Universidad del País Vasco
Euskal Herriko Unibertsitatea

PhD Thesis

Unsupervised learning approaches for disease progression modeling

---

Onintze Zaballa

Supervised by

Aritz Pérez and Jose A. Lozano

Donostia - San Sebastián, 2023

PhD Thesis

Unsupervised learning approaches for disease progression modeling

Onintze Zaballa

Supervised by

Aritz Pérez and Jose A. Lozano

Donostia - San Sebastián, 2023

Funding in direct support of this work has been provided by the Basque Government through the BERC 2022-2025 program and BMTF project, and by the Ministry of Science, Innovation and Universities: BCAM Severo Ochoa accreditation CEX2021-001142-S / MICIN / AEI/ 10.13039/501100011033. Jose A. Lozano is also supported by the Basque Government under grant IT1504-22 and Ministry of Science and Innovation under grant PID2022-137442NB-I00. Onintze Zaballa also holds a predoctoral grant (EJ-GV 2019) from the Basque Government.

Acknowledgement

First and foremost, I would like to thank my advisors, Aritz Pérez and Jose Antonio Lozano, for their guidance and support throughout my Ph.D. I am truly grateful for their time, profound insights and invaluable advices, all of which have significantly influenced my growth as a researcher. It has been a privilege to have both of you as mentors, continually pushing me to achieve my best.

I am also profoundly grateful to Elisa Gómez and Teresa Acaiturri from Hospital Universitario de Cruces. Their interest in researching in the field of healthcare has served as our inspiration. Your insightful discussions and constructive feedback have consistently contributed to the enhancement of this dissertation.

I would like extend my sincere thanks to Mihaela van der Schaar for providing me with the outstanding opportunity to collaborate as a visiting researcher in the Department of Applied Mathematics and Theoretical Physics at the University of Cambridge.

I am eternally thankful to my colleagues at BCAM. I could not have asked for a more exceptional team to accompany me along this journey, your friendship and encouragement have been invaluable to me. Thank you for enhancing the academic research in this thesis through countless discussions, but, above all, for making this time a memorable experience.

Last but not least, I would like to express my deepest gratitude to my family and friends. You have consistently been my unwavering support, standing by my side and encouraging me to persevere until the very end. Thank you for being there through every challenge and celebrating the highs with me. You are my best gift.

Abstract

Electronic Health Records (EHRs), which store extensive patient and treatment data, provide an opportunity for machine learning models to capture disease progression patterns over time. Each medical record in these repositories is composed by a set of clinical variables, including a medical action, a diagnosis, and a timestamp. The medical action describes the trajectory of a patient in the healthcare system and the diagnosis associates each medical event with a specific disease. Therefore, a patient's treatment trajectory is characterized by a chronological sequence of medical records.

The primary objective of this dissertation is to develop methodologies that provide an understanding of patients' treatment progression through meaningful pattern recognition in EHRs. Generative models are powerful approaches for this purpose, as they enable the learning of the underlying data distribution, and offer an interpretable representation of disease dynamics from data. These models have additional benefits, including pattern identification, data augmentation, anomaly detection, and uncertainty estimation in predictions, among others.

In contrast to generative approaches, most existing deep learning models in healthcare focus on accurately predicting future events rather than comprehensively modeling disease progression. Understanding disease progression remains challenging for these methods due to various factors, including limited data availability, data quality problems like missing diagnosis data, and the need for interpretable results in healthcare settings. Generative models provide more interpretable patterns of disease dynamics, require less quantity of data and work properly even in the presence of missing data. Although previous generative models have advantages over deep learning models, they often make simplified assumptions for capturing the evolution of diseases. Further research is required to appropriately model key medical aspects such as the sequential occurrence and relationship of consecutive medical events, the irregular time intervals between records, and the coexistence of multiple diseases when diagnoses are missing.

This dissertation presents unsupervised methodologies to provide interpretable understanding of the progression of disease trajectories. To this end, we develop methods based on different sequence classification techniques. On the one hand, we propose a methodology based on partitional clustering for identifying disease treatment subtypes from EHRs with missing diagnosis information. Specifically, the methodology is based on the K-medoids approach with an adaptation of the edit distance, which enables to determine a representative for each subtype of treatments. On the other hand, we pro-

pose various probabilisitic generative models for sequences of medical events to analyze different scenarios in disease dynamics. The models include latent variables to capture treatment progression, temporal irregularity and comorbidities in medical data. We introduce efficient methods for learning these models, combining the Expectation-Maximization algorithm and dynamic programming.

The effectiveness of the methodological proposals is evaluated using a real-world dataset from Osakidetza, the public healthcare system in the Basque Country, Spain. Each patient in these EHRs is represented by a sequence of medical services over time, with only 19% of these medical events having an associated diagnosis value. We include practical applications involving breast cancer patients, demonstrating the relevance and potential impact of the models. In summary, this dissertation presents methodologies that offer valuable insights into disease dynamics while addressing the unique challenges presented in EHRs.

Contents

Chapter 1

Introduction

Disease progression, which refers to the natural evolution of medical conditions over time, is a critical area of research in healthcare [1]. In recent years, machine learning models have gained substantial significance within this context, offering personalized insights into disease trajectories and progression trends [2, 3]. These insights are particularly valuable for diseases like cardiovascular disease, cancer, and diabetes, which evolve slowly throughout a patient's lifetime.

Healthcare institutions regularly record medical data in repositories known as Electronic Health Records (EHRs) for monitoring patients' health status throughout their clinical history. These EHRs contain a vast amount of patient and treatment information, including demographics, diagnoses, medications, procedures, costs, medical resources and so on (see Figure 1.1). While their primary purpose is efficient medical management, EHRs present the opportunity to develop machine learning methods that can effectively capture disease progression patterns. Indeed, these models can help discover associations between the shared characteristics of similar patients, identify a data-driven taxonomy of the progression of treatments associated with a disease, reduce the uncertainty in a patient's expected treatment trajectory and timing, and analyze comorbidities by uncovering the relationships among them [4].

Developing models for EHRs faces a variety of data challenges, limitations, and quality issues [1, 4]. In this dissertation, we highlight and address the following ones:

- Heterogeneity: EHRs contain numerous distinct medical events associated with different diseases, and the occurrence of these events can be influenced by the individual preferences and characteristics of patients and healthcare professionals. Furthermore, patient responses to treatments can differ, even for the same disease, leading to variations in the sequence and medical events that occur during their treatment trajectories [5]. This variability in medical events and patient responses makes each patient's medical history unique.

- Incomplete information: it is likely that many observations will be missing in a healthcare dataset. Moreover, EHRs are commonly limited to a specific period of

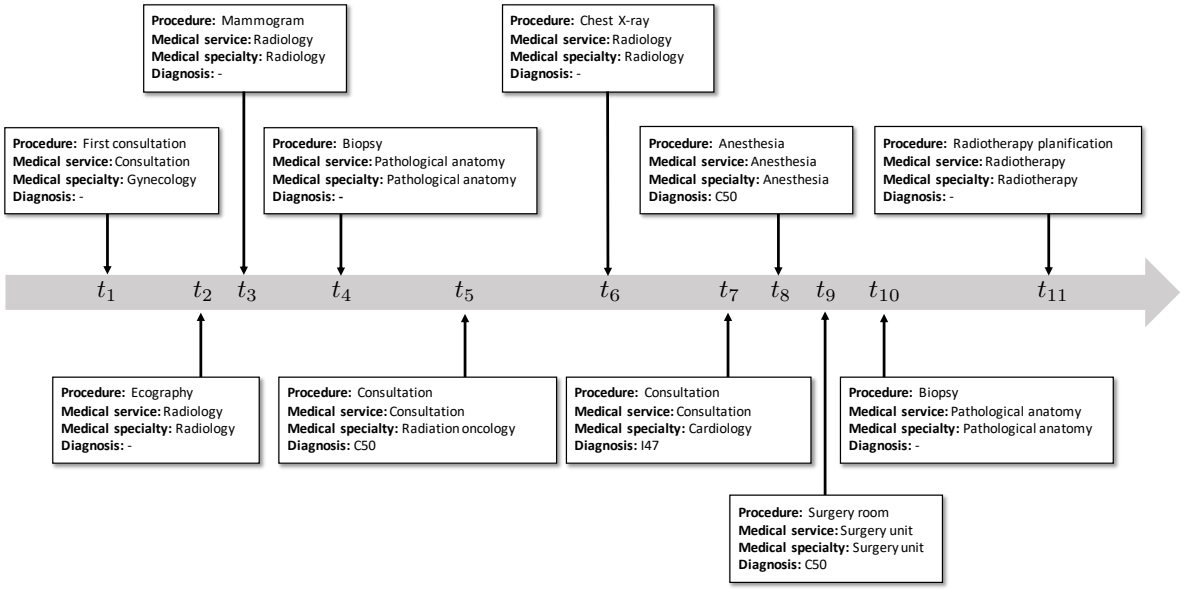


Figure 1.1: A simplified EHR structure represented as a timeline. Diagnosis code C50 corresponds to *malignant neoplasm of breast*, and I47 to *paroxysmal tachycardia*.

time, and therefore, significant events may occur outside this period, leading to incomplete clinical histories and treatment trajectories.

- Irregularity: the events in healthcare settings occur randomly and irregularly, as patients visit the hospital when clinical care is needed. Therefore, in EHRs, the time elapsed between patient's visits is irregular.
- Interpretability: it is crucial that both the machine learning models and their generated outcomes from EHRs are not only accurate but also easily understandable and interpretable. Thus, healthcare professionals will rely on these models to make informed decisions about patient care, treatment strategies, and disease management.
- Uncertainty: the absence of diagnostic values in many medical records, together with the prevalence of comorbidities (the coexistence of multiple diseases) among patients, creates uncertainty regarding the association between medical events and specific diseases. In other words, since patients may suffer from coexisting diseases, certain medical records lack clear associations with specific disease treatments due to missing diagnosis values in the EHRs (Figure 1.2).

1.1 Dataset

Our research is conducted in collaboration with the public healthcare system (Osakidetza) in the Basque Country, Spain. Specifically, with the economic-financial de-

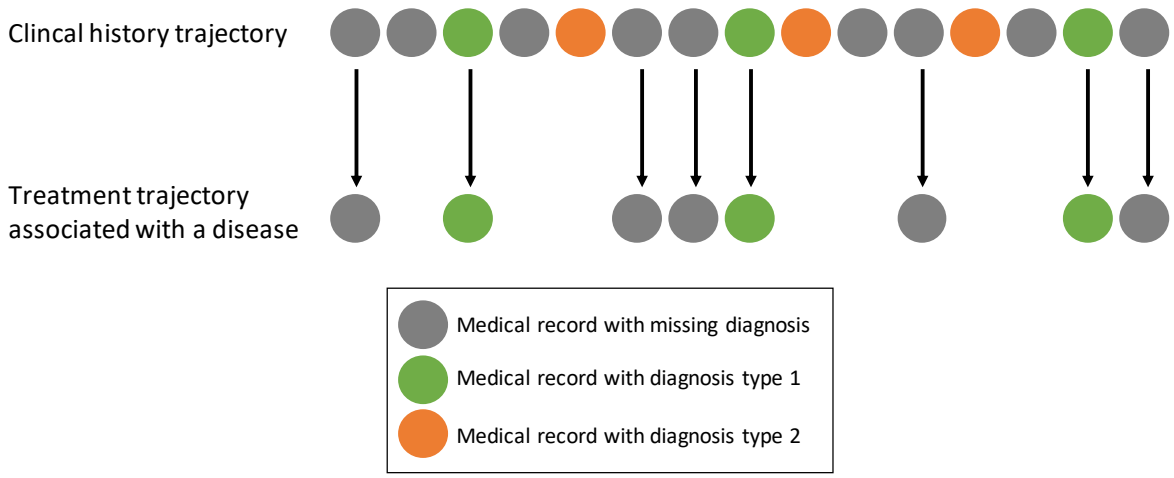


Figure 1.2: Uncertainty in medical actions associated with a diagnosis due to comorbidity and missing values.

partment of the Hospital Universitario de Cruces. They provide us with a dataset that integrates care information with economic information. These EHRs enable the tracking of all the resources used in the treatment of a disease throughout a patient's clinical history, and presents the traceability of the whole clinical care process. In contrast to several publicly available EHR datasets, such as MIMIC-III [6] or eICU [7], our dataset does not include clinical outcomes.

This dataset collects 82,712,233 records from 2016 to 2019, involving a total of 729,134 patients treated at various levels of healthcare, including one hospital, eleven outpatient clinics, and emergency care. The information captured in this dataset is related to billing data, mostly in categorical form, and includes information about the medical specialties, procedures, diagnoses represented using ICD-10 codes (International Classification of Diseases) [8], among others (Figure 1.1).

Each patient's clinical history is characterized by a chronological sequence of medical events from EHRs. In turn, each medical event is defined by a medical action, a diagnosis, and a timestamp. In this dissertation, we use medical services as medical actions (see Table 1.1), but any other variable that represents the clinical trajectory of a patient could be used. The diagnosis variable allows us to associate each medical event with a specific disease, although it frequently contains missing values. In fact, only 19% of these recorded events have associated diagnoses.

This dissertation will propose methodological approaches for this sequential data, such as the segmentation of disease treatments, the identification of treatment subtypes and their progression stages, modeling the irregular time intervals between medical events within a treatment, and tracking the evolution of comorbidities when various diseases coexist.

Abbreviated form	Full Form
ANES	Anesthesia
CONS	Consultation
DHOSP	Day Hospital
EXTC	External Consultation
FUNT	Functional Testing
HCRI	Critical Care Hospitalization
HOMEH	Home Hospitalization
HOSP	Hospitalization
INCO	Interconsultation
LABO	Laboratory
NUCM	Nuclear Medicine
NURS	Nursing Unit
OSAT	Osatek (Magnetic Resonance Service)
PATH	Pathological Anatomy
PAU	Post Anesthesia Care Unit
PHAR	Pharmacy
PHARAMB	Hospital Pharmacy Services
RADI	Radiology
REHA	Rehabilitation
RTER	Radiotherapy
SURG	Surgery Unit
SWH	Surgery without Hospitalization
UCRI	Nursing Critical Care Unit

Table 1.1: Description of the medical services of the dataset.

1.2 Existing methods for disease progression modeling

This section provides a brief overview of machine learning models developed for disease progression tasks. We discuss how these methods have addressed EHR data challenges when making outcome predictions through supervised learning and extracting meaningful patterns through unsupervised learning.

Recently, deep learning techniques have been introduced to identify sequential and temporal patterns within a patient's medical history, enabling them to predict future scenarios, including diagnosis [9–17], procedures [10, 14, 16, 18], and hospital readmissions [14,16,19,20]. The effectiveness of these models is often attributed to the capacity of the neural networks to learn nonlinear distribution and representation of data, as well as to capture long-term dependency in sequences [21]. However, the complexity of these approaches often limits their interpretability, making it challenging for healthcare professionals to gain a deep understanding of the temporal evolution of a disease [21,22].

Some works have attempted to develop interpretable models based on Recurrent Neural Networks (RNNs) in the field of healthcare [9, 11, 12]. These models use attention mechanisms to understand underlying disease dynamics and provide explanations for their discriminative predictions. However, their problem scope often differs from our research, as their ambiguity regarding the association between medical events and diagnoses is related to entire hospitalization episodes, as illustrated in Scenario 2 of Figure 1.3. In addition, they might not be suitable for handling missing data or providing a probabilistic framework for addressing outcome uncertainties due to their deterministic internal structure. In [13] the authors propose a predictive model to overcome these major uncertainty issues and produce a comprehensive estimate of future disease progression trajectories. Nevertheless, this method relies on complete medical data without missing values, which might not be realistic in practice [23].

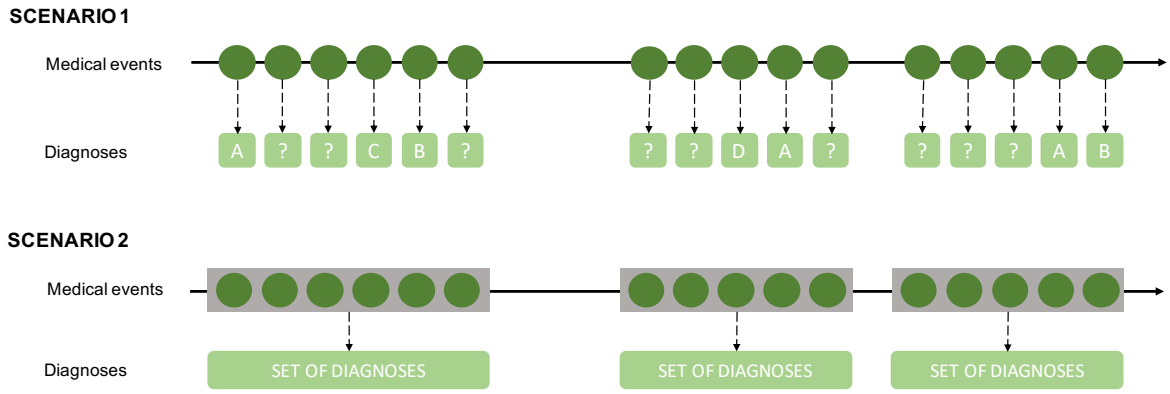


Figure 1.3: Different uncertainty problems and scenarios regarding the association of diagnosis to medical events. The circles refer to the medical events, the green squares to diagnoses (some of them are missing) and gray rectangles to hospitalization episodes. The contributions of this dissertation address the problem in scenario 1.

Another research line in disease progression modeling involves the use of probabilistic methods to capture disease dynamics in an accurate and interpretable manner. For instance, probabilistic topic models based on Latent Dirichlet Allocation (LDA) [24] have been proposed to discover disease clusters and patient subgroups [25–28]. These methods highlight the heterogeneous nature of a disease and the importance of developing models based on disease subtyping. Their main goal is to identify distinct treatment subgroups, but they do not adequately take into account the temporal progression of a treatment as a time series. They do not model the order of events in a sequence and focus only on event frequency within the sequences. Therefore, they are not well-suited for capturing disease dynamics.

Regarding the temporal dynamics of diseases, one aspect that has received limited attention is the estimation of irregular time intervals between consecutive medical events. Despite the high prediction accuracy of deep learning methods when it comes to future medical events, they rarely estimate the irregular time elapsed between medical

events. However, modeling irregular time intervals in EHRs can lead to more efficient healthcare and improved clinical resource management. Existing models, such as Doctor AI [10], which is a RNNs-based approach, analyze patients' medical history to predict both the next diagnosis and the timing of the next medical visit. Other methods incorporate irregularities into the model, although their primary goal is prediction of medical outcomes rather than time estimation. For instance, DeepCare [14], which is built upon Long Short Term Memory units, incorporates temporal decay and attention mechanisms to account for temporal irregularity and importance variation in hospital visits for diagnosis prediction. Deepr [19] is a Convolutional Neural Network based approach that detects clinical motifs while handling irregular timing in EHRs to predict readmission within a time window. Even if these models include the irregular timing in their structure to learn sequential and temporal patterns, ongoing research is needed to address irregular time-related challenges and enhance temporal representation of disease progression.

Certain methodologies focus on a broader range of diseases and consider the simultaneous occurrence of multiple conditions in a patient [10, 17, 21, 25, 29–31], which is often referred to as comorbidities. While these approaches can predict and assess potential future diseases for healthcare providers, they do not offer a complete understanding of how comorbidities interact, evolve, or dynamically influence each other. Consequently, there is a need for dynamic progression methods that account for disease interactions over time, particularly in cases of comorbidity progression, as patients with one chronic disease typically develop other conditions over time [32–34].

A potential approach to model sequences is the use of Hidden Markov Models (HMMs) [30, 35–42]. These generative models are based on latent variables that are capable of uncovering disease evolution patterns from heterogeneous types of treatments in EHRs. As a result, they assist clinicians in obtaining more informative assessments of patients' clinical health status by relating these latent variables to meaningful clinical information. In addition, they are practical models for purposes such as imputing missing values and simulating new treatment trajectories. A specific type of Markovian models, the continuous-time hidden Markov models, attempt to address the irregular timing between events in sequences, capturing the time intervals between hidden variables as a means to model disease progression [30, 39, 40]. Some models account for comorbidities but they overlook the fact that medical data may contain missing diagnostic values (Scenario 1 in Figure 1.3) [30, 34]. Additional research is necessary in this field to appropriately address the shortcomings of the existing generative models.

1.3 Unsupervised learning from sequences of events

This section provides a brief introduction to the fundamental unsupervised learning techniques that support our models. These methodologies are essential for segmenting and modeling sequences of medical events, allowing us to uncover hidden patterns and relationships in EHRs. Sequence segmentation achieves the purpose of addressing the

heterogeneity and variability present in treatment trajectories, while sequence modeling enables to extract valuable patterns of progression from sequences of medical events. Both challenges are tackled by including latent variables into our models.

1.3.1 Sequence segmentation

Given the heterogeneity in treatment trajectories, our goal is to group patients with similar treatment patterns and establish a representative for each category. To achieve this, we build our models upon different sequence segmentation methods [43]: partitional clustering based on distances and probabilistic clustering using mixtures of distributions.

Partitional clustering. Partitional clustering refers to the process of dividing a population into disjoint groups whose union forms the original set. The partitional clustering technique we use is the K-medoids algorithm, which aims to identify K clusters in a dataset and represents clusters with actual data points (medoids). To do so, the algorithm iteratively minimizes the dissimilarity of each data point to all other points within the same cluster and chooses the data point with the lowest total dissimilarity as the medoid. At each iteration of the algorithm, each data point is assigned to only one cluster, and the medoids of the clusters are redefined.

Formally, suppose we have a set $\boldsymbol{A}$, then, the goal is to partition $\boldsymbol{A}$ into K clusters. That is, create a partition $\boldsymbol{A}_k$ for $k = 1, ..., K$, where $\bigcup_{k=1}^{K} \boldsymbol{A}_k = \boldsymbol{A}$ and $\boldsymbol{A}_i \cap \boldsymbol{A}_j = \emptyset$ for any $i \neq j$. Each $\boldsymbol{A}_k$ is represented by a medoid.

The set $\boldsymbol{A}$ is commonly compossed by $\mathbb{R}^d$ vectors, that is, $\boldsymbol{A} = \{\boldsymbol{a}^1, ..., \boldsymbol{a}^N\}$ with $\boldsymbol{a}^i \in \mathbb{R}^d$. Then, the distance is often measured by the L_1 or L_2^2 norm. In our case, the set $\boldsymbol{A}$ is compossed by discrete sequences with varying length. Then, an appropriate distance measure must be employed, such as the edit distance [44]. This metric calculates the minimum number of editing operations required to transform one discrete sequence into another. These operations are usually defined in terms of insertion, deletion, and substitution of one symbol for another, often with different costs for each of these operations.

In the following section, we explain how using probabilistic approaches leads to more flexible assignments of the sequences to clusters, in a way that captures the level of uncertainty over the most appropriate assignment.

Probabilistic clustering. Probabilistic clustering is a technique that involves partitioning a dataset into groups based on probabilistic models. Such methods are often based on the assumption that the data are generated by a mixture of underlying probability distributions, for instance, mixture of Gaussians. These generative models include a latent variable that probabilistically associates data points with clusters learned from the data. As a result, each cluster can be represented in various manners, for instance, by using the mean and variance, or by the data point with the highest probability of

belonging to that cluster in the dataset, among others. Note that the probability distributions can be generalized to different kinds of data, such as numerical, categorical or sequential data.

In our case, instead of data points we use sequences of different lengths. Then, suppose we have a set of sequences $\boldsymbol{A} = \{\boldsymbol{a}^1, ..., \boldsymbol{a}^N\}$, which correspond to the treatments in EHRs. The random variable $\boldsymbol{a}$ is assumed to be distributed according to a mixture of K components. Each component (cluster) is represented by a parametric distribution, and therefore, the entire data set is modeled by a mixture of these distributions. Formally, the mixture distribution of $\boldsymbol{a}$ can be expressed as follows:

$$p(\boldsymbol{a}) = \sum_{k=1}^{K} p(c_k) p(\boldsymbol{a}|c_k),$$

where $\boldsymbol{a}$ is the observed sequence, $p(c_k)$ is the probability of belonging to the cluster k, and $p(\boldsymbol{a}|c_k)$ is the conditional probability distribution of the observation $\boldsymbol{a}$ given that it belongs to the cluster k. It must be satisfied that $\sum_{k=1}^{K} p(c_k) = 1$.

The objective is to estimate the parameters of the underlying probabilistic model that best fit the observed data. This is achieved using the Expectation-Maximization (EM) algorithm [45], which maximizes the likelihood from the given dataset considering that the data is incomplete. EM algorithm iteratively follows these two steps until convergence: the E-step determines the expected probability of assignment of sequences to clusters with the use of current model parameters; and the M-step determines the optimum model parameters of each mixture by using the assignment probabilities as weights.

1.3.2 Sequence modeling

Given the significance of the order of events in a sequence, our research focuses on Hidden Markov Models (HMMs), which capture dependencies among sequence elements. An HMM extends the concept of Markov model by introducing hidden states in their structure. In an HMM, it is assumed that there is a set of hidden states generating observations. An HMM has two primary components: the transition model, which describes the evolution of states over time; and the observation model, which describes the manifestation of the state in the observed space. In Figure 1.4, the latent states are denoted as $\boldsymbol{s}$, with s_t corresponding to patient's state at time t, and observations are denoted as $\boldsymbol{a}$ where a_t represents the observed event at time t.

Formally, consider a sequence of m observations $\boldsymbol{a} = (a_1, \ldots, a_m)$ and its underlying sequence of latent states $\boldsymbol{s} = (s_1, \ldots, s_m)$ where s_t belongs to the set of latent states S for all t. Then, the joint probability distribution of an HMM is given by

$$p(\boldsymbol{a}, \boldsymbol{s}|\boldsymbol{\theta}) = p(s_1) p(a_1|s_1) \prod_{t=2}^{m} p(s_t|s_{t-1}) p(a_t|s_t).$$

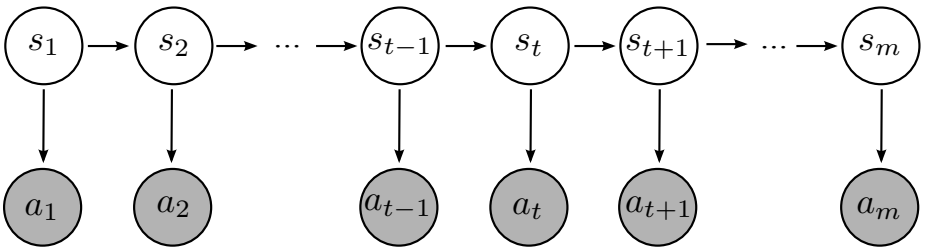


Figure 1.4: Hidden Markov Model structure.

EM algorithm for HMM. EM algorithm is used to maximize the likelihood function of the HMM, which efficiently learns the parameters of models with latent variables [46]. The likelihood function is obtained from the joint distribution by marginalizing over the latent variables, that is, summing over all possible latent state sequences, $\mathcal{S}$:

$$p(\boldsymbol{a}|\boldsymbol{\theta}) = \sum_{\boldsymbol{s}\in\mathcal{S}} p(\boldsymbol{a}, \boldsymbol{s}|\boldsymbol{\theta}).$$

The maximization of the complete-data log-likelihood is not directly feasible due to the unavailability of the complete dataset. That is, we do not know the corresponding values of the latent variables for each observation. Our knowledge of the values of the latent variables is given only by the posterior distribution $p(\boldsymbol{s}|\boldsymbol{a}, \boldsymbol{\theta})$. Therefore, since it is not possible to use the complete-data log-likelihood, the EM algorithm considers its expected value under the posterior distribution of the latent variables.

The EM algorithm starts with some initial selection for the model parameters, $\boldsymbol{\theta}^{old}$, and iterates until convergence as follows:

- **Expectation (E-step):** In the E-step, the aim is to find the expected value of the complete-data log-likelihood with respect to the latent states $\boldsymbol{s}$ given the observed sequence $\boldsymbol{a}$ and the current parameter estimates. This involves calculating the posterior distribution of the latent states given the observations, $p(\boldsymbol{s}|\boldsymbol{a}, \boldsymbol{\theta}^{old})$. Then we use this posterior distribution to find the expectation of the complete-data log-likelihood function, as a function of the parameters, $\boldsymbol{\theta}$. This expectation, denoted $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$, is given by

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \mathbb{E}_{\boldsymbol{s}|\boldsymbol{a},\boldsymbol{\theta}^{old}}[\log p(\boldsymbol{a}, \boldsymbol{s}|\boldsymbol{\theta})] = \sum_{\boldsymbol{s}\in\mathcal{S}} p(\boldsymbol{s}|\boldsymbol{a}, \boldsymbol{\theta}^{old}) \log p(\boldsymbol{a}, \boldsymbol{s}|\boldsymbol{\theta}).$$

- **Maximization (M-step):** In the M-step, the goal is to maximize $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$ with respect to the parameters $\boldsymbol{\theta}$ in which $p(\boldsymbol{s}|\boldsymbol{a}, \boldsymbol{\theta}^{old})$ are treated as constants. That is,

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}).$$

This optimization problem with respect to $\boldsymbol{\theta}$ can be solved in closed form using the method of Lagrange multipliers.

Each iteration of the EM is guaranteed to increase the log-likelihood and the algorithm is guaranteed to converge to a local maximum of the likelihood function [46].

We can see an HMM as a special kind of mixture model in which the different components of the mixture are dependent on each other through transitions. Thus, the clustering of sequence data with a mixture of HMMs can be considered a two-level mixture model.

1.4 Contributions

This section describes the contributions and provides an outline of the dissertation. The main objective is to develop methodologies for unsupervised learning from sequences of medical events that effectively uncover the underlying patterns of disease dynamics. Our contributions can be summarized as follows: firstly, we develop a methodology to extract the treatment of a specific disease from the whole medical history of a patient considering missing diagnosis data in EHRs; then, we propose both partitional and probabilistic clustering methods for identifying subtypes of treatments; finally, within the probabilistic approaches, we present generative models to capture the progression of disease treatments, incorporate the time variable, and manage patients with comorbidities.

1.4.1 Methodology for identifying representative treatment patterns from EHRs.

Chapter 2 introduces a general methodology with a twofold objective. Firstly, to extract complete treatments associated with a specific disease from EHRs, taking into account that these repositories contain patients' entire medical histories with co-occurring diseases (Figure 1.2). Secondly, to identify the treatments that are representative of the dataset. The methodology is specifically designed to address missing diagnosis data and the variability observed in treatment trajectories within EHRs. This method serves as a preliminary framework for the unsupervised classification of disease treatments.

To address the first objective, given that the association between events and diagnoses is often not explicitly documented (Figure 1.2), we introduce a relevance measure. This measure effectively identifies and represents treatment trajectories associated with a specific diagnosis. Subsequently, we establish several selection criteria to ensure the selection of complete end-to-end treatments within the dataset.

To achieve the second goal, we propose to use the K-medoids algorithm, which groups similar treatments and represents these groups using actual sequences of medical actions (treatments) from EHRs. To enable the comparison of discrete sequences of varying lengths, we propose the normalized edit distance metric.

The effectiveness of this methodology is demonstrated using breast cancer patients as a case study, obtaining five groups of treatment patterns. These results have been compared with clinical practice guidelines and validated by healthcare professionals, which highlight the robustness and practical relevance of the proposed methodology. Furthermore, it can be easily applied to other types of diseases.

1.4.2 Modeling disease progression patterns

Chapter 3 introduces a probabilistic generative model to discover treatment subtypes of a disease and their progression stages. To do so, the model classifies sequences of medical actions into different subtypes based on their evolution over time. This is a probabilistic extension of the partitional approach outlined in Chapter 2 that also incorporates sequence modeling based on the progression of the medical events.

To achieve this, the model incorporates a hierarchical structure of latent variables associated with each sequence of actions. These latent variables have a twofold purpose: classifying sequences and segmenting them into distinct stages based on their progression patterns. The model parameters are learned using the EM algorithm. We propose an adaptation of the conventional forward-backward algorithm [47] for the learning process to reduce the complexity to be polynomial.

The evaluation of our generative model consists of two parts: initially, we use synthetic data to demonstrate that the learning procedure recovers the generative model underlying the data. Subsequently, we assess the model's potential to provide treatment classification and staging information using real-world data of breast cancer patients. To validate its practical utility, we compare the results with clinical guidelines and validate them with medical professionals. This model can be seen also as a tool for classification, simulation, data augmentation, and imputation of missing data in healthcare applications.

1.4.3 Modeling time-dependent disease progression patterns

Chapter 4 proposes an extension of the probabilistic generative model presented in Chapter 3. This extension incorporates temporal information to capture the irregular time intervals between consecutive medical actions within the sequence of medical events.

For this purpose, the structure of the model considers latent variables that classify treatments into subtypes based on the patient sequence of medical events and the time intervals, segment treatments into subsequences of patterns of disease progression, and model the irregular time between every pair of medical events. It offers flexibility in modeling the time distribution, allowing the choice of the most appropriate distribution based on the available data. To ensure efficient learning of the parameters, we use the EM algorithm with an adaptation of the forward-backward algorithm to the characteristics of our generative model.

Through synthetic and real data experiments, we demonstrate the effectiveness of our approach in learning the model underlying the data, estimating the irregular timing between medical actions and classifying treatments into different subtypes. To show the significant impact of including temporal data in our approach, we conduct both qualitative and quantitative comparisons of the model against the one proposed in Chapter 3, which does not incorporate irregular temporal information. By considering this information, the model provides healthcare professionals with a more informative view of how a disease may progress over time.

1.4.4 Modeling treatments of coexisting diseases with frequently missing diagnosis

Chapter 5 presents a probabilistic generative approach for modeling the progression of comorbidites, which is specifically designed to handle EHRs with substantial missing data in the diagnosis variable (Scenario 1 in Figure 1.3). This model is a generalization of the method proposed in Chapter 3 to handle multiple co-existing diseases. The main objectives of this model include disentangling the medical history of patients into treatments associated with comorbidities, learning the model associated with each identified disease treatment, and grouping subtypes of patients with similar coevolution of comorbidities.

To this end, the model considers a latent structure for the sequences: a latent class to define the evolution of the comorbidities; and a latent sequence of diagosis to relate each observed medical event of a clinical history to a disease. Additionally, the model describes the different joint evolution of coexisting diseases based on the active comorbidities of the patient at each moment of their clinical history. The learning process is performed through the EM algorithm, which efficiently addresses the exponential complexity of the latent variable configurations with a proposed dynamic programming-based approach.

The evaluation of the method is carried out both on synthetic and real-world data. The experiments using synthetic data demonstrate that the learning process effectively learns the generative model that underlies the data. Furthermore, the experiments conducted on real medical data, for patients with breast cancer and cardiovascular diseases, show accurate results in the segmentation of sequences into different treatments, subtyping of patients and diagnosis imputation.

Chapter 2

A methodology for identifying representative treatment patterns from EHRs

2.1 Introduction

The increasing availability of EHRs offers the opportunity to improve healthcare by learning from past patient information. One important step towards this objective is to learn data-driven representations of diseases. In this sense, disease subtyping has gained significant attention in healthcare research, focusing on the identification and classification of subgroups of patients who share similar characteristics within a specific disease [48]. The discovery of disease subtypes can benefit healthcare management tasks, such as reducing uncertainty in an individual's expected treatment, estimating the expected costs of care or evaluating adherence to medical guidelines [1].

Process mining techniques have been applied to identify representative clinical trajectories and treatment patterns within EHRs [26, 49–52]. These methods involve extracting knowledge from sequences of events, with individual medical activities considered as such events [5]. Due to the heterogeneous behavior of medical data, these models often result in complex outcomes that are hard to interpret [51]. Furthermore, the representative treatments obtained from these models are artificial trajectories, which do not describe appropriately the actual treatments recorded in EHRs.

Machine learning methods provide a potential solution to address these challenges by grouping patients into more homogeneous subgroups. In the healthcare domain, clustering techniques have been widely used for this purpose. For instance, some works focus on hierarchical clustering using similarity metrics like the longest common subsequence distance [53], DBScan with Levenshtein distance [54] or fuzzy c-means [55]. Others use K-means to cluster patients and then represent the trajectories of each cluster with directed graphs where the edges indicate the flow of the events in the trajectories [52]. These methodologies do not assume missing data and often fail in

adequately representing the clusters with actual treatment trajectories from EHRs.

In addition to this clustering methods, probabilistic generative models have also been developed to capture the heterogeneous nature of diseases. Some of these models include LDA-based approaches [25–28], which do not account for the order of medical events. Additionally, HMM-based techniques [30, 40, 42] have been proposed for identifying disease clusters and patient subgroups based on the evolution of continuous medical variables or the occurrence of comorbidities. Rather than extracting representative treatment trajectories from EHRs, these models are developed to identify shared characteristics within each cluster, such as similar values for the continuous variables or common comorbidities within each cluster.

The objective of this chapter is to introduce a general methodology for identifying the representative treatment trajectories for a disease from EHRs. In this methodology, we specifically tackle the challenges in EHRs of both missing data and heterogeneous treatments trajectories among patients. The main contribution of this chapter is the proposal of a general framework that allows us to: (*i*) identify the medical actions in EHRs associated with a particular disease; (*ii*) extract the complete end-to-end treatment of patients related to the target disease from EHRs; and (*iii*) discover the typical treatment trajectories followed by patients with a specific diagnosis.

To illustrate this in a real scenario, we apply the methodology to the real-world dataset described in Section 1.1. We then compare the outcomes with clinical practice guidelines and discuss the results with healthcare professionals to assess their alignment with the treatments administered in practice.

The rest of this chapter is organized as follows. Section 2.2 briefly describes the problem formulation and notation. Section 2.3 presents the methodology for identifying the representative treatments in EHRs. Section 2.4 discusses the outcomes, and finally, Section 2.5 draws the conclusions of the chapter.

2.2 Problem formulation

A patient's treatment trajectory associated with a disease, denoted by $\boldsymbol{a}$, is a sequence of medical actions collected during repeated hospital visits. In our context, these actions indicate the medical service that a patient has visited, including primary care, surgery unit, hospitalization, and more (Table 1.1). Let A be the set of all the possible medical actions, then, we define a (disease) treatment trajectory as

$$\boldsymbol{a} = (a_1, ..., a_m),$$

where $a_i \in A$ represents the i-th medical action of a patient. In addition, each medical action a is related to a more detailed medical specialty x (see Figure 1.1). Therefore, each sequence of medical actions is related to a more detailed sequence of medical specialties, defined as

$$\boldsymbol{x} = (x_1, , ..., x_m),$$

where x_i belongs to the set of all the medical specialties, X, such as gynecology, hematology, radiation oncology, and so on.

Note that the treatment trajectory, $\boldsymbol{a}$, represents a subsequence of a patient's entire healthcare trajectory, excluding actions that are not related to the target disease and preserving those directly relevant to the disease. Extracting these specific subsequences from EHRs is not straightforward due to the frequent presence of incomplete diagnosis codes in these repositories. The lack of diagnostic values in many medical records, combined with the presence of comorbidities, introduces uncertainty regarding the association between medical events and specific diseases (as illustrated in Figure 1.2). Therefore, the first problem of this chapter is to establish a method for directly extracting disease treatment trajectories from EHRs related to a target disease, especially in cases where diagnoses are missing.

Furthermore, the heterogeneity among patients often leads to different progression patterns and a variety of treatments for the same disease. The second problem focuses on the unsupervised classification of the disease treatment trajectories and the subsequent data-driven representation of the trajectories in EHRs.

2.3 Methodology

This section presents the methodology (Figure 2.1) for identifying and representing typical treatments of a disease that patients follow within the healthcare system. For this purpose, we develop a methodology to extract complete end-to-end treatments associated with a diagnosis of interest from the entire medical history of the patients. Subsequently, we apply a clustering method with the aim of discovering the different subtypes of disease treatment trajectories. Note that clustering is an unsupervised technique and it is performed without prior knowledge about the disease. Therefore, although the validation of the clusters could be performed in terms of compactness or coherence, we determined that the most appropriate evaluation of our approach was by checking the results with medical guidelines and physicians. We proceed in this way to validate the applicability of the whole methodology.

2.3.1 Creation of healthcare trajectories from EHRs

The first step is to convert the original EHRs into healthcare trajectories, which represent the clinical history of a patient (Figure 1.1). This structure involves transforming EHRs into chronological sequences of medical actions, in such a way that each patient has an associated healthcare trajectory. These sequences are discrete and of different lengths, leading to significant variations in patients' medical histories. For instance, one patient might have only two hospital visits for routine check-ups, while another patient with a chronic disease may frequently visit the hospital for therapy, medical tests, and other procedures.

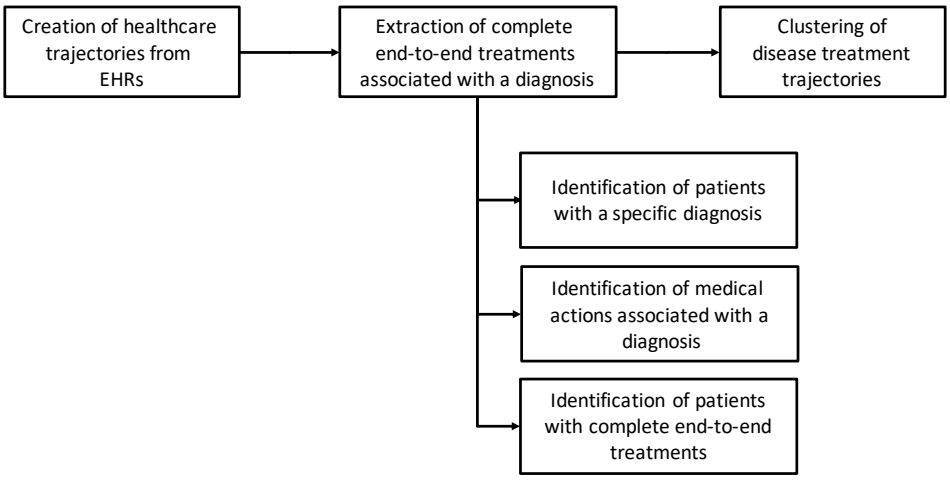


Figure 2.1: Methodology of the study.

2.3.2 Extraction of complete end-to-end treatments associated with a diagnosis

This section explains the extraction of complete end-to-end treatment trajectories associated with a target diagnosis from the entire healthcare trajectories of patients. We achieve this through the following steps: (i) identifying the medical actions directly associated with a disease, thereby defining the disease treatment trajectory for the diagnosis of interest by excluding actions related to other coexisting diseases; (ii) selecting patients with a high probability of having the entire diagnosis-related treatment recorded in the dataset through the application of specific selection criteria. Ultimately, this process enables us to obtain eligible disease treatment trajectories from EHRs for the subsequent unsupervised classification and identification of typical treatment patterns for the disease.

Identification of medical actions associated with a diagnosis. We define a relevance measure to identify the typical medical actions related to the diagnosis of interest. These medical actions are then used to create the disease treatment trajectories $\boldsymbol{a}$.

To formulate the relevance measure, we analyze the sequence of medical specialties visited during patients' healthcare trajectory. The dataset is divided into two groups of sequences: sequences with at least one diagnosis of interest recorded, and sequences without it. Within each group, we calculate the average frequency of the medical specialties by patient, and the relevance is then determined by the ratio of the mean frequency of these individual values between the two groups. A higher relevance value indicates greater importance of the specialty for the disease. Therefore, we establish a threshold λ in such a way that if the relevance is higher than λ, the medical specialty is considered typical of the disease. That is, the medical specialty $x \in X$ is considered

typical of the disease if

$$\frac{f_{x_D}}{f_{x_R}} \geq \lambda. \tag{2.1}$$

Both f_{x_D} and f_{x_R} represent the mean frequency of visits to a medical specialty $x \in X$ among patients with the diagnosis of interest and the remaining of the patients, respectively. Finally, to obtain the disease treatment trajectory $\boldsymbol{a}$, we extract the medical actions from the entire healthcare trajectory whose medical specialties are considered typical of the disease.

Identification of patients with complete end-to-end treatments. Healthcare trajectories might contain medical actions related to similar diseases (e.g., different types of cancer), making it challenging to discern the specific diagnosis targeted in the treatment trajectory. Additionally, there might be treatments that began before or concluded after the recording period of the dataset, or even incomplete treatments with lost follow-up. We propose various general selection criteria to address these issues:

- Ensuring that the disease treatment trajectory is directly focused on the aimed diagnosis: a requirement to exclude patients with similar coexisting diseases (and therefore, treatments) recorded in their medical histories.
- Avoiding treatments started before the recording period of the dataset or treatments which did not finish before the closing date: medical procedures that are crucial for diagnosing a disease must be required in every disease treatment trajectory. Likewise, the absence of diagnosis-related actions in the first and last months of the recording period is an important requirement to obtain end-to-end treatments.
- Avoiding treatments with incomplete follow-up: a minimum follow-up time and a minimum amount of actions recorded are essential.

These selection criteria must be adjusted specifically to each disease, taking into account that the initial or final medical actions, as well as the typical time intervals between initial and final actions, vary depending on the specific diagnosis being considered. Once these criteria are established, they are applied one by one to the dataset to exclude patients that do not meet the specified requirements. The primary motivation behind this data reduction is to identify patients who are highly likely to have the complete end-to-end treatment for the disease recorded in the dataset.

2.3.3 Clustering: K-medoids with edit distance

This section describes the process to identify the subtypes of treatments and their representatives. The main idea is to group together treatments in such a way that

trajectories within a group are similar to each other but are dissimilar to trajectories assigned to other groups. Therefore, the use of a clustering method [56] seems to be a logical and promising approach.

We need to select a suitable distance measure that enables the comparison of discrete sequences of actions with variable lengths. For this purpose, the most commonly used sequence distance is the Levenshtein distance (or edit distance) [44], which enables us to calculate the similarity (or dissimilarity) between pairs of sequences.

The definition of the distance is as follows. Given two strings $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ over a finite alphabet, the edit distance between $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ can be defined as the minimum weight of transforming $\boldsymbol{a}_1$ into $\boldsymbol{a}_2$ through a sequence of weighted edit operations. These operations are usually defined in terms of insertion, deletion, and substitution of one symbol for another, possibly with different costs for each of these operations. In this work, the cost of insertion and deletion is 1, whereas the cost of substitution is 2. Nevertheless, the edit distance is not sufficient for many applications comparing strings with different lengths. Hence, normalization should be applied to appropriately rate the weight of the edit errors concerning the sizes of the objects that are compared [44, 57].

We use the K-medoids clustering method [43] to divide a dataset of N sequences $\boldsymbol{A} = \{a_i\}_{i=1}^{N}$ into distinct groups based on the similarity or dissimilarity between sequences. This method is a variation of the K-means algorithm but more appropriate for making clusters of sequences of actions for serveral reasons: i) it can be computed using distances between every pair of sequences of actions; ii) it does not require to compute the centroid of a given set of sequences, which is computationally intractable and can generate senseless sequences; iii) each cluster of sequences is characterized by a real sequence of actions, known as the medoid; and iv) it is more robust to noise and outliers.

Specifically, a common used K-medoids clustering algorithm is Partitioning Around Medoids (PAM) [58]. The fundamental concept behind PAM is as follows: it seeks to identify K representative medoids (representative treatments) in a dataset, and subsequently assigns each data point to the closest medoid, thereby creating clusters (subgroups of treatments). The primary objective is to minimize the sum of dissimilarities between the objects in a cluster and the medoid of that cluster.

- *Step 1.* Initial step: arbitrarily choose K of the N sequences as the medoids to form initial clusters.
- *Step 2.* Assignment step: associate each sequence to the closest medoid.
- *Step 3.* Update step: for each medoid $\boldsymbol{m}$ and each sequence $\boldsymbol{a}$ associated to $\boldsymbol{m}$, swap $\boldsymbol{m}$ and $\boldsymbol{a}$ and compute the average dissimilarity of $\boldsymbol{a}$ to all the sequences associated with $\boldsymbol{m}$. Select as the medoid of the cluster the sequence $\boldsymbol{a}$ with the lowest average dissimilarity.

  Repeat alternating steps 2 and 3 until there is no change in the assignments.

Thus, by using K-medoids clustering, we avoid generating artificial sequences of actions for characterizing each group. In fact, the representative sequences are actual

treatment trajectories belonging to the dataset. Obtaining these sequences of actions that minimize the mean distance relative to the rest of the sequences of the group is an NP-hard problem.

2.4 Experimental evaluation

This section describes the evaluation of the proposed methodology for extracting, segmenting and representing treatment trajectories from EHRs.

2.4.1 Dataset

We conduct a case study involving breast cancer patients to validate the proposed methodology. This analysis is based on the dataset provided by the public health care system Osakidetza, which has been introduced in Section 1.1. The methodology is applied on data exclusively from the years 2016 and 2017, in which the 75% of diagnoses are missing.

2.4.2 Extraction of complete end-to-end treatments associated with breast cancer

First of all, the target population comprised 1456 patients with breast cancer diagnosis out of 579,798 patients between January 1, 2016, and December 31, 2017. This selection of patients from the dataset is made according to the *International Statistical Classification of Diseases and Related Health Problems (10th revision)* [8], where every code starting by *C50* corresponds to breast cancer diagnosis.

Identification of medical actions associated with breast cancer. The association of actions with a diagnosis is made through the relevance of the medical specialties (Equation (2.1)). Table 2.1 shows those medical specialties whose relevance is higher than $\lambda = 3$, that is, the medical specialties given at least 3 times more frequently in breast cancer patients. Only the medical actions carried out in these 18 medical specialties are included when creating the final treatment trajectories of patients with breast cancer. Once these treatment trajectories are extracted, 21 patients out of 1456 had no action occurring in these medical specialties, therefore, they are excluded from the study.

Identification of patients with complete end-to-end treatments. Once the association between actions and breast cancer diagnosis is known, we can extract for each patient the subsequence of actions that describe the treatment of breast cancer, $\boldsymbol{a}$. However, these sequences may be incomplete. Hence, we will select the sequences of actions that have high probability of describing complete treatment trajectories of

Medical Specialties	Relevance
Gynecologic Oncology	85.9
Radiotherapy	78.3
Plastic Surgery	66.0
Medical Oncology	36.4
Day Hospital	16.5
Nuclear Medicine	11.7
Day Surgical Hospital	10.5
Genetics	10.2
Major Burns Unit	9.5

Medical Specialties	Relevance
Gynecology	9.3
Genetic Laboratory	8.8
Surgery Unit	5.8
Anesthesia	5.6
Home Hospitalization	4.8
Pathological Anatomy	4.2
Hospitalization	3.5
Others	3.3
Radiology	3.0

Table 2.1: Relevance of the medical actions associated with breast cancer diagnosis.

breast cancer. In order to do that, we propose some selection criteria, listed in Figure 2.2 and explained as follows.

First of all, the patients with any other type of cancer diagnosis apart from breast cancer are filtered out, otherwise, we could not distinguish which cancer diagnosis the treatment is focused on. Moreover, to ensure that the pathology has been diagnosed in the recording period of our dataset, at least one record of a breast biopsy procedure is required. It is the only definitive diagnostic procedure to determine if the suspicious area is cancerous [59], and therefore, should be performed for every breast cancer diagnosed patient.

Regarding the recording time of treatments, we consider that a treatment is completely recorded in the dataset if there is no diagnosis in the first and last months. Therefore, the breast cancer diagnosis must be between the 1st February 2016 and the 30th September 2017. If any patient with a breast cancer diagnosis record out of this period was included, we assume that it is the continuation of the treatment previously started or the continuation after 2017.

For the same reason, we need to avoid radiotherapy or chemotherapy actions in the last period of the dataset. Radiotherapy is delivered daily or every 2 days, and chemotherapy every 1-3 weeks [59]. Therefore, if there exists any radiotherapy or chemotherapy action in the last 3 weeks of 2017, it means that it is an unfinished treatment.

Likewise, the period of medical assistance recorded must be at least 3 months once the patient has been diagnosed with breast cancer. Additionally, the minimum number of associated actions in their treatment trajectories must be at least 15 in order to avoid incomplete sequences of actions, this could mean that patients abandoned the treatment or their follow-up was lost for some reason.

After applying these selection criteria, there are in total 440 out of 1456 patients (30.2%) with a high probability to present a complete treatment of breast cancer in the EHRs. These breast cancer treatment trajectories are made up of the actions occurred in the medical specialties in Table 2.1 and they are the eligible sequences for

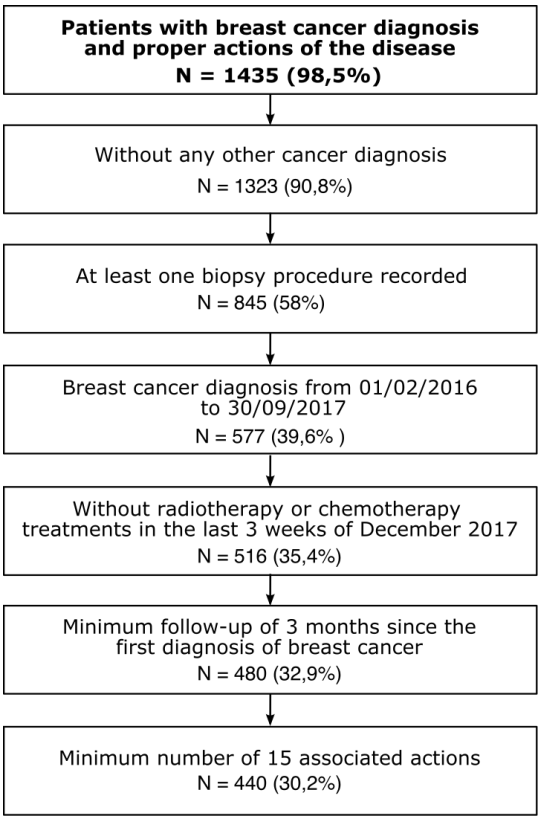


Figure 2.2: Proposal for the selection criteria for breast cancer diagnosis.

the unsupervised classification. The treatment trajectories are of variable lengths, in fact, the minimum treatment trajectory is made of 15 actions and the maximum one of 217 actions. The distribution of these durations of treatments is shown in Figure 2.3.

2.4.3 Representative treatments and their adherence to clinical practice guidelines

K-medoids algorithm is applied to the eligible disease treatment trajectories to identify the treatment patterns for breast cancer patients, with K ranging from 2 to 10. From 5 clusters on, the treatment patterns are repeated, and therefore, we consider a total of 5 subtypes, which are shown in Figure 2.4. The 5 horizontal lines are the representative disease treatment trajectories (medoids), and the vertical lines correspond to the hospital services visited by the representative patients over time.

To validate the results, the representative trajectories are compared with clinical practice guidelines, specifically, with the *European Society for Medical Oncology* breast cancer guideline [59, 60]. These guidelines provide updated state-of-the-art recommen-

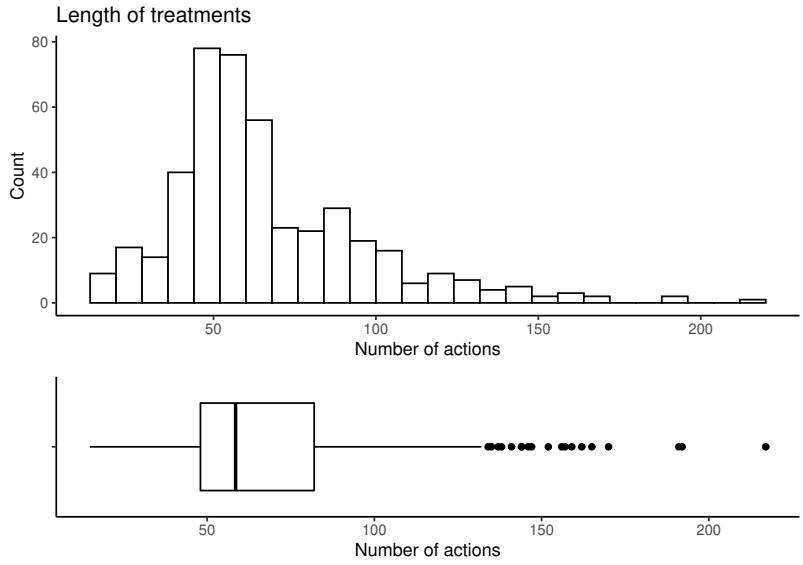


Figure 2.3: Distribution of the length of the breast cancer treatment trajectories.

dations on management of breast cancer (diagnosis, treatment and follow-up). Besides, the outcomes have been also contrasted and approved by physicians.

The 5 sequences obtained fundamentally represent different treatment trajectories to deal with breast cancer. We can see in Figure 2.4 that all of them start with Consultation, Pathological Anatomy, Nuclear Medicine and Radiology visits. In these hospital services, the breast examinations and tests are carried out: in Radiology tests such as sonography, mammogram or even some radiography; in the case of Pathological Anatomy and Nuclear Medicine, the biopsy test and cancer diagnosis. According to the clinical practice guideline, a biopsy must be done before any type of treatment is initiated and the five groups accomplish it in Pathological Anatomy actions.

The main therapies of each group are as follows (Figure 2.5):

- **Group 1: Surgery + Chemotherapy + Radiotherapy** (66 patients, 15 %). The representative disease treatment trajectory involves a 15-week course of chemotherapy (within the recommended duration of 12-24 weeks) after breast-conserving surgery, and then, a month of radiotherapy is administered. According to the guideline suggestions, if both therapies are used, chemotherapy should usually precede radiotherapy, as done in this case.

- **Group 2: Surgery + Radiotherapy + Hormonal Therapy** (89 patients, 20.3 %). This representative patient combines radiotherapy and hormonal therapy. The representative patient in this case undergoes a combination of radiotherapy and hormonal therapy. According to medical guidelines, hormonal therapy can be safely administered concurrently with radiotherapy and typically lasts 5-10 years. However, it's important to note that the dataset collects information over a

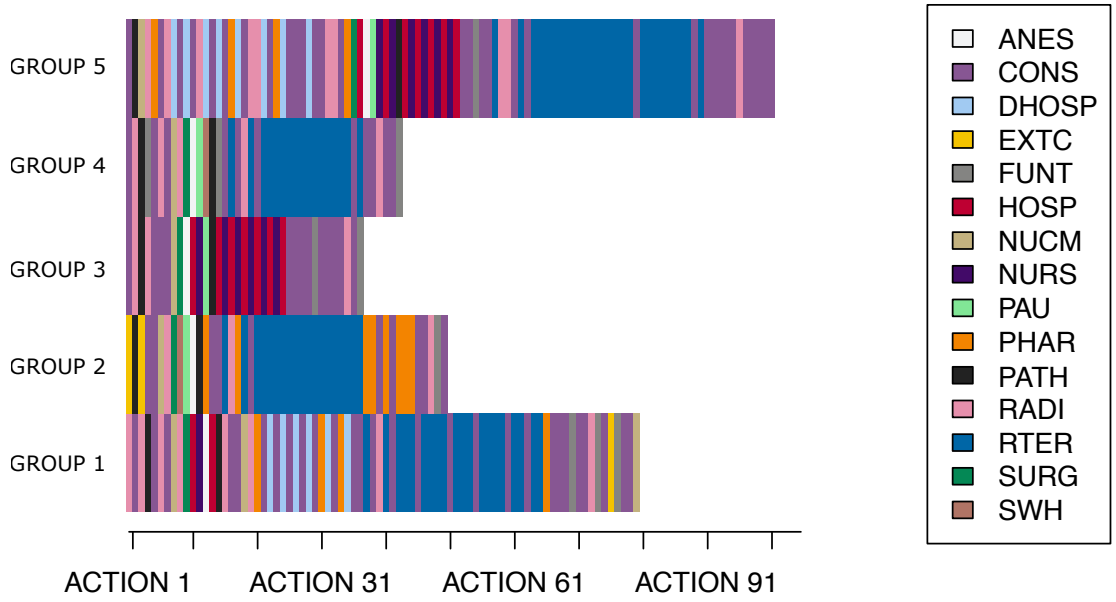


Figure 2.4: Clustering results. Representative medoids considering 5 groups. See Table 1.1 for the description of the medical actions in the legend.

period of up to 2 years, making it challenging to corroborate long-term follow-up outcomes.

Hitherto, it is worth mentioning that there exist two types of surgery when it comes to breast cancer: breast-conserving surgery, in which the surgical team removes the tumor but tries to keep as much of the breast as possible (it is the preferred local treatment option for the majority of early breast cancer patients, in fact, this procedure is performed in most of the groups); or mastectomy, in which the whole breast is removed. In this latter case it is possible to have no therapy after surgery, and in general terms, these are commonly early invasive breast cancer patients [59].

- **Group 3: Surgery + Hospitalization** (108 patients, 24.6%). We suspect that this particular class corresponds to the group of patients who undergo mastectomy, as they receive no further therapy after the surgical procedure. Instead, their post-surgical care involves a series of hospitalization actions combined with nursing interventions. These hospitalizations after undergoing surgery might be due to complications, that is, deviations from guidelines since nothing is explic-

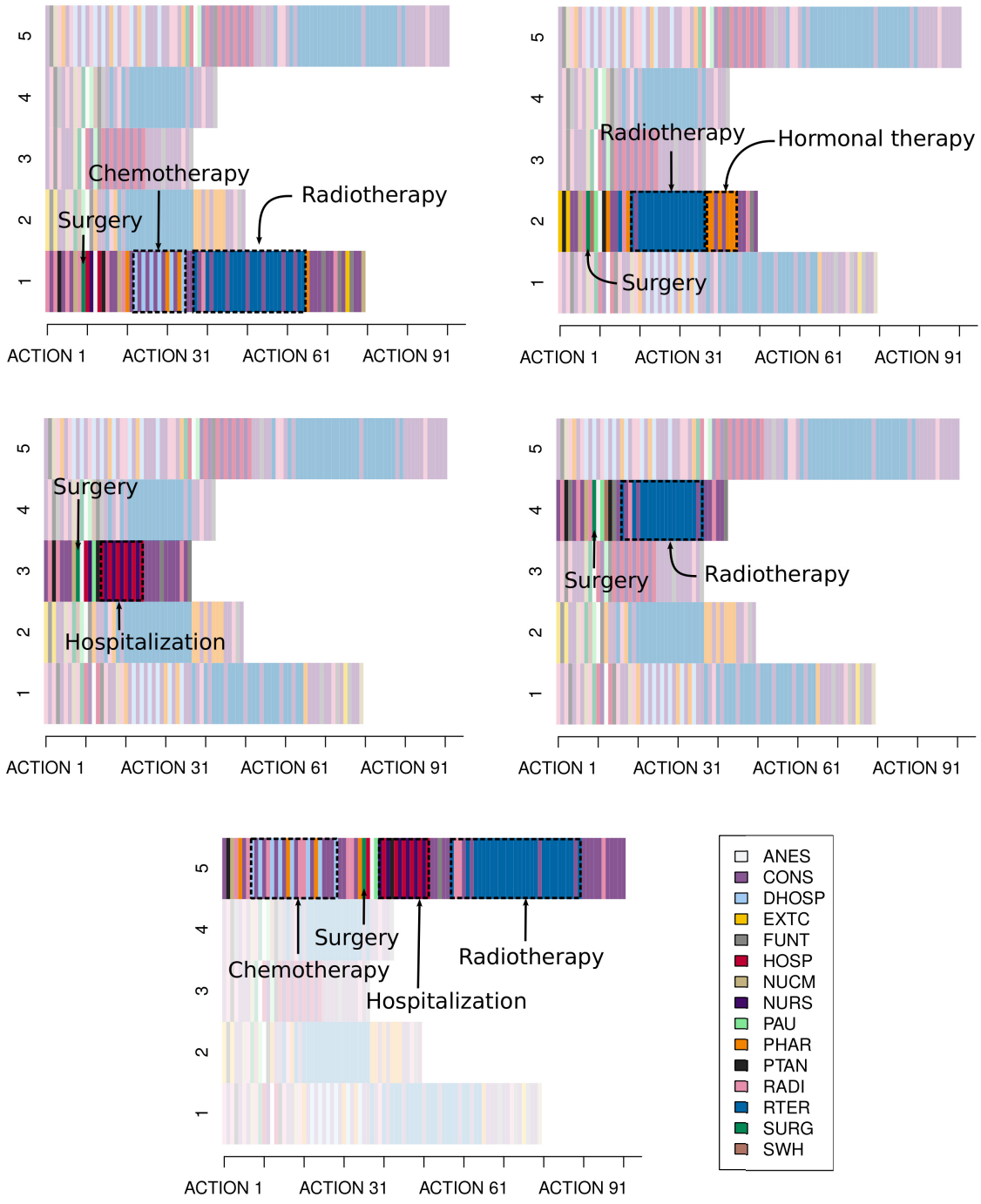


Figure 2.5: Clustering results. Treatment patterns of each class. See Table 1.1 for the description of the medical actions in the legend.

itly mentioned there about hospital stays. This group constitutes one of the

highest number of patients, however, we suspect that some of these patients may come from other hospitals only for surgical treatment. This conclusion arises from discussions with practitioners and clinicians who have indicated that it is not common to have such a significant number of patients without post-surgical therapy in their regular practices.

- **Group 4: Surgery + Radiotherapy** (137 patients, 31.2%). This group of patient describes the most simple and common delivered treatment. The representative patient undergoes breast-conserving surgery, and then receives postoperative radiotherapy, which is highly recommended in practice guidelines.

  Until now, all the representative treatments start therapy after undergoing surgery, which known as Adjuvant Systemic Treatment. However, the remaining group is the only one that also receives therapy before undergoing surgery. This type of treatment is called Neoadjuvant Systemic Treatment and should be used to reduce the extent of surgery in locally advanced and large operable cancers.

- **Group 5: Chemotherapy + Surgery + Hospitalization + Radiotherapy** (40 patients, 9.1%). According to the guidelines, when Neoadjuvant Systemic Treatment is used, all chemotherapy should be delivered preoperatively as done in this case. In particular, 8 rounds of chemotherapy were delivered in 16 weeks, which comes with the recommendation of 12-24 weeks. Furthermore, they mention that magnetic resonance imaging of the breast, which is a test used to detect breast cancer and other abnormalities, is the most accurate modality for assessing the extent of residual disease following Neoadjuvant Systemic Treatment. It should also be carried out before initializing the treatment for proper comparative evaluation. In this patient, it was conducted in the Radiology unit after the first 5 sessions of chemotherapy and once the entire therapy was completed. After breast-conserving surgery, postoperative radiotherapy was delivered, strongly recommended by the clinical guideline. We can observe also in this group some hospitalization actions that deviate from established medical practice guidelines.

The follow-up of the patients is not clearly defined since our dataset only covers 2 years. However, in these 2 years, based on the clinical guideline recommendations, regular visits should be made every 3-4 months. These regular visits correspond to Consultations in the final part of the representative disease treatment trajectories. Furthermore, annual bilateral (after breast-conserving treatment) and/or contralateral mammography (after mastectomy) is also recommended. Bilateral mammography in Radiology was performed in the 5 groups. In some cases, they also have Functional Testing actions (groups 1, 2, 3 and 4) or Nuclear Medicine actions (group 1), which are also likely to be related to the follow-up.

2.5 Discussion

The methodology is designed to tackle the missing information and heterogeneity in EHRs. In addition to that, we also face the difficulty of having comorbidity together with missing diagnosis. The effectiveness and applicability of this methodology have been tested using breast cancer patients, but it can be applied to identify distinct treatment patterns for various other medical conditions, including short-duration diseases. Subsequently, a comparison of the outcomes with clinical practice guidelines can be conducted to determine whether they are adhered in practice. It is also worth mentioning that the identified treatment patterns might be useful for detecting deviations in the treatments from these guidelines.

However, there are some limitations to the application of the proposed methodology. For common diagnoses, such as acute sinusitis, it may fail to identify associated actions effectively. Patients with this type of usual pathologies may visit regular medical specialists (e.g., primary care or consultations), and therefore are unlikely to present high relevance values. That is, they will have no distinctive medical specialty in order to extract the associated disease treatment trajectories (see Equation (2.1)).

Another limitation arises when attempting to extract complete treatments from EHRs for long-duration diseases, exceeding the recording time of the EHRs. These pathologies will have no complete treatments in the dataset as required in the proposed methodology. In fact, in the particular case of breast cancer, some treatments usually finish with hormonal therapy for 5-10 years, however, the recording time of the dataset is of 2 years. In [30] the authors designed a method for creating complete treatments of pseudopatients by merging partial treatments. That is, they align the final part of some patients' disease treatment trajectories that coincide, to some extent, with the initial part of others.

Furthermore, the proposed methodology does not take full advantage of the temporal information. Incorporating the time variable could enhance the results in multiple ways [61], such as better identification of diagnosis-associated actions and improved clustering outcomes. Timestamps could be included in the definition of an action as$\tau = t_i - t_{i-1}$). Then, actions with a τ value higher than a threshold ρ could be excluded from disease treatment trajectories. For instance, in breast cancer cases, it would not make sense to have a surgical action without any prior breast cancer-related action (e.g., a biopsy procedure) within a 2-month period. Likewise, the cluster outcomes might be improved if the time were considered when defining the proper distance for comparing sequences: the larger the τ value, the larger the penalization between actions, even if the hospital services match.

In the subsequent chapters we extend this methodology to capture the progression of medical actions over time. This means that the methodology would not only consider the chronological order of medical actions within treatment trajectories but also how medical actions evolve as a patient undergoes a treatment. This enhancement involves tracking changes in patients' health states and the progression patterns of medical actions. By doing so, we obtain a more comprehensive and dynamic representation of

disease treatment trajectories, which offers valuable insights into the temporal progression patterns of treatments.

2.6 Conclusion

This chapter proposes a methodology to identify treatment patterns for a disease of interest using EHRs, even when diagnosis information is incomplete or missing. Through the definition of a relevance measure and several selection criteria, we extract complete end-to-end treatments composed by the medical actions directly associated with a specific diagnosis. Then, we use the K-medoids algorithm with the normalized edit distance as a distance metric to group patients and identify representative treatment patterns from EHRs.

Practical applications with breast cancer patients demonstrate the model's ability to extract complete end-to-end treatments from clinical data with missing values, segment treatment populations, and depict this population with a set of representative treatments from EHRs. It is important to highlight the potential applicability of the methodology to a wide range of diseases. The validation of the experimental results by healthcare professionals and the alignment of the treatments with clinical practice guidelines further improves the reliability of the proposed methodology.

Chapter 3

A probabilistic generative model for disease progression

3.1 Introduction

Disease progression research aims to improve the understanding of complex and heterogeneous pathologies. This is achieved by modeling the evolution of disease trajectories over time, taking into account changes in patients' health states and considering the chronological order of medical events. Generative models have shown the potential to capture these disease dynamics from sequential medical data. However, creating accurate models to understand this progression in sequences of events remains a fundamental challenge in the field of medical informatics.

In Chapter 2, we revealed substantial variability in treatment trajectories, highlighting the need for models that account for disease treatment subtypes. To tackle this variability, the method proposed in the previous chapter employs a partitional clustering approach based on sequence distances. This technique associates each treatment trajectory with a unique cluster, and represents the clusters through treatments extracted from EHRs. However, it does not explicitly model the evolution of treatment trajectories as disease progression models do. A probabilistic clustering model could effectively capture the progression dynamics, while simultaneously considering diverse subtypes of treatments.

The importance of addressing the heterogeneity in clinical trajectories has been also evidenced by works based on the conventional LDA that aim to identify subgroups of patients with similar trajectory characteristics [25–28]. However, these approaches often assume that all individuals are at a unique treatment progression stage, limiting their ability to account for treatment progression. Additionally, they face challenges in capturing the temporal order of medical events, as they primarily model the frequency of each event type rather than being generative models of the sequential medical events.

HMMs have been widely used for disease progression due to their easy interpretability and their temporal relation assumption in data. Most existing HMMs [12, 35–39, 62]

assume that all patients evolve through the same latent state transition dynamics, thus ignoring the heterogeneity of different subtypes of disease progression. Other probabilistic approaches that simultaneously address disease state progression and treatment subtyping [30, 40–42] are limited to model the evolution of observed data through a latent process and do not directly handle the sequential dependence within medical actions, that is, the order in which the medical events occur.

Various predictive deep learning models have also been developed for healthcare settings [10, 12–15, 19, 20, 63]. They not only ignore the variability in treatments, but also their hidden states do not correspond to clinically meaningful variables such as the treatment evolution patterns provided by probabilistic models. While these methods succeed in predicting a target outcome, they do not provide a generative model of the disease progression to identify patients with similar disease progression patterns, to understand the evolution of treatments through interpretable distributions of stage transitions, or to simulate populations of treatment trajectories.

This chapter introduces a probabilistic generative model that employs latent classes to cluster treatment trajectories and latent stages to identify their temporal progression within each subtype. In summary, the key contributions of this work are as follows:

- We model EHRs using a probabilistic generative model built on Markov models to capture the order of occurrence of the events. The model discovers the subtypes of treatments by grouping the sequences of medical actions into different classes according to their evolution and identifies the progression stages of the treatments over time.

- We efficiently learn the model with the EM algorithm [45] and a dynamic programming-based method that reduces the complexity of the model learning process from exponential to polynomial.

- We evaluate the learning performance of the model in multiple simulated datasets of different sizes with to demonstrate that the model underlying the data is recovered.

- We apply the model on a breast cancer dataset to represent the progression of the different classes of treatments and their phases. The results are contrasted with clinical guidelines and approved by physicians.

The remainder of this chapter is organized as follows. Section 3.2 describes the problem formulation. Section 3.3 introduces the novel probabilistic generative model and the learning process of the parameters by means of the EM algorithm. Section 3.4 presents the results of the synthetic data experiments that evaluate the performance of the proposed method, and the application of the model on a real-world dataset. Section 3.5 discusses the contributions and limitations of our approach. Finally, Section 3.6 draws the conclusions.

3.2 Problem formulation

A patient's treatment trajectory, represented as $\boldsymbol{a}$, consists of a sequence of medical actions accumulated during multiple hospital visits associated with a specific disease. In our context, these medical actions indicate the medical service that a patient has visited, including primary care, surgery unit, hospitalization, and more (see Table 1.1). Let A be the set of all the possible medical actions, then, we define a (disease) treatment trajectory as

$$\boldsymbol{a} = (a_1, ..., a_m),$$

where $a_i \in A$ represents the i-th medical action of a patient.

Given a dataset of medical records, the objective is to develop a probabilistic generative model to effectively capture the disease subtypes and the progression patterns from a set of sequences.

3.3 Methodology

This section describes the proposed probabilistic generative model and the procedure for the inference and the parameter estimation.

3.3.1 Model definition

The general idea is to develop a probabilistic generative model to learn the underlying distribution of a set of discrete sequences of different lengths. We assume that sequences of actions have an associated hierarchical structure of latent variables: at the top-level, we consider that sequences belong to latent classes representing the different subtypes of treatments; at the lower-level, we assume that the sequences of actions progress through a set of latent ordinal-valued stages over time, that is, each action of a sequence has an associated stage that indicates the phase of progression of the treatments at that time point. The goal, therefore, is to simultaneously infer these latent classes of treatments and their progression stages to capture the heterogeneity of the sequences of medical actions.

For the definition of the generative model, we consider that an action depends on the sequence's most recent action and stage within a class. Furthermore, a stage within a class depends on the current action and the previous stage. The duration of the progression stages for each sequence is likely to be different because each patient evolves at their own rate, and consequently, the lengths of the sequences of actions vary. For that reason, we introduce the virtual end-of-treatment action a_m, which allows to implicitly model the length of a population of sequences of actions. The inclusion of this end-of-treatment action prevents the generative model from creating sequences of infinite length. Besides, we consider that the sequences of actions always start in the first stage, representing the initial steps of the treatment. We assume that all the classes of treatments have the same number of stages. The definition of such stages makes it

possible to segment each class of treatments into subsequences that are related to their progression. Note that equivalent stage values across different treatment classes denote distinct subsequences, which allows the model to be more flexible and to better fit a population of sequences of actions. With these assumptions, we develop a generative process built on Markov models that classifies and segments sequences automatically.

Let $\boldsymbol{a} = (a_1, ..., a_m)$ be the sequence of medical actions representing a treatment of a patient associated with a disease. The medical actions a_i belong to a set A which is the set of all the possible medical actions including the virtual end-of-treatment action. Let $\boldsymbol{s} = (s_1, ..., s_m)$ be the sequence of latent stages of the treatment associated with the sequence of actions $\boldsymbol{a}$. The stages s_i belong to a set $S = \{1, ..., r\}$ that represents all the possible stages of a treatment trajectory. Finally, let c be the latent class of treatment which $\boldsymbol{a}$ belongs to. The classes c belong to a set $C = \{1, ..., k\}$ that represents the subtypes of treatments for a disease. Furthermore, we assume that the progression stages of a sequence of actions are non-decreasing, that is, a sequence can not progress backward. Therefore, $s_t \leq s_{t+1}$ for all $t = 1, ..., m-1$.

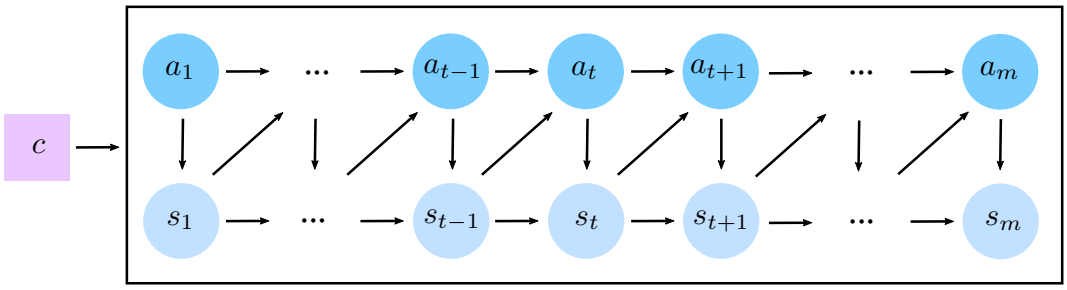


Figure 3.1: Generative model defined by the conditional distributions $p(a_t|a_{t-1}, s_{t-1}, c)$ and $p(s_t|a_t, s_{t-1}, c)$ for sequences of actions $\boldsymbol{a}$, latent sequences of stages $\boldsymbol{s}$ and latent classes c.

The proposal for the probabilistic generative model is as follows (see Figure 3.1):

a) Draw a class of treatment $c \sim Mult(\boldsymbol{\theta}_C)$

b) Draw the initial medical action and the initial stage

$$a_1|c \sim Cat(\boldsymbol{\pi}_A^c), \quad s_1|a_1, c \sim Cat(\boldsymbol{\pi}_S^{a_1,c}).$$

c) For each timestamp index t:

i) Draw a medical action from $p(a_t|a_{t-1}, s_{t-1}, c)$, the transition matrix of the Markov model conditioned on the action a_{t-1}, the stage s_{t-1} and the class c. That is,

$$a_t|a_{t-1}, s_{t-1}, c \sim Cat(\boldsymbol{\theta}_A^{a_{t-1},s_{t-1},c})$$

ii) Draw a stage s_i from $p(s_t|a_t, s_{t-1}, c)$, the transition matrix of the Markov model conditioned on the action a_t, the stage s_{t-1}, and to the class c, that is,

$$s_t|a_t, s_{t-1}, c \sim Cat(\boldsymbol{\theta}_S^{a_t,s_{t-1},c})$$

Translating the generative process into a joint probability model results in the expression:

$$p(\boldsymbol{a}, \boldsymbol{s}, c) = p(c) \prod_{t=1}^{m} p(a_t, s_t|a_{t-1}, s_{t-1}, c) \tag{3.1}$$

where

$$p(a_t, s_t|a_{t-1}, s_{t-1}, c) = p(a_t|a_{t-1}, s_{t-1}, c) \cdot p(s_t|a_t, s_{t-1}, c)$$

and $p(a_1, s_1|a_0, s_0, c) = p(a_1, s_1|c)$. Furthermore, $s_1 = 1$, $a_m = end$, and $s_{t-1} \leq s_t$ for all t.

In light of the above, $p(c)$ is a multinomial distribution that describes the probability of drawing a class from the set of classes of treatments C. We define $\boldsymbol{\theta}_C$ as the set of such probabilities:

$$\boldsymbol{\theta}_C = \{p(c) : c \in C\} \tag{3.2}$$

In addition, we define the Markov models from which the actions and stages are drawn as follows (see Figure 3.1). The first conditional distribution is given by a set of $|C|$ transition matrices of size $|A||S| \times |A|$ whose model parameters are:

$$\boldsymbol{\theta}_A = \{\boldsymbol{\theta}_A^{a,s,c} : a \in A, s \in S, c \in C\} = \{p(a'|a, s, c) : a, a' \in A, s \in S, c \in C\}. \tag{3.3}$$

The other conditional distribution is given by a set of $|C|$ transition matrices of size $|A||S| \times |S|$ whose model parameters are:

$$\boldsymbol{\theta}_S = \{\boldsymbol{\theta}_S^{a,s,c} : a \in A, s \in S, c \in C\} = \{p(s'|a, s, c) : a \in A, s, s' \in S, c \in C\}. \tag{3.4}$$

Finally, the parameters of the initial generative model for medical actions and stages are defined as $\boldsymbol{\pi}_A^c$ and $\boldsymbol{\pi}_S^{a,c}$, respectively.

For the sake of simplicity, we define the classes of treatments with a fixed number of stages. This way, the notation is simplified and it is easier to understand the main idea of the model. However, it is possible to define a more flexible model in terms of stages. It may be the case that some sequences are incomplete because the treatment of a patient is still in progress by the closing date of the dataset. With this flexibility, the model manages to segment the complete sequences into the maximum number of stages r^+, but also the incomplete sequences into a lower number of stages, ranging from r^- to r^+.

3.3.2 Maximum likelihood parameter estimation

This section introduces the learning procedure of the parameters of the model. Given an observed sequence of actions $\boldsymbol{a} = (a_1, ..., a_m)$ and its underlying latent sequence of stages $\boldsymbol{s}$ and class c, we can compute the likelihood function from the joint distribution in Equation (3.1) by marginalizing over the latent variables

$$p(\boldsymbol{a}; \boldsymbol{\theta}) = \sum_{c} \sum_{\boldsymbol{s}} p(\boldsymbol{a}, \boldsymbol{s}, c; \boldsymbol{\theta}).$$

As discussed in Section 1.3.2, the complete dataset, including the respective latent variable values for each observation in $\mathcal{D}$, is unavailable. Hence, we use the EM algorithm [46] to find an effective framework for maximizing the likelihood function. This involves considering the maximization of the expected value of the complete-data log-likelihood concerning the posterior distribution of the latent variables, denoted as $p(\boldsymbol{s}, c|\mathbf{a})$:

$$\max_{\boldsymbol{\theta}} \sum_{\boldsymbol{a} \in D} \sum_{\boldsymbol{s} \in \mathcal{S}_{\boldsymbol{a}}} \sum_{c \in C} p(\boldsymbol{s}, c|\boldsymbol{a}) \cdot \log p(\boldsymbol{a}, \boldsymbol{s}, c; \boldsymbol{\theta}) \tag{3.5}$$

where $\mathcal{S}_{\boldsymbol{a}}$ is the set of all the potential configurations of the sequences of stages for $\boldsymbol{a}$, and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_A, \boldsymbol{\theta}_S, \boldsymbol{\theta}_C\}$ the parameters of the model to be learned. Note that each sequence $\boldsymbol{a}$ contributes equally to the model regardless of its length, and this is achieved because

$$\sum_{c \in C} \sum_{\boldsymbol{s} \in \mathcal{S}_{\boldsymbol{a}}} p(\boldsymbol{s}, c|\boldsymbol{a}) = \sum_{c \in C} \sum_{\boldsymbol{s} \in \mathcal{S}_{\boldsymbol{a}}} p(\boldsymbol{s}|\boldsymbol{a}, c) \cdot p(c|\boldsymbol{a}) = 1. \tag{3.6}$$

The EM algorithm starts with some initial selection for the model parameters and iterates as follows:

E-step. In this step, the goal is to calculate the posterior distribution of the latent variables given the observed sequence of actions $\boldsymbol{a}$, that is, $p(\boldsymbol{s}, c|\boldsymbol{a})$. We then use this posterior distribution to evaluate the expectation of the logarithm of the complete-data likelihood function, as a function of the parameters $\boldsymbol{\theta}$ (Equation (3.5)).

M-step. In the maximization step the aim to update the parameters of the generative model to maximize the likelihood of the observed data in Equation (3.5), based on the expected values of the latent variables computed in the E-step.

3.3.2.1 Efficient learning of the parameters of the model

Suppose that we have a training set $\mathcal{D} = \{\boldsymbol{a}^i\}_{i=1}^N$ that consists of a set of sequences of actions $\boldsymbol{a} = (a_1, ..., a_m)$, a latent variable of stages $\boldsymbol{s} = (s_1, ..., s_m)$ and a latent variable of classes c.

In the E-step, we are interested in finding the marginal posterior distribution $p(s'|\boldsymbol{a}, c)$ and $p(s, s'|\boldsymbol{a}, c)$ for $s', s \in S$ to learn the maximum likelihood estimate parameters. To

achieve this, we need to marginalize $p(\boldsymbol{s}|\boldsymbol{a}, c)$ and compute the probability of all the sequences of stages with the form $(s_1, ..., s_{t-2}, s, s', s_{t+1}, ..., s_m)$ in c for each $s, s' \in S$.

Recall that this requires $\binom{m-2}{r-1}$ number of configurations for $\boldsymbol{s}$ (the last stage is fixed), which is exponential. Adopting the notion of the forward-backward algorithm used for learning HMMs [46], we develop a generalization of this dynamic programming method for the specific characteristics of our model, which avoids the exponential complexity. The conventional algorithm does not suffice for constructing our forward-backward filtering algorithm since we need to account for the direct sequential relation between the observations, as well as the classes and the latent correlation structures of stages on observed actions.

Let us assume that $f_c(t, s)$ is the sum of the probabilities of all the sequences of stages $(s_1, ..., s_t)$ in the class c that ends at $s_t = s$, and $g_c(t, s)$ is the sum of the probabilities of all the sequences of stages $(s_{t+1}, ..., s_m)$ that starts at $s_t = s$ in the class c. Then,

$$f_c(t, s) = \sum_{\boldsymbol{s}_{1:t-1}} p(\boldsymbol{a}_{1:t-1}, \boldsymbol{s}_{1:t-1}|c) \cdot p(a_t|a_{t-1}, s_{t-1}, c)p(s_t = s|a_t, s_{t-1}, c) \tag{3.7}$$

$$g_c(t, s) = \sum_{\boldsymbol{s}_{t+1:m}} p(\boldsymbol{a}_{t+1:m}, \boldsymbol{s}_{t+1:m}|s_t = s, c), \tag{3.8}$$

where $\boldsymbol{a}_{i:j} = (a_i, \ldots, a_j)$ and $\boldsymbol{s}_{i:j} = (s_i, \ldots, s_j)$.

Now, we can express the sum of the probabilities of the sequences for which $s_{t-1} = s$ and $s_t = s'$ as

$$p(s_{t-1} = s, s_t = s'|\boldsymbol{a}, c) = \frac{p(s_{t-1} = s, s_t = s', \boldsymbol{a}|c)}{p(\boldsymbol{a}|c)}. \tag{3.9}$$

Using Equations (3.7) and (3.8),

$$\begin{aligned}
&p(s_{t-1} = s, s_t = s', \boldsymbol{a}|c) = \\
&\quad = \sum_{\substack{\boldsymbol{s}_{1:t-2} \\ \boldsymbol{s}_{t+1:m}}} p(\boldsymbol{a}_{1:t-1}, \boldsymbol{s}_{1:t-2}, s_{t-1} = s|c) \cdot p(a_t|a_{t-1}, s_{t-1} = s, c) \cdot p(s_t = s'|a_t, s_{t-1} = s, c) \cdot \\
&\qquad\qquad p(\boldsymbol{a}_{t+1:m}, \boldsymbol{s}_{t+1:m}|s_t = s', c) \\
&\quad = f_c(t-1, s) \cdot p(a_t|a_{t-1}, s_{t-1} = s, c) \cdot p(s_t = s'|a_t, s_{t-1} = s, c) \cdot g_c(t, s')
\end{aligned}$$

We can store the values obtained from the functions f_c and g_c for $t \in \{1, ..., m\}$ and $s \in S$ in a matrix of size $r \times m$ associated with each function. Using dynamic programming, we efficiently compute f_c and g_c and reduce the number of computations for the parameter estimation. The functions f_c and g_c are defined as recursive functions as follows (see Figure 3.2):

$$
\begin{aligned}
f_c(t,s) =& p(a_t|a_{t-1},s,c)\cdot p(s|a_t,s,c)\cdot f_c(t-1,s)\\
&+ p(a_t|a_{t-1},s-1,c)\cdot p(s|a_t,s-1,c)\cdot f_c(t-1,s-1)\\
g_c(t,s) =& p(a_{t+1}|a_t,s+1,c)\cdot p(s+1|a_{t+1},s,c)\cdot g_c(t+1,s+1)\\
&+ p(a_{t+1}|a_t,s,c)\cdot p(s|a_{t+1},s,c)\cdot g_c(t+1,s)
\end{aligned}
$$

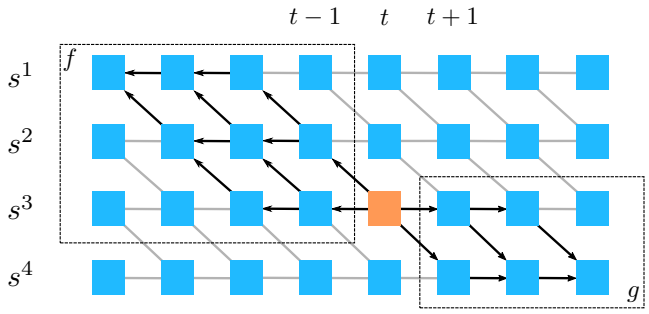


Figure 3.2: Dynamic programming procedure developed to learn the parameters of the model. The orange box represents Equation (3.9), and f and g correspond to the recursive functions. The black arrows generate all the possible sequences of stages that pass through the orange box. Note that in this example the maximum stage r^+ is the same as the minimum stage r^-.

The functions f_c and g_c are defined in such a way that consecutive stages s_{t-1} and s_t are non-decreasing, $s_{t-1} \leq s_t$ for $t = 1, ..., m$. Intuitively, we use dynamic programming and marginalize over $p(\boldsymbol{s}|\boldsymbol{a}, c)$ in an exponential number of stages in order to obtain $p(s_t = s|\boldsymbol{a}, c)$ and $p(s_{t-1} = s', s_t = s|\boldsymbol{a}, c)$.

In the M-step, we use the posterior distributions computed using Equation (3.9) as constants to maximize Equation (3.5) with respect to the parameters $\boldsymbol{\theta}$. This maximization is achieved using Lagrange multipliers (see Appendix A.1). If $\theta_{a'}^{a,s,c}, \theta_{s'}^{a',s,c}$ denote a component in $\boldsymbol{\theta}_A^{a,s,c}$, $\boldsymbol{\theta}_S^{a,s,c}$, respectively, the model parameters corresponding to the transition from the pair (a, s) to (a', s') given the class c where $a, a' \in A$ and $s, s' \in S$ are updated as follows:

$$
\theta_{a'}^{a,s,c} = \frac{\sum_{\boldsymbol{a}\in\mathcal{D}}\sum_{t=1}^{m_{\boldsymbol{a}}} \mathbb{1}_{a,a'}(a_{t-1},a_t)\cdot p(s_{t-1}=s|c,\boldsymbol{a})}{\sum_{a'\in A}\sum_{\boldsymbol{a}\in\mathcal{D}}\sum_{t=1}^{m_{\boldsymbol{a}}} \mathbb{1}_{a,a'}(a_{t-1},a_t)\cdot p(s_{t-1}=s|c,\boldsymbol{a})} \tag{3.10}
$$

$$
\theta_{s'}^{a',s,c} = \frac{\sum_{\boldsymbol{a}\in\mathcal{D}}\sum_{t=1}^{m_{\boldsymbol{a}}} \mathbb{1}_{a'}(a_t)\cdot p(s_{t-1}=s, s_t=s'|c,\boldsymbol{a})}{\sum_{s'\in S}\sum_{\boldsymbol{a}\in\mathcal{D}}\sum_{t=1}^{m_{\boldsymbol{a}}} \mathbb{1}_{a'}(a_t)\cdot p(s_{t-1}=s, s_t=s'|c,\boldsymbol{a})} \tag{3.11}
$$

where

$$
\mathbb{1}_{a,a'}(a_{t-1},a_t) = \begin{cases} 1 & \text{if } a_{t-1}=a, a_t=a' \\ 0 & \text{otherwise} \end{cases}.
$$

and

$$
\mathbb{1}_{a'}(a_t) = \begin{cases} 1 & \text{if } a_t=a' \\ 0 & \text{otherwise.} \end{cases}.
$$

Finally, if θ_c denotes a component in $\boldsymbol{\theta}_C$, we update the probability of the class of treatments $c \in C$ as follows:

$$\theta_c = \frac{\sum_{\boldsymbol{a} \in \mathcal{D}} p(c|\boldsymbol{a})}{\sum_{c \in C} \sum_{\boldsymbol{a} \in \mathcal{D}} p(c|\boldsymbol{a})}. \tag{3.12}$$

At each iteration of the algorithm, we combine the expectation and maximization steps for each sequence of actions $\boldsymbol{a}$ in such a way that we avoid storing, in the E-step, the exponential number of probabilities of all the possible sequences of stages and classes for the entire dataset $\mathcal{D}$. In addition, note that the proposed dynamic programming based method allows the EM algorithm to be solved considering the exponential number of sequences of stages with a computational complexity of $O(N \cdot m^2)$, where m is the length of the longest sequence of actions.

The large amount of possibilities in the combination of pairs of sequences of actions and stages creates problems of sparsity in the Markov models. Once the maximum likelihood estimation of the parameters assigns zero probability to some transition, there is no possibility to obtain in the subsequent step a different value for that pair of action-stages. We solve this problem by smoothing the parameters of the Markov models in each iteration of the EM algorithm.

3.3.3 Inference on latent classes and stages

Given the proposed model and the observed sequences of actions, we can efficiently make inference regarding the latent classes and stages by means of the dynamic programming based algorithm (see Section 3.3.2.1) in spite of their exponential number of configurations. In this way, we can compute:

- The probability of the latent classes given a sequence of actions $p(c|\boldsymbol{a})$ or the entire dataset $p(c)$.
- The probability of a latent sequence of stages given a sequence of actions and a class, $p(\boldsymbol{s}|\boldsymbol{a}, c)$.
- The probability of being in each latent stage of a class at each time point given the observed sequences of actions, that is, $p(s_t = s|\boldsymbol{a}, c)$ for $t = 1, ..., m_{\boldsymbol{a}}$.
- The probability of a sequence of actions given a class, $p(\boldsymbol{a}|c)$.
- The probabilities $p(s_t, c|\boldsymbol{a})$ and $p(s_{t-1}, s_t, c|\boldsymbol{a})$ computed in the EM algorithm (Equations (3.10) and (3.11)) for the parameter estimation.
- Expectations such as $\mathbb{E}_{p(\boldsymbol{s},c|\boldsymbol{a};\boldsymbol{\theta}')}[\log p(\boldsymbol{a}, \boldsymbol{s}, c; \boldsymbol{\theta})]$.

Subsequently, these inferences can be used to find the most probable latent class for each sequence of actions, and group together those with common evolution patterns. In addition, in order to show the general behavior of a class, the groups can be represented by the most probable sequences of actions. All these probabilities are calculated with a polynomial time complexity using the dynamic programming based method.

3.4 Experimental results

This section empirically shows two types of results. Firstly, we use synthetic datasets of different sizes to evaluate the behavior of the learning algorithm by comparing the learned models with the original generative model underlying the data. The corresponding source code is publicly available[1]. Secondly, we apply the model on real-world EHRs involving breast cancer patients to classify their treatment trajectories and segment them in different progression stages.

3.4.1 Results on synthetic data

We firstly create a probabilistic generative model $p_{\boldsymbol{\theta}}$, whose parameters are generated as follows: $p(c)$ is sampled from a uniform Dirichlet distribution with parameters $\alpha = 1$; $p(a'|a, s, c)$ is also sampled from a uniform Dirichlet distribution with parameters $\alpha = 1$ for $a, a' \in A$, $s \in S$ and $c \in C$; and $p(s'|a, s, c)$ is sampled from a Dirichlet distribution setting $\alpha = 0.7$ for the parameters whose corresponding transition stays in the same stage ($s' = s$) and setting $\alpha = 0.3$ for those that progress to a different stage ($s' \neq s$), for $a \in A$, $s, s' \in S$ and $c \in C$. The fundamental reason for setting a lower value when the transition progresses to a different stage is to generate more realistic phases by avoiding subsequences of stages which are too short.

For the sake of simplicity, we fix the total number of classes $|A| = 3$, the minimum number of stages $r^- = 3$, and the maximum number of stages $r^+ = 4$ to sample the training sets of sizes $N = \{300, 400, 500, 600, 700, 800, 900, 1000, 2000, 3000\}$ using the randomly generated model $p_{\boldsymbol{\theta}}$ (see Appendix A.2 for more details about the training sets). In particular, we use 10 unique actions to generate these sequences. Apart from that, we also sample a test set of 4000 sequences from p_θ in order to evaluate the learning process.

The objective is to show that the proposed learning algorithm is able to recover the generative model. Therefore, we fit the model on the training sets using the EM-based procedure proposed in Section 3.3.2. In the initialization of the EM algorithm, we segment the sequences of actions into equal-length intervals of stages and we initialize the probability of each sequence to belong to the classes with the uniform distribution. We then add a probability $\epsilon = 0.1$ to the true class to avoid relabeling in the results. After training the model, we analyze the evolution of the quality of the learned models as the training set size $n \in N$ increases. For each value $n \in N$ we obtain a new model $\boldsymbol{\theta}^n = \{\boldsymbol{\theta}^n_A, \boldsymbol{\theta}^n_S, \boldsymbol{\theta}^n_C\}$ and we measure the quality of such a model by using the log likelihood of Equation (3.5) normalized by n to make the datasets comparable.

The experiment is carried out five times, considering in each of them a different random generative model $p_{\boldsymbol{\theta}}$, from which the training sets and the test sets are generated. Figure 3.3 shows the fitting and generalization ability of our model by means of the average log likelihood. The average log likelihood of the learned models on the

[1] https://github.com/onintzezaballa/ProbGenerativeModel

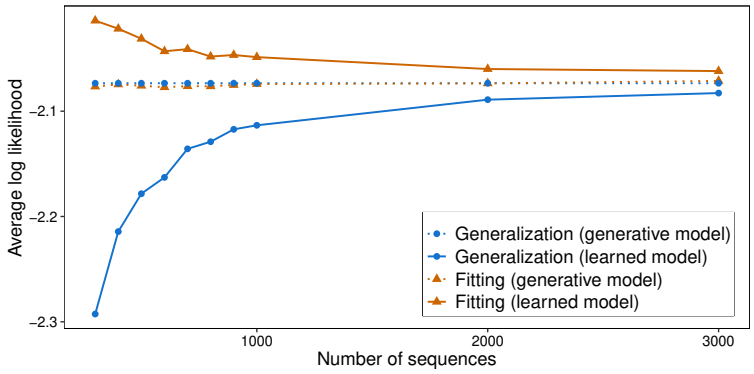


Figure 3.3: Fitting and generalization of synthetic models.

training sets (solid orange line) quantifies the fitting of the models to the data, while on the test set (solid blue lines) it measures its ability of generalization. The dotted lines correspond to the average log likelihood of the 5 original generative models evaluated in the training (orange) and test (blue) datasets. We can see that as N increases, the curves that quantify the fitting and generalization of the learned models converge to the curves of the original generative models. This means that, given a sufficiently large dataset, the proposed learning algorithm recovers the original generative model underlying the data.

3.4.2 Results on real data

This section shows the application of the model on a real-world dataset of breast cancer, where we represent the classification and stage progression of the sequences of actions associated with such disease. The achieved results were compared with clinical practice guidelines [59] and discussed in detail with physicians to check their coherence and validity.

3.4.2.1 Dataset

We use a dataset provided by the public health care system Osakidetza, introduced in Section 1.1. This dataset records the sequences of medical actions of patients for any diagnosed disease from 2016 to 2019. As in Chapter 2, we focus our attention on the breast cancer treatment population. Note that the dataset contains complete and incomplete sequences of actions. Therefore, individuals with treatments which have already started are excluded from this study, however, those that continue their treatments are included. The resulting dataset consists of 645 sequences of actions, whose average length is of 115 actions, the minimum sequence length is 63 and the maximum is 369 (see Figure 3.4 for more details). They are generated by 23 unique medical actions (Table 1.1), whose frequency in patients and their transition frequency are shown in Appendix A.3.

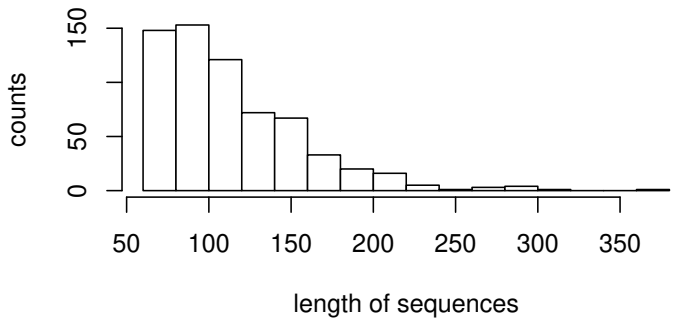


Figure 3.4: Histogram of the lengths of treatments in the EHRs.

3.4.2.2 Hyperparameters

The hyperparameters (classes, minimum stage and maximum stage) of the model are set before the learning procedure. Regarding the class, we use the method developed in Chapter 2 to appropriately pick the number of different classes of treatments and initialize in the same group those treatments with similar trajectories. We obtain a total of 5 classes of treatments and we set the minimum and maximum stages as $r^- = 3$ and $r^+ = 4$ respectively. For the initialization of these stages, the sequences of actions are divided into equal-length intervals of stages.

We replicate the experiment of Section 3.4.1 with the breast cancer dataset. In this case we randomly create the training sets of sizes $N = \{100, 200, 300, 400, 500,\ 600\}$, leaving 45 sequences of actions out to create the test set. Figure 3.5 shows the results of 5 experiments where the generalization curve and the fitting curve of the models converge to the same point. Therefore, we can conclude that the size of the dataset is large enough to learn the generative model, and the hyperparameters chosen beforehand are appropriate for the breast cancer dataset, as well as the smoothing parameter with value 0.2.

3.4.2.3 Analysis of breast cancer treatments

The first application of the generative model is the representation of the evolution of the breast cancer disease, by classifying the different sequences of actions and identifying their multiple phases of progression over time.

Considering the hyperparameters of the previous section and randomly initializing the sequences of stages, we trained the model using the EM-based procedure described in Section 3.3.2. The classification of sequences of actions is carried out by associating each sequence of actions $\boldsymbol{a}$ with the most probable class c^* (Section 3.3.3), that is,

$$c^* = \underset{c}{\operatorname{argmax}}\, p(c|\boldsymbol{a}). \tag{3.13}$$

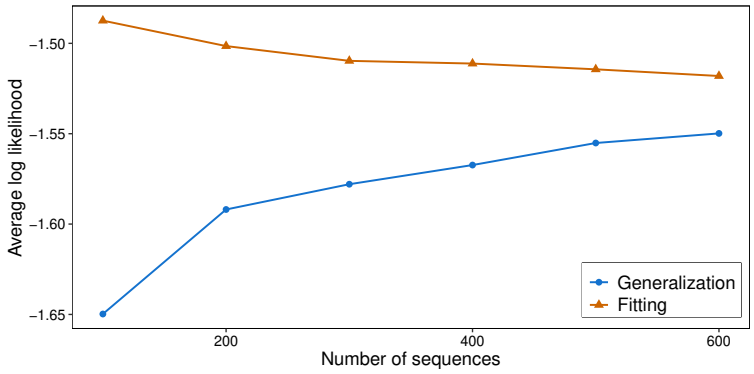


Figure 3.5: Fitting and generalization of the breast cancer generative model.

The evolution patterns of the sequences of actions of each class are characterized by a representative sequence. This is defined as the most probable sequence of actions $\boldsymbol{a}$ within each class (Section 3.3.3) normalized by the length of $\boldsymbol{a}$, in order to avoid the probability $p(\boldsymbol{a}|c)$ to exponentially decrease as long as the length of $\boldsymbol{a}$ increases. That is,

$$\boldsymbol{a}^* = \underset{\boldsymbol{a}}{\operatorname{argmax}} \frac{\log p(\boldsymbol{a}|c)}{|\boldsymbol{a}|}. \tag{3.14}$$

Finally, the sequence of stages associated with the representative sequence $\boldsymbol{a}^*$ is given by the most probable stage at each time point (Section 3.3.3), that is,

$$s_t^* = \underset{s \in S}{\operatorname{argmax}}\, p(s_t = s|\boldsymbol{a}^*, c^*) \tag{3.15}$$

in such a way that the representative sequence of stages associated with the representative sequences of actions $\boldsymbol{a}^*$ is $\boldsymbol{s}^* = (s_1^*, ..., s_m^*)$.

We show in Figure 3.6 the five representative breast cancer treatments (sequences of actions) that characterize the progression classes and stages. The width of the horizontal lines refers to the size of the groups. The vertical lines refer to the medical actions ordered in time. To get a better insight into the behavior of the sequences of actions, we explain the major patterns of the representative treatments, which are real sequences of actions from EHRs, as follows (see Table 3.1).

To begin with, the diagnosis of breast cancer is based on clinical examination in combination with imaging and confirmed by pathological assessment [59]. Every class of treatments in Stage 1 includes this diagnosis process (performed on radiology, nuclear medicine and pathological anatomy medical services), and before any type of treatment is initiated, as recommended.

There exist two types of surgeries when it comes to breast cancer: breast-conserving surgery, in which the surgical team removes the tumor but tries to keep as much of the breast as possible (it is the preferred local treatment option for the majority of early breast cancer patients); or mastectomy, in which the whole breast is removed [59].

The main patterns identified by the model witihin each group are as follows:

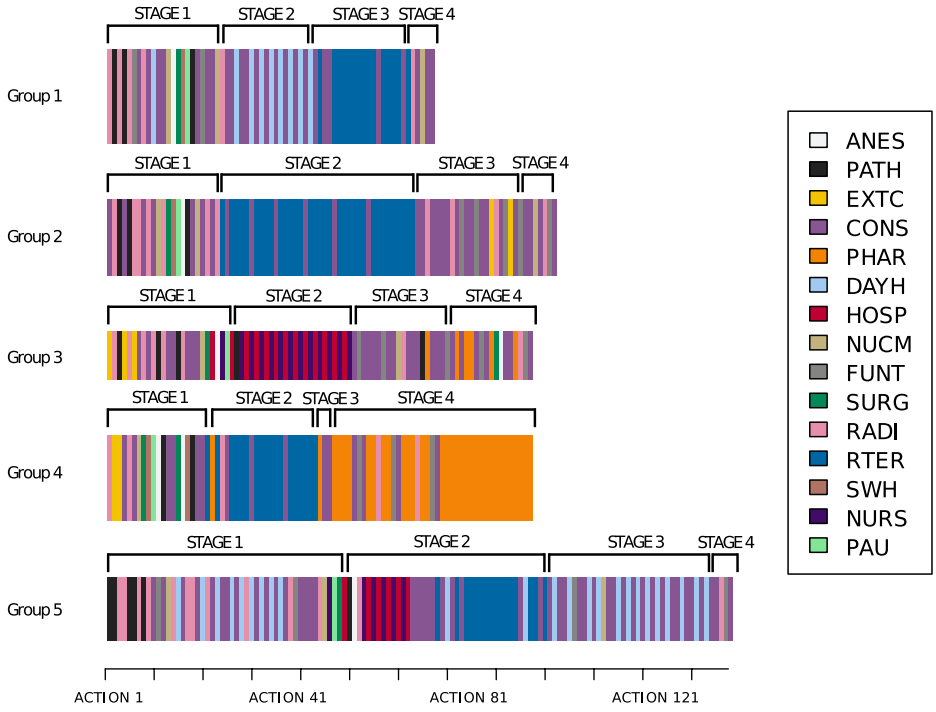


Figure 3.6: Representative treatments of breast cancer segmented in the different phases of evolution.

- **Group 1: Surgery + Chemotherapy + Radiotherapy** (166 patients, 25.7 %). The vast majority of these sequences of actions undergo breast-conserving surgery (Stage 1), followed by chemotherapy (Stage 2) and radiotherapy (Stage 3). According to the guideline suggestions, if both therapies are used, chemotherapy should usually precede radiotherapy, as done here. This type of treatment used after primary treatments, such as surgery, is called adjuvant treatment and its aim is to decrease the chance of cancer recurrence. Some of these patients also include adjuvant hormonal therapy in their Stage 4.

- **Group 2: Surgery + Radiotherapy** (134 patients, 20.7 %). The sequences of actions in this group begin with breast-conserving surgery (Stage 1). This is followed by radiation therapy (Stage 2), which is highly recommended after this type of surgery by the medical guidelines. Regular follow-up actions are given in Stages 3 and 4.

- **Group 3: Surgery + Hospitalization + Hormonal Therapy** (84 patients, 13.1%). This group represents patients undergoing mastectomy (Stage 1). Hospitalization actions (Stage 2) and additional surgical events (Stage 4) are due to breast reconstruction. These patients are followed up with diagnostic tests

	N	STAGE 1	STAGE 2	STAGE 3	STAGE 4
GROUP 1	25.7%	Medical examinations Diagnostic tests Surgery	Chemotherapy	Radiotherapy	Medical examinations Diagnostic tests
GROUP 2	20.7%	Medical examinations Diagnostic tests Surgery	Radiotherapy	Medical examinations Diagnostic tests	Medical examinations
GROUP 3	13.1%	Medical examinations Diagnostic tests Surgery	Hospitalization	Medical examinations Diagnostic tests	Hormonal therapy Surgery
GROUP 4	23.3%	Medical examinations Diagnostic tests Surgery	Radiotherapy	Hormonal therapy	Hormonal therapy Medical examinations Diagnostic tests
GROUP 5	17.2%	Medical examinations Diagnostic tests Chemotherapy Surgery	Radiotherapy Hospitalization	Chemotherapy Diagnostic tests	Medical examinations Diagnostic tests

Table 3.1: Evolution patterns of the breast cancer treatments obtained from the learned generative model.

and physical examinations in Stage 3. Finally, they have hormonal therapy as adjuvant treatment (Stage 4).

- **Group 4: Surgery + Radiotherapy + Hormonal Therapy** (150 patients, 23.3%). Individuals in this group undergo breast-conserving surgery (Stage 1) and postoperative radiotherapy (Stage 2), as suggested. Additionally, they take hormonal therapy as adjuvant systemic treatment (Stage 3) and followed up with clinical examinations (Stage 4).
- **Group 5: Chemotherapy + Surgery + Radiotherapy + Chemotherapy** (111 patients, 17.2%). Neoadjuvant systemic therapy is treatment administered preoperatively to reduce the extent of surgery in locally advanced and large operable cancers. This is the case for this group of patients, who receive neoadjuvant chemotherapy before breast-conservative surgery or mastectomy (Stage 1). Afterwards, they complete their adjuvant treatment with radiotherapy (Stage 2) and chemotherapy (Stage 3). They are followed up in Stage 4.

See Appendix A.3 for more details about the behavior of the medical actions within each class of treatments.

3.5 Discussion

The main contribution of this chapter is the development of a novel probabilistic generative model, which characterizes the progression of the treatment trajectories of a

disease. State-of-the-art disease progression approaches [12–14, 25–28, 30, 36–42] partially adopt the main properties of our model, which we consider essential in order to describe and understand the behavior of the treatment trajectories. In particular, our model simultaneously classifies the heterogeneous sequences of actions based on their treatment evolution over time, segments the sequences of actions in different progression stages of the disease, and captures the sequential dependence between medical actions.

Another contribution of this work is the proposal of an efficient learning process of the parameters of the model to make the computation of the EM algorithm feasible. Exact inference often requires high computational cost for learning, in fact, an *ad hoc* algorithm would require an exponential complexity. We propose a generalization of the forward-backward algorithm for the learning process to reduce this complexity to be polynomial.

Treatment subtyping and phase identification are useful to extract potential information, such as essential or critical treatment behaviors and their causal dependencies in treatment sequences, as well as to understand disease mechanisms and health practices. Apart from classification and segmentation of treatment trajectories, another benefit of our model is the simulation of artificial sequences of actions that resemble original treatments. Then, the model can be regarded as a data augmentation tool when little information is available, for example, for rare diseases. In addition to this, since healthcare datasets are frequently incomplete and the removal of missing values may result in a dataset that is too small or induce statistical bias [1], the model has the ability to impute such missing values in the trajectories of patients or reconstruct incomplete sequences of actions. In terms of interpretability, our model provides easier comprehension and explanation for healthcare professionals than other approaches developed in the healthcare setting [10, 15, 19, 20, 63].

Let us also mention some limitations of our approach. The stages are defined as ordered discrete values of progression and in their evolution only two steps are allowed: to be increased in one stage with respect to the previous stage; or be maintained in the same one. In a more realistic scenario, diseases with recurrent stages would be considered, and, consequently, the sequences of actions could pass through the same stage more than once or move from one stage to another without setting an ordered progression. However, this assumption requires a modification in the dynamic programming procedure that would exponentially increase the complexity of the model. On the other hand, as in many other classification machine learning methods, the number of classes is not a flexible parameter and has to be chosen beforehand. Despite this, we solved this problem by initializing the classes of treatments with the clustering of sequences outlined in Chapter 2, where the number of classes that best fits the data was selected. For the minimum and maximum stages, we could estimate their value by including them in the learning process of the model, assuming again an increase in its complexity.

Finally, addressing the irregular timing between medical actions is crucial for assessing a patient's health condition. In fact, temporal patterns can reveal important insights into disease progression. From a clinical perspective, it can lead to more efficient management, better personalized patient care and more accurate predictions [1].

Chapter 4 proposes an extension of this generative model by including the modeling of the irregular temporal gaps between medical events.

3.6 Conclusion

This chapter proposes a probabilistic generative model to capture the treatment variability of a disease and its progression. The generative model is defined as a mixture of models for sequences of medical events. These models incorporate a latent variable representing the progression stage, capturing the underlying dynamics of the medical events. We efficiently learn the model using the EM algorithm and a dynamic programming-based method. The proposed model enables to identify subtypes of treatments for a disease, determine the stage of progression of treatments, and simulate new treatment trajectories.

We demonstrate the potential of our generative model as a treatment classification and stage identification tool in breast cancer patients. We further validate the proposed learning process by a simulation experiment, where the original model is recovered. Note that the proposed approach can be applied to any progressive disease, such as other types of cancers, respiratory diseases or neurodegenerative diseases.

Chapter 4

Time-dependent probabilistic generative models for disease progression

4.1 Introduction

EHRs contain a large amount of essential information for monitoring patients' health status throughout their clinical history. The temporal component of EHRs, which collects the sequence of medical events in the healthcare system over time, is important for understanding patients' treatment trajectories and identifying patterns in them. In contrast to other types of time series data with regularly recorded observations, EHRs exhibit irregular time intervals between patients' visits [29]. This requires the development of models that effectively handle the variability and irregularities inherent in EHRs [1].

Chapter 3 presents a probabilistic generative approach for modeling both subtypes of treatments and their progression over time, considering regularly observed medical events. However, learning the irregular time intervals between patients' visits would achieve a more accurate representation of disease dynamics. This modeling process is essential to provide a deeper understanding of the diverse temporal characteristics that may exist depending on the subtype of treatment the patient is undergoing.

Recently, deep learning techniques have been introduced to predict specific outcomes based on the progression of a disease [10,11,21], with high prediction accuracy in future events but often overlooking the irregular temporality inherent in EHRs. While some methods have incorporated the irregular time information in their models [10, 13, 14, 16, 21], they rarely focus on estimating the time intervals between consecutive medical events. Moreover, their lack of interpretability makes challenging the understanding of the underlying temporal dynamics of diseases. Consequently, there is a need for more interpretable models in the context of time-dependent disease progression [21, 22].

In this regard, probabilistic generative models enable to make representations of

the temporal progression within sequences of medical events through parametric modeling, resulting in more interpretable outcomes for healthcare professionals. In the literature, various adaptations of Markov models have been used to capture disease state transitions and model the temporal progression of diseases [30, 35, 36, 38–40, 42]. However, some of these approaches primarily focus on modeling the time intervals between hidden variables and do not explicitly address the time elapsed between observed events [30, 39, 40]. The latter consideration is critical for accurately estimating the time between consecutive medical events in real-world scenarios, and therefore, for estimating the temporal progression of an entire treatment.

This chapter presents an extension of the probabilistic generative model introduced in Chapter 3. This method employs a latent class of treatments to categorize sequences of medical events into different subtypes and a latent sequence of stages to segment the sequence of events into subsequences of progression patterns. One of the key contributions of the present work is the incorporation of the time elapsed between medical actions within the disease treatment trajectory. With this approach, we aim to achieve the following objectives: (i) model the irregular time intervals between medical events; (ii) discover the different subtypes of disease progression in terms of the sequence of medical events and the time elapsed between them; and (iii) segment the sequences into progression patterns of treatments.

The main contributions of this chapter are as follows:

- We propose a probabilistic generative model based on Markov models that incorporates temporal information between medical events to model the underlying dynamics of disease treatments. Our model is flexible in terms of time distribution, allowing for the incorporation of the most appropriate distribution based on the available data. Specifically, we propose three parametric distributions to effectively model the irregular time intervals between medical actions: the geometric, exponential, and Weibull distributions.
- The model includes a class of treatments, which is a hidden variable that enables the grouping of patients. While in Chapter 3 the class is based on the sequence of medical events, in this chapter the class also has influence on the time intervals between these events. Additionally, it incorporates a hidden sequence of progression stages, which segments treatments into distinct patterns of evolution. To efficiently learn the parameters of our generative model, we use the EM algorithm [45] with a dynamic programming-based method.
- We demonstrate the effectiveness of our approach in uncovering the underlying data distribution, predicting the irregular timing between medical events, and classifying treatments into different subtypes using synthetic and real data (Section 1.1).

The remainder of the chapter is organized as follows: Section 4.2 introduces the problem formulation, Section 4.3 presents our proposed generative model and describes the methodology in detail. Section 4.4 presents the experimental setup and the results.

Section 4.5 discusses the contributions and limitations of our approach. Finally, Section 4.6 draws the conclusions.

4.2 Problem formulation

A patient's treatment associated with a disease, denoted by $\boldsymbol{a}$, is a sequence of medical actions collected during repeated hospital visits. Let A be the set of medical specialties (for instance, radiology, radiotherapy, hospitalization, etc), we define a patient's treatment as

$$\boldsymbol{a} = (a_1, ..., a_m)$$

where $a_i \in A$ represents the i-th medical action of a patient. Each sequence of medical actions has an associated sequence of time intervals,

$$\boldsymbol{\tau} = (\tau_1, ..., \tau_m)$$

where $\tau_i \in \mathbb{R}$ is the time interval between a_{i-1} and a_i, for $i = 2, ...m$. We initialize τ_1 as 0 to indicate the starting point of the treatment.

Given a dataset of medical records, the objective is to develop a probabilistic generative model to effectively capture the temporal dynamics of the disease and the variability in treatment patterns.

4.3 Methodology

This section describes the proposed probabilistic generative model and the learning procedure of the model.

4.3.1 Model definition

Based on the proposed model in Chapter 3, the time-dependent generative model is also built on Markovian assumptions and considers that a sequence of actions has a structure of latent variables. These latent variables include the classes of treatments, which identify similar subtypes of treatments, and the stages, which segment each treatment into different progression patterns. We assume that all sequences of actions begin in the first stage, representing the initial steps of the treatment, and all classes of treatments have an equal number of stages. These stages segment the sequences within each class of treatments into subsequences that are associated with their progression patterns. As in Chapter 3, the same stage values from different classes of treatments represent different subsequences, which allows the model to be more flexible. Our primary contribution lies in expanding this model to include the irregular timing between consecutive medical actions, assuming that this timing varies depending on the latent class of treatment.

To define the time-dependent generative model, consider $\boldsymbol{a} = (a_1, ..., a_m)$ as the treatment sequence associated with a disease, where $a_i \in A$, and let $\boldsymbol{\tau} = (\tau_1, ..., \tau_m)$ represent the corresponding sequence of time intervals, where $\tau_i \in \mathbb{R}$. Let $\boldsymbol{s} = (s_1, ..., s_m)$ denote the sequence of latent stages associated with $\boldsymbol{a}$. The stages, denoted as s_i, belong to a set $S = \{1, ..., r\}$ that represents all the possible progression stages of a treatment. Finally, let c be the latent class of treatments which $\boldsymbol{a}$ belongs to. The class of treatments c belongs to a set $C = \{1, ..., k\}$ that represents all the possible classes, corresponding to distinct subtypes of treatments for a specific disease.

It is assumed that the progression stages are non-decreasing, implying that a sequence cannot go backward. Thus, for any given time point $i = 1, ..., m-1$, we have $s_i \leq s_{i+1}$. This assumption guarantees that the treatment moves forward without skipping any stage.

The proposal for the time-dependent probabilistic generative model is as follows (see Figure 4.1):

a) Draw a class of treatments $c \sim Mult(\boldsymbol{\theta}_C)$

b) Draw the initial medical action and the initial stage

$$a_1|c \sim \ Cat(\boldsymbol{\pi}_A^c), \quad s_1|a_1, c \sim \ Cat(\boldsymbol{\pi}_S^{a_1,c}).$$

c) For each timestamp index i:

i) Draw a medical action from $p(a_i|a_{i-1}, s_{i-1}, c)$, that is,

$$a_i|a_{i-1}, s_{i-1}, c \sim Cat(\boldsymbol{\theta}_A^{a_{i-1},s_{i-1},c})$$

ii) Draw a stage s_i from $p(s_i|a_i, s_{i-1}, c)$,

$$s_i|a_i, s_{i-1}, c \sim Cat(\boldsymbol{\theta}_S^{a_i,s_{i-1},c})$$

iii) Draw the time interval from $p(\tau_i|a_{i-1}, a_i, c)$, that is,

$$\tau_i|a_{i-1}, a_i, c \sim F_T(\boldsymbol{\theta}_T^{a_{i-1},a_i,c})$$

The time-dependent generative model provides flexibility in capturing the time intervals between pairs of medical actions by utilizing an appropriate parametric distribution $F_T(\boldsymbol{\theta}_T^{a,a',c})$. It assumes that the time intervals depend on the latent class of treatments and pairs of actions, but not on the stage of progression.

Translating the generative process into a joint probability model results in the expression:

$$p(\boldsymbol{a}, \boldsymbol{\tau}, \boldsymbol{s}, c) = p(c) \prod_{i=1}^{m} p(a_i, s_i|a_{i-1}, s_{i-1}, c) \cdot p(\tau_i|a_{i-1}, a_i, c), \tag{4.1}$$

where

$$p(a_i, s_i|a_{i-1}, s_{i-1}, c) = p(a_i|a_{i-1}, s_{i-1}, c) \cdot p(s_i|a_i, s_{i-1}, c)$$

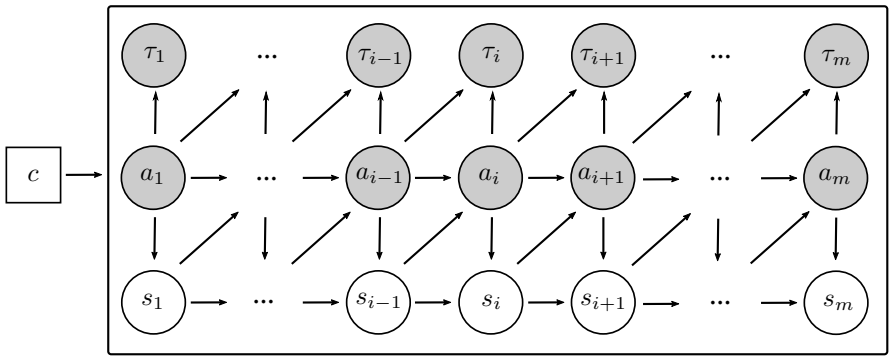


Figure 4.1: Probabilistic generative model defined by the conditional distributions $p(a_i|a_{i-1}, s_{i-1}, c)$, $p(s_i|a_i, s_{i-1}, c)$ and $p(\tau_i|a_{i-1}, a_i, c)$ for sequences of actions $\boldsymbol{a}$, sequences of time intervals $\boldsymbol{\tau}$, latent sequences of stages $\boldsymbol{s}$ and latent classes c. The gray figures represent the observed variables.

and $p(a_1, s_1|a_0, s_0, c) = p(a_1, s_1|c)$. Furthermore, $s_1 = 1$, $a_m = end$, and $s_{i-1} \leq s_i$ for all $i = 2, \ldots, m$.

We use a Markov model to generate actions based on the previous action and stage in the sequence, and another Markov model to generate stages based on the previous stage and current action. These dependencies allow to maintain the consistency of the sequences of events over time. The distributions $F(\boldsymbol{\theta}_T)$ that we consider are the geometric, exponential and Weibull distributions.

The parameters of the initial model for medical actions and stages are denoted as $\boldsymbol{\pi}_A^c$ and $\boldsymbol{\pi}_S^{a,c}$, respectively. Our goal is to estimate the parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_A, \boldsymbol{\theta}_T, \boldsymbol{\theta}_S, \boldsymbol{\theta}_C, \boldsymbol{\pi}_A, \boldsymbol{\pi}_S\}$ to capture the underlying dynamics and distributions in the data.

4.3.2 Maximum likelihood parameter estimation

This section introduces the procedure for learning the model parameters. Let $\mathcal{D} = \{(\boldsymbol{a}^i, \boldsymbol{\tau}^i)\}_{i=1}^N$ be the set of observed sequences of medical actions and time intervals, let C be the set of latent classes of treatments and S the set of latent stages of progression. We use the EM algorithm [46] to obtain the maximum likelihood estimate of the model's parameters in the presence of the latent variables, that is, the treatment classes and progression stages. Note that we are not given the complete data set, meaning that for each observation in $\mathcal{D}$ we lack the corresponding values of the latent variables. Therefore, we will instead consider the expected value of the log likelihood for the complete dataset under the posterior distribution of the latent variables, denoted as $p(\boldsymbol{s}, c|\mathbf{a}, \boldsymbol{\tau})$. This involves considering all possible configurations for the hidden variables to solve the following maximization:

$$\max_{\boldsymbol{\theta}} \sum_{(\boldsymbol{a},\boldsymbol{\tau})\in D} \sum_{\boldsymbol{s}\in\mathcal{S}_{\boldsymbol{a}}} \sum_{c\in C} p(\boldsymbol{s}, c|\boldsymbol{a}, \boldsymbol{\tau}) \cdot \log p(\boldsymbol{a}, \boldsymbol{\tau}, \boldsymbol{s}, c; \boldsymbol{\theta}), \tag{4.2}$$

where $\mathcal{S}_{\boldsymbol{a}}$ is the set of the all possible configurations of sequences of stages for $\boldsymbol{a}$. Note that every pair $(\boldsymbol{a}, \boldsymbol{\tau}) \in \mathcal{D}$ contributes equally to the model regardless of its length due to

$$\sum_{\substack{c \in C \\ \boldsymbol{s} \in \mathcal{S}_{\boldsymbol{a}}}} p(\boldsymbol{s}, c|\boldsymbol{a}, \boldsymbol{\tau}) = \sum_{\substack{c \in C \\ \boldsymbol{s} \in \mathcal{S}_{\boldsymbol{a}}}} p(\boldsymbol{s}|c, \boldsymbol{a}, \boldsymbol{\tau}) \cdot p(c|\boldsymbol{a}, \boldsymbol{\tau}) = 1. \tag{4.3}$$

The EM algorithm allows to efficiently find the parameters that maximize the log-likelihood following the subsequent iterative process:

E-step. In this step, we calculate the posterior distribution of the latent variables given the observed data, that is, $p(\boldsymbol{s}, c|\boldsymbol{a}, \boldsymbol{\tau})$. Then we use this posterior distribution to evaluate the expectation of the complete-data log-likelihood function as a function of the parameters $\boldsymbol{\theta}$ (Equation (4.2)). The efficient learning procedure of these posterior distributions is similar to the dynamic programming-based method described in Section 3.3.2.1. Appendix B.1 shows the adaptation of this method to the specific temporal characteristics of this time-dependent generative model.

M-step. In the maximization step, we maximize the Equation (4.2) using the posterior distributions computed in the E-step. This maximization is achieved using the Lagrange multiplier method, similar to that in Appendix A.1. If $\theta_{a'}^{a,s,c}, \theta_{s'}^{a',s,c}$ denote a component in $\boldsymbol{\theta}_A^{a,s,c}$, $\boldsymbol{\theta}_S^{a',s,c}$, respectively, the model parameters corresponding to the transition from the pair (a, s) to (a', s') given the class c, where $a, a' \in A$ and $s, s' \in S$ are updated as follows:

$$\theta_{a'}^{a,s,c} = \frac{\sum_{(\boldsymbol{a},\boldsymbol{\tau}) \in \mathcal{D}} \sum_{i=1}^{m_{\boldsymbol{a}}} \mathbb{1}_{a,a'}(a_{i-1}, a_i) \cdot p(s_{i-1} = s|c, \boldsymbol{a}, \boldsymbol{\tau})}{\sum_{a' \in A} \sum_{(\boldsymbol{a},\boldsymbol{\tau}) \in \mathcal{D}} \sum_{i=1}^{m_{\boldsymbol{a}}} \mathbb{1}_{a,a'}(a_{i-1}, a_i) \cdot p(s_{i-1} = s|c, \boldsymbol{a}, \boldsymbol{\tau})} \tag{4.4}$$

where

$$\mathbb{1}_{a,a'}(a_{i-1}, a_i) = \begin{cases} 1 & \text{if } a_{i-1} = a, a_i = a' \\ 0 & \text{otherwise.} \end{cases}$$

$$\theta_{s'}^{a',s,c} = \frac{\sum_{(\boldsymbol{a},\boldsymbol{\tau}) \in \mathcal{D}} \sum_{i=1}^{m_{\boldsymbol{a}}} \mathbb{1}_{a'}(a_i) \cdot p(s_{i-1} = s, s_i = s'|c, \boldsymbol{a}, \boldsymbol{\tau})}{\sum_{s' \in S} \sum_{(\boldsymbol{a},\boldsymbol{\tau}) \in \mathcal{D}} \sum_{i=1}^{m_{\boldsymbol{a}}} \mathbb{1}_{a'}(a_i) \cdot p(s_{i-1} = s, s_i = s'|c, \boldsymbol{a}, \boldsymbol{\tau})} \tag{4.5}$$

where

$$\mathbb{1}_{a'}(a_i) = \begin{cases} 1 & \text{if } a_i = a' \\ 0 & \text{otherwise.} \end{cases}$$

If θ_c denotes a component in $\boldsymbol{\theta}_C$, the probability of the classes of treatments $c \in C$ is updated as

$$\theta_c = \frac{\sum_{(\boldsymbol{a},\boldsymbol{\tau}) \in \mathcal{D}} p(c|\boldsymbol{a}, \boldsymbol{\tau})}{\sum_{c \in C} \sum_{(\boldsymbol{a},\boldsymbol{\tau}) \in \mathcal{D}} p(c|\boldsymbol{a}, \boldsymbol{\tau})}. \tag{4.6}$$

As mentioned earlier, various distributions, such as geometric, exponential, or Weibull, can be used to model the time interval between each pair of actions within each class. For each transition from a to a' given a class c, the parameters of the geometric distribution are updated as follows:

$$\theta_T^{a,a',c} = \sum_{(\boldsymbol{a},\boldsymbol{\tau})\in\mathcal{D}} \sum_{i=1}^{m_{\boldsymbol{a}}} \frac{n_{\boldsymbol{a}}}{\tau_i \cdot \mathbb{1}_{a,a'}(a_{i-1}, a_i) \cdot p(c|\boldsymbol{a},\boldsymbol{\tau}) + n_{\boldsymbol{a}}} \tag{4.7}$$

where $n_{\boldsymbol{a}} = \mathbb{1}_{a,a'}(a_{i-1}, a_i) \cdot p(c|\boldsymbol{a},\boldsymbol{\tau})$.

For the exponential distribution, which is the continuous analogue of the geometric distribution,

$$\theta_T^{a,a',c} = \sum_{(\boldsymbol{a},\boldsymbol{\tau})\in\mathcal{D}} \sum_{i=1}^{m_{\boldsymbol{a}}} \frac{n_{\boldsymbol{a}}}{\tau_i \cdot \mathbb{1}_{a,a'}(a_{i-1}, a_i) \cdot p(c|\boldsymbol{a},\boldsymbol{\tau})}$$

Finally, due to the absence of a closed-form solution for the maximum likelihood estimation of the Weibull distribution, it is necessary to employ numerical optimization methods to estimate the parameters (see [64] for more details).

At each iteration of the algorithm, we combine the expectation and maximization steps for each $(\boldsymbol{a},\boldsymbol{\tau}) \in \mathcal{D}$ without the need to store the exponential number of probabilities for all configurations of sequences of stages and classes. Additionally, using a dynamic programming-based method (Apprendix B.1) enables the EM algorithm to be solved while considering the exponential number of sequences of stages, with a computational complexity of $O(N \cdot m^2)$, where m represents the length of the longest sequence of actions.

To simplify the notation and provide a clearer understanding of the model's main idea, we establish a fixed number of stages for all classes of treatments. Nevertheless, a more adaptable model can be defined to accommodate varying numbers of stages. With this flexibility, the model can segment complete sequences into the maximum number of stages, denoted as r^+, while also handling incomplete sequences by using a reduced number of stages, ranging from r^- to r^+.

4.4 Experimental results

This section presents the results obtained from a series of experiments conducted on both synthetic data and real-world data. Firstly, the experiments using synthetic data demonstrate the capability of our learning procedure to achieve a close approximation of the original generative model. Secondly, the experiments conducted on breast cancer patients show the applicability of the proposed model in gaining insights into the varying time intervals between consecutive medical records, as well as in the unsupervised classification of the treatments. The source code of the probabilistic generative model is publicly available[1].

[1]https://github.com/onintzezaballa/TimeDependentDiseaseProgressionModel

4.4.1 Results on synthetic data

In this experiment, we demonstrate the learning performance of the proposed procedure concerning the number of training samples in practical scenarios. To do so, we use a set of artificially generated treatments derived from a randomly generated model.

First, we create a probabilistic generative model $p_{\boldsymbol{\theta}}$, where the model's parameters are generated using the following procedure: $\boldsymbol{\theta}_C$ is sampled from a uniform Dirichlet distribution with parameters $\alpha = 1$; similarly, $\boldsymbol{\theta}_A^{a,s,c}$ is sampled from a uniform Dirichlet distribution with parameters $\alpha = 1$ for each $a \in A$, $s \in S$ and $c \in C$; additionally, $\boldsymbol{\theta}_S^{a,s,c}$ is sampled from a Dirichlet distribution setting $\alpha = 0.7$ for the parameters corresponding to transitions that remain in the same stage ($s' = s$) and setting $\alpha = 0.3$ for the parameters related to transitions progressing to a different stage ($s' \neq s$), for $a \in A$, $s, s' \in S$ and $c \in C$. The reason for setting a lower value when the transition progresses to a different stage is to generate more realistic sequences, avoiding excessively short subsequences of stages.

This experiment is repeated for each time distribution: geometric, exponential, and Weibull distributions. The parameters for the geometric distribution are sampled from a Beta(5,2) distribution, for the exponential distribution they are sampled from a Gamma(2,1) distribution, and for the Weibull distribution, the shape parameters are sampled from $\mathcal{U}(2, 5)$, and the scale parameters are sampled from $\mathcal{U}(1, 1.5)$.

For the sake of simplicity, we set a fixed total number of classes, $|C| = 2$, and define a range of stages from a minimum of $r^- = 3$ to a maximum of $r^+ = 4$. These models allow us to generate training sets of various sizes, specifically $N = \{300, 500, 800, 1000, 1200, 1500, 2000, 3000\}$, using the randomly generated model $p_{\boldsymbol{\theta}}$. We consider a set of 10 unique actions to create these sequences. Additionally, we sample a test set of 4000 sequences from $p_{\boldsymbol{\theta}}$ to evaluate the learning process.

To demonstrate that the learning algorithm can provide a good approximation of the original model with realistic training set sizes, we employ the EM-based procedure proposed in Section 4.3.2 to fit the model on the training sets. For the EM initialization, we divide the observed sequences of actions into equal-length stage intervals. The initial parameters for the time distribution are uniform across all classes and are estimated with the observed time intervals between actions. For the initial class model, we initialize the probability of each sequence belonging to each class of treatments with the uniform distribution. We then add a probability $\epsilon = 0.1$ to the true class to which they belong to prevent relabeling in the results. After learning the model, we analyze the evolution of the method's quality as the size of the training set, $n \in N$, increases. For each value of n, we obtain a new model $\boldsymbol{\theta}^n = \{\boldsymbol{\theta}_A^n, \boldsymbol{\theta}_T^n, \boldsymbol{\theta}_S^n, \boldsymbol{\theta}_C^n\}$ and assess its quality by computing the log likelihood of Equation (4.2) normalized by n, making the datasets of different sizes comparable.

The experiment is conducted five times for each time distribution, with each experiment considering a different random generative model, denoted as $p_{\boldsymbol{\theta}}$, from which the training sets and test sets are generated. Figure 4.2 shows the fitting and generalization capabilities of our models by presenting the average log likelihood for the three

time distributions. The solid orange line represents the average log likelihood of the learned models on the training sets, indicating how well the models fit the data. On the other hand, the solid blue lines represent the average log likelihood of the learned models on the test set, showing their ability to generalize to unseen data. The dotted lines correspond to the average log likelihood of the original generative models, with the orange line representing the training dataset and the blue line representing the test dataset. As we can see in Figure 4.2, as $n \in N$ increases, the curves representing the fitting and generalization of the learned models converge to the curves of the original generative models. This convergence indicates that, given a sufficiently large dataset, the proposed learning algorithm successfully recovers the original generative model that underlies the data.

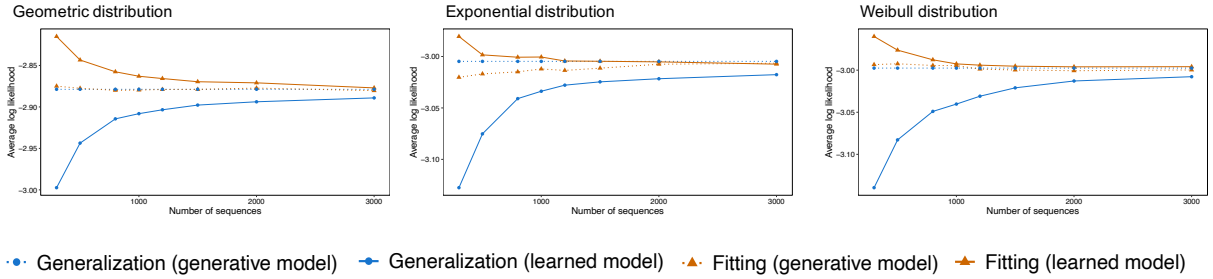


Figure 4.2: Synthetic data results for different time distributions.

4.4.2 Results on real-world data

This section shows the utility of the generative model in real EHRs. We use the generative model in two different applications: for time interval prediction and for treatment classification.

4.4.2.1 Dataset

We validate the model on the EHRs described in Section 1.1, which stores every outpatient and hospital visit of patients from 2016 to 2019. As a use case, we focus our attention on the breast cancer population, which comprises 645 patients. Their treatments average 115 medical actions, and they are generated by 23 unique medical specialties (selected following the procedure in Chapter 2). In total, there are 73150 transitions between pairs of actions, with a mean time interval of 10 days and a standard deviation of 31 days.

4.4.2.2 Time prediction performance

The goal of this experiment is to determine which parametric model provides better predictions for the time intervals between medical actions. To achieve this, our objec-

tive is to estimate the time interval until the next medical action as time progresses.

Experiment setup: We use a cross-validation approach to assess the predictive performance of the generative model. Following the results obtained in Chapter 2 and Chapter 3, we consider 5 classes of treatments, with a minimum of 3 stages and a maximum of 4 stages for each treatment. In all training models, including the baselines, we use 90% of the patients as the training set and 10% as the test set.

We train the models using the three time distributions: geometric, exponential, and Weibull. The initial parameters for the stages and time distributions are the same as in the synthetic experiments. However, for the initial class model, we use the K-medoids method proposed in Chapter 2 for real-world data. Subsequently, we make predictions for each time step by sampling a set of time intervals from the learned generative model and using their median as the prediction for that time step. Let $\boldsymbol{a}_t = (a_1, \ldots, a_t)$ be the observed subsequence of actions up to time step t, and $\boldsymbol{\tau}_t = (\tau_1, \ldots, \tau_t)$ the observed subsequence of time intervals up to time step t. We define $q_t(c)$ as the probability distribution of classes given the subsequence of actions $\boldsymbol{a}_t$ and the subsequence of time intervals $\boldsymbol{\tau}_t$, in such a way that $q_t(c)$ changes as time progresses:

$$q_t(c) = p(c|\boldsymbol{a}_t, \boldsymbol{\tau}_t).$$

We estimate the time interval between medical actions, $\hat{\tau}_{t+1}$ for $t = 2, \ldots, m$, by sampling time intervals from the generative model in the following two ways:

(a) Using the mixture of classes of treatments of the model,

$$\sum_{c \in C} q_t(c) \cdot p(\tau_{t+1}|a_t, a_{t+1}, c) \tag{4.8}$$

(b) Using the class of treatments of maximum probability,

$$p(\tau_{t+1}|a_t, a_{t+1}, c^*), \quad c^* = \underset{c}{\operatorname{argmax}}\, q_t(c) \tag{4.9}$$

The final prediction of the time interval $\hat{\tau}_{t+1}$ is given by the median of the samples obtained using Equations (4.8) and (4.9).

Evaluation metrics: We evaluate the prediction error using the mean absolute error, that is, $|\tau - \hat{\tau}|$.

Baselines: On the one hand, we use parametric and non-parametric approaches to make predictions of the time interval until the next medical action. In the parametric approaches, we fit the data to geometric, exponential and Weibull distributions, using $p(\tau|a, a')$ to estimate the time intervals. In the non-parametric approach, we predict the time using the median of the observed time intervals between each pair of medical actions. On the other hand, we compare our model against the one proposed in

Chapter 3. Since this model is not time-dependent, we first learn the generative model and then fit the geometric, exponential, and Weibull distributions to the training data as described in Section 4.3.2. We then use both the mixture of classes of the model (Equation (4.8)) and the class of maximum probability (Equation (4.9)) to sample time intervals and make the prediction with the median of these samples.

Prediction performance: Table 4.1 compares the results from various algorithms, confirming that our proposed approach outperforms baseline models in the parametric setting. Specifically, predictions using the Weibull distribution show the lowest mean absolute error among these models. For more details on errors made by different approaches when predicting the most frequent pairs of actions, refer to Apprendix B.2. In summary, we can conclude that the Weibull distribution performs better than other parametric approaches, and our time-dependent model enhances prediction accuracy for irregular time intervals.

	Parametric			**Non-param.**
	Geometric	**Exponential**	**Weibull**	**Median**
Empirical	16.36	17.06	18.03	3.86
Model in Chapter 3 (mixture)	4.64	4.62	4.24	
Model in Chapter 3 (argmax)	4.54	4.55	4.17	
Proposed model (mixture)	4.45	4.89	**4.12**	
Proposed model (argmax)	4.57	5.21	4.25	

Table 4.1: Mean absolute error in predicting the time interval until the next medical action.

4.4.2.3 Treatment classification

In this second experiment, we aim to explore the impact of incorporating time modeling on the representation of treatment subtypes. Using the same hyperparameters as in the previous section, we trained the model using the EM-based procedure described in Section 4.3.2. The classification of treatments is carried out by associating each sequence of actions $\boldsymbol{a}$ and its corresponding $\boldsymbol{\tau}$ with the most probable class c^*, that is,

$$c^* = \underset{c}{\operatorname{argmax}}\, p(c|\boldsymbol{a}, \boldsymbol{\tau}). \tag{4.10}$$

The dynamics of the sequences of actions of each class are characterized by a representative sequence. This is defined as the most probable pair $(\boldsymbol{a}, \boldsymbol{\tau})$ within each class normalized by the length of $\boldsymbol{a}$, in order to avoid the probability $p(\boldsymbol{a}, \boldsymbol{\tau}|c)$ to exponentially decrease as long as the length of $\boldsymbol{a}$ increases. That is,

$$\boldsymbol{a}^* = \underset{\boldsymbol{a}}{\operatorname{argmax}}\, \frac{\log p(\boldsymbol{a}, \boldsymbol{\tau}|c)}{|\boldsymbol{a}|}. \tag{4.11}$$

Figure 4.3 presents the five representative breast cancer treatments obtained using the Weibull distribution, which is the distribution with the best results in the previous experiment. These treatments characterize different progression subtypes. Figure 4.4 shows the same results as in Figure 4.3 without displaying the time intervals between medical events (*No event*).

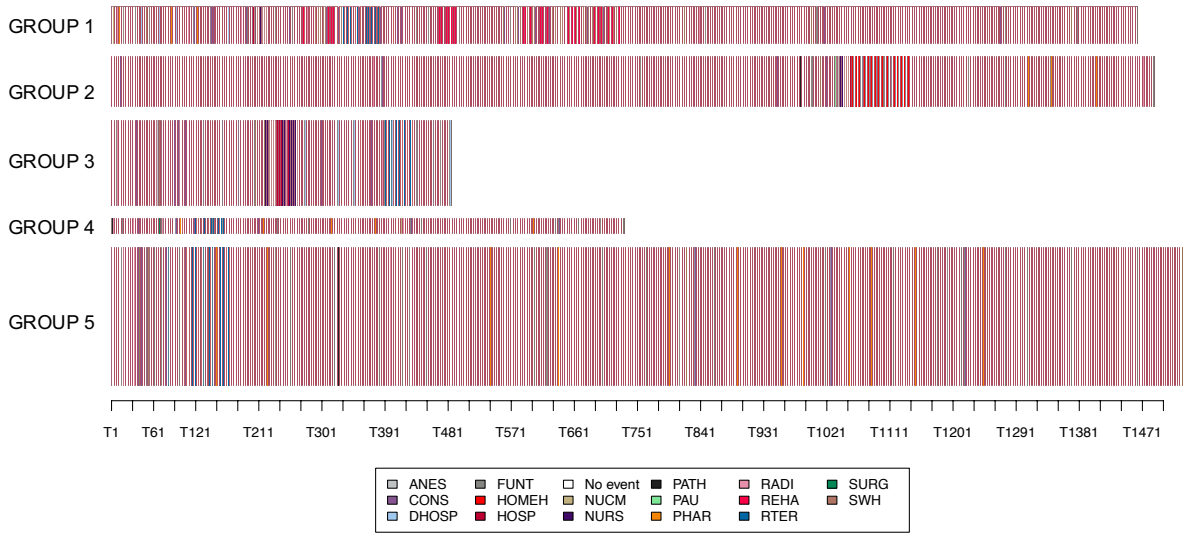


Figure 4.3: Classification results for treatments associated with breast cancer considering the time between medical events. See Table 1.1 for the description of the medical actions.

The major patterns of the representative treatments, which consists of real sequences of actions from EHRs, are as follows:

- **Group 1.** Chemotherapy + Surgery + Hospitalization + Radiotherapy + Rehabilitation (11.3 %)
- **Group 2.** Surgery + Hospitalization + Home hospitalization + Hormonotherapy (18.2 %)
- **Group 3.** Surgery + Chemotherapy + Hospitalization + Radiotherapy (24%)
- **Group 4.** Surgery + Radiotherapy + Hormonotherapy (5%)
- **Group 5.** Surgery + Radiotherapy + Hormonotherapy (41.5%)

Figures 4.3 and 4.4 show that all the treatments start with the diagnosis process (conducted through radiology, nuclear medicine and pathological anatomy medical services). After receiving the specific therapy for each group, patients undergo regular

follow-up consultations and medical tests. Note that Group 4 and Group 5 seem to be similar subtypes, however, their primary distinction lies in the longer duration of the treatment for patients in Group 5. All these findings related to the treatment patterns and their duration align with clinical practice guidelines [59].

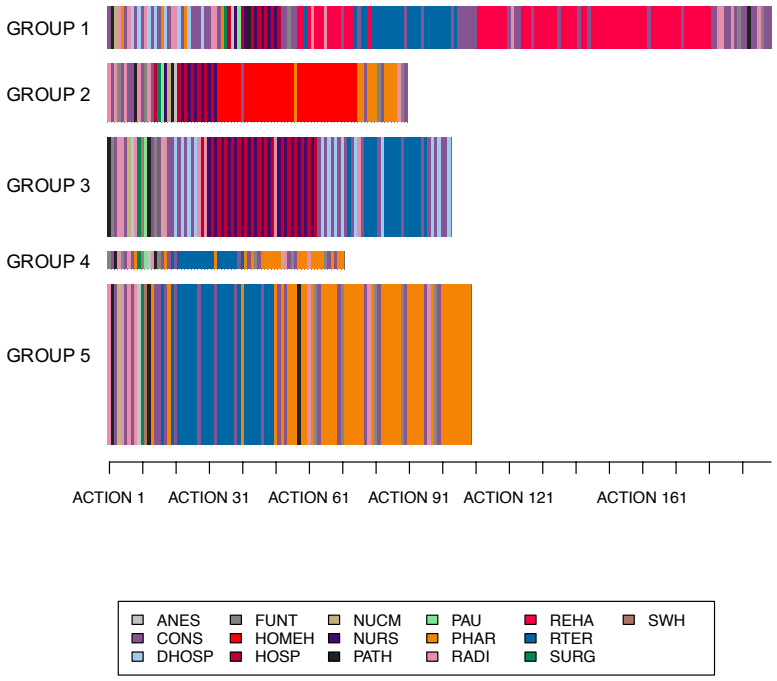


Figure 4.4: Classification results for treatments associated with breast cancer without representing the time intervals between the medical actions.

4.5 Discussion

This chapter proposes a probabilistic generative model that incorporates temporal information between medical events to model the underlying dynamics of disease treatments. This model is flexible in terms of time distribution, enabling the adoption of the most suitable distribution for the available data. Specifically, we propose three parametric distributions to effectively model the irregular time intervals between medical actions: the geometric, exponential, and Weibull distributions. The model includes a latent class variable, which makes the time modeling a mixture of these parametric distributions.

Unlike existing disease progression models [10, 13, 21, 30, 40], this is the first generative model of sequences that primarily aims to comprehend the temporal evolution of a disease, taking into account the temporal irregularities between observed medical

events. Our approach provides interpretable representations of the temporal progression within sequences of actions through parametric modeling, by simultaneously capturing both disease stage transitions and distinct disease subtypes. We would like to emphasize that the main focus of this model is on learning the underlying distribution of a set of sequences of medical events. By capturing the temporal dynamics of these sequences, we open up a wide range of potential applications, including the prediction of medical variables, treatment classification, and the generation of new treatments, as demonstrated in our experiments.

This identification of representative treatments is shown in Figure 4.3. Figure 4.4 offers a more interpretable view of these results in terms of treatment patterns, displaying the same outcomes as Figure 4.3 but without showing the time intervals. Comparing these results with the representative sequences of actions obtained using the model developed in Chapter 3, which does not consider the temporal component, we can identify several similarities. For instance, we can observe that the treatment patterns in Group 5, obtained from the time-dependent model, match those in Group 4, obtained using the model in Chapter 3, although the proportion of patients assigned to these groups is different. Similarly, Group 2 from the time-dependent model and Group 3 from the model in Chapter 3 are also similar, with the exception that patients in Group 2 receive home hospitalization. However, there are slight variations in the remaining treatments between the two models.

The model proposed in this chapter significantly outperforms the parametric baselines in predicting time intervals between medical events, as shown in Table 4.1. These results highlight the importance of considering treatment classes for modeling the irregular time gaps within sequences of actions. The second set of experiments uses a modification of the model presented in Chapter 3. This model originally does not consider time information, however, to be able to compare our model with a baseline, we introduced time interval estimation after the original model was already learned. Note that the structure of both models is similar in terms of classes and stages, which may explain their similar predictive results. However, our proposed model is able to slightly improve the predictive results by jointly learning the time intervals and latent variables, and provides a more informative representation of data in terms of medical actions and the treatment duration.

The more accurate predictive performance of the non-parametric method in Table 4.1 can be attributed to the robustness of the median when handling extreme time interval values that deviate significantly from the mean. Our proposed parametric probability distributions are more sensitive to these outliers and may not adequately approximate to these extreme time intervals. Nevertheless, the difference in the mean absolute error of the non-parametric method and our model is just 0.26 days.

4.6 Conclusion

This chapter presents a comprehensive framework for incorporating temporality into disease progression modeling. The main contribution is the proposal of a time-dependent probabilistic generative model for unsupervised classification of treatments with irregular time intervals. The generative model allows to: (i) model the irregular time intervals between medical events; (ii) discover the different subtypes of disease progression in terms of the sequence of medical events and the time elapsed between them; and (iii) segment the sequences into progression patterns of treatments.

We validate this approach through a simulation experiment, successfully recovering the original model. Additionally, we demonstrate, using real EHRs, that the model accurately captures underlying temporal dynamics and variability within treatment subtypes. Practical applications of this model include the assessment of the adherence of the treatment trajectories to medical practice guidelines, the simulation of new treatments, the prediction of the next hospital visit, and the interpretable representation of a set of treatments.

Chapter 5

A probabilistic generative model for comorbidity progression

5.1 Introduction

In previous chapters, we focused on modeling sequences based on their subtypes and temporal progression within the context of a single disease. We now shift our focus to a more complex challenge: understanding and modeling the comorbidities within patients' clinical history. *Comorbidity* refers to the co-occurrence of multiple diseases within the same patient. Considering the joint evolution of diseases offers several benefits, including a deeper understanding of disease interactions, joint progression, and relationships between diseases [29].

This chapter addresses the specific problem of modeling the joint progression of coexisting diseases when most of the diagnoses in EHRs are missing. Probabilistic models are a practical solution to face this challenge. Not only because they can handle missing data, but also because they account for temporal relationships in data. Furthermore, they are interpretable models capable of extracting clinically meaningful representations from the inferred latent variables, as demonstrated in Chapters 3 and 4.

In the literature, most probabilistic models developed for disease progression are extensions of LDA [25, 65] or variants of Hidden Markov models [30, 34, 35, 39, 42, 66] that capture the evolution of disease trajectories through latent states. While medical events are time-dependent variables, these models generally ignore the direct stochastic dependence between such observations and are limited to modeling sequential correlations of data only through latent states [66].

In general, existing models describe the evolution of single-disease trajectories instead of their evolution in multiple co-existing diseases (comorbidities) settings [13, 23, 35, 39]. Including comorbidities in the structure of the methods is crucial for a detailed insight into the co-occurrence patterns of diseases, and in this sense, there still remains a need for developing an interpretable framework to capture and explain their joint pro-

gression patterns [4]. The works that model the coexistence of diseases [25,30,32–34,65] assume that diagnosis labels are available at each patient visit, which might not be true in reality in the EHRs (see Figure 1.1 and Section 1.1). Moreover, diagnostic information is recorded at the specific time the diagnosis is reported, however, in the subsequent records it might not be specified.

There also exist some comorbidity progression approaches based on deep learning techniques that have been specifically built for predicting future outcomes [21]. Some of them construct comorbidity networks or learn multilevel embeddings of hospital visits to predict the onset of new diseases without providing insights into disease coevolution patterns over time [32,33,67,68]. The main purpose of these latter models is to recognize the underlying structure within each hospital visit rather than identifying the hidden diagnosis of most of the visits based on the dynamics of the clinical history. Some other works have attempted to create interpretable Recurrent Neural Network-based models [9, 11, 12] using attention mechanism to interpret hidden disease dynamics and provide an explanation of their discriminative predictions. In general, these methods are not motivated from a generative perspective and do not face common challenges in the healthcare setting, such as limited data availability, missingness or uncertainty in medical data [4, 23].

This chapter proposes a novel probabilistic generative model to address the challenges posed by EHRs, paying special attention to missing data. The objective of such a model is threefold: (i) identify and segment the medical history of patients into treatments associated with each disease they suffer from; (ii) learn the model associated with each identified disease treatment; and (iii) discover subtypes of patients with similar patterns of coevolution of comorbidities. For this purpose, the model considers a latent structure for temporal sequences, where patients are modeled by a latent class defined by the evolution patterns of their comorbidities, and each observed medical event of their clinical histories is associated with a latent diagnosis. In other words, we seek to extract diagnosis-associated subsequences from the complete sequence of medical events (i.e., from the clinical history), where classes represent similar coevolution of these subsequences of latent diagnoses.

The main contributions of this work are as follows:

- We propose a probabilistic generative model of treatment trajectories for patients suffering from several comorbidities. The model builds on Markov models to capture the transitions between medical events of the different diseases.

- The generative model considers a latent class variable that identifies different subtypes of patients according to their evolution patterns of comorbidities. In addition, the model includes a generative submodel for the treatment associated with each comorbidity.

- The generative model is trained on EHRs that are characterized by a significant amount of missing data related to the diagnosis variable. To address missing data, the model considers this diagnosis variable as latent. Therefore, we propose an

EM scheme with a dynamic programming-based method as an efficient learning algorithm for the parameters of the model.

We use synthetic and real-world data (Section 1.1) to demonstrate the validity and practical significance of the model. The experiments show the ability of our method to model the progression of coexisting diseases and to extract meaningful and individualized representations of the different treatments.

The remainder of this chapter is organized as follows. Section 5.2 introduces the problem formulation. Section 5.3 describes the probabilistic generative model and the model learning procedure. Section 5.4 presents the results of the synthetic data experiments that evaluate the performance of the proposed method and the application of the model to real-world EHRs. Section 5.5 discusses the contributions and limitations of our approach and Section 5.6 draws the conclusions.

5.2 Problem formulation

A patient's clinical history, denoted by $\boldsymbol{h}$, is a sequence of medical data collected during repeated hospital visits. Let A be the set of medical actions and D the set of diagnoses, we define a patient's EHRs as

$$\boldsymbol{h} = (h_1, ..., h_m),$$

where $h_t = (a_t, d_t)$ represents the t-th medical event of the patient, $a_t \in A$ is the medical specialty (for instance, oncology, hematology, cardiology, etc.) attended and $d_t \in D$ the diagnosis/disease, for $t = 1, ..., m$. The sequence of medical specialties $\boldsymbol{a} = (a_1, ..., a_m)$ is an observable variable, while the sequence of diagnoses $\boldsymbol{d} = (d_1, ..., d_m)$ is partially observed since it often presents missing values.

The ultimate objective is to capture the different subtypes of joint evolution of comorbidities in EHRs. For that, we first seek to identify and segment the medical history of patients into treatments associated with each comorbidity $d \in D$. This is not a straightforward task as $\boldsymbol{d}$ is incomplete (Figure 5.1), and therefore, requires to estimate the diagnosis $d_t \in \boldsymbol{d}$ for each medical specialty visit $a_t \in \boldsymbol{a}$. Furthermore, the priority of treating a disease or ongoing medical therapies often involves the modification or interruption of other treatments. For instance, the majority of anticancer therapies are associated with some cardiovascular toxicities, ranging from asymptomatic and transient to more clinically significant and long-lasting cardiac events [69]. Depending on the previous existence of cardiovascular diseases and their progression, patients are at higher risk for the development of subsequent cardiovascular injuries (e.g., heart failure), which would lead to closer and more intense monitoring of such pathology and may affect the cancer treatment. This means that the transition dynamics of comorbidities depends not only on the subtype of patient, but also on the coexisting diseases of the patient at each moment.

The problem can be seen as an unsupervised classification of a set of treatments with different progression dynamics of their comorbidities.

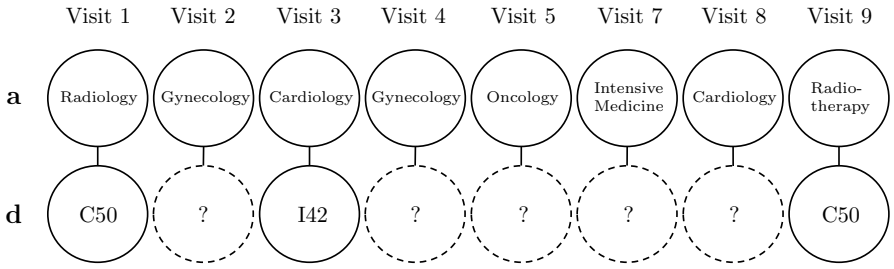


Figure 5.1: Example of EHRs with missing diagnosis information. ICD-10 code C50 corresponds to breast cancer diagnosis and I42 is a cardiomyopathy diagnosis.

5.3 Methodology

This section outlines the proposed probabilistic generative model and details the procedure for inference and parameter estimation.

5.3.1 Model definition

The proposed comorbidity generative model is built on a Markov model, which enables the description of the sequential evolution of data through a series of transitions between medical events (see Figure 5.2). Let $\boldsymbol{a} = (a_1, ..., a_m)$ be the observed sequence of medical actions that describe a patient's trajectory, where a_t belongs to the set of medical specialties A. We assume that $\boldsymbol{a}$ has an associated hidden structure of comorbidities that relates medical actions to diseases. This means that a patient trajectory consists of subsequences of medical actions associated with different diseases, $\boldsymbol{a}_d$ for $d \in D$, and these subsequences are mixed in a way that constitutes the clinical history $\boldsymbol{h}$. However, extracting the subsequences $\boldsymbol{a}_d$ for $d \in D$ is not trivial since most of the diagnosis are missing.

In this hidden structure, the presence or absence of comorbidities over time is captured by a sequence of active disease states $\boldsymbol{s} = (s_1, ..., s_m)$ associated with $\boldsymbol{a}$, where each state s_t is the set of active diseases at each time $t = 1, ..., m$ and represents the comorbidity patterns of a patient in t. The set of active disease states is defined as $S = \{0, 1\}^{|D|}$ where 1 indicates that the disease $d \in D$ is active at a specific time and 0 means that disease is not active in the patient at that time. The transition dynamics of these active disease states define the activation and deactivation of diseases, and therefore, the possible occurrence of diseases over time. Let $\boldsymbol{d} = (d_1, ..., d_m)$ be the latent sequence of diseases, where d_t belongs to the set of diagnoses $D = \{1, ..., r\}$ for $t = 1, ..., m$. The active disease states determine the distribution of such diseases over time, since the dynamics of the diseases depend on which comorbidities are active at the same time. Therefore, when a comorbidity is activated or deactivated, the distribution of the remaining active diseases changes. We further consider that once an active disease is deactivated, it cannot be present in the patient again.

Finally, let c be the latent class which $\boldsymbol{a}$ belongs to. The class c belongs to a set $C = \{1, ..., k\}$, which represents the subtypes of similar coevolution patterns of comorbidities among patients. The role of this latent variable is to capture the heterogeneity among the clinical histories based on the joint evolution of diseases. By doing so, it enables the classification of patients into distinct groups characterized by diverse comorbidity patterns over time. The classes influence the distribution of diseases but do not affect the transition dynamics of medical actions. That is, the generative model assumes that the stochastic model of the treatment of a disease is common to all patients, while it is the evolution of diseases over time that creates the different subgroups of patients.

The proposal for the generative model is as follows (see Figure 5.2):

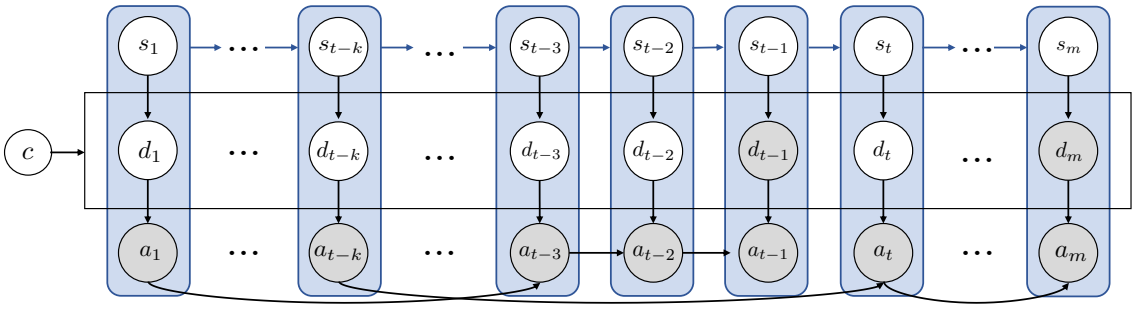


Figure 5.2: Proposed comorbidity model defined by the conditional distributions $p(a_{d_t:t}|a_{d_t:t'}, d_t)$, $p(d_t|c, s_t)$ and $p(s_t|s_{t-1}, d_{t-1}, a_{t-1})$ for observed sequences of actions $\boldsymbol{a}$, latent sequence of active disease states $\boldsymbol{s}$, latent sequences of diseases $\boldsymbol{d}$ and latent classes c.

a) Draw a class $c \sim Mult(\boldsymbol{\theta}_C)$

b) Sample the initial active disease state (set of potential comorbidities), the initial disease, and the initial medical action,

$$s_1 \sim Cat(\boldsymbol{\pi}_S),$$

$$d_1|s_1, c \sim Cat(\boldsymbol{\pi}_D^{s_1,c}), \quad a_1|d_1 \sim Cat(\boldsymbol{\pi}_A^{d_1})$$

c) For each time t:

i) Sample an active disease state from $p(s_t|s_{t-1}, d_{t-1}, a_{t-1})$, that is,

$$s_t|s_{t-1}, d_{t-1}, a_{t-1} \sim Cat(\boldsymbol{\theta}_S^{s_{t-1},d_{t-1},a_{t-1}})$$

ii) Sample a disease d_t from $p(d_t|s_t, c)$,

$$d_t|s_t, c \sim Cat(\boldsymbol{\theta}_D^{s_t,c})$$

iii) Sample an action $a_{d_t:t}$ from $p(a_{d_t:t}|d_t, a_{d_t:t'})$, that is,

$$a_{d_t:t}|d_t, a_{d_t:t'} \sim Cat(\boldsymbol{\theta}_A^{d_t, a_{d_t:t'}})$$

where $a_{d:t}$ is the t-th action associated with the disease d and $a_{d:t'}$ the previous action associated with the same disease d, so that $\boldsymbol{a}_d$ is the treatment of the disease d.

Translating the generative process into a joint probability model results in the following expression (Figure 5.2):

$$p(\boldsymbol{a}, \boldsymbol{s}, \boldsymbol{d}, c) = p(c) \prod_{t=1}^{m} p(s_t|s_{t-1}, d_{t-1}, a_{t-1}) \cdot p(d_t|c, s_t) \cdot p(a_{d:t}|d_t, a_{d:t'}) \tag{5.1}$$

where $p(s_1|s_0, d_0, a_0) = p(s_1)$ and $p(a_{d:t}|d_t = d, a_{d:0}) = p(a_{d:t}|d_t = d)$ for any value of $t = 1, ..., m$.

In light of the above, $p(c)$ is a multinomial probability distribution that describes the probability of drawing a class from the set of classes of treatments C. We define $\boldsymbol{\theta}_C$ as the set of such probabilities that we have to learn:

$$\boldsymbol{\theta}_C = \{p(c) : c \in C\}.$$

The active disease states determine the coexisting diseases at each time t. The probability of transition from a state s to s' is defined by a Markov model, whose parameters are:

$$\boldsymbol{\theta}_S = \{\boldsymbol{\theta}_S^{s,d,a} : s \in S, d \in D, a \in A\} = \{p(s'|s, d, a) : s, s' \in S, d \in D, a \in A\}.$$

Diseases follow a categorical distribution conditioned to the set of coexisting diseases $s_t \in S$ at time t and the class of patient $c \in C$. Thus, for each active disease state $s \in S$ and each class $c \in C$, we have the following parameters:

$$\boldsymbol{\theta}_D = \{\boldsymbol{\theta}_D^{s,c} : s \in S, c \in C\} = \{p(d|s, c) : d \in D, s \in S, c \in C\}.$$

In addition, we define a Markov model from which the medical actions are drawn. The conditional distributions of this model are given by a set of $|D|$ transition matrices of size $|A| \times |A|$ whose model parameters are:

$$\boldsymbol{\theta}_A = \{\boldsymbol{\theta}_A^{d,a} : d \in D, a \in A\} = \{p(a'|a, d) : a, a' \in A, d \in D\}.$$

Finally, $\boldsymbol{\pi}_S$, $\boldsymbol{\pi}_D^{s,c}$ and $\boldsymbol{\pi}_A^d$ are the parameters of the initial model for the active disease states, diseases and medical actions, respectively.

5.3.2 Maximum likelihood parameter estimation

This section introduces the learning procedure of the model parameters. Let $\mathcal{A} = \{\boldsymbol{a}^1, ..., \boldsymbol{a}^N\}$ be the set of observed sequences of actions and let $\mathcal{S} = \{\boldsymbol{s}^1, ..., \boldsymbol{s}^N\}$ be the

associated set of sequences of active disease states. As we mentioned in Section 5.2, the sequence of diseases $\boldsymbol{d}$ is partially observed, providing an intuition about the onset and end of the diseases, and therefore, about their activation and deactivation timestamps. However, note that the activation time of a disease tends to be inherently unobservable in EHRs since the first and last records of a diagnosis may not reliably indicate the real time of disease onset and end. We define a time parameter τ to determine the time interval in which a disease is active $(t_{init} - \tau, t_{end} + \tau)$, where t_{init} and t_{end} are the first and last time a diagnosis is observed in EHRs, respectively. Thus, we determine the sequence of active disease states for each sequence of actions $\boldsymbol{a} \in \mathcal{A}$ in such a way that the corresponding set of sequences of diseases, $\mathcal{D}_{\boldsymbol{a}}$, is limited to all the possible sequences of diseases that coherently fit the existing diagnoses in EHRs.

The complete dataset, including the respective latent variable values for each observation in $\mathcal{A}$, is unavailable. Hence, we use the EM algorithm to find an efficient framework for maximizing the likelihood function. To learn the distribution underlying the sequences, we maximize the following expected value of the complete-data log-likelihood:

$$\max_{\boldsymbol{\theta}} \sum_{\substack{\boldsymbol{a}\in\mathcal{A} \\ \boldsymbol{s}\in\mathcal{S}}} \sum_{\boldsymbol{d}\in\mathcal{D}_{\boldsymbol{a}}} \sum_{c\in C} p(\boldsymbol{d}, c|\boldsymbol{a}, \boldsymbol{s}) \cdot \log p(\boldsymbol{a}, \boldsymbol{s}, \boldsymbol{d}, c; \boldsymbol{\theta}) \tag{5.2}$$

where $\mathcal{D}_{\boldsymbol{a}}$ is the set of possible sequences of diseases for $\boldsymbol{a}$, and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_A, \boldsymbol{\theta}_S, \boldsymbol{\theta}_D, \boldsymbol{\theta}_C\}$. Each sequence $\boldsymbol{a} \in \mathcal{A}$ contributes equally to the model regardless of its length due to

$$\sum_{c\in C} \sum_{\boldsymbol{d}\in\mathcal{D}_{\boldsymbol{a}}} p(\boldsymbol{d}, c|\boldsymbol{a}, \boldsymbol{s}) = 1. \tag{5.3}$$

Note that the maximum size of the set $\mathcal{D}_{\boldsymbol{a}}$ is $|D|^{|\boldsymbol{a}|}$ and exponentially increases with the length of the sequence $\boldsymbol{a}$. Indeed, the parameters depend on the number of diseases we jointly model, and, in this work, we assume that the number of coexisting diseases at a specific time, s_t, is moderate even though the total number of diseases $|D|$ can be large.

To find the parameters that maximize the log-likelihood in Equation (5.2), the EM algorithm iterates as follows:

E-step: the objective is to find the posterior distribution of the latent variables given the observed sequences of actions $\boldsymbol{a} \in \mathcal{A}$ and the sequences of active states $\boldsymbol{s} \in \mathcal{S}$. Afterwards, the expectation of the logarithm of the complete-data likelihood function is computed using these values, as a function of the parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_A, \boldsymbol{\theta}_S, \boldsymbol{\theta}_D, \boldsymbol{\theta}_C\}$.

M-step: In the maximization step we update the parameters of the model with the expected values computed in the previous E-step. This maximization is achieved using the Lagrange multiplier method.

5.3.2.1 Efficient learning of the parameters of the model

Suppose that we have an observed sequence of actions $\boldsymbol{a}$, an active disease state sequence $\boldsymbol{s}$, a latent sequence of diagnosis $\boldsymbol{d}$ and a latent class variable c. A brute force learning of the parameters of the model with latent variables is computationally expensive. We propose an alternative learning algorithm based on dynamic programming to considerably reduce the number of computations, and thus, the complexity of the model from exponential to polynomial.

Suppose that, for a sequence of actions $\boldsymbol{a} \in \mathcal{A}$, we observe the transition from $a_{t'} = a'$ to $a_t = a$ between two any time points t' and t, $t' < t$. Transitions between medical actions are only allowed if they are associated with the same diagnosis. Therefore, for the transition from $a_{t'} = a'$ to $a_t = a$ it must be satisfied that the sequence of latent diagnosis has the form

$$(d_1, ...d_{t'-1}, d, d_{t'+1}, ..., d_{t-1}, d, d_{t+1}, ..., d_m)$$

where $d_{t'+1}, ..., d_{t-1} \neq d$.

In the E-step, we marginalize $p(\boldsymbol{d}, c|\boldsymbol{a}, \boldsymbol{s})$ and compute the sum of the probabilities of all the possible sequences of diseases for which $d_t = d$. We can express the probabilities of these sequences as

$$p(d_t = d|\boldsymbol{a}, \boldsymbol{s}, c) = \frac{p(d_t = d, \boldsymbol{a}, \boldsymbol{s}|c)}{p(\boldsymbol{a}, \boldsymbol{s}|c)}$$

Let us define $f_c(t_1, ..., t_r)$ as the function that computes the sum of probabilities of all the possible sequences of diseases $\boldsymbol{d} = (d_1, ..., d_t)$ in the class c, where $(t_1, ..., t_r)$ $(r = |D|)$ indicates the last time that the diseases in D appear in the sequence $\boldsymbol{d}$. We compute the probability of all the sequences of diseases that have the disease $d \in D$ at time t as follows:

$$f_c(t_1, ..., t_r) = \sum_{\boldsymbol{d}_{1,...,t-1}} p(\boldsymbol{a}_{1,...,t}, \boldsymbol{s}_{1,...,t}, \boldsymbol{d}_{1,...,t-1}, d_t = d|c) \tag{5.4}$$

where $\boldsymbol{d}_{1,...,t-1} = (d_1, ..., d_{t-1})$, $\boldsymbol{s}_{1,...,t} = (s_1, ..., s_t)$ and $\boldsymbol{a}_{1,...,t} = (a_1, ..., a_t)$.

Let us define $g_c(t_1, ..., t_r)$ as the function that computes the sum of probabilities of all the possible sequences of diseases $\boldsymbol{d} = (d_{t+1}, ..., d_m)$ in c, where $(t_1, ..., t_r)$ $(r = |D|)$ indicates the first time each disease $d \in D$ appears in the sequence $\boldsymbol{d}$. We compute the probability of all the sequences of diseases that start with $d_t = d$ as:

$$g_c(t_1, ..., t_r) = \sum_{\boldsymbol{d}_{t+1,...,m}} p(\boldsymbol{a}_{t+1,...,m}, \boldsymbol{s}_{t+1,...,m}, \boldsymbol{d}_{t+1,...,m}|c, d_t = d). \tag{5.5}$$

Using Equations (5.4) and (5.5), we can express the sum of the probabilities of all the sequences for which $d_t = d$ as follows:

$$\begin{aligned} p(d_t = d, \boldsymbol{a}, \boldsymbol{s}|c) =& f_c(t_1, ..., t_i = t', ..., t_r) \cdot p(s_t|s_{t-1}, d_{t-1}, a_{t-1}) \cdot p(d_t = d|s_t, c) \\ & \cdot p(a_{d:t}|a_{d:t'}, d_t = d) \cdot g_c(t_1, ..., t_i = t, ..., t_r) \end{aligned} \tag{5.6}$$

where t' is the previous time where the disease d is allocated. In Equation (5.6), certain constraints need to be taken into account to determine the set of compatible sequences of diagnoses for $\boldsymbol{a}$, $\mathcal{D}_{\boldsymbol{a}}$. See Appendix C.1 for more details.

We propose to create a matrix of size $|D| \times |D|$ associated with each function f_c and g_c, each of them calculated with the recursive functions in Algorithm 1 and Algorithm 2.

Algorithm 1 Computation of f_c matrix

Input: $\{t_1, ..., t_r\}$: set of the last time we saw each disease
$\boldsymbol{\theta} = \{\boldsymbol{\theta}_A, \boldsymbol{\theta}_S, \boldsymbol{\theta}_D, \boldsymbol{\theta}_C\}$: model parameters.
Output: $f_c(t_1, ..., t_r)$
$t_i \leftarrow \max\{t_1, ..., t_r\}$
$t_j \leftarrow \max\{t_1, ..., t_{i-1}, t_{i+1}, ..., t_r\}$
$t \leftarrow t_i$
if $t_i - t_j > 1$ **then**
$f_c(t_1, ..., t_r) \leftarrow p(s_t|s_{t-1}, d_{t-1}, a_{t-1}) \cdot p(d^i|c, s_t) \cdot p(a_{d^i:t}|a_{d^i:t-1}, d^i) \cdot f_c(t_1, ..., t_{i-1}, t - 1, t_{i+1}, ..., t_r)$
else
$f_c(t_1, ..., t_r) \leftarrow p(s_t|s_{t-1}, d_{t-1}, a_{t-1}) \cdot p(d^i|c, s_t) \cdot \sum_{t'=0}^{t_j-1} p(a_{d^i:t}|a_{t'}, d^i) \cdot f_c(t_1, ..., t_{i-1}, t', t_{i+1}, ..., t_r)$
end if

Notice that in Algorithm 1 the statement $t_i - t_j > 1$ means that the action at time $t_i = t$ comes from the same disease as the action in the previous time $t - 1$, while the statement $t_i - t_j = 1$ means that we do not know from which previous action (or time) the action at time t comes.

Algorithm 2 Computation of g_c matrix

Input: $\{t_1, ..., t_r\}$: set of the first time we saw each disease
$\boldsymbol{\theta} = \{\boldsymbol{\theta}_A, \boldsymbol{\theta}_S, \boldsymbol{\theta}_D, \boldsymbol{\theta}_C\}$: model parameters.
Output: $g_c(t_1, ..., t_r)$
$t_i \leftarrow \max\{t_1, ..., t_r\}$
$t \leftarrow t_i + 1$
$g_c(t_1, ..., t_r) \leftarrow \sum_i p(s_t|s_{t-1}, d_{t-1}, a_{t-1}) \cdot p(d^i|c, s_t) \cdot p(a_t|a_{d^i:t_i}, d^i) \cdot g_c(t_1, ..., t_{i-1}, t_i = t, t_{i+1}, ...t_r)$

In the M-step, we use the posterior distributions computed in E-step as constants to maximize Equation (5.2) with respect to the parameters $\boldsymbol{\theta}$. This maximization is achieved using Lagrange multipliers (Appendix C.2). If $\theta_a^{d,a'}, \theta_s^{s',d,a}, \theta_d^{s,c}$, θ_c denote a component in $\boldsymbol{\theta}_A^{d,a'}$, $\boldsymbol{\theta}_S^{s',d,a}$, $\boldsymbol{\theta}_D^{s,c}$, $\boldsymbol{\theta}_C$, respectively, the parameters of the model are updated as follows:

$$\theta_a^{d,a'} = \frac{\sum_{\boldsymbol{a}\in\mathcal{A}} \sum_{t=1}^{|\boldsymbol{a}|} \sum_{t'<t} \mathbb{1}_{a',a}(a_{d:t'}, a_{d:t}) \cdot p(d_t = d|\boldsymbol{a}, \boldsymbol{s})}{\sum_{a\in A} \sum_{\boldsymbol{a}\in\mathcal{A}} \sum_{t=1}^{|\boldsymbol{a}|} \sum_{t'<t} \mathbb{1}_{a',a}(a_{d:t'}, a_{d:t}) \cdot p(d_t = d|\boldsymbol{a}, \boldsymbol{s})} \tag{5.7}$$

where

$$\mathbb{1}_{a',a}(a_{d:t'}, a_{d:t}) = \begin{cases} 1 & \text{if } a_{d:t'} = a', a_{d:t} = a \\ 0 & \text{otherwise.} \end{cases} \tag{5.8}$$

$$\theta_s^{s',d,a} = \frac{\sum_{\boldsymbol{a}\in\mathcal{A}} \sum_{t=1}^{|\boldsymbol{a}|} \mathbb{1}_{a,s',s}(a_{t-1}, s_{t-1}, s_t) \cdot p(d_{t-1} = d|\boldsymbol{a}, \boldsymbol{s})}{\sum_{s\in S} \sum_{\boldsymbol{a}\in\mathcal{A}} \sum_{t=1}^{|\boldsymbol{a}|} \mathbb{1}_{a,s',s}(a_{t-1}, s_{t-1}, s_t) \cdot p(d_{t-1} = d|\boldsymbol{a}, \boldsymbol{s})} \tag{5.9}$$

where

$$\mathbb{1}_{a,s',s}(a_{t-1}, s_{t-1}, s_t) = \begin{cases} 1 & \text{if } a_{t-1} = a, s_{t-1} = s', s_t = s \\ 0 & \text{otherwise.} \end{cases}$$

$$\theta_d^{s,c} = \frac{\sum_{\boldsymbol{a}\in\mathcal{A}} \sum_{t=1}^{|\boldsymbol{a}|} \mathbb{1}_s(s_t) \cdot p(d_t = d, c|\boldsymbol{a}, \boldsymbol{s})}{\sum_{d\in D} \sum_{\boldsymbol{a}\in\mathcal{A}} \sum_{t=1}^{|\boldsymbol{a}|} \mathbb{1}_s(s_t) \cdot p(d_t = d, c|\boldsymbol{a}, \boldsymbol{s})} \tag{5.10}$$

$$\theta_c = \frac{\sum_{\boldsymbol{a}\in\mathcal{A}} p(c|\boldsymbol{a}, \boldsymbol{s})}{\sum_{c\in C} \sum_{\boldsymbol{a}\in\mathcal{A}} p(c|\boldsymbol{a}, \boldsymbol{s})} \tag{5.11}$$

The proposed learning algorithm based on dynamic programming allows the E-step to be polynomially solved, where the exponential number of configurations of diseases and classes for a given sequence of actions is considered. Furthermore, the complexity of the M-step is of order $\mathcal{O}(\sum_{\boldsymbol{a}\in\mathcal{A}} |\boldsymbol{a}|)$, that is, the total number of medical actions of the set $\mathcal{A}$.

A large amount of configurations of diseases, classes, and actions creates problems of sparsity in the parameters of the model. Once a parameter reaches a value of 0, that parameter cannot obtain a different value in the subsequent iterations. We add a smoothing parameter to the model in each iteration of the EM algorithm to prevent this sparsity problem.

5.4 Experimental evaluation

This section presents two sets of experiments to validate the model. The goal of the first set of experiments is to evaluate the ability of our learning algorithm to recover the original generative model underlying the data, for which we use synthetic data. The second set of experiments show some applications of the generative comorbidity model on real-world data, such as the segmentation of the medical history of a patient into different treatments, the identification of the different classes based on the joint progression of comorbidities, and the imputation of missing diagnoses. The corresponding

source code is publicly available[1].

5.4.1 Results on synthetic data

We perform experiments on synthetic data to show the behavior of the learning algorithm in controlled environments. In these experiments the diagnoses are considered unknown in the learning process. Since this is an artificial domain, the evaluation of the learned model is carried out using the log-likelihood in training and test data so that we can quantify the fitting and generalization abilities, respectively.

To this end, the first step of the experiment is to build a original generative model. In order to do that, we consider random parameters. For simplicity, we perform experiments with 2 and 3 comorbidities. In both cases, we set 2 classes and 10 medical actions. The parameters of the generative model are created as follows: $p(c)$, $p(a'|a,d)$ and $p(d|s,c)$ are sampled from a uniform Dirichlet distribution for $c \in C$, $a, a' \in A$ and $d \in D$; and $p(s'|s,d,a)$ is also sampled from a Dirichlet distribution with $\alpha = 1$ but limiting the active disease states to only activate or deactivate a single disease in each transition. To avoid the generative model taking values too close to 0, we smooth the sufficient statistics $p(c)$, $p(a'|a,d)$ and $p(d|s,c)$ by adding 10^{-2}, and $p(s|s',d,a)$ by adding 10^{-3}.

From the generative model we sample training sets of sizes $N = \{100, 300,$
$500, 800, 1000, 1200, 1500\}$ and a test set of size 1500. We learn the parameters of the model $\boldsymbol{\theta}^n = \{\boldsymbol{\theta}^n_A, \boldsymbol{\theta}^n_S, \boldsymbol{\theta}^n_D, \boldsymbol{\theta}^n_C\}$ for each training set of size $n \in N$ using the EM algorithm proposed in Section 5.3.2. At each iteration of the EM algorithm the sufficient statistics are smoothed by adding 10^{-2} to $p(c)$, $p(a|a',d)$ and $p(d|s,c)$, and 10^{-3} to $p(s|s',a,d)$. Once the model has converged, we measure the quality of these learned models with the log-likelihood of the data (Equation (5.2)) normalized by the total number of actions in each dataset of size $n \in N$ to make the results comparable.

This experiment is repeated five times, considering, for each of them, a different original generative model. Figure 5.3 and Figure 5.4 show the fitting and generalization ability of the method through the average log-likelihood of 2 and 3 comorbidities. The average log-likelihood of the learned models on the training sets (orange solid line) quantifies the fitting of the models to the data, while on the test set (blue solid lines) it measures its ability of generalization. The dotted lines correspond to the average log-likelihood of the 5 original generative models evaluated in the training (orange) and test (blue) datasets. We can see that as the sample size increases, the curves that quantify the fitting and generalization of the learned models converge to the curves of the original generative models. This means that, given a sufficiently large dataset, the proposed learning algorithm can reach the original generative model underlying the data.

[1] https://github.com/onintzezaballa/ComorbidityGenerativeModel

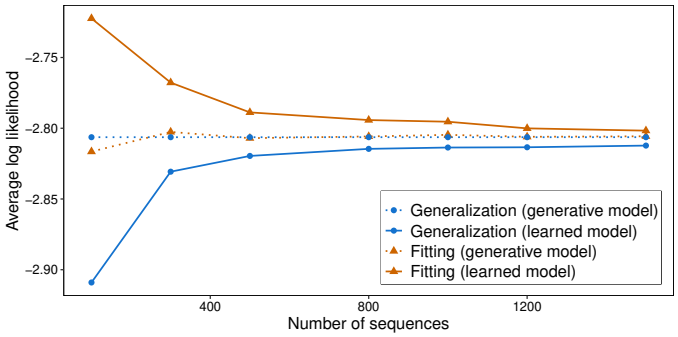


Figure 5.3: Fitting and generalization of synthetic generative models with 2 comorbidities.

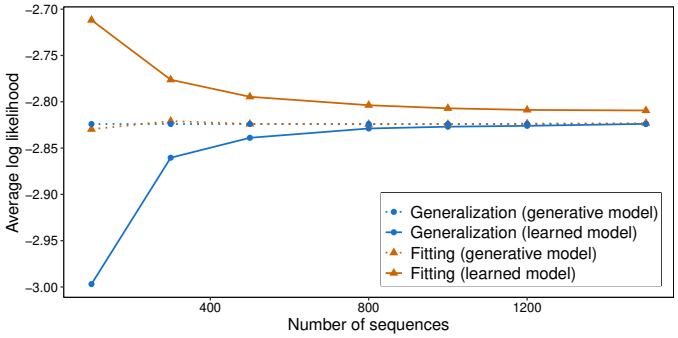


Figure 5.4: Fitting and generalization of synthetic generative models with 3 comorbidities.

5.4.2 Results on real-world data

This section shows the utility of the generative model on patients with breast cancer and cardiovascular diseases, which are highly related comorbidities [69]. We use the generative model in two different applications: we first perform an experiment to show the segmentation of individual clinical histories into disease treatments; and then, a population-level experiment to obtain the coevolution patterns of these two comorbidities. We further assess the results of these experiments by predicting the diagnosis of unseen instances.

5.4.2.1 Dataset

We apply the model on the EHRs described in Section 1.1. As a use case, we focus our attention on the comorbidities of the breast cancer population, specifically on cardiovascular diseases. These diseases are biologically connected through some adverse effects of cancer treatment on cardiovascular health [69]. The resulting dataset consists of 90 clinical histories, whose average length is 140 medical actions, and they are generated by 29 unique medical specialties (selected following the process in Chapter 2).

The percentage of missing diagnoses of these EHRs is 81%.

5.4.2.2 Hyperparameters and model specifications

We consider breast cancer patients with any diagnosis related to cardiovascular diseases, that is, $|D| = 2$. According to clinical guidelines [69], patients evolve according to their severity of short-term cardiotoxic effects caused by anticancer therapies. In order to have a sufficient number of patients per class and after conducting experiments for different values of the latent class, we have concluded that $|C| = 2$ is appropriate for the available data.

Besides, since we are in a realistic scenario, we include prior diagnosis knowledge in the model, so that we can obtain more accurate results and reduce the model complexity. Since 19% of the diagnoses are available, we force them to remain fixed in their original time position in the latent sequences of diseases. Varying the value of τ can have a significant influence on both accuracy and computational efficiency. Through experiments conducted with different values of $\tau = \{90, 180, 360, 720, 1080, 1440\}$, we observed that setting τ to 720 days gets a good balance between computational efficiency and model performance. Therefore, to establish the active disease states, we assume that the transition between two medical actions of the same disease may occur within a maximum interval of $\tau = 720$ days.

	ACTION 1	ACTION 2	ACTION 3	ACTION 4	ACTION 5	ACTION 6	ACTION 7	ACTION 8	ACTION 9	ACTION 10	ACTION 11
BREAST CANCER	Radiology		Oncology	**Oncology**	**Oncology**	Oncology		Oncology	**Oncology**	Oncology	Radiology
CARDIOVASCULAR DISEASE		Cardiology					Cardiology				

	ACTION 12	ACTION 13	ACTION 14	ACTION 15	ACTION 16	ACTION 17	ACTION 18	ACTION 19	ACTION 20	ACTION 21	ACTION 22
BREAST CANCER		Gynecology	Oncology	Anesthesia	**Anesthesia**	**Surgery**	**Oncological gynecology**		Pathological anatomy	Gynecology	**Radiotherapy**
CARDIOVASCULAR DISEASE	**Cardiology**							Radiology			

	ACTION 23	ACTION 24	ACTION 25	ACTION 26	ACTION 27	ACTION 28	ACTION 29	ACTION 30	ACTION 31	ACTION 32	ACTION 33
BREAST CANCER	Oncology				Radiotherapy	**Radiotherapy**	Cardiology	**Radiotherapy**	Radiotherapy	Radiotherapy	Radiotherapy
CARDIOVASCULAR DISEASE		**Rehabilitation**	Rehabilitation	Rehabilitation							

Figure 5.5: Disentangle of a partial clinical history of a patient with the diagnosis of breast cancer and cardiovascular disease. The bold medical specialties represent the real diagnosis collected in EHRs. The results are obtained from the model learned in Section 5.4.2.3.

5.4.2.3 Individualized segmentation of clinical histories

The first objective of the model is to segment the sequence of actions, $\boldsymbol{a}$, into subsequences associated with the different comorbidities. This is useful, for instance, for understanding the evolution of a single disease in a patient, extracting its associated treatment dynamics from the clinical history, or even for an informing forecast of expected costs of care and medical resources for specific diseases and patients by simulating trajectories from each disease related model.

In this experiment we train the model with the whole dataset. Then, the association between medical specialties and diagnosis at each time t of the sequence $\boldsymbol{a} \in \mathcal{A}$ is given by the diagnosis of maximum probability at time t, that is,

$$\max_{d \in D} p(d|\boldsymbol{a}, s_t) \tag{5.12}$$

where s_t is the set of active diseases at time t.

Thus, we can extract the subsequence associated with each diagnosis from a patient's clinical trajectory $\boldsymbol{h}$. An example of that is the segmentation of a partial clinical history of a real patient that we show in Figure 5.5. Although in Figure 5.5 we attribute a diagnosis to each medical event, the model allows us to assign to each medical action the probability of belonging to any disease. In reality, a fundamental aspect of caring for a patient undergoing potentially cardiotoxic anticancer therapy is to be treated by a multidisciplinary team of oncologists, cardiologists, and other healthcare professionals [69]. This means that a medical event may not be the consequence of a single disease, but is caused by a set of diseases that co-exist over time.

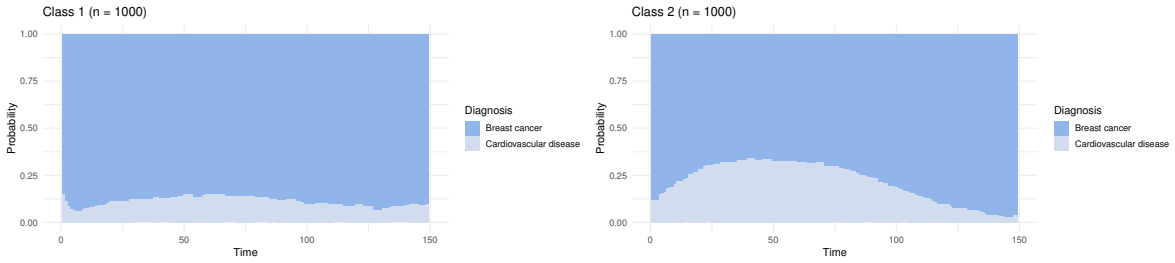
5.4.2.4 Representation of the joint progression of comorbidities at population-level

The learned generative model enables to extract knowledge about comorbidity evolution patterns at population-level regarding the subtypes of treatments. This is a simulation experiment to provide a representation of the different joint evolution of breast cancer and cardiovascular diseases.

Following the generative process in Section 5.3.1, we randomly sample a set of 1000 clinical histories for each class from the learned model in the previous paragraph. The clinical histories are of variable length and we set the maximum number of actions to be 140. We show the joint evolution of comorbidities by calculating the probability of a disease-related event occurring at each time point, that is,

$$p(d_t = d|c), \quad \text{for all } t. \tag{5.13}$$

In Figure 5.6 we show the joint evolution of the breast cancer and cardiovascular diseases for the 2 classes. Although breast cancer treatment clearly dominates in both classes, the occurrence of cardiovascular treatment is different depending on the class. The probability of treating cardiovascular diseases remains constant in the first class



(a) Class 1 of coevolution treatments.

(b) Class 2 of coevolution treatments.

Figure 5.6: Joint evolution of comorbidities at a population-level.

(Figure 5.6a), while it increases in the initial part of the medical records in the second class (Figure 5.6b). Therefore, class 1 may refer to patients with pre-existing cardiovascular disease or cardiovascular risk factors undergoing potentially cardiotoxic anticancer therapy that requires routine monitoring [69]. On the contrary, class 2 may indicate more severe cardiovascular complications as a consequence of the harmful effect of anticancer therapies on the cardiovascular system [69].

5.4.2.5 Imputation of diagnosis

Another application of our generative model is the imputation of missing diagnosis values of EHRs. In other words, we seek to label a new patient's medical events with a diagnosis for each timestep. To assess the diagnosis assignment of the model, we carry out a 10-fold cross-validation, where we split the dataset into training and test sets in a 90:10 proportion. We train the model as in previous experiments, including the diagnoses collected in the EHRs. However, in this experiment we propose the most complex scenario for the test set, considering every diagnosis to be unknown. The problem consists of setting a diagnosis label for each medical specialty of the test set with Eq. 5.12, and afterward, checking them with 19% of available diagnoses as ground truth.

We replicated the cross-validation experiment using two simplified versions of the model (Equation (5.1)) to demonstrate the significance and utility of the latent class and activation state variables in the assignment of diagnoses to medical events. On the one hand, the first simplification we carry out to our original model is to delete the class information. In this sense, we assume that there are no subtypes of progression in comorbidities, and therefore, there are no patients with higher probability of developing one disease over another. Then,

$$p(\boldsymbol{a}, \boldsymbol{s}, \boldsymbol{d}) = \prod_{t=1}^{m} p(s_t|s_{t-1}, d_{t-1}, a_{t-1}) \cdot p(d_t|s_t) \cdot p(a_{d:t}|d_t, a_{d:t'}). \tag{5.14}$$

On the other hand, in the second baseline model we eliminate the activation states from the original model, while still considering class information. This model assumes

that comorbidities are always active throughout the entire medical history of a patient. The joint probability of this model is defined as

$$p(\boldsymbol{a}, \boldsymbol{d}, c) = p(c) \prod_{t=1}^{m} p(d_t|c) \cdot p(a_{d:t}|d_t, a_{d:t'}). \tag{5.15}$$

Model	AUC	Accuracy	F1-score	
			Breast cancer	Cardiovasc.
Proposed model (Eq. 5.1)	0.81	0.84	0.90	0.75
Model without classes (Eq. 5.14)	0.76	0.80	0.85	0.71
Model without activation states (Eq. 5.15)	0.75	0.78	0.83	0.68

Table 5.1: Comparative evaluation of the models.

We can observe in Table 5.1 the improved assignment performance of our model, which achieves higher AUC, accuracy, and F1-score values. These results highlight the significance of including both latent classes and activation states in our model. In addition, this experiment not only supports the quality of the segmentation of clinical histories into treatments of individual patients (Section 5.4.2.3), but also the comorbidity evolution dynamics captured in the simulation experiment (Section 5.4.2.4).

5.5 Discussion

This chapter proposes a novel probabilistic generative model for patients with comorbidities, that is, co-existing diseases. Modeling comorbidity dynamics from EHRs is not straightforward and involves addressing challenges such as small datasets, uncertainty, and missingness [1, 4]. We face the challenging problem where the diagnosis is missing in most of the EHRs. This requires to contruct a model where diseases coexist without precise information indicating when they occurred. Hence, the model is specifically focused on the identification of the diagnoses associated with medical events and the discovery of subtypes of similar disease coevolution patterns. To the best of our knowledge, this is the first method to learn the dynamics of underlying comorbidity without observing the entire clinical history of diagnoses.

Experiments show that the generative model can accurately estimate the diagnosis of medical records. These results emphasize the model's ability to extract treatment subsequences from EHRs and capture the main subtypes of comorbidity evolution dynamics based on medical specialties. This correct diagnosis imputation is of great interest for training models that require complete EHRs or avoiding loss of information observed in other imputation methods [1].

A limitation of the proposed model is its complexity when the number of diseases is too large. The number of parameters of the disease distribution $\boldsymbol{\theta}_D$ to be learned is $2^{|D|}$. Nevertheless, the number of coexisting comorbidities that we consider is not so large as to become an intractable problem. One way to control the number of

parameters is to assume that the number of coexisting diseases is limited, that is, only $j \leq |D|$ diseases can be active at the same time. This would imply $\sum_{i=1,\dots,j} \binom{|D|}{i}$ possible combinations of active diseases at the same time. In this case, the number of the disease distribution parameters would be $|D| \cdot \sum_{i=1,\dots,j} \binom{|D|}{i}$. Another alternative to deal with a larger amount of diseases would be to simplify the model by assuming the same disease distribution throughout the entire clinical history, instead of being dependent on the active diseases at each time.

5.6 Conclusion

Comorbidity refers to the co-occurrence of multiple diseases within the same patient. Considering the joint evolution of diseases offers several benefits, including a deeper understanding of disease interactions, joint progression, and relationships between diseases. However, learning the joint progression of comorbidities in the presence of significant missing diagnoses is a challenging task. This missing data introduces uncertainty regarding the association between medical events and specific diseases. Therefore, many medical records are not directly related to any specific disease treatments.

This chapter introduces an interpretable probabilistic generative model developed to capture the comorbidity dynamics within the context of incomplete EHRs. This model has a specific focus on identifying missing diagnoses related to medical events and discovering various subtypes of disease coevolution patterns. It proves to be effective in scenarios where coexisting diseases follow diverse progressions based on the patient's active comorbidities. Practical applications involving patients with breast cancer and cardiovascular diagnoses showcase the model's success in diagnosis imputation, identification of treatment subsequences from clinical histories and representation of various subtypes of comorbidity progression dynamics.

Chapter 6

Conclusions and future work

To conclude the thesis, this last chapter introduces the main conclusions of the dissertation, as well as some further research directions motivated by the contributions. Finally, the main achievements of the thesis are summarized at the end of the chapter.

6.1 Conclusions

This dissertation presents novel methodologies for unsupervised learning from sequences of medical events. These methodologies effectively analyze disease treatments, identify treatment patterns and model sequences of events of variable lengths. In addition, our approaches deal with various complexities of EHRs, including missing diagnosis, the heterogeneous nature of diseases, the irregular time intervals between actions, and the presence of co-existing diseases.

This dissertation uses an administrative dataset provided by the public healthcare system in the Basque Country, Spain. These EHRs enable the tracking of all the resources used in the treatment of a disease throughout a patient's clinical history, and presents the traceability of the whole clinical care process. However, this repository does not include clinical outcomes. Each patient's clinical history is characterized by a chronological sequence of medical events, and each medical event is represented by a medical action, a diagnosis, and the timestamp. Within a sequence, the diagnosis variable allows us to associate each medical event with a specific disease, although in the 19% of the medical events the variable presents missing values. The experimental results based on this dataset validate the reliability of the proposed models and demonstrate their diverse applications, including new data generation, missing diagnosis imputation, treatment segmentation and time estimation.

Chapter 2 introduces a partitional methodology developed to identify representative treatments for any disease of interest using EHRs. It systematically extracts end-to-end treatment trajectories from EHRs using a relevance measure and multiple selection criteria. Then, it classifies these treatments to create a comprehensive representation of disease treatments within the healthcare system. Practical applications in breast

cancer patients demonstrate the model's ability to extract complete end-to-end treatments from clinical data with missing values, segment treatment populations, and depict this population with a set of representative treatments from EHRs. These experimental results, validated by healthcare professionals and compared with clinical practice guidelines, demonstrates the reliability of the methodology. This research improves the understanding of treatment patterns for a disease and highlights the importance of considering the heterogeneous types of diseases in disease progression models.

Chapter 3 proposes a probabilistic generative model to characterize treatment variability in a disease through progression patterns. Unlike Chapter 2, which performs a partition of the dataset for treatment classification, this chapter also focuses on understanding the evolution of treatments. The model classifies sequences of medical events into distinct subtypes and segments these sequences into various progression stages. Practical applications involving breast cancer patients demonstrate the model's capacity to classify treatments, identify their progression stages, and generate new treatment trajectories for a disease. The model application could be extended to other progressive diseases like other cancers, respiratory conditions, or neurodegenerative disorders that evolve slowly over time. This probabilistic model also allows to uncover associations between treatment trajectories of similar patients, establish data-driven taxonomies for disease progression, and reduce the uncertainty in predicting a patient's treatment trajectory.

Chapter 4 presents an extension of the probabilistic generative model in Chapter 3 including the temporal information to capture the irregular time intervals between consecutive medical events. The model classifies treatments into different subtypes based on the order of medical events and their time intervals, segments the treatments into subsequences of patterns of disease progression, and model the irregular time between every pair of medical events. It offers flexibility in modeling the time distribution, allowing the choice of the most appropriate distribution based on the available data. The experimental results involving breast cancer patients for both treatment classification and time prediction demonstrate the model's efficacy and reliability in providing meaningful insights into disease progression patterns and accurate estimations of time intervals between medical events.

Chapter 5 proposes a probabilistic generative model to understand comorbidity dynamics considering incomplete EHRs. Unlike the model developed in Chapter 3 which focuses on single diseases, this chapter considers the modeling of multiple diseases coexisting simultaneously. The model mainly focuses on the identification of missing diagnoses associated with medical events and the classification of subtypes of similar disease coevolution patterns. It is particularly suitable for scenarios where coexisting diseases evolve differently depending on the active comorbidities of the patient. Practical applications involving patients with breast cancer and cardiovascular diagnoses showcase the model's success in diagnosis imputation, identification of treatment subsequences from clinical histories and representation of various subtypes of comorbidity progression dynamics.

6.2 Future work

This section proposes various research directions based on the contributions of this dissertation.

In the methodology proposed in Chapter 2, it would be valuable to establish a new distance metric based on expert knowledge for comparing sequences of actions. For instance, the relevance measure of medical specialties could be useful. Specifically, we calculate the relevance of the medical specialties for the patients with the target diagnosis, which essentially indicates how more frequently each medical specialty is visited by patients with the disease. This measurement could be interpreted as the significance of the medical specialties for the specific disease. By including these relevance values into the edit distance as weights, similar to the Edit Distance with Real penalties [70], we can assign higher weights to actions that are particularly relevant to the disease. Consequently, these actions would carry more significance when clustering sequences.

Throughout the remainder of the dissertation, we have developed our generative models based on Markov models to capture the transition dynamics of diseases and ensure interpretable results. However, it is essential to acknowledge a limitation of Markov models, which is their memoryless assumption. They consider that an individual's current action depends only on the previous medical action, rather than considering their entire or partial clinical history. Then, for simulation and predictive purposes, in this type of models an error can not be corrected after it is made and any error will be cascaded through all the subsequent predictions. To solve this problem, in [71], the authors introduce a neural probabilistic model that combines an autoregressive base model with an energy function. The base model generates predictions, and then, a transformer energy function learns to reweight the generated proposals to assign higher probabilities to more realistic predictions. To do this adjustment of weights, they account for the entire complete sequence, that is, past events together with predicted future events. Future work will focus on relaxing the memoryless structure of our generative models to capture long-term dependencies within patients' medical history. This could provide more informative insights into the progression and relationships between medical events.

Regarding the uncertainty between diagnosis and medical events, we propose to extend our probabilistic generative models in Chapters 3, 4 and 5 to address the Scenario 2 in Figure 1.3. That is, consider the diagnoses collectively as a set for each hospitalization or ICU episode instead of a single diagnosis for each medical event. This extension would be particularly relevant for publicly available datasets like MIMIC-III [6] or eICU [7], where diagnoses are not associated with individual medical actions but rather with entire episodes. The problem formulation changes regarding the type of missing data. Instead of assigning from a set of diagnoses a diagnosis to each medical event with missing value, the new approach involves assigning a subset of diagnoses to a subsequence of medical events. Addressing this challenge requires a modification of the EM algorithm and the dynamic programming approach. The extension of our generative models will not only enhance their applicability but also facilitate their val-

idation and comparative assessments against other methods designed for this specific scenario.

These publicly available datasets offer additional patient information that can significantly improve our patient classification and disease progression modeling results. For instance, they contain diverse data types, including laboratory results and vital signs, among others. The incorporation of such a variety of data sources would provide the model with a more informative representation of patients' health status and medical history over time. The model could capture continuous variations in a patient's health status, and therefore, improve the predictions of the onset of new diseases or potential readmissions in the hospital.

Chapter 4 introduces a generative model to learn the irregular time intervals between pairs of medical events. A limitation of the current approach is its reliance on parametric modeling of the time intervals. Although parametric models, in our particular case the Weibull distribution, have shown favorable results for time estimation, they may not capture the full complexity and variability present in data. In future work, we propose to address this limitation by incorporating non-parametric techniques into our approach. For instance, non-parametric kernel density estimation [72] could provide even more flexibility to the model and potentially capture a wider range of patterns and distributions in the time intervals.

Finally, Chapter 5 presents a generative model to learn the co-evolution of multiple diseases. However, certain diseases may have different activity patterns, with periods of activity alternating with periods of inactivity. A future direction is to enhance the comorbidity model by incorporating these variations, thereby enabling diseases to be reactivated once their initial treatment has been completed. This extension will be particularly valuable in scenarios where diseases have a cyclic or recurring nature. By accounting for disease reactivation, our model will more accurately capture the dynamic nature of diseases and how they interact with other diseases in a patient's medical history.

6.3 Main achievements

The research work conducted during this thesis has resulted in the following publications:

6.3.1 Journal papers

- Zaballa, O., Pérez, A., Gómez-Inhiesto, E., Acaiturri-Ayesta, T., Lozano, J. A. (2020). Identifying common treatments from electronic health records with missing information: An application to breast cancer. *PLOS ONE*, 15(12), e0244004.
- Zaballa, O., Pérez, A., Gómez-Inhiesto, E., Acaiturri-Ayesta, T., Lozano, J. A. (2022). Learning the progression patterns of treatments using a probabilistic generative model. *Journal of Biomedical Informatics*, 137, 104271.

- Zaballa, O., Pérez, A., Gómez-Inhiesto, E., Acaiturri-Ayesta, T., Lozano, J. A. (2023). A Probabilistic Generative Model to Discover the Treatments of Coexisting Diseases with Missing Data. *Computer Methods and Programs in Biomedicine*, 107870.

6.3.2 Conferences

- Zaballa, O., Pérez, A., Gómez-Inhiesto, E., Acaiturri-Ayesta, T., Lozano, J. A. (2023). Time-dependent probabilistic generative models for disease progression. *Machine Learning for Healthcare (ML4H)*, New Orleans, United States of America.

- Zaballa, O., Pérez, A., G'omez-Inhiesto, E., Acaiturri-Ayesta, T., Lozano, J. A. (2023). Probabilistic generative model for disease progression and healthcare cost estimation. *Sociedad de Estadística e Investigación Operativa (SEIO)*, Elche, Spain.

6.3.3 Posters

- Zaballa, O., Pérez A., Lozano, J. A. (2019). Identifying breast cancer treatments from electronic health records with missing information. *4th edition of the Bilbao Data Science Workshop*, Bilbao, Spain.

6.3.4 Short Visits

- 30 September - 29 December 2022: University of Cambridge, Deparment of Applied Mathematics and Theoretical Physics. Supervisor: Mihaela van der Schaar.

Appendix A

Probabilistic generative model for disease progression

A.1 Lagrange multiplier method

This appendix describes the process followed to obtain the update of the model parameters.

We consider $\boldsymbol{a} = (a_1, ..., a_m)$ to be the observed data, $\boldsymbol{s} = (s_1, ..., s_m)$ the underlying latent sequence of stages and c the latent class. We aim to obtain the model parameters $\boldsymbol{\theta}$ that maxime the following function,

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{\boldsymbol{s} \in \mathcal{S}_{\boldsymbol{a}}} \sum_{c \in C} p(\boldsymbol{s}, c|\boldsymbol{a}) \cdot \log p(\boldsymbol{a}, \boldsymbol{s}, c; \boldsymbol{\theta}) \tag{A.1}$$

where $\mathcal{S}_{\boldsymbol{a}}$ is the set of all the potential sequences of stages for $\boldsymbol{a}$, and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_A, \boldsymbol{\theta}_S, \boldsymbol{\theta}_C\}$.

From Equation (A.1), we obtain

$$\begin{aligned}
&\sum_{\boldsymbol{s} \in \mathcal{S}_{\boldsymbol{a}}} \sum_{c \in C} p(\boldsymbol{s}, c|\boldsymbol{a}) \cdot \log p(\boldsymbol{a}, \boldsymbol{s}, c; \boldsymbol{\theta}) = \\
&\quad = \sum_{\boldsymbol{s} \in \mathcal{S}_{\boldsymbol{a}}} \sum_{c \in C} p(\boldsymbol{s}, c|\boldsymbol{a}) \cdot \log \left(p(c) \prod_{t=1}^{m} p(a_t|a_{t-1}, s_{t-1}, c) \cdot p(s_t|s_{t-1}, a_t, c) \right) \\
&\quad = \sum_{\boldsymbol{s} \in \mathcal{S}_{\boldsymbol{a}}} \sum_{c \in C} p(\boldsymbol{s}, c|\boldsymbol{a}) \left(\log p(c) + \sum_{t=1}^{m} \log p(a_t|a_{t-1}, s_{t-1}, c) + \sum_{t=1}^{m} \log p(s_t|s_{t-1}, a_t, c) \right)
\end{aligned} \tag{A.2}$$

Since the parameters we want to optimize are now independently split into three terms in the sum, we can optimize them individually.

For the first term in Equation (A.2),

$$\sum_{\boldsymbol{s}\in\mathcal{S}_{\boldsymbol{a}}}\sum_{c\in C} p(\boldsymbol{s},c|\boldsymbol{a})\sum_{t=1}^{m}\log p(c) = \sum_{t=1}^{m}\sum_{c\in C}\sum_{s\in S} p(s_t=s,c|\boldsymbol{a})\log\theta_c$$
$$= \sum_{c\in C} p(c|\boldsymbol{a})\log\theta_c$$

To maximize with respect to θ_c we introduce the Lagrange multipliers ε. The Lagrangian is then given by:

$$L(\boldsymbol{\theta}_C) = \sum_{c\in C} p(c|\boldsymbol{a})\log\theta_c + \varepsilon(\sum_{c\in C}\theta_c - 1)$$

where $\sum_{c\in C}\theta_c = 1$. We get $p(c|\boldsymbol{a})$ from the E-step and use it as a constant. Setting the derivative equal to zero, we obtain:

$$\frac{\partial L(\boldsymbol{\theta}_C)}{\partial\theta_c} = \frac{p(c|\boldsymbol{a})}{\theta_c} + \varepsilon = 0 \implies \varepsilon = -\frac{p(c|\boldsymbol{a})}{\theta_c} \tag{A.3}$$

Multiplying each side by θ_c and summing over $c \in C$, we obtain that

$$\varepsilon = -\sum_{c\in C} p(c|\boldsymbol{a}). \tag{A.4}$$

From Equations (A.3) and (A.4), we obtain

$$\hat{\theta}_c = \frac{p(c|\boldsymbol{a})}{\sum_{c\in C} p(c|\boldsymbol{a})} = \frac{\sum_{t=1}^{m}\sum_{s\in S} p(s_t=s,c|\boldsymbol{a})}{\sum_{c\in C}\sum_{s\in S}\sum_{t=1}^{m} p(s_t=s,c|\boldsymbol{a}).}$$

For the second term in Equation (A.2),

$$\sum_{\boldsymbol{s}\in\mathcal{S}_{\boldsymbol{a}}}\sum_{c\in C} p(\boldsymbol{s},c|\boldsymbol{a})\sum_{t=1}^{m}\log p(a_t|a_{t-1},s_{t-1},c) =$$
$$= \sum_{t=1}^{m}\sum_{a\in A}\sum_{s\in S}\sum_{a'\in A} p(s_{t-1}=s,a_{t-1}=a,a_t=a',c)\log\theta_{a'}^{a,s,c}$$

To maximize with respect to $\theta_{a'}^{a,s,c}$ we introduce the Lagrange multipliers $\lambda_{a,s}$ for $a \in A$ and $s \in S$. The Lagrangian is then given by:

$$L(\boldsymbol{\theta}_A) = \sum_{t=1}^{m}\sum_{a\in A}\sum_{s\in S}\sum_{a'\in A} p(s_{t-1}=s,a_{t-1}=a,a_t=a',c)\log\theta_{a'}^{a,s,c} + \sum_{a\in A}\sum_{s\in S}\lambda_{a,s}(\sum_{a'\in A}\theta_{a'}^{a,s,c} - 1)$$

where $\sum_{a'\in A}\theta_{a'}^{a,s,c} = 1$. We get $p(s_{t-1}=s,a_{t-1}=a,a_t=a',c)$ from the E-step and use it as a constant. Setting the derivative equal to zero, we obtain:

$$\frac{\partial L(\boldsymbol{\theta}_A)}{\partial \theta_{a'}^{a,s,c}} = \frac{\sum_{t=1}^{m} p(s_{t-1}=s, a_{t-1}=a, a_t=a', c)}{\theta_{a'}^{a,s,c}} + \lambda_{a,s} = 0$$
$$\implies \lambda_{a,s} = -\frac{\sum_{t=1}^{m} p(s_{t-1}=s, a_{t-1}=a, a_t=a', c)}{\theta_{a'}^{a,s,c}} \tag{A.5}$$

Multiplying each side by $\theta_{a'}^{a,s,c}$ and summing over $a' \in A$, we obtain that

$$\lambda_{a,s} = -\sum_{a' \in A} \sum_{t=1}^{m} p(s_{t-1}=s, a_{t-1}=a, a_t=a', c) \tag{A.6}$$

From Equations (A.5) and (A.6), we obtain

$$\hat{\theta}_{a'}^{a,s,c} = \frac{\sum_{t=1}^{m} p(s_{t-1}=s, a_{t-1}=a, a_t=a', c)}{\sum_{a' \in A} \sum_{t=1}^{m} p(s_{t-1}=s, a_{t-1}=a, a_t=a', c)}$$
$$= \frac{\sum_{t=1}^{m} \mathbb{1}(a_{t-1}=a, a_t=a') p(s_{t-1}=s, c|\boldsymbol{a})}{\sum_{a' \in A} \sum_{t=1}^{m} \mathbb{1}(a_{t-1}=a, a_t=a') p(s_{t-1}=s, c|\boldsymbol{a})}$$

Finally, for the third term in Equation (A.2),

$$\sum_{\boldsymbol{s} \in \mathcal{S}_{\boldsymbol{a}}} \sum_{c \in C} p(\boldsymbol{s}, c|\boldsymbol{a}) \sum_{t=1}^{m} \log p(s_t|a_t, s_{t-1}, c) =$$
$$= \sum_{t=1}^{m} \sum_{s \in S} \sum_{a' \in A} \sum_{s' \in S} p(s_{t-1}=s, a_t=a', s_t=s') \log \theta_{s'}^{a',s,c}$$

To maximize with respect to $\theta_{s'}^{a',s,c}$ we introduce the Lagrange multipliers $\lambda_{s,a'}$ for $s \in S$ and $a' \in A$. The Lagrangian is then given by:

$$L(\boldsymbol{\theta}_S) = \sum_{t=1}^{m} \sum_{s \in S} \sum_{a' \in A} \sum_{s' \in S} p(s_{t-1}=s, a_t=a, s_t=s', c) \log \theta_{a'}^{a',s,c} + \sum_{s \in S} \sum_{a' \in A} \lambda_{s,a'} \left(\sum_{s' \in S} \theta_{s'}^{a',s,c} - 1\right)$$

where $\sum_{s' \in S} \theta_{s'}^{a',s,c} = 1$. We get $p(s_{t-1}=s, a_t=a', s_t=s', c)$ from the E-step and use it as a constant. Setting the derivative equal to zero, we obtain:

$$\frac{\partial L(\boldsymbol{\theta}_S)}{\partial \theta_{s'}^{a',s,c}} = \frac{\sum_{t=1}^{m} p(s_{t-1}=s, a_t=a', s_t=s', c)}{\theta_{s'}^{a',s,c}} + \lambda_{s,a'} = 0$$
$$\implies \lambda_{s,a'} = -\frac{\sum_{t=1}^{m} p(s_{t-1}=s, a_t=a', s_t=s', c)}{\theta_{s'}^{a',s,c}} \tag{A.7}$$

Multiplying each side by $\theta_{s'}^{a',s,c}$ and summing over $s' \in S$, we obtain that

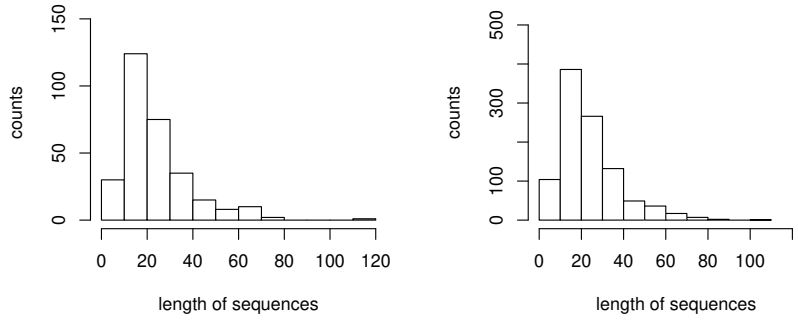
$$\lambda_{s,a'} = -\sum_{s' \in S} \sum_{t=1}^{m} p(s_{t-1}=s, a_t=a', s_t=s', c) \tag{A.8}$$

From Equations (A.7) and (A.8), we obtain

$$\begin{aligned}\hat{\theta}^{a',s,c}_{s'} =& \frac{\sum_{t=1}^{m} p(s_{t-1}=s, a_t=a', s_t=s', c)}{\sum_{s'\in S}\sum_{t=1}^{m} p(s_{t-1}=s, a_t=a', s_t=s', c)} \\ =& \frac{\sum_{t=1}^{m} \mathbb{1}(a_t=a') p(s_{t-1}=s, s_t=s', c|\boldsymbol{a})}{\sum_{s'\in S}\sum_{t=1}^{m} \mathbb{1}(a_t=a') p(s_{t-1}=s, s_t=s', c|\boldsymbol{a})}\end{aligned}$$

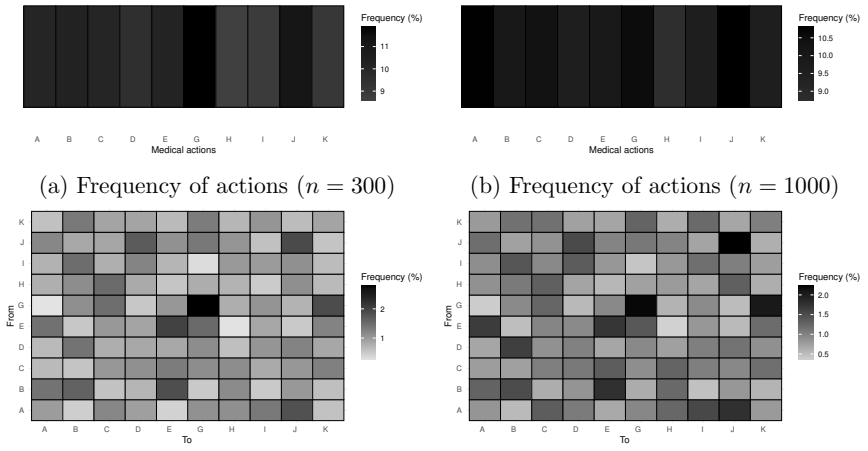
A.2 Heterogeneity on synthetic sequences

This appendix aims to show the variability of the synthetic sequences generated for the experiments in Section 3.4.1. For each experiment we represent the distribution of the lengths of the sequences, the frequency of actions and the frequency of the transition between actions for two different sizes of the dataset.



(a) $n = 300$ (b) $n = 1000$

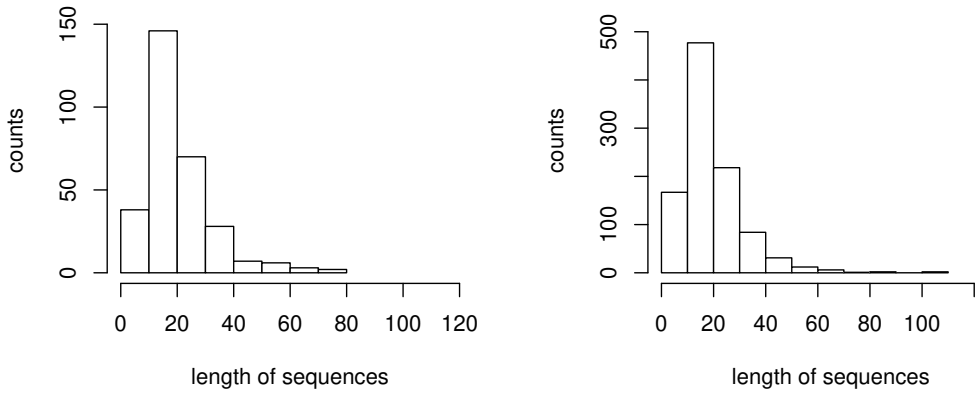
Figure A.1: Experiment 1: histogram of the lengths of the sequences of actions.



(a) Frequency of actions ($n = 300$) (b) Frequency of actions ($n = 1000$)

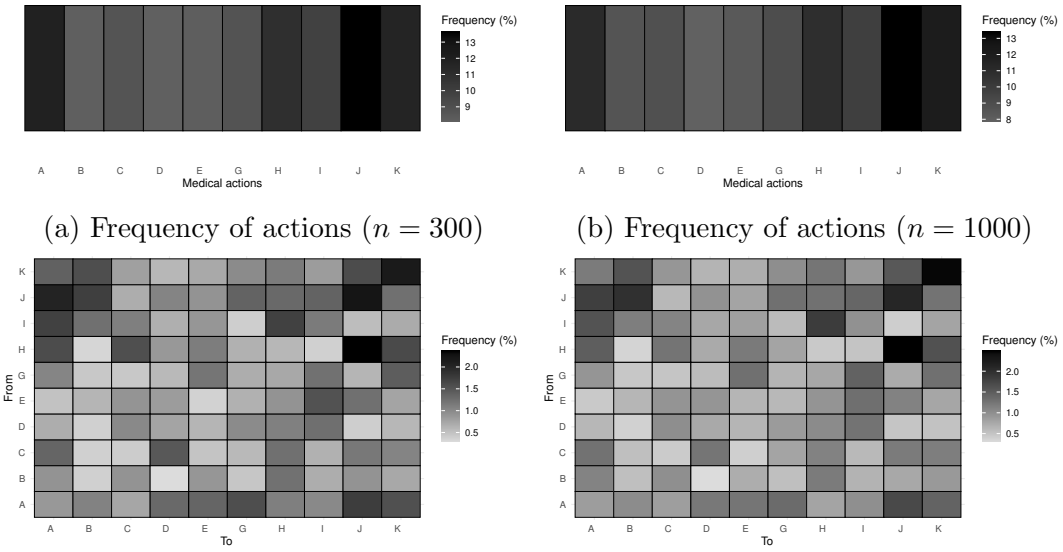
(c) Frequency of transitions ($n = 300$) (d) Frequency of transitions ($n = 1000$)

Figure A.2: Experiment 1: frequency of actions and their transitions.



(a) $n = 300$ (b) $n = 1000$

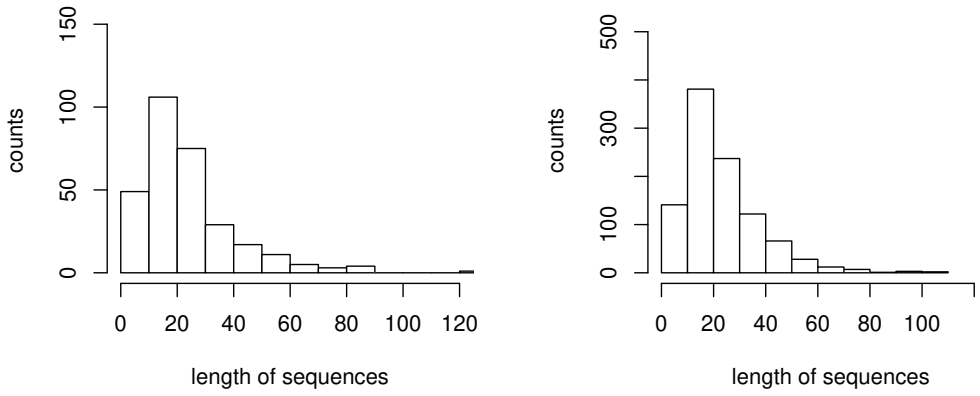
Figure A.3: Experiment 2: histogram of the lengths of the sequences of actions.



(a) Frequency of actions ($n = 300$) (b) Frequency of actions ($n = 1000$)

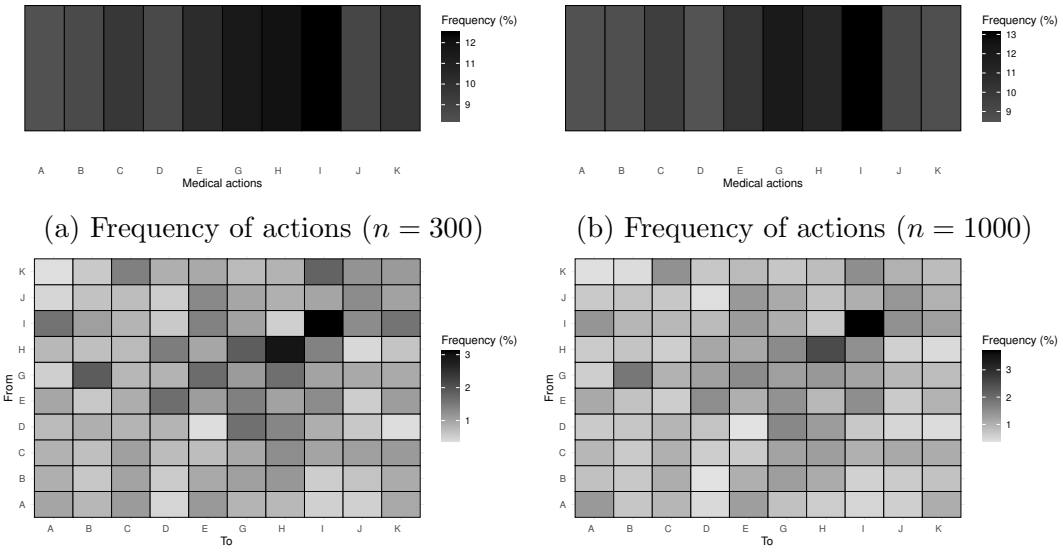
(c) Frequency of transitions ($n = 300$) (d) Frequency of transitions ($n = 1000$)

Figure A.4: Experiment 2: frequency of actions and their transitions.



(a) $n = 300$ (b) $n = 1000$

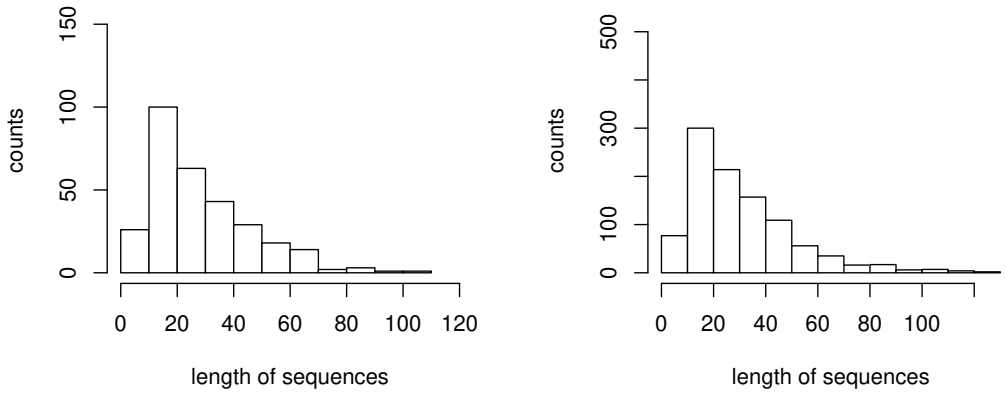
Figure A.5: Experiment 3: histogram of the lengths of the sequences of actions.



(a) Frequency of actions ($n = 300$) (b) Frequency of actions ($n = 1000$)

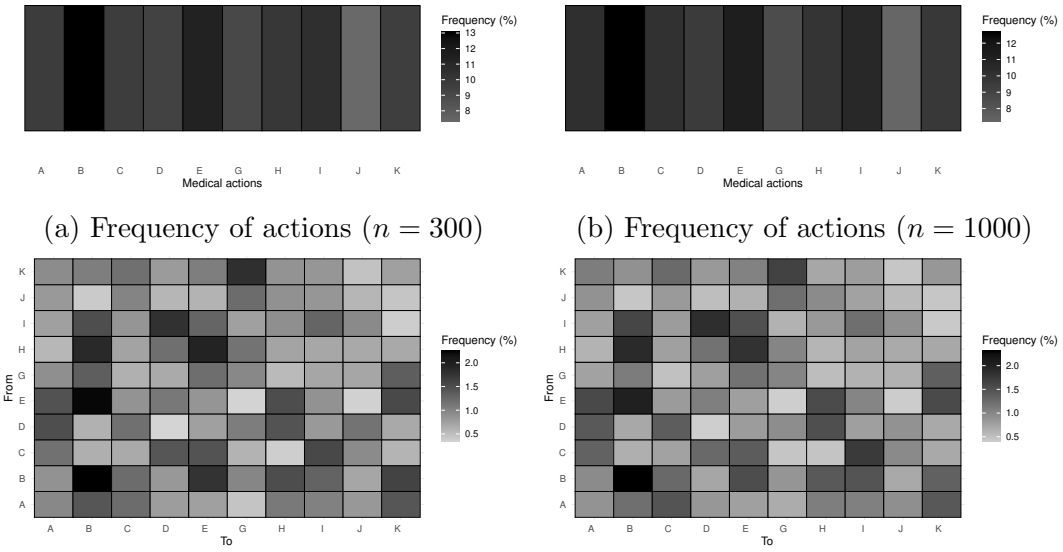
(c) Frequency of transitions ($n = 300$) (d) Frequency of transitions ($n = 1000$)

Figure A.6: Experiment 3: frequency of actions and their transitions.



(a) $n = 300$ (b) $n = 1000$

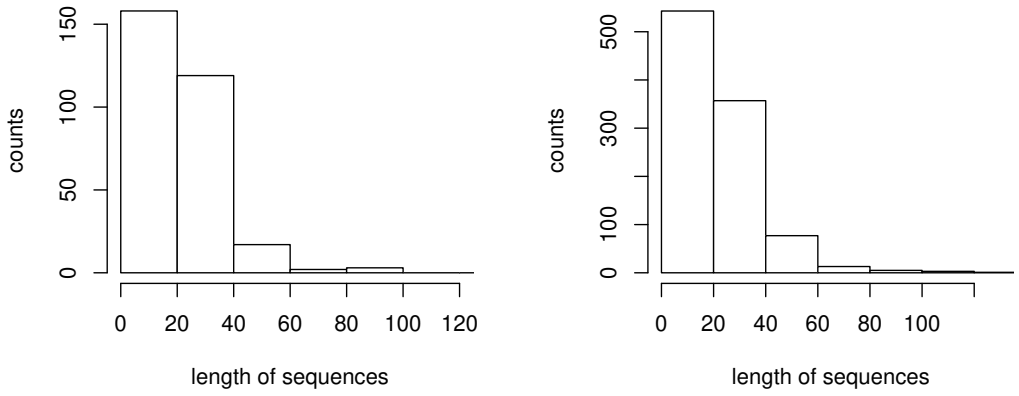
Figure A.7: Experiment 4: histogram of the lengths of the sequences of actions.



(a) Frequency of actions ($n = 300$) (b) Frequency of actions ($n = 1000$)

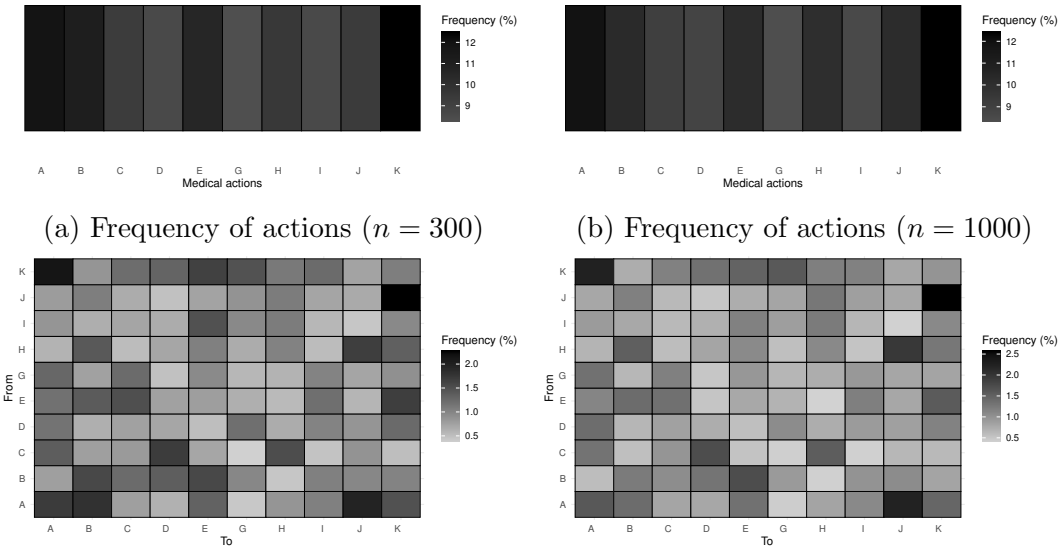
(c) Frequency of transitions ($n = 300$) (d) Frequency of transitions ($n = 1000$)

Figure A.8: Experiment 4: frequency of actions and their transitions.



(a) $n = 300$ (b) $n = 1000$

Figure A.9: Experiment 5: histogram of the lengths of the sequences of actions.



(a) Frequency of actions ($n = 300$) (b) Frequency of actions ($n = 1000$)

(c) Frequency of transitions ($n = 300$) (d) Frequency of transitions ($n = 1000$)

Figure A.10: Experiment 5: frequency of actions and their transitions.

A.3 Heterogeneity in sequences of real EHRs

This appendix shows the frequency of medical actions and their transitions in real EHRs. Then, we represent these frequencies within each class of treatments that we obtained in the experiment of Section 3.4.2.3.

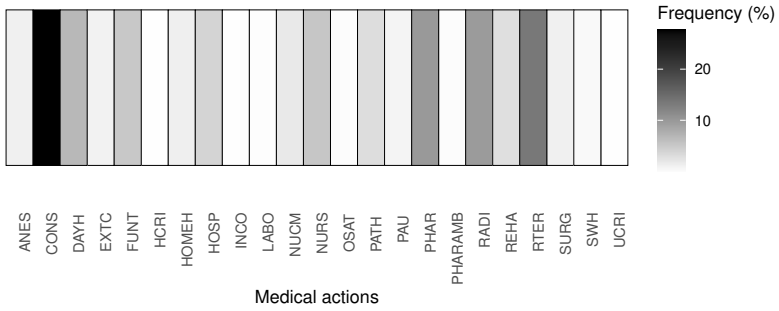


Figure A.11: Frequency (%) of medical actions in real EHRs.

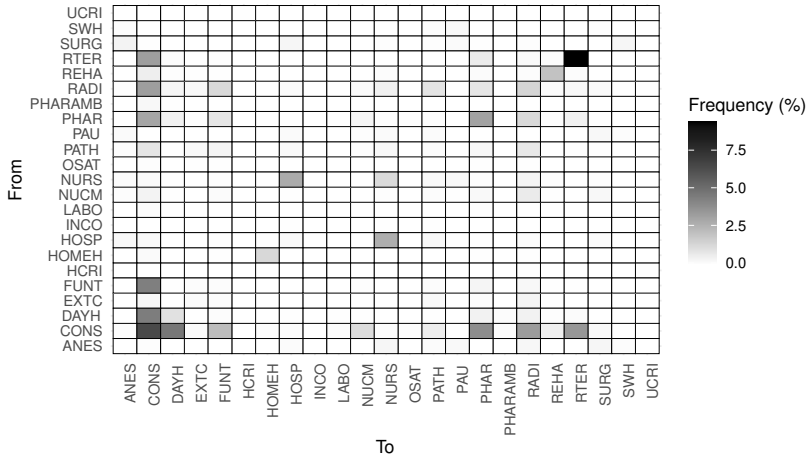
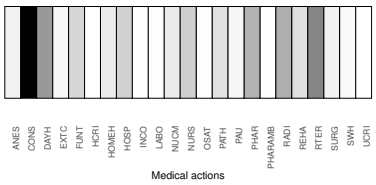


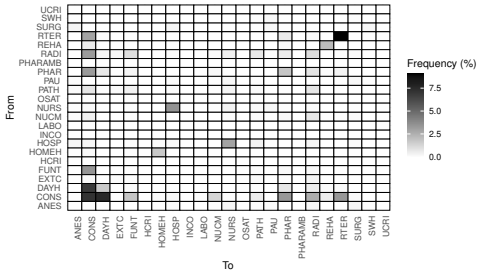
Figure A.12: Frequency (%) of the transitions between medical actions in real EHRs.

A.3.1 Inter-class heterogeneity

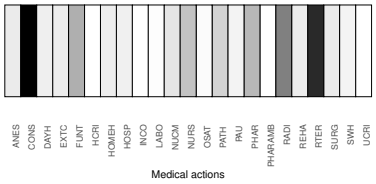
The objective of Figure A.13 is to show the variety of medical actions that can typically be executed for each class, as well as the transition between them.



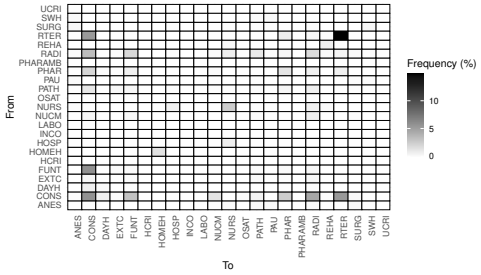
(a) Class 1: Frequency (%) of actions.



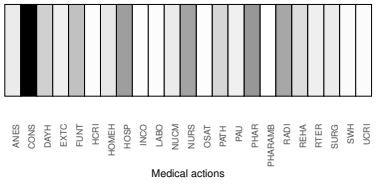
(b) Class 1: Frequency (%) of transitions.



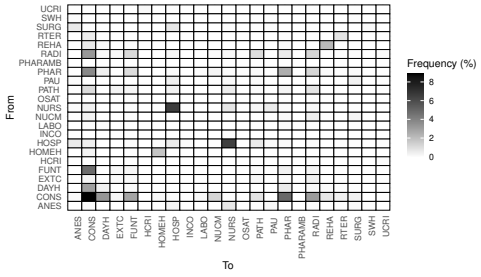
(c) Class 2: Frequency (%) of actions.



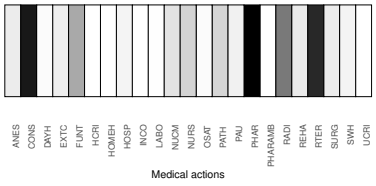
(d) Class 2: Frequency (%) of transitions.



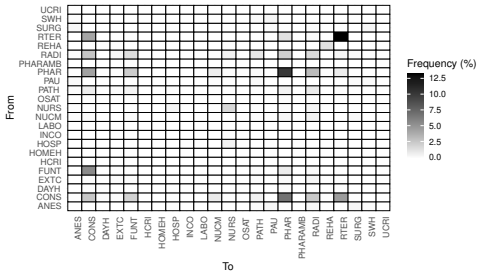
(e) Class 3: Frequency (%) of actions.



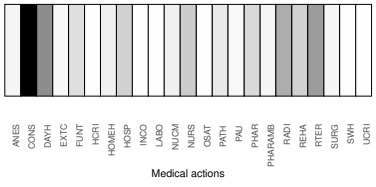
(f) Class 3: Frequency (%) of transitions.



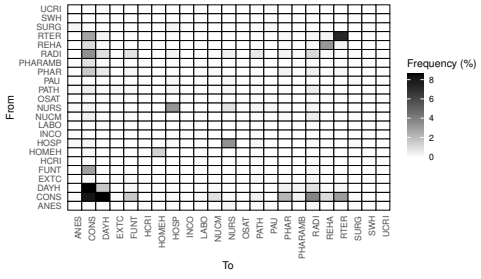
(g) Class 4: Frequency (%) of actions.



(h) Class 4: Frequency (%) of transitions.



(i) Class 5: Frequency (%) of actions.



(j) Class 5: Frequency (%) of transitions.

Figure A.13: Frequency (%) of actions and their transitions of each class.

Appendix B

Time-dependent probabilistic generative models for disease progression

B.1 Efficient inference based on dynamic programming

Exact parameter learning of a generative model can be computationally expensive for long sequences. We adapt the learning procedure developed in Section 3.3.2.1 to the specific characteristics of the time-dependent generative model described in Equation (4.1). In this case, we need to find the posterior distribution of the latent variables, $p(\boldsymbol{s}, c|\boldsymbol{a}, \boldsymbol{\tau})$. We then use this posterior distribution to evaluate the expectation of the logarithm of the complete-data likelihood function in Equation (B.1), as a function of the parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_A, \boldsymbol{\theta}_T, \boldsymbol{\theta}_S, \boldsymbol{\theta}_C\}$:

$$\max_{\boldsymbol{\theta}} \sum_{(\boldsymbol{a},\boldsymbol{\tau})\in D} \sum_{\boldsymbol{s}\in\mathcal{S}_{\boldsymbol{a}}} \sum_{c\in C} p(\boldsymbol{s}, c|\boldsymbol{a}, \boldsymbol{\tau}) \cdot \log p(\boldsymbol{a}, \boldsymbol{\tau}, \boldsymbol{s}, c; \boldsymbol{\theta}) \tag{B.1}$$

where $\mathcal{S}_{\boldsymbol{a}}$ is the set of all the potential sequences of stages for $\boldsymbol{a}$.

Let us assume that we have a training set $\mathcal{D} = \{(\boldsymbol{a}^i, \boldsymbol{\tau}^i)\}_{i=1}^N$ that consists of a set of treatments $\boldsymbol{a} = (a_1, ..., a_m)$ and their corresponding sequences of time intervals $\boldsymbol{\tau} = (\tau_1, ..., \tau_m)$. Let consider the underlying sequence of latent stages $\boldsymbol{s} = (s_1, ..., s_m)$ where $s_i \in S$, and a latent variable of classes $c \in C$ for each pair $(\boldsymbol{a}, \boldsymbol{\tau}) \in \mathcal{D}$. We aim to estimate the maximum likelihood parameters $\boldsymbol{\theta}$ of the model in each iteration of the EM algorithm.

In the E-step, we compute the expected values of the latent variables, which can be thought of as the probabilities of each possible stage $s \in S$ at time i in each possible class $c \in C$. That is, the probability of all the sequences of stages with the form $(s_1, ..., s_{i-1}, s, s_{i+1}, ..., s_m)$ in c.

Let us assume that $f_c(i,s)$ is the sum of the probabilities of all the sequences of stages $(s_1,...,s_i)$ in the class c that end at $s_i = s$, and $g_c(i,s)$ is the sum of the probabilities of all the sequences of stages $(s_{i+1},...,s_m)$ that start at $s_i = s$ in the class c. Then,

$$f_c(i,s) = \sum_{\boldsymbol{s}_{1:i}} p(\boldsymbol{s}_{1:i}, \boldsymbol{a}_{1:i}, \boldsymbol{\tau}_{1:i}|c) \tag{B.2}$$

$$g_c(i,s) = \sum_{\boldsymbol{s}_{i+1:m}} p(\boldsymbol{s}_{i+1:m}, \boldsymbol{a}_{i+1:m}, \boldsymbol{\tau}_{i+1:m}|s,c), \tag{B.3}$$

where $\boldsymbol{a}_{j:k} = (a_j,....,a_k)$, $\boldsymbol{\tau}_{j:k} = (\tau_j,...,\tau_k)$ and $\boldsymbol{s}_{j:k} = (s_j,....,s_k)$.

Now, we can express the sum of the probabilities of the sequences for which $\boldsymbol{s}_{i-1,i} = (s,s')$ as

$$p(s_{i-1} = s, s_i = s'|\boldsymbol{a},\boldsymbol{\tau},c) = \frac{p(s_{i-1} = s, s_i = s', \boldsymbol{a},\boldsymbol{\tau}|c)}{p(\boldsymbol{a},\boldsymbol{\tau}|c)}$$

Using Equations (B.2) and (B.3),

$$\begin{aligned}
&p(s_{i-1} = s, s_i = s', \boldsymbol{a},\boldsymbol{\tau}|c) = \\
&\quad = \sum_{\substack{\boldsymbol{s}_{1:i-2} \\ \boldsymbol{s}_{i+1:m}}} p(\boldsymbol{s}_{1:i-2}, s_{i-1} = s, \boldsymbol{a}_{1:i-1}, \boldsymbol{\tau}_{1:i-1}|c) \cdot p(a_i|a_{i-1}, s_{i-1} = s, c) \cdot \\
&\qquad\qquad \cdot p(s_i = s'|a_i, s_{i-1} = s, c) \cdot p(\tau_i|a_{i-1}, a_i, c) \cdot p(\boldsymbol{s}_{i+1:m}, \boldsymbol{a}_{i:m}, \boldsymbol{\tau}_{i:m}|s_i = s', c) \\
&\quad = f_c(i-1,s) \cdot p(a_i|a_{i-1}, s_{i-1} = s, c) \cdot p(s_i = s'|a_i, s_{i-1} = s, c) \cdot p(\tau_i|a_{i-1}, a_i, c) \cdot g_c(i,s')
\end{aligned}$$

We propose to create a matrix associated with each function f and g. These functions are defined as recursive functions:

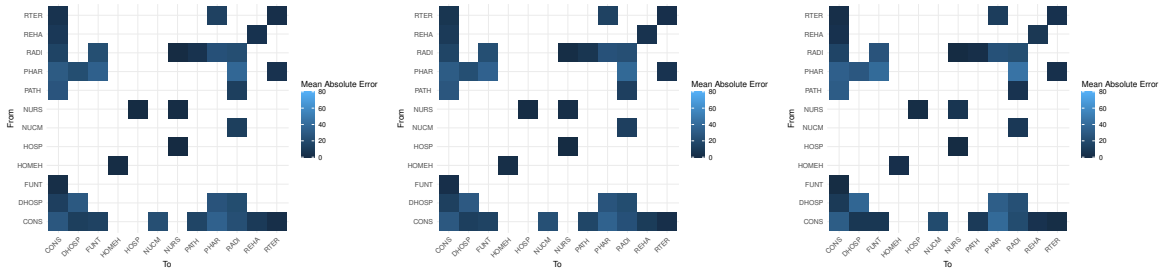
$$\begin{aligned}
f_c(i,s) =& f_c(i-1,s) \cdot p(a_i|a_{i-1}, s-1, c) \cdot p(s|a_i, s, c) \cdot p(\tau_i|a_{i-1}, a_i, c) \\
&+ f_c(i-1,s-1) \cdot p(a_i|a_{i-1}, s-1, c) \cdot p(s|a_i, s-1, c) \cdot p(\tau_i|a_{i-1}, a_i, c) \\
g_c(i,s) =& g_c(i+1,s+1) \cdot p(a_{i+1}|a_i, s+1, c) \cdot p(s+1|a_{i+1}, s, c) \cdot p(\tau_{i+1}|a_i, a_{i+1}, c) \\
&+ g_c(i+1,s) \cdot p(a_{i+1}|a_i, s, c) \cdot p(s|a_{i+1}, s, c) \cdot p(\tau_{i+1}|a_i, a_{i+1}, c)
\end{aligned}$$

The functions f_c and g_c are defined in such a way that the stages are non-decreasing. The dynamic programming method significantly reduces the number of computations for the parameter estimation. In essence, rather than independently computing the probability for all possible combinations of $(\boldsymbol{a},\boldsymbol{\tau},\boldsymbol{s},c)$, the dynamic programming approach reuses the transition probabilities that the sequences share.

Finally, note that, to model the time, we use the cumulative distribution function $F(x;\boldsymbol{\theta})$ for the exponential and Weibull distributions, given their continuous nature. In these cases, $p(\tau|a,a',c)$ is computed as $1 - F(\tau;\boldsymbol{\theta}_T)$. However, for the geometric distribution, we use the probability density function, that is, $p(\tau|a,a',c) = f(\tau;\boldsymbol{\theta}_T)$.

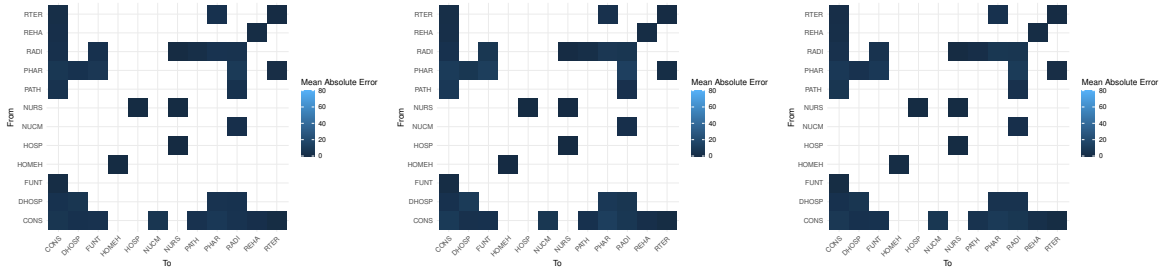
B.2 Time prediciton error in real EHRs

This appendix presents the mean absolute errors for time interval predictions in Figures B.1, B.2 and B.3. These mean absolute errors are calculated for the most frequent transitions between pairs of medical specialties, allowing us to demonstrate the improvements in predictions made by our model compared to empirical parametric methods. In the following figures, lighter blue squares indicate higher predictive errors.



(a) Geometric distribution (empirical) (b) Exponential distribution (empirical) (c) Weibull distribution (empirical)

Figure B.1: Heatmap of the mean absolute errors of the prediction of time intervals using the empirical distributions.



(a) Geometric distribution (our model) (b) Exponential distribution (our model) (c) Weibull distribution (our model)

Figure B.2: Heatmap of the mean absolute errors of the prediction of time intervals using the proposed generative model. The results of our model are obtained from the mixture of classes Equation (4.8).

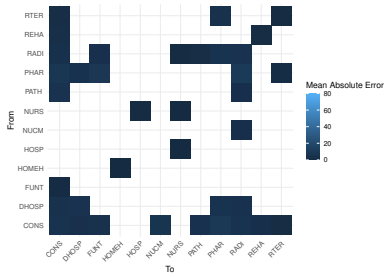


Figure B.3: Heatmap of the mean absolute errors of the prediction of time intervals using the non-parametric model (median).

Appendix C

Probabilistic generative model for comorbidity progression

C.1 Requirements for the transitions between medical actions

This appendix outlines the requirements necessary for the transition between pairs of medical actions. Recall that transitions between medical actions are only allowed if they are associated with the same diagnosis. Therefore, for the transition from $a_{t'} = a'$ to $a_t = a$ it must be satisfied that the sequence of latent diagnosis has the form

$$(d_1, ...d_{t'-1}, d, d_{t'+1}, ..., d_{t-1}, d, d_{t+1}, ..., d_m)$$

where $d_{t'+1}, ..., d_{t-1} \neq d$.

In terms of the recursive Equations 5.4 and 5.5, we have that

$$\begin{aligned} p(d_t = d, \boldsymbol{a}, \boldsymbol{s}|c) = \\ = f_c(t_1, ..., t_i = t', ..., t_r) \cdot p(s_t|s_{t-1}, d_{t-1}, a_{t-1}) \cdot p(d_t = d|s_t, c) \cdot p(a_{d:t}|a_{d:t'}, d_t = d) \\ \cdot g_c(t_1, ..., t_i = t, ..., t_r) \end{aligned}$$

where t' is the previous time where the same disease d is allocated. To account for the constraints on the set of possible configurations in the sequences of diagnosis, $\mathcal{D}_{\boldsymbol{a}}$, in the computation of the matrices f_c and g_c using the proposed dynamic programming-based method, we follow the subsequent procedure:

If a is observed at time t in the sequence $\boldsymbol{a}$, let t' be the set of times such that we can find the action a' in the subsequence $\boldsymbol{a}_{1,...,t-1}$, that is, $t' = \{y < t : a_y = a'\}$. Let $T = (t_1, ..., t_r)$ be the vector that indicates the last time each type of disease $d^i \in D$, $i = 1, ..., r$, appears in the sequence $\boldsymbol{d} = (d_1, ..., d_t)$, and let $h = \max t' = \max\{y < t : a_y = a'\}$.

For each time t where the action a is observed, and for each disease $d^i \in D$, $i = 1, ..., r$, the two following options can occur:

1. If $t - h > 1$:

   1.1) If at least one disease has already finished before t:

   For the set of finished diseases before t, $d^f \in D$, we use their endpoint in the sequence $\boldsymbol{d}$ to set in the vector T the last time we have seen that disease.

   For the set of unfinished diseases that are already initialized, we fix each disease, $d^j \in D$, at $t_j = t - 1$ $(j \neq f, i)$ while we set $t_{r'} = 0, ..., t - 2$ in the rest of the unfinished diseases ($d^{r'} \in D$, $r' \neq j, i, f$). Take into account that if any disease's endpoint is fixed at $t - 1$, we do not have to set any unfinished disease t_j in $t - 1$, rather they are all fixed at $t_{r'} = 0, ..., t - 2$ $(r' \neq i, f)$.

   For those diseases that have not already been initialized their position in T is fixed at 0.

   1.2) If no disease has finished before t, we fix for each disease $d^j \in D$ their last position in T as $t_j = t - 1$ and the rest of the initialized diseases' position at $t_{r'} = 0, ..., t - 2$ $(r' \neq j, i)$.

   Let $J = \{1, ..., i - 1, i + 1, ..., r\}$, then we can compute $p(d_t = d^i, \boldsymbol{a}, \boldsymbol{s}|c)$ as

$$\sum_{y \in t'} \sum_{j \in J} \sum_{\substack{t_1,...,t_r \\ t_j = t-1 \\ t_{r'}, r' \neq j,i,f}} f_c(t_1, ..., t_i = y, ..., t_r) \cdot p(s_t|s_{t-1}, d_{t-1}, a_{t-1}) \cdot p(d_t = d^i|c, s_t) \cdot p(a_t|a_{d^i:t'} = a', d_t = d^i) \cdot g_c(t_1, ..., t_i = t, ..., t_r)$$

2. If $t - h = 1$:
   We fix for each disease $d^j \in D$ their position t_j at the maximum position t', that is, $t_j = h$. In addition, $t_{r'} = 0, ..., t - 2$ for all $r' \neq i, j$. Then, let $J = \{1, ..., i - 1, i + 1, ..., r\}$, then we can compute $p(d_t = d^i, \boldsymbol{a}, \boldsymbol{s}|c)$ as

$$\sum_{y \in t'} \sum_{j \in J} \sum_{\substack{t_1,...,t_r \\ t_j = h \\ t_{r'}, r' \neq j}} f_c(t_1, ..., t_i = y, ..., t_r) \cdot p(s_t|s_{t-1}, d_{t-1}, a_{t-1}) \cdot p(d_t = d^i|c, s_t) \cdot p(a_t|a_{d^i:t'} = a', d_t = d^i) \cdot g_c(t_1, ..., t_i = t, ..., t_r)$$

C.2 Lagrange multiplier method

This appendix describes the process followed to obtain the update of the model parameters of the comorbidity progression model.

We consider $\boldsymbol{a} = (a_1, ..., a_m)$ to be the observed data, $\boldsymbol{s} = (s_1, ..., s_m)$ the corresponding sequence of active disease states, $\boldsymbol{d} = (d_1, ..., d_m)$ the latent sequence of diagnosis and c the latent class. We aim to obtain the model parameters $\boldsymbol{\theta}$ that maxime the following function,

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{\boldsymbol{d} \in \mathcal{D}_{\boldsymbol{a}}} \sum_{c \in C} p(\boldsymbol{d}, c|\boldsymbol{a}, \boldsymbol{s}) \cdot \log p(\boldsymbol{a}, \boldsymbol{s}, \boldsymbol{d}, c; \boldsymbol{\theta}) \tag{C.1}$$

where $\mathcal{D}_{\boldsymbol{a}}$ is the set of all the potential sequences of diagnosis for $\boldsymbol{a}$, and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_A, \boldsymbol{\theta}_S, \boldsymbol{\theta}_D, \boldsymbol{\theta}_C\}$.

From Equation (C.1), we obtain

$$\begin{aligned}
&\sum_{\boldsymbol{d} \in \mathcal{D}_{\boldsymbol{a}}} \sum_{c \in C} p(\boldsymbol{d}, c|\boldsymbol{a}, \boldsymbol{s}) \cdot \log p(\boldsymbol{a}, \boldsymbol{s}, c; \boldsymbol{\theta}) \\
&= \sum_{\boldsymbol{d} \in \mathcal{D}_{\boldsymbol{a}}} \sum_{c \in C} p(\boldsymbol{d}, c|\boldsymbol{a}, \boldsymbol{s}) \cdot \log \left(p(c) \prod_{t=1}^{m} p(s_t|s_{t-1}, a_{t-1}, d_{t-1}) \cdot p(d_t|s_t, c) \cdot p(a_{d:t}|a_{d:t'}, d_t) \right) \\
&= \sum_{\boldsymbol{d} \in \mathcal{D}_{\boldsymbol{a}}} \sum_{c \in C} p(\boldsymbol{d}, c|\boldsymbol{a}, \boldsymbol{s}) \left(\log p(c) + \sum_{t=1}^{m} \log p(s_t|s_{t-1}, a_{t-1}, d_{t-1}) + \sum_{t=1}^{m} \log p(d_t|s_t, c) \right. \\
&\qquad \left. + \sum_{t=1}^{m} \log p(a_{d:t}|a_{d:t'}, d_t) \right)
\end{aligned} \tag{C.2}$$

Since the parameters we want to optimize are now independently split into four terms in the sum, we can optimize them individually.

For the first term in Equation (C.2),

$$\begin{aligned}
\sum_{\boldsymbol{d} \in \mathcal{D}_{\boldsymbol{a}}} \sum_{c \in C} p(\boldsymbol{d}, c|\boldsymbol{a}, \boldsymbol{s}) \sum_{t=1}^{m} \log p(c) &= \sum_{t=1}^{m} \sum_{c \in C} \sum_{d \in D} p(d_t = d, c|\boldsymbol{a}, \boldsymbol{s}) \log \theta_c \\
&= \sum_{c \in C} p(c|\boldsymbol{a}, \boldsymbol{s}) \log \theta_c
\end{aligned}$$

To maximize with respect to θ_c we introduce the Lagrange multipliers ε. The Lagrangian is then given by:

$$L(\boldsymbol{\theta}_C) = \sum_{c \in C} p(c|\boldsymbol{a}, \boldsymbol{s}) \log \theta_c + \varepsilon (\sum_{c \in C} \theta_c - 1)$$

where $\sum_{c \in C} \theta_c = 1$. We get $p(c|\boldsymbol{a}, \boldsymbol{s})$ from the E-step and use it as a constant. Setting the derivative equal to zero, we obtain:

$$\frac{\partial L(\boldsymbol{\theta}_C)}{\partial \theta_c} = \frac{p(c|\boldsymbol{a}, \boldsymbol{s})}{\theta_c} + \varepsilon = 0 \implies \varepsilon = -\frac{p(c|\boldsymbol{a}, \boldsymbol{s})}{\theta_c} \tag{C.3}$$

Multiplying each side by θ_c and summing over $c \in C$, we obtain that

$$\varepsilon = -\sum_{c \in C} p(c|\boldsymbol{a}, \boldsymbol{s}) \tag{C.4}$$

From Equations (C.3) and (C.4), we obtain

$$\hat{\theta}_c = \frac{p(c|\boldsymbol{a},\boldsymbol{s})}{\sum_{c\in C} p(c|\boldsymbol{a},\boldsymbol{s})} = \frac{\sum_{t=1}^{m}\sum_{d\in D} p(d_t=d,c|\boldsymbol{a},\boldsymbol{s})}{\sum_{c\in C}\sum_{d\in D}\sum_{t=1}^{m} p(d_t=d,c|\boldsymbol{a},\boldsymbol{s})}$$

For the second term in Equation (C.2),

$$\sum_{\boldsymbol{d}\in\mathcal{D}_{\boldsymbol{a}}}\sum_{c\in C} p(\boldsymbol{d},c|\boldsymbol{a},\boldsymbol{s})\sum_{t=1}^{m}\log p(s_t|s_{t-1},d_{t-1},a_{t-1}) =$$
$$= \sum_{t=1}^{m}\sum_{s'\in S}\sum_{d\in D}\sum_{a\in A}\sum_{s\in S} p(s_{t-1}=s',d_{t-1}=d',a_{t-1}=a,s_t=s)\log\theta_s^{s',d,a}$$

To maximize with respect to $\theta_s^{s',d,a}$ we introduce the Lagrange multipliers $\lambda_{s',d,a}$ for $s'\in S$, $d\in D$ and $a\in A$. The Lagrangian is then given by:

$$L(\boldsymbol{\theta}_S) = \sum_{t=1}^{m}\sum_{s'\in S}\sum_{d\in D}\sum_{a\in A}\sum_{s\in S} p(s_{t-1}=s',d_{t-1}=d,a_{t-1}=a,s_t=s)\log\theta_s^{s',d,a} \tag{C.5}$$
$$+\sum_{s'\in S}\sum_{d\in D}\sum_{a\in A}\lambda_{s',d,a}\Big(\sum_{s\in S}\theta_s^{s',d,a}-1\Big)$$

where $\sum_{s\in S}\theta_s^{s',d,a}=1$. We get $p(s_{t-1}=s',d_{t-1}=d,a_{t-1}=a,s_t=s)$ from the E-step and use it as a constant. Setting the derivative equal to zero, we obtain:

$$\frac{\partial L(\boldsymbol{\theta}_S)}{\partial\theta_s^{s',d,a}} = \frac{\sum_{t=1}^{m} p(s_{t-1}=s',d_{t-1}=d,a_{t-1}=a,s_t=s)}{\theta_s^{s',d,a}} + \lambda_{s',d,a} = 0$$
$$\implies \lambda_{s',d,a} = -\frac{\sum_{t=1}^{m} p(s_{t-1}=s',d_{t-1}=d,a_{t-1}=a,s_t=s)}{\theta_s^{s',d,a}} \tag{C.6}$$

Multiplying each side by $\theta_s^{s',d,a}$ and summing over $s\in S$, we obtain that

$$\lambda_{s',d,a} = -\sum_{s\in S}\sum_{t=1}^{m} p(s_{t-1}=s',d_{t-1}=d,a_{t-1}=a,s_t=s) \tag{C.7}$$

From Equations (C.6) and (C.7), we obtain

$$\hat{\theta}_s^{s',d,a} = \frac{\sum_{t=1}^{m}\mathbb{1}(a_{t-1}=a,s_{t-1}=s',s_t=s)\cdot p(d_{t-1}=d|\boldsymbol{a},\boldsymbol{s})}{\sum_{s\in S}\sum_{t=1}^{m}\mathbb{1}(a_{t-1}=a,s_{t-1}=s',s_t=s)\cdot p(d_{t-1}=d|\boldsymbol{a},\boldsymbol{s})}$$

Similarly, we obtain the update for the third and last term in Equation (C.2). The update of the corresponding model parameters are as follows:

$$\hat{\theta}_d^{s,c} = \frac{\sum_{t=1}^{m}\mathbb{1}(s_t=s)\cdot p(d_t=d,c|\boldsymbol{a},\boldsymbol{s})}{\sum_{d\in D}\sum_{t=1}^{m}\mathbb{1}(s_t=s)\cdot p(d_t=d,c|\boldsymbol{a},\boldsymbol{s})}$$
$$\hat{\theta}_a^{d,a'} = \frac{\sum_{t=1}^{m}\sum_{t'<t}\mathbb{1}(a_{d:t}=a,a_{d:t'}=a')\cdot p(d_t=d|\boldsymbol{a},\boldsymbol{s})}{\sum_{a\in A}\sum_{t=1}^{m}\sum_{t'<t}\mathbb{1}(a_{d:t}=a,a_{d:t'}=a')\cdot p(d_t=d|\boldsymbol{a},\boldsymbol{s})}$$

Bibliography

[1] Tabinda Sarwar, Sattar Seifollahi, Jeffrey Chan, Xiuzhen Zhang, Vural Aksakalli, Irene Hudson, Karin Verspoor, and Lawrence Cavedon. The secondary use of electronic health records for data mining: Data characteristics and challenges. *ACM Computing Surveys (CSUR)*, 55(2):1–40, 2022.

[2] Irene Y Chen, Shalmali Joshi, Marzyeh Ghassemi, and Rajesh Ranganath. Probabilistic machine learning for healthcare. *Annual review of biomedical data science*, 4:393–415, 2021.

[3] Bjoern M Eskofier and Jochen Klucken. Predictive models for health deterioration: Understanding disease pathways for personalized medicine. *Annual Review of Biomedical Engineering*, 25:131–156, 2023.

[4] Peter B Jensen, Lars J Jensen, and Søren Brunak. Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*, 13(6):395, 2012.

[5] Eric Rojas, Jorge Munoz-Gama, Marcos Sepúlveda, and Daniel Capurro. Process mining in healthcare: A literature review. *Journal of biomedical informatics*, 61:224–236, 2016.

[6] Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3(1):160035, 2016.

[7] Tom J Pollard, Alistair E W Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. The eICU Collaborative Research Database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):1–13, 2018.

[8] World Health Organization. *ICD-10: international statistical classification of diseases and related health problems: tenth revision*. World Health Organization, 2 edition, 2004.

[9] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional re-

current neural networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1903–1911, 2017.

[10] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine learning for healthcare conference*, pages 301–318. PMLR, 2016.

[11] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in neural information processing systems*, volume 29, 2016.

[12] Ahmed M Alaa and Mihaela van der Schaar. Attentive state-space modeling of disease progression. *Advances in neural information processing systems*, 32, 2019.

[13] Xian Teng, Sen Pei, and Yu-Ru Lin. Stocast: Stochastic disease forecasting with progression uncertainty. *IEEE Journal of Biomedical and Health Informatics*, 25(3):850–861, 2020.

[14] Trang Pham, Truyen Tran, Dinh Phung, and Svetha Venkatesh. Predicting healthcare trajectories from medical records: A deep learning approach. *Journal of biomedical informatics*, 69:218–229, 2017.

[15] Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6(1):1–10, 2016.

[16] Huilong Duan, Zhoujian Sun, Wei Dong, Kunlun He, and Zhengxing Huang. On clinical event prediction in patient treatment trajectory using longitudinal electronic health records. *IEEE Journal of Biomedical and Health Informatics*, 24(7):2053–2063, 2019.

[17] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

[18] Zhaohong Sun, Zhoujian Sun, Wei Dong, Jinlong Shi, and Zhengxing Huang. Towards predictive analysis on disease progression: a variational hawkes process model. *IEEE Journal of Biomedical and Health Informatics*, 25(11):4195–4206, 2021.

[19] Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. Deepr: a convolutional net for medical records. *IEEE journal of biomedical and health informatics*, 21(1):22–30, 2016.

[20] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Michaela Hardt, Peter J Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, et al. Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, 1(1):1–10, 2018.

[21] Benjamin Shickel, Patrick James Tighe, Azra Bihorac, and Parisa 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*, 22(5):1589–1604, 2017.

[22] Liang Zhao. Event prediction in the big data era: A systematic survey. *ACM Computing Surveys (CSUR)*, 54(5):1–37, 2021.

[23] Gerard Martí-Juan, Gerard Sanroma-Guell, and Gemma Piella. A survey on machine and statistical learning for longitudinal analysis of neuroimaging data in alzheimer's disease. *Computer methods and programs in biomedicine*, 189:105348, 2020.

[24] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[25] Yanshan Wang, Yiqing Zhao, Terry M Therneau, Elizabeth J Atkinson, Ahmad P Tafti, Nan Zhang, Shreyasee Amin, Andrew H Limper, Sundeep Khosla, and Hongfang Liu. Unsupervised machine learning for the discovery of latent disease clusters and patient subgroups using electronic health records. *Journal of biomedical informatics*, 102:103364, 2020.

[26] Zhengxing Huang, Wei Dong, Lei Ji, Chenxi Gan, Xudong Lu, and Huilong Duan. Discovery of clinical pathway patterns from event logs using probabilistic topic models. *Journal of biomedical informatics*, 47:39–57, 2014.

[27] Zhengxing Huang, Wei Dong, Peter Bath, Lei Ji, and Huilong Duan. On mining latent treatment patterns from electronic medical records. *Data mining and knowledge discovery*, 29(4):914–949, 2015.

[28] Jingfeng Chen, Leilei Sun, Chonghui Guo, and Yanming Xie. A fusion framework to extract typical treatment patterns from electronic medical records. *Artificial Intelligence in Medicine*, 103:101782, 2020.

[29] Pranjul Yadav, Michael Steinbach, Vipin Kumar, and Gyorgy Simon. Mining electronic health records (ehrs) a survey. *ACM Computing Surveys (CSUR)*, 50(6):1–40, 2018.

[30] Xiang Wang, David Sontag, and Fei Wang. Unsupervised learning of disease progression models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 85–94, 2014.

[31] Bryan Lim and Mihaela van der Schaar. Disease-atlas: Navigating disease trajectories using deep learning. In *Machine Learning for Healthcare Conference*, pages 137–160. PMLR, 2018.

[32] Zhaozhi Qian, Ahmed Alaa, Alexis Bellot, Mihaela Schaar, and Jem Rashbass. Learning dynamic and personalized comorbidity networks from event data using deep diffusion processes. In *International Conference on Artificial Intelligence and Statistics*, pages 3295–3305. PMLR, 2020.

[33] Edward Choi, Nan Du, Robert Chen, Le Song, and Jimeng Sun. Constructing disease network and temporal progression model via context-sensitive hawkes process. In *2015 IEEE International Conference on Data Mining*, pages 721–726. IEEE, 2015.

[34] Basil Maag, Stefan Feuerriegel, Mathias Kraus, Maytal Saar-Tsechansky, and Thomas Züger. Modeling longitudinal dynamics of comorbidities. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 222–235, 2021.

[35] Kristen A Severson, Lana M Chahine, Luba Smolensky, Kenney Ng, Jianying Hu, and Soumya Ghosh. Personalized input-output hidden markov models for disease progression modeling. In *Machine Learning for Healthcare Conference*, pages 309–330. PMLR, 2020.

[36] Rafid Sukkar, Elyse Katz, Yanwei Zhang, David Raunig, and Bradley T Wyman. Disease progression modeling using hidden markov models. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2845–2848. IEEE, 2012.

[37] Christopher H Jackson, Linda D Sharples, Simon G Thompson, Stephen W Duffy, and Elisabeth Couto. Multistate markov models for disease progression with classification error. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52(2):193–209, 2003.

[38] Zhengxing Huang, Zhenxiao Ge, Wei Dong, Kunlun He, and Huilong Duan. Probabilistic modeling personalized treatment pathways using electronic health records. *Journal of biomedical informatics*, 86:33–48, 2018.

[39] Yu-Ying Liu, Shuang Li, Fuxin Li, Le Song, and James M Rehg. Efficient learning of continuous-time hidden markov models for disease progression. *Advances in neural information processing systems*, 28, 2015.

[40] Nikhil Galagali and Minnan Xu-Wilson. Patient subtyping with disease progression and irregular observation trajectories. *arXiv preprint arXiv:1810.09043*, 2018.

[41] Jaewon Yang, Julian McAuley, Jure Leskovec, Paea LePendu, and Nigam Shah. Finding progression stages in time-evolving event sequences. In *Proceedings of the 23rd international conference on World wide web*, pages 783–794, 2014.

[42] Taha Ceritli, Andrew P Creagh, and David A Clifton. Mixture of input-output hidden markov models for heterogeneous disease progression modeling. In *Workshop on Healthcare AI and COVID-19*, pages 41–53. PMLR, 2022.

[43] Chandan K Reddy. *Data clustering: algorithms and applications*. Chapman and Hall/CRC, 2018.

[44] Andres Marzal and Enrique Vidal. Computation of Normalized Edit Distance and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):926–932, 1993.

[45] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[46] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[47] Pierre A Devijver. Baum's forward-backward algorithm revisited. *Pattern Recognition Letters*, 3(6):369–373, 1985.

[48] Suchi Saria and Anna Goldenberg. Subtyping: What it is and its role in precision medicine. *IEEE Intelligent Systems*, 30(4):70–75, 2015.

[49] Zhengxing Huang, Xudong Lu, and Huilong Duan. On mining clinical pathway patterns from medical behaviors. *Artificial intelligence in medicine*, 56(1):35–50, 2012.

[50] Carlos Fernández-Llatas, Teresa Meneu, Jose Miguel Benedi, and Vicente Traver. Activity-based process mining for clinical pathways computer aided design. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 6178–6181. IEEE, 2010.

[51] Wil Van der Aalst, Ton Weijters, and Laura Maruster. Workflow mining: Discovering process models from event logs. *IEEE transactions on knowledge and data engineering*, 16(9):1128–1142, 2004.

[52] Sergey V Kovalchuk, Anastasia A Funkner, Oleg G Metsker, and Aleksey N Yakovlev. Simulation of patient flow in multiple healthcare units using process and data mining techniques for model identification. *Journal of biomedical informatics*, 82:128–142, 2018.

[53] Yiye Zhang, Rema Padman, and Nirav Patel. Paving the cowpath: Learning and visualizing clinical pathways from electronic health record data. *Journal of biomedical informatics*, 58:186–197, 2015.

[54] Geetika T Lakshmanan, Szabolcs Rozsnyai, and Fei Wang. Investigating clinical care pathways correlated with outcomes. In *Business process management*, pages 323–338. Springer, 2013.

[55] Rafał Deja, Wojciech Froelich, Grażyna Deja, and Alicja Wakulicz-Deja. Hybrid approach to the generation of medical guidelines for insulin therapy for children. *Information Sciences*, 384:157–173, 2017.

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

[57] Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095, 2007.

[58] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.

[59] F Cardoso, S Kyriakides, S Ohno, F Penault-Llorca, P Poortmans, I T Rubio, S Zackrisson, and E Senkus. Early breast cancer: ESMO Clinical Practice Guidelines for diagnosis, treatment and follow-up? *Annals of Oncology*, 30(8):1194–1220, 2019.

[60] F Cardoso, E Senkus, A Costa, E Papadopoulos, M Aapro, F André, N Harbeck, B Aguilar Lopez, CH9 Barrios, J Bergh, et al. 4th eso–esmo international consensus guidelines for advanced breast cancer (abc 4). *Annals of Oncology*, 29(8):1634–1657, 2018.

[61] Pierre-François Marteau. Time warp edit distance with stiffness adjustment for time series matching. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):306–318, 2008.

[62] Preetish Rath, Gabriel Hope, Kyle Heuton, Erik B Sudderth, and Michael C Hughes. Prediction-constrained markov models for medical time series with missing data and few labels. In *NeurIPS 2022 Workshop on Learning from Time Series for Health*, 2022.

[63] Zhengping Che, Sanjay Purushotham, Guangyu Li, Bo Jiang, and Yan Liu. Hierarchical deep generative models for multi-rate multivariate time series. In *International Conference on Machine Learning*, pages 784–793. PMLR, 2018.

[64] Jerald F Lawless. *Statistical models and methods for lifetime data*. John Wiley & Sons, 2011.

[65] Hsin-Min Lu, Chih-Ping Wei, and Fei-Yuan Hsiao. Modeling healthcare data using multiple-channel latent dirichlet allocation. *Journal of biomedical informatics*, 60:210–223, 2016.

[66] Ioan Stanculescu, Christopher KI Williams, and Yvonne Freer. Autoregressive hidden markov models for the early detection of neonatal sepsis. *IEEE journal of biomedical and health informatics*, 18(5):1560–1570, 2013.

[67] Edward Choi, Cao Xiao, Walter Stewart, and Jimeng Sun. Mime: Multilevel medical embedding of electronic health records for predictive healthcare. *Advances in neural information processing systems*, 31, 2018.

[68] Edward Choi, Zhen Xu, Yujia Li, Michael Dusenberry, Gerardo Flores, Emily Xue, and Andrew Dai. Learning the graphical structure of electronic health records with graph convolutional transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 606–613, 2020.

[69] Giuseppe Curigliano, D Lenihan, M Fradley, Sarju Ganatra, A Barac, A Blaes, J Herrmann, C Porter, AR Lyon, Patrizio Lancellotti, et al. Management of cardiac disease in cancer patients throughout oncological treatment: Esmo consensus recommendations. *Annals of Oncology*, 31(2):171–190, 2020.

[70] Lei Chen and Raymond Ng. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 792–803, 2004.

[71] Siqiao Xue, Xiaoming Shi, James Zhang, and Hongyuan Mei. Hypro: A hybridly normalized probabilistic model for long-horizon prediction of event sequences. *Advances in Neural Information Processing Systems*, 35:34641–34650, 2022.

[72] Peter Malec and Melanie Schienle. Nonparametric kernel density estimation near the boundary. *Computational Statistics & Data Analysis*, 72:57–76, 2014.