, "Clustering demographics and sequences of diagnosis codes," *IEEE Journal of Biomedical and Health Informatics*, 2021.

[244] E. W. Myers, "The fragment assembly string graph," *Bioinformatics*, vol. 21, no. suppl_2, pp. ii79–ii85, 2005.

[245] S. Jun, G. Sims, G. A. Wu, and S. Kim, "Whole-proteome phylogeny of prokaryotes by feature frequency profiles: An alignment-free method with optimal feature resolution," *PNAS*, vol. 107, no. 1, pp. 133–138, 2010.

[246] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *AAAI*, 2014, pp. 2149–2155.

[247] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information." in *IJCAI*, 2015, pp. 2111–2117.

[248] X. Wang, D. Jin, X. Cao, L. Yang, and W. Zhang, "Semantic community identification in large attribute networks." in *AAAI*, 2016, pp. 265–271.

[249] L. Akoglu, H. Tong, B. Meeder, and C. Faloutsos, "PICS: Parameter-free identification of cohesive subgroups in large attributed graphs," in *SDM*, 2012, pp. 439–450.

[250] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *IJCAI*, 2018, p. 2609–2615.

[251] S. Yang and B. Yang, "Enhanced network embedding with text information," in *ICPR*, 2018, pp. 326–331.

[252] H. Yang, S. Pan, P. Zhang, L. Chen, D. Lian, and C. Zhang, "Binarized attributed network embedding," in *ICDM*, 2018, pp. 1476–1481.

[253] T. Warnow, *Computational Phylogenetics: An Introduction to Designing Methods for Phylogeny Estimation*, 1st ed. USA: Cambridge University Press, 2017.

[254] D. H. Huson and C. Scornavacca, "A survey of combinatorial methods for phylogenetic networks," *Genome biology and evolution*, vol. 3, pp. 23–35, 2011.

[255] C. C. Aggarwal and H. Wang, "A survey of clustering algorithms for graph data," in *Managing and mining graph data*. Springer, 2010, pp. 275–301.

[256] C. Bothorel, J. D. Cruz, M. Magnani, and B. Micenkova, "Clustering attributed graphs: Models, measures and methods," *Network Science*, vol. 3, no. 3, p. 408–444, 2015.

[257] J. A. Carriço, M. Crochemore, A. P. Francisco, S. P. Pissis, B. Ribeiro-Gonçalves, and C. Vaz, "Fast phylogenetic inference from typing data," *Algorithms for Molecular Biology*, vol. 13, no. 1, pp. 4:1–4:14, 2018.

[258] T. Xiong, S. Wang, Q. Jiang, and J. Z. Huang, "A new Markov model for clustering categorical sequences," in *ICDM*, 2011, pp. 854–863.

[259] H. N. Djidjev and M. Onus, "Scalable and accurate graph clustering and community structure detection," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 1022–1029, 2013.

[260] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[261] J. Laeuchli, "Fast community detection with graph sparsification," in *PAKDD*, 2020, pp. 291–304.

[262] J. Pei, D. Jiang, and A. Zhang, "On mining cross-graph quasi-cliques," in *KDD*, 2005, p. 228–238.

[263] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014, p. 701–710.

[264] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton, "Gemsec: Graph embedding with self clustering," in *ASONAM*, 2019, pp. 65–72.

[265] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels." *Journal of Machine Learning Research*, vol. 12, no. 9, 2011.

[266] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowledge-Based Systems*, vol. 151, pp. 78–94, 2018.

[267] M. Crochemore, C. Hancart, and T. Lecroq, *Algorithms on strings*. Cambridge University Press, 2007.

[268] M. Potamias, F. Bonchi, C. Castillo, and A. Gionis, "Fast shortest path distance estimation in large networks," in *CIKM*. ACM, 2009, pp. 867–876.

[269] B. Youngmann, T. Milo, and A. Somech, "Boosting SimRank with semantics," in *EDBT*, 2019, pp. 37–48.

[270] W. Yu, W. Zhang, X. Lin, Q. Zhang, and J. Le, "A space and time efficient algorithm for SimRank computation," *World Wide Web*, vol. 15, p. 327–353, 2012.

[271] W. Yu, X. Lin, W. Zhang, J. Pei, and J. A. Mccann, "SimRank*: Effective and scalable pairwise similarity search based on graph topology," *VLDBJ*, vol. 28, no. 3, p. 401–426, 2019.

[272] D. A. Brannan, *A First Course in Mathematical Analysis*. Cambridge University Press, 2006.

[273] A. Backurs and P. Indyk, "Edit distance cannot be computed in strongly subquadratic time (unless SETH is false)," in *STOC*, 2015, p. 51–58.

[274] D. Chakraborty, E. Goldenberg, and M. Kouckỳ, “Streaming algorithms for embedding and computing edit distance in the low distance regime,” in *STOC*, 2016, pp. 712–725.

[275] R. A. Wagner and M. J. Fischer, “The string-to-string correction problem,” *Journal of the ACM*, vol. 21, no. 1, p. 168–173, Jan. 1974.

[276] M. Thorup, “Undirected single-source shortest paths with positive integer weights in linear time,” *Journal of the ACM*, vol. 46, no. 3, p. 362–394, May 1999.

[277] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, and C. Zhang, “Attributed graph clustering: A deep attentional embedding approach,” in *IJCAI*, 2019, pp. 3670–3676.

[278] H. Zhang and Q. Zhang, “Embedjoin: Efficient edit similarity joins via embeddings,” in *KDD*, 2017, p. 585–594.

[279] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang, “Image clustering using local discriminant models and global integration,” *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2761–2773, 2010.

[280] A. Amelio and C. Pizzuti, “Is normalized mutual information a fair measure for comparing community detection methods?” in *ASONAM*, 2015, pp. 1584–1585.

[281] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[282] “National Center for Biotechnology Information (NCBI),” https://www.ncbi.nlm.nih.gov/.

[283] Y. Li, L. He, R. L. He, and S. S.-T. Yau, “A novel fast vector method for genetic sequence comparison,” *Scientific reports*, vol. 7, no. 1, pp. 1–11, 2017.

[284] J. Chang, “Biopython: Tutorial and cookbook,” 2020.

[285] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[286] A. Tamersoy, G. Loukides, M. E. Nergiz, Y. Saygin, and B. Malin, “Anonymization of longitudinal electronic medical records,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 3, pp. 413–423, 2012.

[287] C. Che, C. Xiao, J. Liang, B. Jin, J. Zho, and F. Wang, “An RNN architecture with dynamic temporal matching for personalized predictions of parkinson’s disease,” in *SDM*, 2017, pp. 198–206.

[288] International Classification of Diseases - Tenth Revision, https://www.cdc.gov/nchs/icd/icd10cm.htm, Last accessed 2021-06-01.

[289] P. Bühlmann, M. Kalisch, and M. H. Maathuis, “Variable selection in high-dimensional linear models: partially faithful distributions and the pc-simple algorithm,” *Biometrika*, vol. 97, no. 2, pp. 261–278, 2010.

[290] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl *et al.*, “Constrained k-means clustering with background knowledge,” in *Icml*, vol. 1, 2001, pp. 577–584.

[291] J. Schrodt, A. Dudchenko, P. Knaup-Gregori, and M. Ganzinger, “Graph-representation of patient data: a systematic literature review,” *Journal of medical systems*, vol. 44, no. 4, pp. 1–7, 2020.

[292] V. I. Levenshtein *et al.*, “Binary codes capable of correcting deletions, insertions, and reversals,” in *Soviet physics doklady*, vol. 10, no. 8, 1966, pp. 707–710.

[293] W. E. Winkler, “String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage.” 1990.

[294] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970.

[295] V. F. Panoulas, G. S. Metsios, A. Pace, H. John, G. Treharne, M. Banks, and G. D. Kitas, “Hypertension in rheumatoid arthritis,” *Rheumatology*, vol. 47, no. 9, pp. 1286–1298, 2008.

[296] G. Mathon, C. Gagné, D. Brun, P.-J. Lupien, and S. Moorjani, “Articular manifestations of familial hypercholesterolaemia.” *Annals of the Rheumatic Diseases*, vol. 44, no. 9, pp. 599–602, 1985.

[297] M. Epstein and J. R. Sowers, “Diabetes mellitus and hypertension.” *Hypertension*, vol. 19, no. 5, pp. 403–418, 1992.

[298] B. V. Howard, “Lipoprotein metabolism in diabetes mellitus.” *Journal of Lipid Research*, vol. 28, no. 6, pp. 613–628, 1987.

[299] E. B. Schroeder, R. Hanratty, B. L. Beaty, E. A. Bayliss, E. P. Havranek, and J. F. Steiner, “Simultaneous control of diabetes mellitus, hypertension, and hyperlipidemia in 2 health systems,” *Circulation: Cardiovascular Quality and Outcomes*, vol. 5, no. 5, pp. 645–653, 2012.

[300] T. Kushiro, H. Itakura, Y. Abo, H. Gotou, S. Terao, and D. L. Keefe, “Long-term safety, tolerability, and antihypertensive efficacy of aliskiren, an oral direct renin inhibitor, in japanese patients with hypertension,” *Hypertension Research*, vol. 32, no. 3, pp. 169–175, 2009.

[301] R. D. Santos, E. A. Stein, G. K. Hovingh, D. J. Blom, H. Soran, G. F. Watts, J. A. G. López, S. Bray, C. E. Kurtz, A. W. Hamer *et al.*, “Long-term evolocumab in patients with familial hypercholesterolemia,” *Journal of the American College of Cardiology*, vol. 75, no. 6, pp. 565–574, 2020.

[302] A.-S. Rigaud and B. Forette, “Hypertension in older adults,” *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, vol. 56, no. 4, pp. M217–M225, 2001.

[303] S. Gudlaugsdottir, W. M. Verschuren, J. Dees, T. Stijnen, and J. P. Wilson, “Hypertension is frequently present in patients with reflux esophagitis or barrett’s esophagus but not in those with non-ulcer dyspepsia,” *European Journal of Internal Medicine*, vol. 13, no. 6, pp. 369–375, 2002.

[304] M. Bellani, J. P. Hatch, M. A. Nicoletti, A. E. Ertola, G. Zunta-Soares, A. C. Swann, P. Brambilla, and J. C. Soares, "Does anxiety increase impulsivity in patients with bipolar disorder or major depressive disorder?" *Journal of Psychiatric Research*, vol. 46, no. 5, pp. 616–621, 2012.

[305] E.-L. Wu, I.-C. Chien, C.-H. Lin, Y.-J. Chou, and P. Chou, "Increased risk of hypertension in patients with major depressive disorder: a population-based study," *Journal of Psychosomatic Research*, vol. 73, no. 3, pp. 169–174, 2012.

[306] C. R. Roxbury, M. Qiu, J. Shargorodsky, T. D. Woodard, R. Sindwani, and S. Y. Lin, "Association between rhinitis and depression in united states adults," *The Journal of Allergy and Clinical Immunology: In Practice*, vol. 7, no. 6, pp. 2013–2020, 2019.

[307] G. Gadzhimirzaev, Z. Mikhrailova, I. Akhmedov, and G. Muradova, "The role of disturbances of lipid metabolism in pathogenesis of allergic rhinitis," *Vestnik otorinolaringologii*, no. 5, pp. 15–18, 2011.

[308] T. O'Brien, T. T. Nguyen, and B. R. Zimmerman, "Hyperlipidemia and diabetes mellitus," in *Mayo Clinic Proceedings*, vol. 73, no. 10, 1998, pp. 969–976.

[309] J. W. Liew, M. M. Ward, J. D. Reveille, M. Weisman, M. A. Brown, M. Lee, M. Rahbar, S. R. Heckbert, and L. S. Gensler, "Nonsteroidal antiinflammatory drug use and association with incident hypertension in ankylosing spondylitis," *Arthritis Care & Research*, vol. 72, no. 11, pp. 1645–1652, 2020.

[310] D. Robertson, D. Kumbhare, P. Nolet, J. Srbely, and G. Newton, "Associations between low back pain and depression and somatization in a canadian emerging adult population," *The Journal of the Canadian Chiropractic Association*, vol. 61, no. 2, p. 96, 2017.

[311] V. M. Russo, R. T. Dhawan, I. Baudracco, N. Dharmarajah, A. I. Lazzarino, and A. T. Casey, "Hybrid bone spect/ct imaging in evaluation of chronic low back pain: correlation with facet joint arthropathy," *World Neurosurgery*, vol. 107, pp. 732–738, 2017.

[312] D. Colombo, A. Hauser, M. Kalisch, M. Maechler, and M. M. Kalisch, "Package 'pcalg'," 2014.

[313] G. Howard, L. E. Wagenknecht, G. L. Burke, A. Diez-Roux, G. W. Evans, P. McGovern, F. J. Nieto, G. S. Tell, A. investigators, A. Investigators *et al.*, "Cigarette smoking and progression of atherosclerosis: The atherosclerosis risk in communities (aric) study," *JAMA*, vol. 279, no. 2, pp. 119–124, 1998.

[314] G. Bondjers, M. Glukhova, G. Hansson, Y. Postnov, M. Reidy, and S. Schwartz, "Hypertension and atherosclerosis. cause and effect, or two effects with one unknown cause?" *Circulation*, vol. 84, no. 6 Suppl, pp. VI2–16, 1991.

[315] M. A. Lazar, "How obesity causes diabetes: not a tall tale," *Science*, vol. 307, no. 5708, pp. 373–375, 2005.

[316] I. H. De Boer, S. Bangalore, A. Benetos, A. M. Davis, E. D. Michos, P. Muntner, P. Rossing, S. Zoungas, and G. Bakris, "Diabetes and hypertension: a position statement by the american diabetes association," *Diabetes care*, vol. 40, no. 9, pp. 1273–1284, 2017.

[317] C. Walker, “Migraine and its relationship to hypertension,” *British Medical Journal*, vol. 2, no. 5164, p. 1430, 1959.

[318] J. L. Wetherell, M. Gatz, and N. L. Pedersen, “A longitudinal analysis of anxiety and depressive symptoms.” *Psychology and Aging*, vol. 16, no. 2, p. 187, 2001.

[319] A. F. Rubio-Guerra, L. Rodriguez-Lopez, G. Vargas-Ayala, S. Huerta-Ramirez, D. C. Serna, and J. J. Lozano-Nuevo, “Depression increases the risk for uncontrolled hypertension,” *Experimental & Clinical Cardiology*, vol. 18, no. 1, p. 10, 2013.

[320] K. C. Ferdinand, “Substance abuse and hypertension.” *The Journal of Clinical Hypertension*, vol. 2, no. 1, pp. 37–40, 2000.

Appendix A

Appendix of Chapter 3

A.1 The number of patterns vs. $minSup$ for VERMONT and INFORMS

Table A.1 The number of patterns vs. minSup for VERMONT. The results for the default value are in bold.

# of Patterns	minSup (%)					
	0.7	**1**	1.3	1.6	1.9	2.2
MASPC	151	**146**	113	88	65	48
MASPC+	280	**252**	193	146	116	93
MSPC+	292	**278**	209	154	121	95
MSPC	163	**172**	129	96	70	50

Table A.2 The number of patterns vs. minSup for INFORMS. The results for the default value are in bold.

# of Patterns	minSup (%)							
	0.1	0.2	0.3	0.4	0.5	0.6	**0.7**	1
MASPC	245	151	105	134	154	102	**71**	49
MASPC+	445	301	292	303	311	207	**172**	153
MSPC+	467	330	304	322	334	221	**184**	162
MSPC	267	180	117	153	177	116	**83**	58

A.2 The number of unclustered record vs. $minSup$ for VERMONT and INFORMS

Table A.3 The number of unclustered records vs. minSup for VERMONT. The results for the default value are in bold.

# of unclustered records	**minSup (%)**					
	0.7	**1**	1.3	1.6	1.9	2.2
All methods	24209	**24586**	24935	25050	27234	30339

Table A.4 The number of unclustered records vs. minSup for INFORMS. The results for the default value are in bold.

# of unclustered records	**minSup (%)**							
	0.1	0.2	0.3	0.4	0.5	0.6	**0.7**	1
All methods	5423	9864	12346	11045	10098	12567	**13321**	16490

A.3 The number of patterns vs. $minAc$ for VERMONT and INFORMS

Table A.5 The number of patterns vs. minAc for VERMONT. The results for the default value are in bold.

# of Patterns	**minAc**				
	0.08	0.09	**0.1**	0.11	0.12
MASPC	170	156	**146**	131	117
MASPC+	276	262	**252**	237	223
MSPC+	278	278	**278**	278	278
MSPC	172	172	**172**	172	172

Table A.6 The number of patterns vs. minAc for INFORMS. The results for the default value are in bold.

# of Patterns	**minAc**								
	0.01	0.03	0.05	0.07	0.09	**0.11**	0.13	0.15	0.17
MASPC	81	79	77	74	72	**71**	62	42	23
MASPC+	182	180	178	175	173	**172**	163	143	124
MSPC+	184	184	184	184	184	**184**	184	184	184
MSPC	83	83	83	83	83	**83**	83	83	83

A.4 The number of unclustered record vs. $minAc$ for VERMONT and INFORMS

Table A.7 The number of unclustered records vs. minAc for VERMONT. The results for the default value are in bold.

# of unclustered records	**minAc**				
	0.08	0.09	**0.1**	0.11	0.12
All methods	23099	23624	**24586**	25273	25880

Table A.8 The number of unclustered records vs. minAc for INFORMS. The results for the default value are in bold.

# of unclustered records	**minAc**								
	0.01	0.03	0.05	0.07	0.09	**0.11**	0.13	0.15	0.17
All methods	12832	12943	13041	13144	13244	**13321**	14548	16362	19434

A.5 The number of patterns vs. $minOv$ for VERMONT and INFORMS

Table A.9 The number of patterns vs. $minOv$ for VERMONT. The results for the default value are in bold.

# of Patterns	$minOv$					
	10	30	**40**	55	70	90
MASPC	155	154	**146**	128	106	85
MASPC+	261	260	**252**	234	212	191
MSPC+	332	301	**278**	244	215	195
MSPC	226	195	**172**	138	109	89

Table A.10 The number of patterns vs. $minOv$ for INFORMS. The results for the default value are in bold.

# of Patterns	$minOv$						
	10	30	50	70	90	120	150
MASPC	**71**	60	48	31	28	28	27
MASPC+	**172**	161	149	132	129	129	128
MSPC+	**184**	164	153	146	143	143	142
MSPC	**83**	63	52	45	42	42	41

A.6 The number of unclustered record vs. $minOv$ for VERMONT and INFORMS

Table A.11 The number of unclustered records vs. $minOv$ for VERMONT. The results for the default value are in bold.

# of unclustered records	$minOv$					
	10	30	**40**	55	70	90
All methods	24002	24239	**24586**	26233	26836	27936

Table A.12 The number of unclustered records vs. $minOv$ for INFORMS. The results for the default value are in bold.

# of unclustered records	$minOv$						
	10	30	50	70	90	120	150
All methods	**13321**	13448	16994	17320	18921	18921	19024

A.7 Runtime of MAS and PC vs. $minSup$, $minAc$, and $minOv$

Table A.13 Runtime (sec) vs. (a) $minSup$, (b) $minAc$, and (c) $minOv$, for VERMONT.

	minSup (%)					
	0.7	1	1.3	1.6	1.9	2.2
MAS	90	126	90	59	37	24
PC	203	147	90	93	87	58

(a)

	minAc				
	0.08	0.09	0.1	0.11	0.12
MAS	188	136	121	95	73
PC	170	178	152	109	105

(b)

	$minOv$				
	10	30	50	70	90
MAS	148	147	109	73	52
PC	133	126	103	119	106

(c)

A.8 Runtime of MAS and PC vs. dataset size

Table A.14 Runtime (sec) vs. dataset size for VERMONT.

	% records			
	25	50	75	100
MAS	13	37	56	104
PC	18	43	97	123

A.9 Runtime of MAS and PC vs. domain size

Table A.15 Runtime (sec) vs. domain size for VERMONT.

	% diag codes			
	25	50	75	100
MAS	4	53	67	92
PC	8	57	101	114

A.10 Full names of ICD codes that appeared in Table 3.10a

Table A.16 Full names of ICD codes.

ICD Code	Full name
038.9	Unspecified septicemia
041.49	Other and unspecified Escherichia coli
174.9	Malignant neoplasm of breast (female), unspecified
182.0	Malignant neoplasm of corpus uteri, except isthmus
185	Malignant neoplasm of prostate
188.9	Malignant neoplasm of bladder, part unspecified
244.9	Unspecified acquired hypothyroidism
250.00	Diabetes mellitus without mention of complication, type II or unspecified type, not stated as uncontrolled
250.60	Diabetes with neurological manifestations, type II or unspecified type, not stated as uncontrolled
272.4	Other and unspecified hyperlipidemia
275.2	Disorders of magnesium metabolism
276.8	Hypopotassemia
278.00	Obesity, unspecified
278.01	Morbid obesity
300.00	Anxiety state, unspecified
305.1	Nondependent tobacco use disorder
311	Depressive disorder, not elsewhere classified
327.23	Obstructive sleep apnea (adult)(pediatric)
338.29	Other chronic pain
357.2	Polyneuropathy in diabetes
401.9	Unspecified essential hypertension
403.90	Hypertensive chronic kidney disease with chronic kidney disease stage I through stage IV
412	Old myocardial infarction
414.01	Coronary atherosclerosis of native coronary artery
428.0	Congestive heart failure, unspecified
491.21	Obstructive chronic bronchitis with (acute) exacerbation
496	Chronic airway obstruction, not elsewhere classified
530.81	Esophageal reflux
584.9	Acute kidney failure, unspecified
599.0	Urinary tract infection, site not specified
645.11	Post term pregnancy, delivered, with or without mention of antepartum condition
659.71	Abnormality in fetal heart rate or rhythm, delivered, with or without mention of antepartum condition
664.11	Second-degree perineal laceration, delivered, with or without mention of antepartum condition
995.91	Sepsis
E78.5	Hyperlipidemia, unspecified
I10	Essential (primary) hypertension
I12.9	Hypertensive chronic kidney disease with stage 1 through stage 4 chronic kidney disease
I25.10	Atherosclerotic heart disease of native coronary artery without angina pectoris
K21.9	Gastro-esophageal reflux disease without esophagitis
N18.3	Chronic kidney disease, stage 3 (moderate)
N52.9	Male erectile dysfunction, unspecified
V05.3	Long ICD9 Description: Need for prophylactic vaccination and inoculation against viral hepatitis
V27.0	Mother with single liveborn
V30.00	Single liveborn, born in hospital, delivered without mention of cesarean section
V46.2	Dependence on supplemental oxygen
V49.86	Do not resuscitate status
V58.66	Long-term (current) use of aspirin
V58.67	Long-term (current) use of insulin
V58.69	Long-term (current) use of other medications
V66.7	Encounter for palliative care
Z79.82	Long term (current) use of aspirin
Z87.891	Personal history of nicotine dependence

Appendix B

Appendix of Chapter 4

B.1 Categories for demographics

Table B.1 (a) Age and (b) Ethnicity group categories.

MIMIC		INFORMS	
Age Group	Definition	Age Group	Definition
1	Birth to 1 month (Newborn)	1	Birth to 1 month (Newborn)
2	1 month to $<$ 24 months (Infant)	2	2 to $<$ 6 years (Preschool)
3	2 to $<$ 6 years (Preschool)	3	6 to $<$ 13 years (Child)
4	6 to $<$ 13 years (Child)	4	13 to $<$ 19 years (Adolescent)
5	13 to $<$ 19 years (Adolescent)	5	19 to $<$ 45 years (Adult)
6	19 to $<$ 45 years (Adult)	6	45 to $<$ 65 years (Middle aged)
7	45 to $<$ 65 years (Middle aged)	7	65 to $<$ 80 years (Aged)
8	65 to $<$ 80 years (Aged)	8	80 years (Aged, 80 and over)
9	80 years (Aged, 80 and over)		

(a)

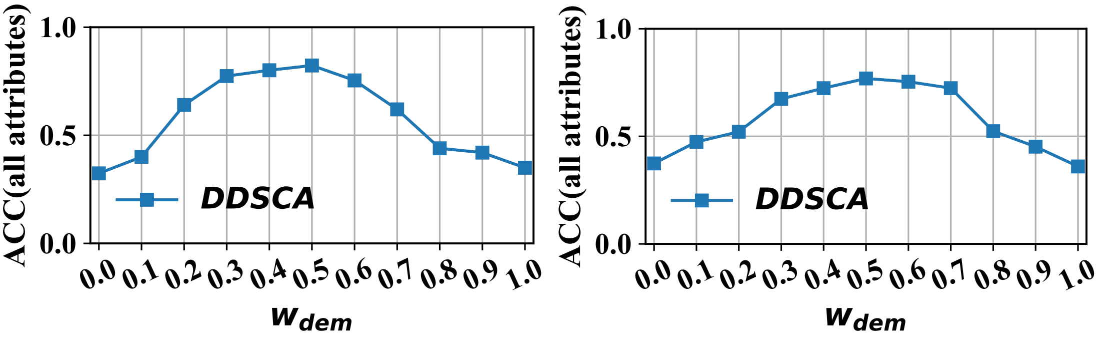
MIMIC		INFORMS	
Ethn. Group	Definition	Ethn. Group	Definition
1	AMERICAN	1	WHITE
2	ASIAN	2	BLACK
3	BLACK	3	AMERICAN INDIAN/ALASKA NATIVE
4	CARIBBEAN	4	ASIAN
5	HISPANIC	5	NATIVE HAWAIIAN / PACIFIC ISLANDER
6	MIDDLE EASTERN	6	MULTI RACE
7	MULTI RACE		
8	PACIFIC		
9	PORTUGUESE		
10	WHITE		
11	SOUTH AMERICAN		

(b)

Age in both MIMIC and INFORMS was discretized using a well-defined taxonomy [208] (see Table B.1a). We have also experimented with two different discretizations of *Age* based on the same taxonomy to demonstrate that our method still outperforms the competitors (see Section B.9 in Appendix B).

B.2 Impact of weights

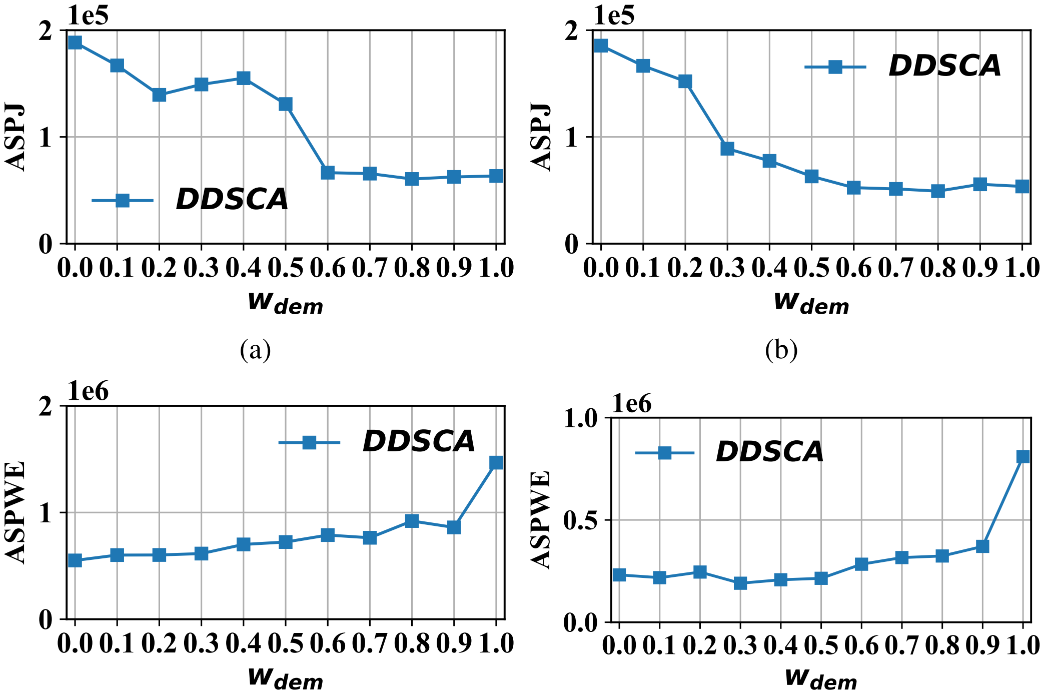
Fig. B.1 shows that the most compact clusters with respect to ACC are produced when equal weights are used.



(a)

(b)

Fig. B.1 ACC vs. w_{dem} for (a) MIMIC and (b) INFORMS.



(a)

(b)

(c)

(d)

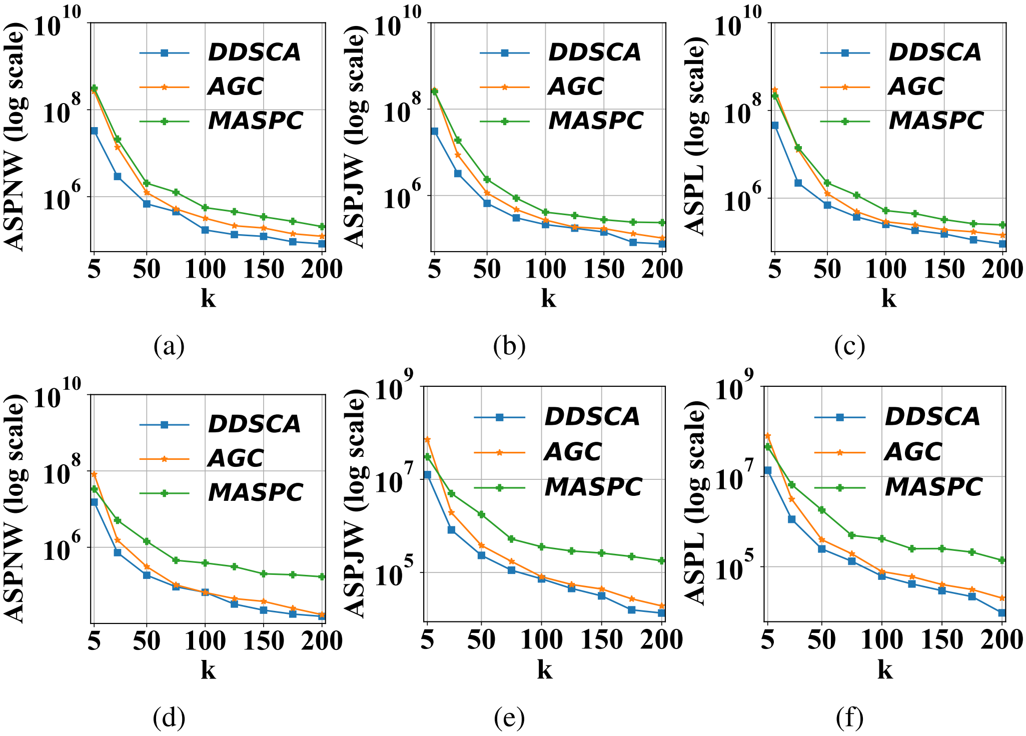
Fig. B.2 ASPJ vs. w_{dem} for (a) MIMIC and (b) INFORMS. ASPWE vs. w_{dem} for (c) MIMIC and (d) INFORMS.

Fig. B.2 shows the impact of w_{dem} on ASPJ, ASPWE. As can be seen, when w_{dem} [1] increases, ASPJ (ASPWE) will decrease (respectively, increase).

[1] $w_{dem} = 1 - w_{diag}$.

B.3 Clustering effectiveness on more edit-distance based metrics

Fig. B.3 shows clustering effectiveness on Levenshtein (d_L) [292], Jaro-Winkler (d_{JW}) [2] [293], and Needleman-Wunsch (d_{NW}) [3] [294] distances. Similarly, we use $\frac{1}{k}\sum_{c\in C}\sum_{r_i,r_j\in C} d(r_i^{seq}, r_j^{seq})$ to report clustering effectiveness, where $d()$ can be d_L, d_{JW} and d_{NW}. Finally, we report ASPNW (Average Sum of Pairwise Needleman-Wunsch distance), ASPJW (Average Sum of Pairwise Jaro-Winkler distance) and ASPL (Average Sum of Pairwise Levenshtein distance) in Fig. B.3, where $w_{dem} = w_{diag} = 0.5$. Lower values are preferred.



(a) (b) (c)

(d) (e) (f)

Fig. B.3 (a) ASPNW, (b) ASPJW, and (c) ASPL vs. k for MIMIC. (d) ASPNW, (e) ASPJW, and (f) ASPL vs. k for INFORMS.

[2] The parameter prefix scale p is set to 0.1.

[3] The gap cost is set to -1, match score is set to 1, and mismatch score is set to 0.

B.4 Compactness results for MIMIC

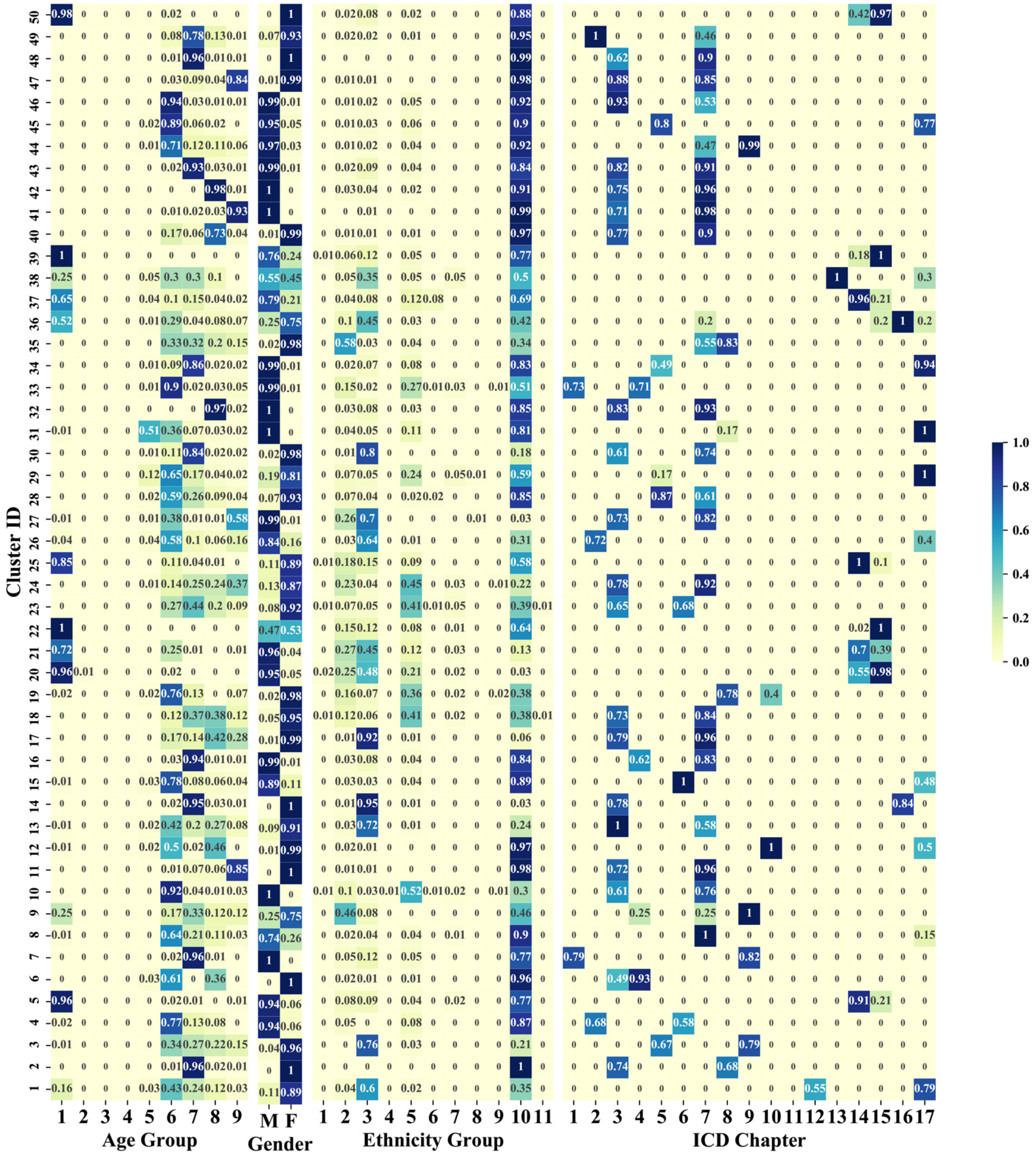


Fig. B.4 Heat map of MIMIC for all clusters when k is 50.

B.5 Compactness results for INFORMS

Fig. B.5 shows the distribution of values in each demographic, and the distribution of ICD Chapters [212] in the clusters, for ten clusters generated when our algorithm was applied to INFORMS with $k = 50$. The reported clusters were selected randomly among the best 25 clusters with respect to ACC, computed over all attributes. Note that most records in each cluster have the same value in a demographic attribute, and their top-2 frequent diagnosis codes belong into two ICD Chapters. This implies that clusters have records with similar demographics and diagnosis codes (i.e., they are compact), which allows meaningful analytic and mining tasks, based on both demographics and diagnosis codes. For example, in Cluster 1 most patients (98%) are male, the age of 82% of patients is between 65 years and 80 years, and 91% of patients are white. Meanwhile, 79% of patients in Cluster 1 have at least one ICD-9 code in ICD Chapter 13 (Diseases of the Musculoskeletal System and Connective Tissue) and 49% of patients have at least one ICD-9 code in ICD Chapter 7 (Diseases of the Circulatory System).

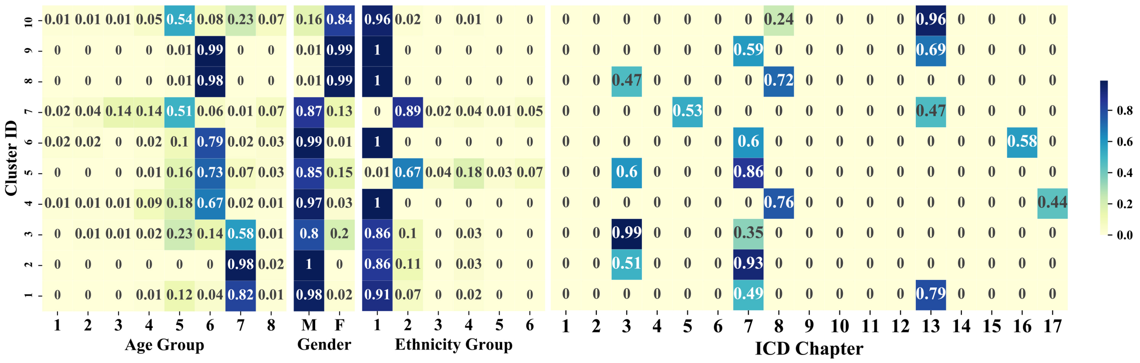


Fig. B.5 Heat map of INFORMS for Age, Gender, Ethnicity, and ICD Chapter. The values in the cells of a heat map are ratios of records in a cluster (e.g., 0.82 of records in cluster with ID 1 have their Age values in Age Group 7 corresponding to Aged people). The sum of ratios for a cluster over the ICD Chapters is not 1 since a record can contain diagnosis codes that belong in multiple ICD Chapters.

Fig. B.6 shows the distribution of values in each demographic and the distribution of ICD Chapters in all clusters constructed by applying DDSCA with k = 50 on INFORMS.

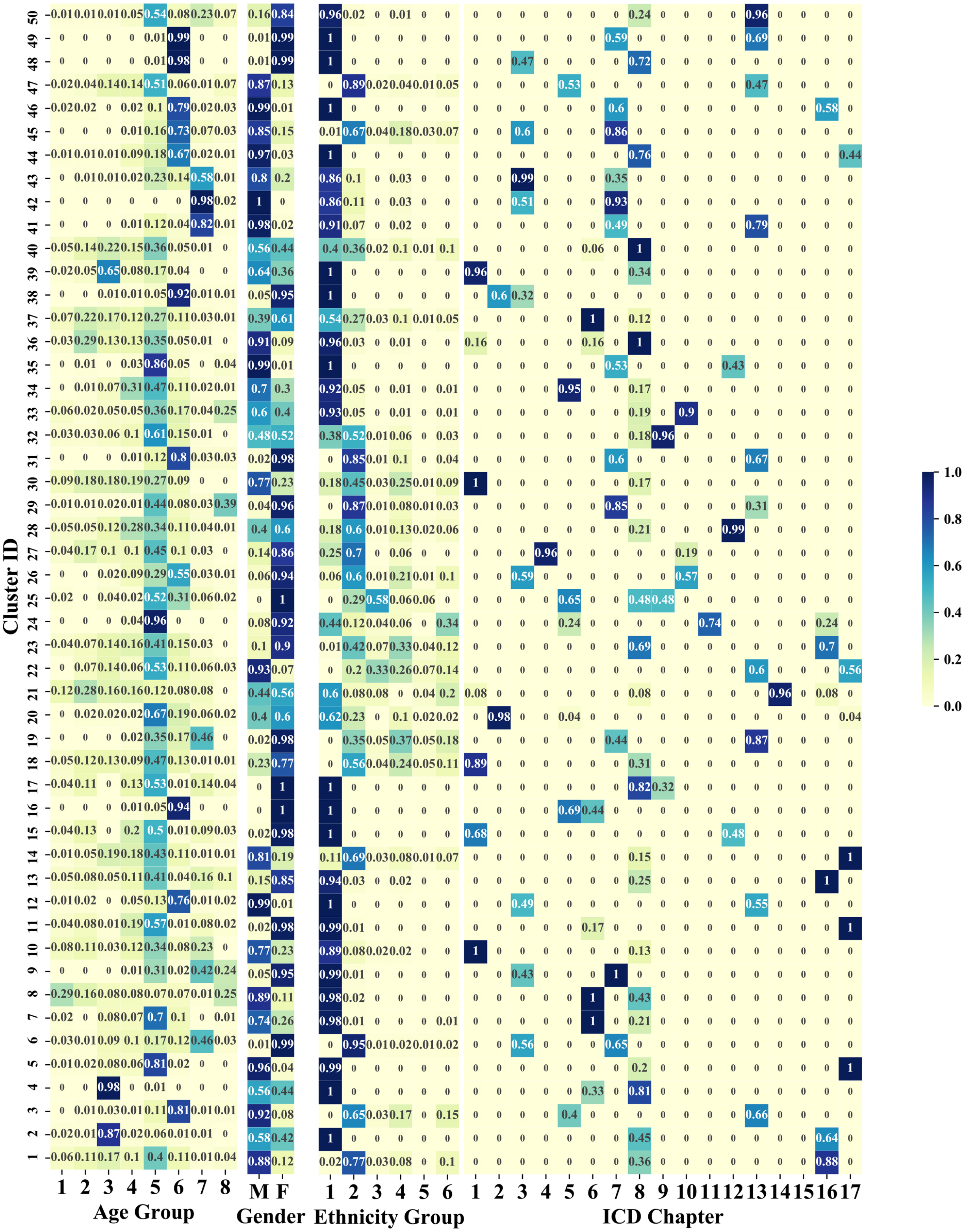


Fig. B.6 Heat map of INFORMS for all clusters when k is 50.

B.6 Separability results for INFORMS

Table B.2 The top-1 (i.e., most) frequent value in each demographic, top-2 frequent ICD Chapters, and top-3 frequent sequential patterns, for ten clusters in INFORMS for DDSCA with k=50.

ID	Gender	Age	Ethnicity	Chapter	Top 3 Sequential Patterns		
1	M	Aged	White	$\{7,13\}$	$(401,716)$	$(272,401)$	$(272,716)$
2	M	Aged	White	$\{3,7\}$	$(272,401)$	$(401,250)$	$(401,780)$
3	M	Aged	White	$\{3,7\}$	$(250,401)$	$(250,272)$	$(250,272,401)$
4	M	Middle aged	White	$\{8,17\}$	$(401,460)$	$(272,460)$	$(460,724)$
5	M	Middle aged	Black	$\{3,7\}$	$(250,401)$	$(272,401)$	$(250,272,401)$
6	M	Middle aged	White	$\{7,16\}$	$(272,401)$	$(401,780)$	$(530,401)$
7	M	Adult	Black	$\{5,13\}$	$(300,311)$	$(311,401)$	$(311,477)$
8	F	Middle aged	White	$\{3,8\}$	$(272,401)$	$(272,477)$	$(250,272)$
9	F	Middle aged	White	$\{7,13\}$	$(272,401)$	$(401,716)$	$(250,401)$
10	F	Adult	White	$\{8,13\}$	$(401,724)$	$(311,724)$	$(716,724)$

B.7 The medical relevance of top-3 frequent sequential patterns in the clusters of INFORMS

Next, we discuss the top-3 frequent sequential patterns in Table III of Supplementary Material, showing that they capture relationships between diagnosis codes that are documented in the medical literature.

Cluster 1: The diagnoses in each of the top-3 frequent sequential patterns in this cluster (see Table III of Supplementary Material) frequently co-occur. Specifically, hypertension (401) that appears in 2 patterns frequently co-occurs with arthropathy (716) [295], or with hypercholesterolemia (a type of disorders of lipoid metabolism denoted by 272) [220]. For hypercholesterolaemia (272) and arthropathy (716), [296] states that arthropathy has been associated mainly with the homozygous form of familial hypercholesterolaemia.

Cluster 2: Hypertension (401) frequently co-occurs with disorders of lipoid metabolism (272) [220], or with diabetes mellitus (250) [297].

Cluster 3: Diabetes mellitus (250) frequently co-occurs with hypertension (401) [297], or with disorders of lipoid metabolism (272) [298]. Also, diabetes (250), hyperlipidemia (272), and hypertension (401) frequently co-occur [299].

Cluster 4: Acute nasopharyngitis (460) frequently co-occurs with hypertension (401) [300], or with disorders of lipoid metabolism (272) [301].

Cluster 5: Two of the three top-3 frequent sequential patterns in this cluster are the same as those in Cluster 3. One difference between this cluster and Cluster 3 is that hypertension in the latter occurs two times more frequently. This is expected as the incidence of hypertension in the elderly population is high [302].

Cluster 6: Hypertension (401) frequently co-occurs with hypercholesterolemia (272) [220], or with diseases of esophagus (530) [303].

Cluster 7: Depressive disorder (311) frequently co-occurs with diabetes mellitus (300) [304], or with hypertension (401) [305], or with allergic rhinitis (477) [306].

Cluster 8: Disorders of lipoid metabolism (272) frequently co-occur with hypertension (401) [220], or with allergic rhinitis (477) [307], or with diabetes mellitus (250) [308].

Cluster 9: Hypertension (401) frequently co-occurs with (272) [220], or with arthropathies (716) [295], or with diabetes mellitus (250) [297]. One difference between this cluster and Cluster 6 is in Gender. In both of these clusters, many patients have hypertension (401), but the number of patients with hypertension in Cluster 6 is 1.5 times larger than that in Cluster 9. This is expected because hypertension is more prevalent among men (the gender of 99% of patients in Cluster 6) than women (the gender of 99% of patients in Cluster 9) [231].

Cluster 10: Disorders of back (724) frequently co-occur with hypertension (401) [309], or with depressive disorder (311) [310], or with arthropathies (716) [311].

B.8 Causal Inference

In this section, PC-select [289] is used to get causal relationships. PC-select is a well-known algorithm that finds the local causal relationships around a given response variable from a learned Bayesian network. In our context, the response variable is a diagnosis code u and the local causal relationships show what other diagnosis codes may cause u. The parameter α (significance level of individual partial correlation tests) is set to 0.05, following [289]. We used the implementation of PC-select from [312].

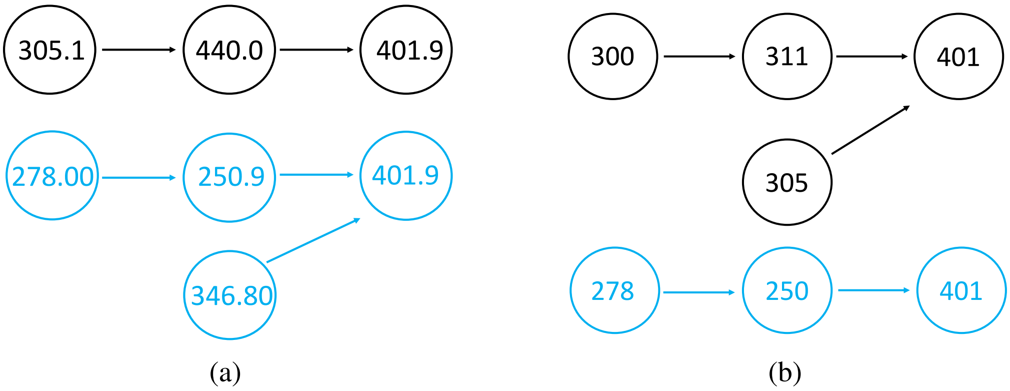


Fig. B.7 (a) Causal relationships from Cluster 5 (black) and Cluster 2 (blue) of MIMIC. (b) Causal relationships from Cluster 5 (black) and Cluster 2 (blue) of INFORMS.

We first examined Clusters 5 and 2 in MIMIC (see Table 4.6 in Chapter 4) and used hypertension (401.9) as a response variable. We have found causal relationships which are documented in the medical literature. In Cluster 5, 89% of patients have age in $[19, 45)$ (Age Group 6 in Table B.1b in Appendix B), and 95% of patients are male. In Cluster 2, 98% of patients have age in $[65, 80)$ (Age Group 8 in Table B.1b in Appendix B), and 100% of patients are male. In Fig. B.7a, the obtained causal relationships are shown in the form of a graph; each node u in the graph corresponds to an ICD-9 code and each edge (u, v) corresponds to a relationship "u causes v". As can be seen, tobacco use disorder (305.1) causes atherosclerosis of aorta (440.0) [313], which causes hypertension (401.9) [314]. This is reasonable, since 87% of patients in Cluster 5 are smokers and smoking can cause atherosclerosis [313], which can cause hypertension [314]. In Cluster 2, 99% of patients do not smoke, but they still have hypertension (440.0). This is because most patients in this cluster (72%) have obesity (278.00), which causes diabetes (250.9) [315] and diabetes (250.9) causes hypertension (440.0) [316]. Also, 21% of patients in this cluster have hypertension (440.0) due to migraine (346.80) [317].

Next, we examined Clusters 5 and 2 in INFORMS (see Table B.2 in Appendix B) and again used hypertension (401.9) as a response variable. We have found causal relationships which are documented in the medical literature. In Cluster 5, 73% of patients have age in $[45, 65)$ and 85% of patients are male. In Cluster 2, 98% of patients have age in $[65, 80)$ and 100% of patients are male. As can be seen in Fig. B.7b, anxiety (300) (74% of patients are associated with it) causes depressive disorder (311) [318], which causes hypertension (401) [319]. Also, nondependent abuse of drugs (305) (32% of patients are associated with it) causes hypertension (401) [320]. In Cluster 2, 89% of patients are associated with obesity (278), which causes diabetes (250) [315] and diabetes (250) causes hypertension (401) [316].

B.9 Impact of different Age discretizations

Table B.3 shows two different ways to discretize *Age*. As can be seen in the results in Table B.4 and B.5, our DDSCA algorithm always outpeforms both competitors. Note also that the first discretization in Table B.3a has more values (i.e., it is more fine-grained) compared to the discretization in Table B.3b (i.e., 5 Age groups vs. 3 Age groups). As can be seen, ASPJ values are higher in Table B.4a compared to those in Table B.5a (and also in Table B.4b compared to those in Table B.5b). This is because it is more difficult to construct clusters with similar age groups in each cluster, when the Age discretization has more values. On the other hand, the values in APSWE, ASPLCS, ASPL, ASPJW, and ASPNW are lower in Table B.4a compared to those in Table B.5a (and also in Table B.4b compared to those in Table B.5b). This is because it is easier to construct clusters that have similar diagnosis code sequences, when the Age discretization has more values. This shows that an overall good clustering cannot be achieved by using clustering algorithms that consider only demographics or only sequences of diagnosis codes.

Table B.3 (a) Age discretization with 5 groups (b) Age discretization with 3 groups

MIMIC		INFORMS	
Age Group	Definition	Age Group	Definition
1	$<$ 2 years	1	$<$ 6 years
2	2 to $<$ 13 years	2	6 to $<$ 19 years
3	13 to $<$ 45 years	3	19 to $<$ 65 years
4	45 to $<$ 80 years	4	65 to $<$ 80 years
5	$>=$ 80 years	5	$>=$ 80 years

(a)

MIMIC		INFORMS	
Age Group	Definition	Age Group	Definition
1	$<$ 13 years	1	$<$ 19 years
2	13 to $<$ 80 years	2	19 to $<$ 80 years
3	$>=$ 80 years	3	$>=$ 80 years

(b)

Table B.4 Effectiveness based on the Age discretization from Table B.3 (a) for $k = 50$ on: (a) MIMIC and (b) INFORMS.

Method	ACC (all attributes)	ASPJ	ASPWE	ASPLCS	ASPL	ASPJW	ASPNW
DDSCA	0.734	$100,789$	$793,357$	$18,145,214$	$754,745$	$726,954$	$741,745$
AGC	0.372	$170,202$	$1,846,489$	$27,285,241$	$1,885,274$	$1,842,142$	$1,912,154$
MASPC	0.322	$643,481$	$2,456,803$	$37,145,127$	$2,954,741$	$2,955,741$	$2,711,223$

(a)

Method	ACC (all attributes)	ASPJ	ASPWE	ASPLCS	ASPL	ASPJW	ASPNW
DDSCA	0.724	$53,034$	$284,792$	$2,541,241$	$297,745$	$291,854$	$221,225$
AGC	0.441	$907,107$	$399,040$	$4,721,514$	$437,854$	$441,742$	$370,541$
MASPC	0.372	$1,098,075$	$1,923,306$	$7,325,685$	$2,552,741$	$2,365,433$	$1,711,541$

(b)

Table B.5 Effectiveness based on the Age discretization from Table B.3 (b) for $k = 50$ on: (a) MIMIC and (b) INFORMS.

Method	ACC (all attributes)	ASPJ	ASPWE	ASPLCS	ASPL	ASPJW	ASPNW
DDSCA	0.706	$88,584$	$841,274$	$21,745,625$	$801,741$	$796,184$	$810,124$
AGC	0.324	$121,745$	$2,274,852$	$32,174,652$	$2,574,154$	$2,354,854$	$2,311,533$
MASPC	0.287	$501,285$	$2,818,562$	$45,685,741$	$3,454,745$	$3,312,473$	$3,022,545$

(a)

Method	ACC (all attributes)	ASPJ	ASPWE	ASPLCS	ASPL	ASPJW	ASPNW
DDSCA	0.698	$44,412$	$327,456$	$2,914,741$	$347,125$	$343,854$	$274,711$
AGC	0.404	$840,107$	$457,452$	$5,134,112$	$494,154$	$503,147$	$421,741$
MASPC	0.341	$897,452$	$2,246,285$	$8,195,412$	$2,915,112$	$2,954,002$	$2,223,782$

(b)

Appendix C

Appendix of Chapter 5

C.1 Phylogenetic Trees With The Ground Truth

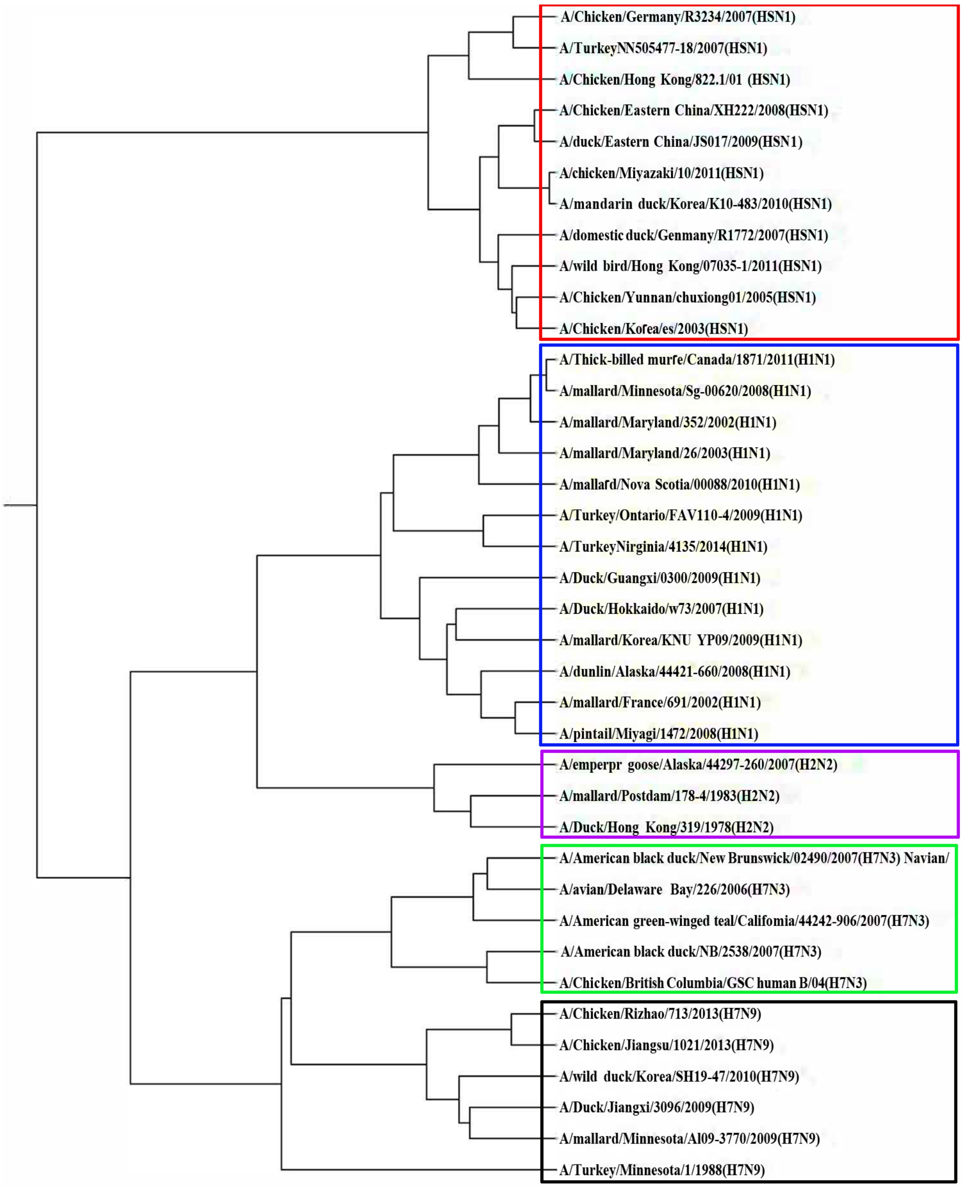

A/Chicken/Germany/R3234/2007(HSN1)
A/TurkeyNN505477-18/2007(HSN1)
A/Chicken/Hong Kong/822.1/01 (HSN1)
A/Chicken/Eastern China/XH222/2008(HSN1)
A/duck/Eastern China/JS017/2009(HSN1)
A/chicken/Miyazaki/10/2011(HSN1)
A/mandarin duck/Korea/K10-483/2010(HSN1)
A/domestic duck/Genmany/R1772/2007(HSN1)
A/wild bird/Hong Kong/07035-1/2011(HSN1)
A/Chicken/Yunnan/chuxiong01/2005(HSN1)
A/Chicken/Korea/es/2003(HSN1)
A/Thick-billed murre/Canada/1871/2011(H1N1)
A/mallard/Minnesota/Sg-00620/2008(H1N1)
A/mallard/Maryland/352/2002(H1N1)
A/mallard/Maryland/26/2003(H1N1)
A/mallard/Nova Scotia/00088/2010(H1N1)
A/Turkey/Ontario/FAV110-4/2009(H1N1)
A/TurkeyNirginia/4135/2014(H1N1)
A/Duck/Guangxi/0300/2009(H1N1)
A/Duck/Hokkaido/w73/2007(H1N1)
A/mallard/Korea/KNU YP09/2009(H1N1)
A/dunlin/Alaska/44421-660/2008(H1N1)
A/mallard/France/691/2002(H1N1)
A/pintail/Miyagi/1472/2008(H1N1)
A/emperpr goose/Alaska/44297-260/2007(H2N2)
A/mallard/Postdam/178-4/1983(H2N2)
A/Duck/Hong Kong/319/1978(H2N2)
A/American black duck/New Brunswick/02490/2007(H7N3) Navian/
A/avian/Delaware Bay/226/2006(H7N3)
A/American green-winged teal/Califomia/44242-906/2007(H7N3)
A/American black duck/NB/2538/2007(H7N3)
A/Chicken/British Columbia/GSC human B/04(H7N3)
A/Chicken/Rizhao/713/2013(H7N9)
A/Chicken/Jiangsu/1021/2013(H7N9)
A/wild duck/Korea/SH19-47/2010(H7N9)
A/Duck/Jiangxi/3096/2009(H7N9)
A/mallard/Minnesota/Al09-3770/2009(H7N9)
A/Turkey/Minnesota/1/1988(H7N9)

Fig. C.1 Phylogenetic tree of INFL with the ground truth. The sequences are shown as leaves. Each cluster in the ground truth clustering is represented with a differently colored rectangle.

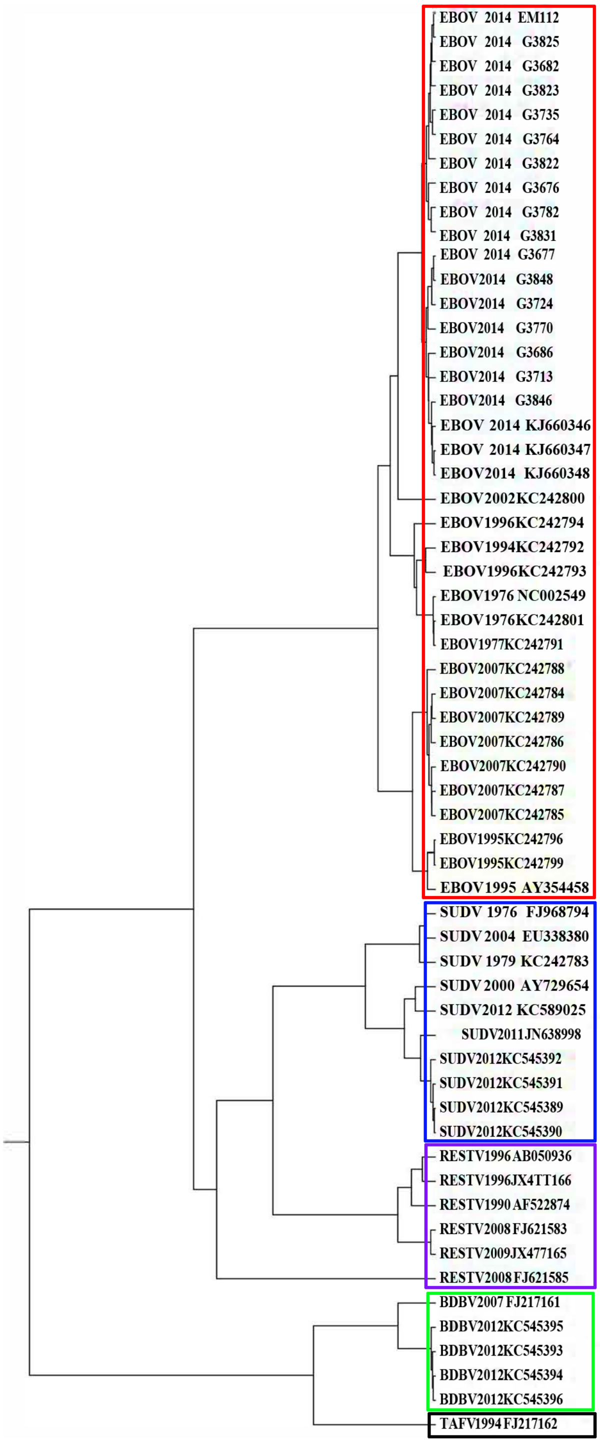


Fig. C.2 Phylogenetic tree of EBOL with the ground truth. The sequences are shown as leaves. Each cluster in the ground truth clustering is represented with a differently colored rectangle.

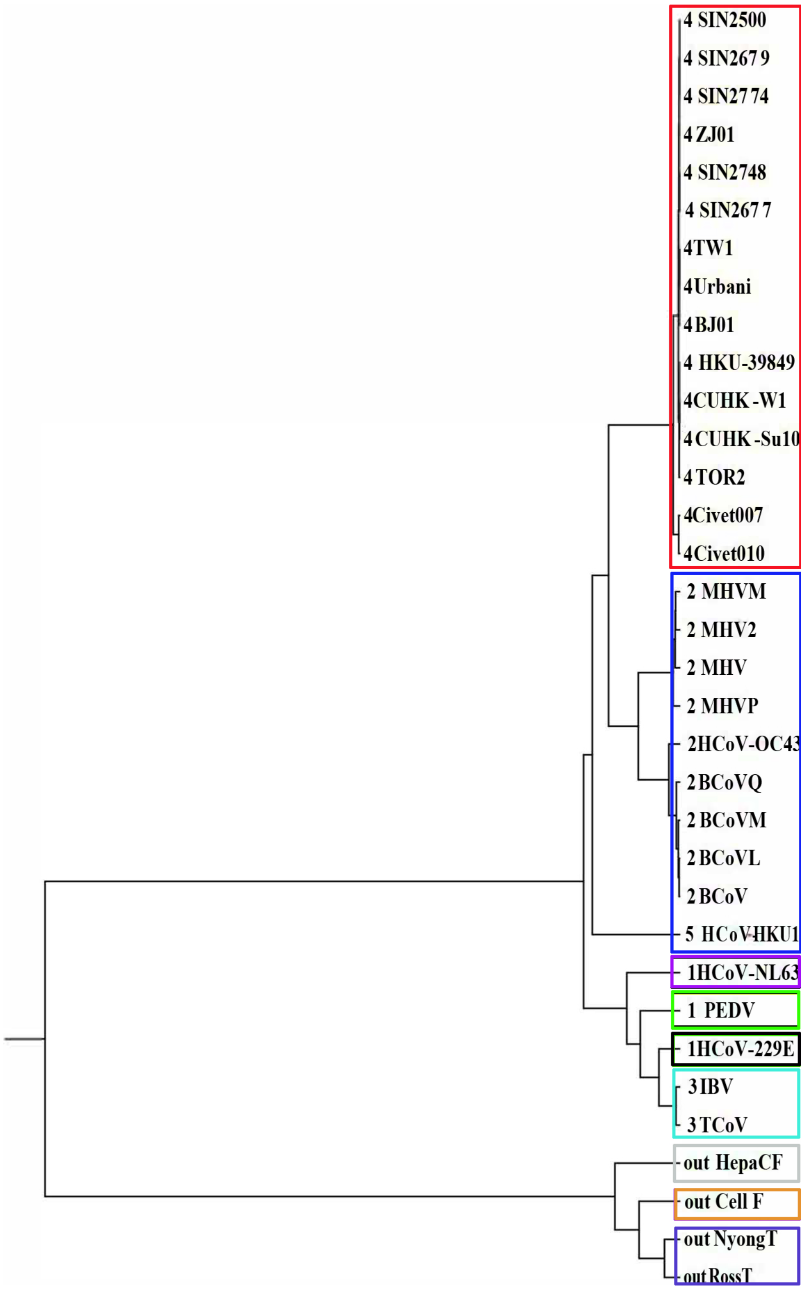


Fig. C.3 Phylogenetic tree of COR with the ground truth. The sequences are shown as leaves. Each cluster in the ground truth clustering is represented with a differently colored rectangle.