Universitat Autònoma de Barcelona

PhD in Computer Science

Research line: Artificial Intelligence

Handling Missing Data in Clinical Decision Support

PhD Student: David Sánchez Pinsach
PhD Advisor: Josep Lluis Arcos Rosell
PhD admission date: October 1, 2015

Acknowledgments

Recalling all the steps done until this moment I have the sensation that I have been lucky to be surrounded by great people.

I would like to thank all the professionals from the Institut Guttmann for the private grant and the funding by project Innobrain, COMRDI-151-0017 (RIS3CAT comunitats), and Feder funds that help me to start the Ph.D. thesis. I knew that the aim was to complete the thesis during that period, but sometimes, research needs more time and support than grants or work environment provide. My gratitude also extends to Jose Maria Tormos, Eloy Opisso, Jaume Lopez, Alejandro Garcia, Marc Morell, Marta Rudilla, Imma Estella, David Hurtado, Joan Sauri, Vicky Ferrer, Roger Marsal, Merce Yuguero, Montse Bernabeu, Narda Murillo and those who already embraced new professional challenges, Maria Almenara, Mariona Gifre, Martin Sassul, Raquel Lopez, and Sara Laxe. I am extremely gratefully for your assistance and suggestions throughout this project.

I would like to express my deepest appreciation to my director Josep Lluis Arcos for all his time, support, and effort during these years, especially when the financial support finished at the Institut Guttmann. I will always remember how many times you said that the Ph.D. thesis is practically done. Thank you also for the opportunity to work in the project called Playing and Singing for the Recovering Brain: Efficacy of Enriched Social-Motivational Musical Interventions in Stroke Rehabilitation (Play&Sing), 299/C/2017, Fundació La Marató de TV3.

I would like also to thank Antoni, Jennifer, and Emma from IDIBELL, it was an absolute pleasure to work together in the Playing and Singing for the Recovering Brain project.

Last but not the least thank you to my family. To Mom, for being always extremely sincere and critical, as this kind of honesty is sometimes hard to be found. And to my wife, to accompany me since my sophomore year at university. I wouldn't be who I am now without you around this last ten years.

David Sanchez Pinsach
Calella, September 2020

Abstract

Deciding which are the best treatments is a complex task when patients suffer multiple impairments and when a multidisciplinary team is involved in the intervention. There is always more than a unique treatment option and the results sometimes can be viewed in a short period or only be capable to be measured when the treatment is finished. In this context, the design of effective Clinical Decision Support Systems (CDSS) to help clinicians to select most appropriate interventions is still a challenge.

The amount of available data is not always the same for all patients, especially in early treatment stages, hindering the inference in CDSS. To improve the capabilities of CDSS, different components are proposed within a CDSS framework for long-term treatments. A first component is focused on improving the quality of the inferences in missing data scenarios. The Dynamic Multiple Imputation (DMI) algorithm is presented as an effective methodology for data enhancement in CDSS. DMI is capable to adapt to different scenarios with a low or high percentage of missing data. Several experiments conducted reveal that DMI is competitive with regression problems. A second component is devoted to weigh confidence measures, given the uncertainty associated to missing information, by incorporating Mutual Information measures in confidence existing estimators. A third component, based on a community detection algorithm, is proposed to find relationships between clinical decisions that are not explicit. Finally, to illustrate the applicability of different proposed components, two real clinical use cases with chronic patients are presented. The first in the hospital context and the other in the home context.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

The development of Decision Support Systems (DSS) has a large tradition in the field of Artificial Intelligence, specially in the clinical domain [9, 39, 48]. The design of DSS has tackled by a variety solutions ranging from probabilistic models [40, 54], possibilistic models [7, 22], machine learning [69], or case-based reasoning [43, 38].

Nevertheless, the intrinsic complexity to clinical problems has limited the development of effective decision support systems [8]. The main reason behind such limitation is that the integration of Clinical Decision Support Systems (CDSS) into healthcare processes requires a detailed and complete understanding of the whole clinical processes involved (e.g. the decisions to be taken, the appropriate time to take a decision, or how the different clinical decisions interact among each other) [50]. An additional challenge for CDSS is the limited amount of available data in early treatment stages. Missing data may be originated by different reasons: errors during data entry; information considered as irrelevant; lack of time or resources, etc [65]. Unfortunately, these different reasons are usually not explicitly reported and bias CDSS in different ways.

In recent years the design of CDSS have resurged as consequence of the massive digitization of clinical data performed by health care centers and hospitals that migrated all their daily activity into structured databases called Electronic Health Records (EHR). The aim of the EHR has changed during the last decade, evolving from simple patient data records to the integration of health data from multiple sources, ensuring security and interoperability capabilities and using internationally recognized standards [23]. Current trends in EHR are the design of more intuitive interfaces, reduction of data duplication, and gradual incorporation of clinical decision support systems to assist professionals when solving different clinical tasks.

Although CDSS may tackle different tasks, the most common use of CDSS is focused on *diagnosis* [48]. Also called description systems, the main goal of diagnosis is to identify the patient disease through knowledge models usually acquired from data. However, diagnosis is not the only clinical task (see Figure 1.1). *Prognosis* is also a key element in which experts estimate the future scenarios that patients can achieve depending on whether some clinical treatments/interventions are or aren't performed. The third main clinical task is *prescription*. Prescription is focused on selecting the most appropriate treatments taking into account diagnoses and prognoses.

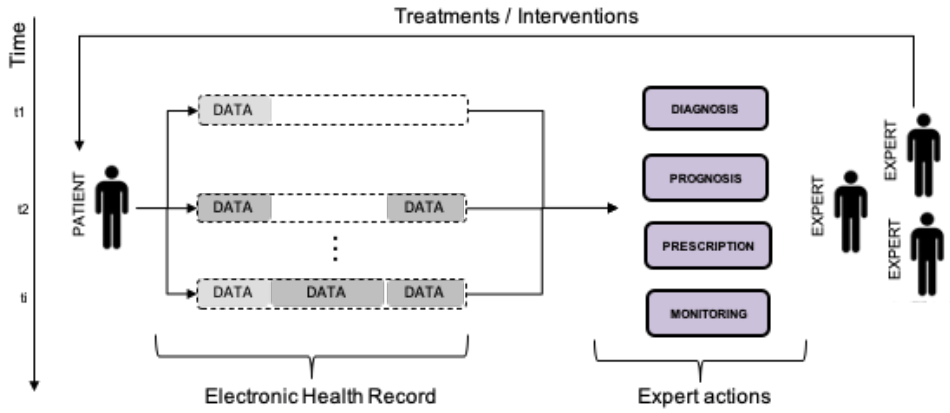

Figure 1.1: Problem description.

A CDSS is designed as a system that receives a set of input variables that are exploited to produce outcome variables. Historically, the working hypothesis of CDSS is to work in a "one-shot" mode. That is, input is received at the same time and then the outcome can be established from this input. Nowadays, the focus moved to systems able to operate in long processes, such as complex clinical treatments, where hypotheses and conclusions are periodically/incrementally determined/reviewed.

Often there is more than a unique treatment option and some of the treatments provide short-term results (few days) whereas other treatments focus on long-term effects (some months) [27]. This implies that the effects of some treatments are not observed until a long time later. Moreover, when patients are subjected to multiple interventions at the same time, part of them may interfere, either positively or negatively, with other interventions. Additionally, over the weeks, new, initially hidden problems might arise.

The clinical environment usually involves multiple professionals (such as doctors, physiotherapists, psychologists, neuropsychologists, occupational therapists) and each one of them identifies and treats the patients' problems from their own perspectives. When interventions are assigned to different professionals, interactions among interventions may remain undetected. Usually, the resulting information from the multiple parallel applied treatments is fragmented into different sources of data silos.

1.1 Current Facilitators and Barriers for CDSS

Clinicians base their decisions taking into account the available information related to a patient. When more data becomes available, treatments are reviewed and/or reformulated. However, sometimes a certain level of uncertainty arises from the patient's individual response capability to treatments [8]. Although there are guidelines and protocols to orient clinical decisions, personalized medicine aims to maximize treatment effectiveness.

Personalized medicine is based on the fact that each individual has different needs and can evolve differently [8]. Nevertheless, it also shall maintain the delicate balance between offering the same treatment opportunities to all patients (equity) and ensuring that each patient is receiving the most adequate and cost effective treatment (resource optimization).

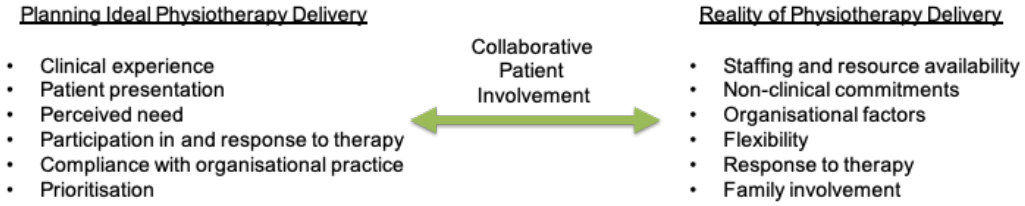

Figure 1.2: Ideal Planning versus planning reality.

FACILITATORS	**BARRIERS**
Measurable patient improvements	System rarely used
Financial incentives	Clinical Workflow changes
User-friendly design	Usability issues
Relevant recommendations	Lack of reliability
Large data volume	Lack of relevant information
	Incremental information
Integration into clinical workflows	Practice-based medicine customization
	Clinical workflow stage

Table 1.1: CDSS barriers and facilitators.

The ideal planning in practice-base medicine is reformulated time to time to adjust it to the current resource availability, organization factors or expected therapy response (see the adapted Figure 1.2 from [51]).

Although CDSS are generally focused to guide the clinicians, nowadays there is a common consent that patients should play an active role in the decisions related to treatments to be applied. This patients' empowerment requires to consider the patients' needs and preferences while most of the CDSS are designed to maximize the result, ignoring patient preferences. Incorporating patient preferences, may lead to deal with situations in which patients feel more comfortable with interventions with lower benefits.

Some facilitators to implement a CDSS (see Table 1.1) are measurable patient improvements, financial incentives, a user-friendly design, relevant recommendations, huge data volume, and their integration with clinical workflows [50]. Most commonly, the facilitators are oriented to cost-effectiveness, i.e. to maximize the use of limited resources to achieve the highest improvement in patients. However, as it was shown previously, not all the healthcare workflow stages have enough precise and relevant information to facilitate the incorporation of CDSS. Additionally, while the lack of reliability and comprehensibility of existing CDSS raise concerns among clinicians, at the same time CDSS may reveal deficiencies in the clinical workflow requiring organizational changes [13, 52].

Due to the potential of CDSS in terms of cost-effectiveness, CDSS awaken the interest of healthcare managers, providers and stakeholders. In the paradigm of efficiency and better management of resources, there is a clinical need to define a more personalized and efficient medicine for patients with multiple-impairments incorporating capabilities such as (1) dealing with partial and incremental information; (2) improved confidence measures when only partial information available; and (3) the identification of unknown interactions when preforming multiple treatments at the same time.

1.2 Motivation

This research started thanks to the opportunity that Hospital Guttmann provided me to dive into the complexity of decisions clinicians have to deal with regarding the neuro-rehabilitation of patients. Patients at neuro-rehabilitation units are long stay patients where a team of multi-disciplinary experts work together to maximize the recovery of people that suffer a chronic condition. Specifically, the aim of this research started from the analysis that current CDSS still have room for improvement in the context of long healthcare processes.

For instance, in the first hospitalization days, clinicians can only partially evaluate patients. This constraint may represent a problem for CDSS because they work under the hypothesis that main relevant information is given. But usually, only through additional assessments, which are performed over several days, a more precise characterization of the patient may be achieved. To deal with this problem, some CDSS generate multiple models from partial views of historical data. But these independent models may produce inconsistencies, changing the predictions every time new data is incorporated. Additionally, because CDSS tend to be used as "one shoot" systems, the revision of previous outcomes or the analysis of the impact of new knowledge may not be adequately addressed.

An important issue in CDSS outcomes is the capability to incorporate explanations and confidence measures over these outcomes. In the clinical context, more important than the solution is the capability to provide trust. There exists a vast literature regarding confidence measures. However, the way missing information should affect confidence measures is still unclear.

In the context of patients with multiple impairments, multiple experts perform several interventions at the same time, focusing on multiple therapeutic goals, and determined by multiple diagnoses. Due to each intervention is usually related to different therapeutic goals, and each therapeutic goal may contribute to solve, to a different extent, several impairments, in some cases it is difficult to precise the clinical evidence of a specific treatment. If we add the fact that, over time, new interventions are incorporated, the volume of information generated for a given patient becomes overwhelming. Thus, providing computational tools to support clinical decisions is still a challenge.

Interventions are performed by patients following a set of activities that clinicians supervise. However, some of these activities do not take into account patient preferences. Actual societies are oriented to empower patients by giving them a more active role. Including patient preferences and abilities in treatments have been shown as a powerful strategy to improve patient adherence and patient recovery.

1.3 Contributions

The main goal of this research is the development and application of AI techniques, and especially machine learning algorithms, to the design of Clinical Decision Support Systems for long healthcare processes. Additionally, the aim of this research is the evaluation of proposed solutions in real long term clinical healthcare processes: the Institut Guttmann and MST-Project. Concretely, the contributions are:

- Define a general architecture of Clinical Decision Support System (CDSS) for long term healthcare processes.
- Propose data enhancement techniques. How the system can anticipate the data that is not already available? How can prediction accuracy be improved?
- Propose confidence measures to support CDSS. How confident is a CDSS about performed predictions taking into account the known/unknown part of the problem?
- Propose tools to assess treatments for patients with multiple impairments. How can a CDSS find non-explicit relationships between patient problems and treatment goals in multi-impairment scenarios?
- Propose solutions to transfer our contributions for long-term healthcare processes to two specific use cases (Institut Guttmann, MST-project).

1.4 Document structure

The document, including this introductory chapter, is composed of seven chapters with the following structure. In Chapter 2 healthcare stages are described showing that a clinical process is sometimes not viewed as a long-term treatment. The lack of clinical decision support systems to guide experts in the entire healthcare process is highlighted. A literature review is performed showing how CDSS systems have been implemented historically. Finally, a proposal for a CDSS framework aimed at long-term treatments is presented.

Chapter 3 is devoted to data enhancement in scenarios with partial information. Clinical domains are very representative examples of these scenarios with a significant amount of missing information. The first part of the chapter describes existing proposals to address this issue, focusing on imputation techniques. We will show that there is still room for improvement in changing environments where the percentage of missing information may be very diverse. We propose a new imputation methodology, Dynamic Multiple Imputation, and report experiments conducted both in regression and classification problems. The aim behind our proposal was to design an algorithm able to adapt to different percentages of missing data.

Chapter 4 describes existing confidence measures and discusses the necessity of extending them to handle partial information. Since current confidence measures usually do not take into account what information is available, we propose a new solution, based on mutual information measures, to mitigate this issue. The proposed solution was evaluated and performed experiments are reported.

Chapter 5 focuses on finding non-explicit relationships between different data taxonomies. In the context of multi-impaired patients several diagnoses have to be treated. In turn, these multiple diagnoses generate several therapeutic goals. Finally, therapeutic goals generate several interventions that are performed at the same time. But the relationships between diagnoses and therapeutic goals and the relationships between therapeutic goals and interventions are “n to m” relations and usually not explicitly reported in patient health records. In Chapter 5, we propose a new methodology to explore and analyze these

implicit relationships. Moreover, we present a tool to support the analysis of a new patient starting the rehabilitation program.

Chapter 6 introduces two real clinical use cases. The first use case applied to Institut Guttmann, a hospital for patient rehabilitation. This use case illustrates how the different proposed solutions can contribute in a hospital environment with patients expending several months in rehabilitation. The second use case is oriented to illustrate the clinical decision support in a home-based environment. In the Music Supported Therapy (MST) project, funded by "La Marato de TV3", a group of chronic patients who suffered a stroke follow a home-based treatment therapy of 30 sessions during ten weeks.

Concluding the document, the last Chapter summarizes the different proposals, highlighting the contributions. With the purpose of continuous improvement, possible future research lines are also presented to try to cover issues aroused during this research.

Chapter 2

Clinical Decision Support System for long-term treatments

Clinical Decision Support Systems (CDSS) have a large research trajectory in AI. However, the heterogeneity of proposed solutions difficults their generalization in a common taxonomy or framework. With the aim of providing a general taxonomy to describe and compare CDSS, some authors defined a framework to classify Clinical Decision Support Systems [71]. Their framework characterizes CDSS in 24 different taxonomy axes grouped into 5 main categories: (1) the clinical context and tasks that a CDSS is tackling; (2) the knowledge and data sources exploited; (3) the methods and algorithms incorporated; (4) the kind of delivery of the outcome of the CDSS; and (5) their integration into the clinical workflow.

The clinical context refers to the specific task a CDSS deals with and most of them are devoted to diagnosis and prognosis problems, although, prescription and prevention problems are gaining popularity in recent years. Another general characteristic of CDSSs is that they are designed as a "one shoot" system. In such systems, every new input is managed independently of the previous ones, without detecting continuity or performing any revision of the outcome previously generated by the CDSS. The one shoot approach has promoted highly specialized CDSS but also CDSS with a narrow scope. While these systems are appropriate for some specialized clinical problems, they are not suitable for dealing with chronic patients and/or patients subjected to long-term treatments where detecting continuity or performing any revision is mandatory.

The knowledge and data sources exploited by a CDSS are a key issue. Electronic Health Records (EHR), potentially, may provide extended information regarding a specific patient, but the heterogeneity of information sources (text, images, taxonomies, ...) is usually a stopper to effectively exploit them. Also privacy issues may additionally restrict the available information. Moreover, in real clinical scenarios, the time and order in which information is available may differ significantly from one patient to another.

Together with data sources, inference methods and algorithms incorporated into CDSS determine the final capabilities and outcomes. As it has been introduced before, the heterogeneity of methods proposed is as diverse as the amount of AI sub-fields. Although maybe more important than the inference methods, the diversity of processes involved

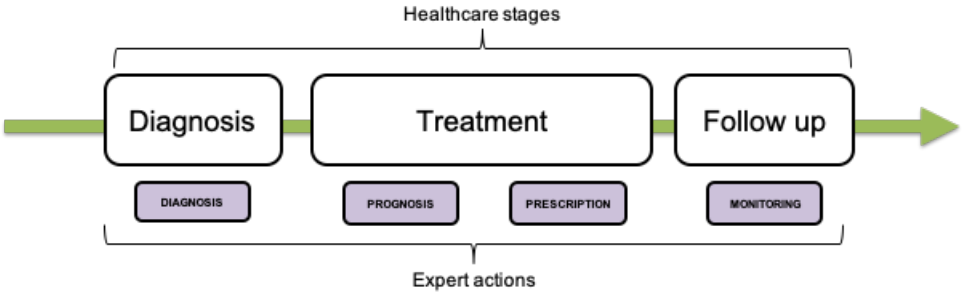

Figure 2.1: Typical clinical workflow.

in clinical decisions and their inter-relations described as clinical protocols and clinical guidelines, determines the roles of CDSS.

2.1 Healthcare workflow

Clinical decisions and actions are all inter-related and they can be framed in a common workflow process guided by established clinical protocols (see Figure 2.1). Each stage of this workflow involves different tasks and professionals. Usually, the healthcare process starts with a diagnosis task. In turn, diagnosis requires that the main information from a given patient has to be acquired. This initial patient information is organized as an electronic health record and can be conceptualized as a set of variables defining a patient profile. From this profile, clinicians diagnose patients' problems from present symptoms [9]. Correctly diagnosing a disease is key to clinical decisions, but it can be overwhelming as time is often crucial to mitigating the risks and repercussions associated with diseases. For instance, stroke is one of the clearest examples where rapid diagnosis is of utmost importance to minimize loss of blood supply to the brain and its possible outcomes. Another example is cancer, where an early-stage diagnosis can dramatically increase the chances of survival.

Once patients have a diagnosis, the next stage focuses on treating the problem [9]. There are three main clinical actions: determine the group of possible treatments, assess the suitability of each treatment (known as a prognosis task) and prescribe the most accurate and appropriate treatment customized for the specific patient profile. Typically, the initial stage of treatment is performed in hospitals where patients are closely monitored in terms of medication, food, and vital signs. The controlled environment of hospitals guarantees that the patient is following the treatments and, at the same time, the effects of the treatments are also monitored. Once patients are stable and the intensive treatment finishes, patients return home but some of them will continue to visit the hospital regularly to perform part of their therapy. This sub-stage is known as "outpatient". It is also worth pointing out that not all diseases or hospitals include this sub-stage and the transition between hospital and home treatment can be emotionally demanding.

Finally, patients' performance is periodically monitored to assess that they are improving as expected. In this stage, patients are screened few months after the initial diagnose to assess their health condition. As a close supervision of the patients no longer exists,

professionals cannot be certain that all patients follow the recommendations given and the way patients are able to perform them. This lack of information related to the environmental context of the patient is sometimes a problem for clinicians to determine what are the best actions or recommendations.

Currently, one of the key clinical stages is prevention. It is clear that preventing diseases before they appear is the winning strategy but, at the same time, one of the most difficult tasks to carry out. Therefore, preventive care focuses on the decisions that should be taken to minimize the risk of future illnesses [9]. However, despite prevention is one of the most important healthcare stages because it reduces the number of people with health problems, it is not well covered. Usually, few people perform regular revisions, even if they have severe antecedents of clinical problems. For the majority of people, this periodic revision tends to be voluntary. Analogously, the follow up stage also is very important but not well covered in some cases [9].

Clinical decisions are based on evidence medicine that guide the experts to make decisions throughout healthcare process. According to [70], there are three main types of evidence-based medicine: literature-based evidence, practice-based evidence, and patient-directed evidence.

Literature-based evidence is based on the exploitation of results reported in literature. Departing from clinical trials and results described in literature, new treatments are proposed and discussed. Moreover, some publications are devoted to analyze the limitations of previous publications (sensitive to a type of population, problems that do not reflect certain specific aspects, ...) [70]. Although new treatments are regularly proposed, the difficulties to update actual clinical treatments are daunting as the volume of related publications is huge but the time that clinicians have to read them is extremely limited [70]. Furthermore, automatizing the analysis of the literature is a hard task because the content is mainly described in natural language, which makes difficult the implementation of specific algorithms [70].

Practice-based evidence is also based on literature. Despite the fact that literature-based decisions are the most important for evidence-based medicine, there exists a gap with practice-based medicine [70]. Therefore, real clinical environments always require to adjust treatments. The ideal treatments described in literature are adjusted to the environmental context, such as the availability of materials, resources, or the previous experience of clinicians. Additionally, time is an important variable, as some studies in literature are conducted with a concrete window treatment. Nevertheless, the experience accumulated in a given hospital may allow to personalize treatments according to specific patient conditions.

Finally, **patient-directed evidence** is based on the differences resulting from each person (patient) preferences and expectations. Although some researches argue that this approach could also be understood as evidence-based medicine, nowadays there is a common agreement that patients should be involved in clinical decisions due to their active role in their health care guarantees a maximal adherence to treatments. In this context, patients gain more autonomy to decide between different treatment options, although they still consider clinicians as experts [70].

Historically, the first CDSS adopted followed the literature-based evidence approach, i.e., finding and providing new medical evidence to the experts by providing, into the daily

practice, tools to exploit knowledge described in literature. Usually, this type of CDSS does not exploit the information acquired in daily practice and tends to be a "One-shot" decision support. Furthermore, these systems focus on a specific health task or clinical action rather than on trying to guide clinicians in the entire healthcare process. Covering such process should require the design of multiple and independent "One-shot" CDSS which might produce inconsistencies between them if applied at the same time, such as, changing predictions every time new information is available. The alternative is to design a unique CDSS covering all the inter-related decisions regarding the different health processes and stages. In this regard, current trends are oriented to apply CDSS in a mixed patient/practice-based medicine approach trying to consider the entire healthcare process.

Regardless of the evidence-medicine approach, clinical domain always has to deal with many uncertainties as not all the data required to take a decision may be available when needed. Specially in initial healthcare stages, when the patient profile can be only established on a partial view as test results and the issuance of corresponding reports takes time, although patients shall be treated from the first day that they arrive to a hospital or health center. Decision making in this stage is also particularly important both in terms of patient recovery and hospital resource allocation. Over the weeks, the patient profile is complemented based on the first completed reports providing additional information about the real status of the patient. Such data in some cases will deviate from the initial impression of the patient's situation and his/her expected recovery plan. The challenge for the whole team of professionals involved in the healthcare process is to prioritize and organize the patient's interventions over time, since not all problems may be treated at the same time.

The temporal order in which the different data may be available vary from one patient to another. Furthermore, the importance of certain information depends on the problem type (i.e. different characteristics may be relevant for assessing different diagnoses) and on timing (not all characteristics may be available at the same time, and some of them may be interrelated or resulting from others) [4]. Thus, reasoning with partial knowledge becomes mandatory as there is the need to take decisions before all the data is available.

Moreover, predicting the same outcome at various temporal moments using different available information can lead to prediction contradictions. Imagine a patient who has been prescribed a set of interventions and, as per the clinicians feedback, the patient does not improve as expected. At this point, a clinician may suspect that the patient is suffering from stress and may consult a psychologist to confirm her hypothesis. After several tests administrated by the psychologist, the results are incorporated into the patient's profile. At this point, the clinician has more information about the patient's condition and may modify the interventions taking into account the patient's stress levels. That is, minimizing the number of activities and giving more importance to rest in therapy. This change might have an impact on the previously planned interventions by increasing performance of the outcome (as the patient has less stress) or by decreasing it (because the patient performs less activities). Therefore, determining causality between interventions and feedback is a key factor in long term treatments, where the revision process is continuous.

The first CDSS come from the expert systems, where a rule-based system performed inferences like an expert clinician in front of a patient [9]. The main idea was to identify the relation of problems with solutions. For instance, if a patient if found to have a set of symptoms/conditions then a solution is proposed. One approximation explored to

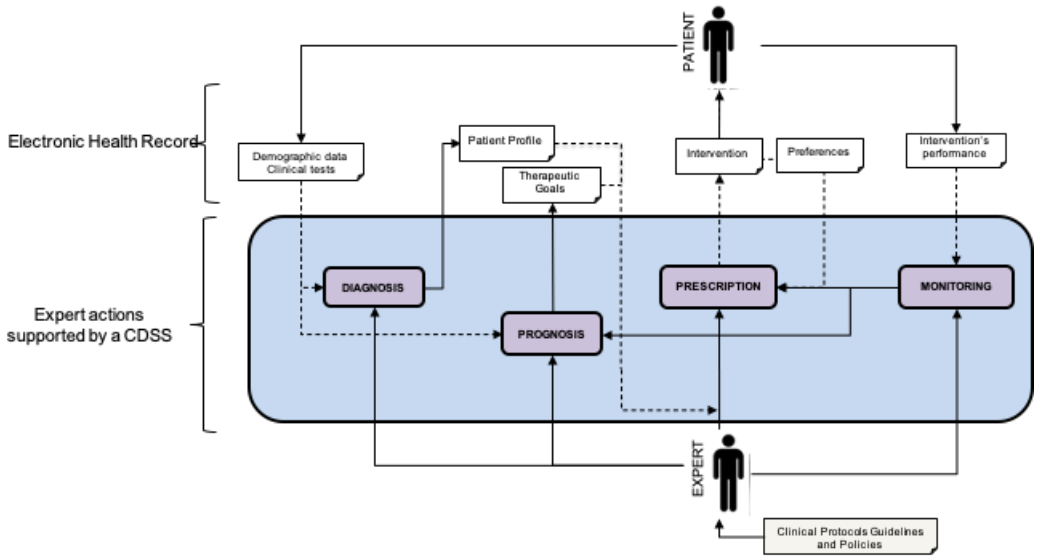

Figure 2.2: Clinical Decision Support System.

implement CDSS was using Case Based Reasoning (CBR). CBR is based on the hypothesis that "similar problems have similar solutions" [2, 1, 17]. The analogy in clinical problems is that "similar patients have similar solutions". However, determining the appropriate similarity measures is not always an easy task. In the CBR community, a strategy to address partial information is with Conversational CBR systems[4] Conversational CBR assume that the data is incomplete and provides a component able to select a question to ask the experts regarding some information that is not yet available. Once question are answered, they tare included into the case library. In [60] a distributed approximation is proposed for clinical prognosis. Proposing a cooperative coordination mechanism, different agents try to make first an individual prognosis and results are sent to a coordinating agent. The coordinating agent constructs a consensual solution. The integration of a CBR system into a health care organization was explored in [5].

2.2 Proposal

In previous Section it we have shown that clinical decisions are taken in the context of a several processes, following standard protocols, and that can be represented as a workflow. Moreover, decisions are influenced by previous ones and by the continuous evaluation of patients. With the aim of proposing a general framework for CDSS, Figure 2.1 is extended to introduce a general architecture for CDSS (see Figure 2.2). The proposal distinguishes two layers: the data layer and the decision layer. In the data layer, the top layer, the different data components stored in EHRs are detailed. The decision layer, bottom layer, makes explicit the processes and data sources involved.

As it was introduced previously, the clinical decision workflow starts with a patient assessment. By collecting initial information, such as, demographic data, physical exploration, or clinical tests, a first initial *Diagnosis* of the patient is performed. The outcome of this diagnosis will be named as the *Patient Profile*.

Then, from patient information and patient profile, clinicians perform an estimation (a *Prognosis*) of patient recovery. For some diagnoses the prognosis may be the return to a healthy condition while for others the prognosis may be limited to reach a better chronic condition. The outcome of the prognosis will be called as *Therapeutic Goals* as they establish the target for clinical interventions. Analyzing the information gathered in the patient profile, and the estimated patient complexity, a set of therapeutic goals (e.g. walk 100 meters without help) may be determined.

Once the therapeutic goals have been established, the next stage in the clinical decision workflow is to *Prescribe* the most appropriate *Interventions*. Interventions may vary from chemical treatments (drugs) to a diversity of physical and psychological activities (physical rehabilitation, brain training, ...). In the context of patients with complex impairments and necessities, the key issue of a multi-disciplinary team of professionals is to prioritize and schedule these different interventions.

Normally, each intervention is detailed in a clinical protocol. Clinical protocols are a catalog of guidelines explaining, step by step, the actions and activities to be followed giving a specific patient condition. They define the flow of activities and the conditions or requirements (patient achievements) to switch from one activity to the following. Interventions are subjected to revision because not all patients respond, evolve, or restore their functionalities in the same way. As a consequence, a *Monitoring* phase is needed to periodically review the evolution of the interventions. Therefore, the monitoring phase is nothing else than a continuous assessment of the performance of the different activities the patients are involved. When a patient achieves a therapeutic goal associated with a specific activity, the activity is replaced by another one to fulfill next therapeutic goals. However, when patients are subjected to multiple interventions at the same time, some of them may interfere, either positively or negatively, with other interventions. Moreover, due to each activity is usually assigned to a different professional, these interactions might remain undetected for a long period of time. Additionally, over the weeks, new, initially hidden problems might also arise.

The proposal of this research does not intend to design a new CDSS. Instead, it aims to provide a layer with several components to help CDSS. As well as CDSS support clinicians, our proposal focuses on supporting CDSS with new capabilities (see Figure 2.3), with the aim to approximate them to the real clinical environment. Specifically, the first proposal is to introduce data enhancement techniques to deal with missing data values, providing the CDSS's capability to predict independently of the available variables (see Chapter 3). Next, an improvement of confidence measures is proposed by assessing the possible impact of missing data (see Chapter 4).

A common problem related with patients with multiple-impairments is that the relations between diagnoses, therapeutic goals and interventions are difficult to analyze. The introduction of community detection techniques is proposed in the prescription phase to support clinicians when deciding which are the best interventions. Specifically, such techniques allow the analysis of non-explicit relationships between patient problems and therapeutic goals (see Chapter 5).

Moreover, to solve the lack of patient traceability, supervising techniques have been introduced to anticipate future available scenarios taking into account the past results in Section 2.4. Since the treatment process is an extensive process where patients evolve over

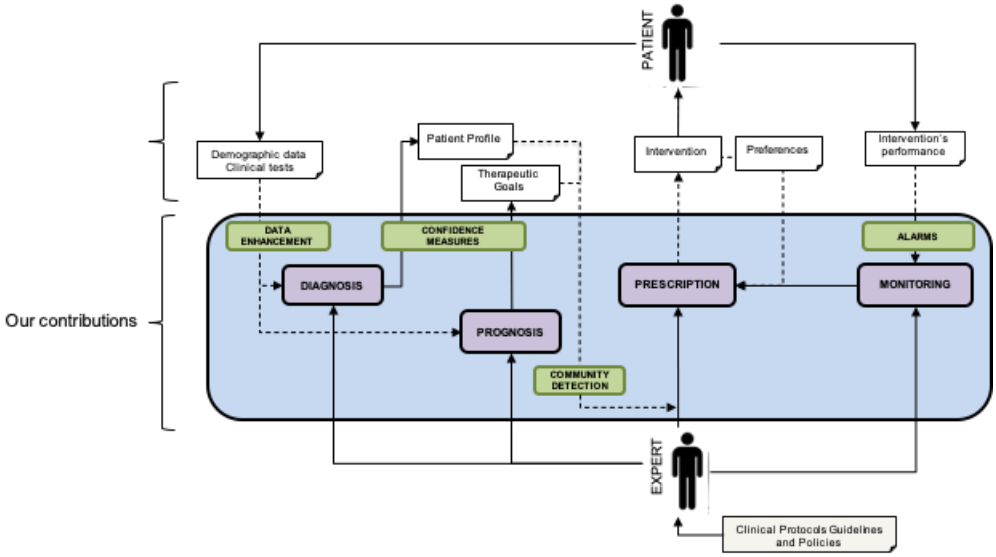

Figure 2.3: Our proposal to enhance CDSS.

time and the interaction between patients and clinicians constantly generates data, the current and past results are reviewed to assess whether diagnoses and prognoses are changing after certain period and why. Also, those systems highly sensitives to the available information tend to be weak as their frequent changes in predictions results may generate confusion to clinicians.

To sum up, several components will be proposed to provide more capabilities to CDSS with the aim to improve the treatment results. Specifically, adding capabilities such as the possibility to improve predictions in the context of incomplete data, by adding confidence measures to diagnoses and supporting the prescription of interventions by analyzing not explicit relationships between clinical data.

2.3 Datasets

To evaluate the algorithms proposed in this dissertation, the focus is placed on two types of problems: regression problems and classification problems where classes are unbalanced. Specifically, experiments have been performed on three types of domains: datasets available at public repositories, clinical data available at eICU research database and clinical data from Innobrain and Play&Sing projects, not publicly available. These last two datasets are described in Chapter 6 being real use cases that have motivated this research.

Publicly available datasets were selected with no missing values. To allow the assessment and comparison of the different algorithms, different percentages of missing values were simulated from the complete data.

Dataset	Features	# Categorical	Instances	Source
AutoPrice	15	0	159	OpenML
Bodyfat	14	0	252	OpenML
Boston	10	0	442	SkLearn
California	8	0	20640	SkLearn
CPS 85 wages	10	7	534	OpenML
CPU small	12	1	8192	OpenML
Diabetes	10	0	442	SkLearn
ICU	19	0	200	OpenML
Plasma retinol	13	3	315	OpenML
Wine Quality	11	0	6497	OpenML

Table 2.1: Properties of regression datasets.

2.3.1 Regression Datasets

Several regression datasets from OpenML [80], Sklearn [59] and UCI [21] will be used to evaluate the methods proposed in this research. The complete list of regression datasets is summarized in 2.1. All the datasets have no missing value and all of them is composed of, at least, 100 instances. The number of categorical features is also reported because existing literature have shown that imputation in categorical features maps to a classification problem.

2.3.2 Unbalanced Classification Datasets

An immense volume of diagnosis problems consist of determining whether a problem is present or not (e.g. presence of cancer). These classification problems have two possible class solutions. Fortunately, the frequency of instances with the problem being present is significantly lower than instances without problem. This fact produces what is known as unbalanced datasets.

To evaluate the sensitivity and confidence of proposed methods several publicly available unbalanced classification datasets were selected (see Table 2.2). In terms of the class distribution, Table 2.2 shows that all the two-class datasets are exaggeratedly unbalanced. For instance, in the *Online intention* dataset, there are 85% of instances from the dominant class and only 15% from the minority class. Intuitively, a classification algorithm determining most instances as from the dominant class, will achieve high accuracy results because in terms of probability there are more instances in the dominant class. In this context, classification algorithms tend to be more biased towards such class. Thus, the real challenge is to determine the minority class being the class that is less frequent, but where an error in this class has a greater impact on patients.

2.3.3 eICU Collaborative Research Database

The eICU Collaborative Research Database is a large multi-center critical care database made available by Philips Healthcare in partnership with the MIT Laboratory for Com-

Dataset	Features	Instances	Dominant class	Minority class
blood-transfusion	4	748	570 (76%)	178 (24%)
contraceptive	8	1473	1364 (93%)	109 (7%)
online-intention	17	12330	10422 (85%)	1908 (15%)
phoneme	5	5404	3818 (71%)	1586 (29%)
pima-diabetes	8	768	500 (65%)	268 (35%)

Table 2.2: Properties of classification datasets.

Dataset	Features	Instances	Problem type
eicu_mortality_1000	10	1000	Classification
eicu_mortality_10000	10	10000	Classification
eicu_los	9	1000	Regression
eicu_ner	4	2000	Non explicit relationships

Table 2.3: Datasets generated from eICU database.

putational Physiology. The eICU Collaborative Research Database has data from a combination of many critical care units from the United States. The data in the collaborative database covers patients admitted to critical care units in 2014 and 2015 [61].

From eICU four datasets have been selected: two classification problems with unbalanced classes, one regression problem, and one dataset with non-explicit relationships (see Table 2.3). Specifically, for the classification problem, two random datasets with different sizes (1000 and 10000) were generated. The classification problem aims to classify correctly if a patient in the hospital will survive or not, being the mortality the classification variable. Mortality has two different labels: alive or expired. Mortality labels were transformed into 0 and 1 values, 0 for alive and 1 for expired. The variables used to train the classification problem are the age, gender, admission weight, admit source, motor, verbal, eyes, apache score and the acute physiology score from the tables `apachepatientresult` and `apachepredvar`. Since missing values are artificially generated and then compared in experiments, only patients without missing data were considered.

Regarding the regression problem, a random dataset with 1000 patients has been generated with the purpose of determining the length of stay (LOS) at the hospital. The LOS is a very frequent regression problem in clinical domains as it is used to compute the patient's cost and resource efficiency. Input variables for the regression problem are the same variables selected for the classification problem extended by an additional variable. The new variable included is related to the discharge location.

The last dataset extracted from the eICU database was generated by combining patient diagnoses (patient profiles) with therapeutic goals clinicians assigned to 2000 different patients. The explicit relation between each specific patient diagnosis and each specific therapeutic goal is not provided in eICU. Thus, this is a clear example of non-explicit relationships between two types of clinical decisions.

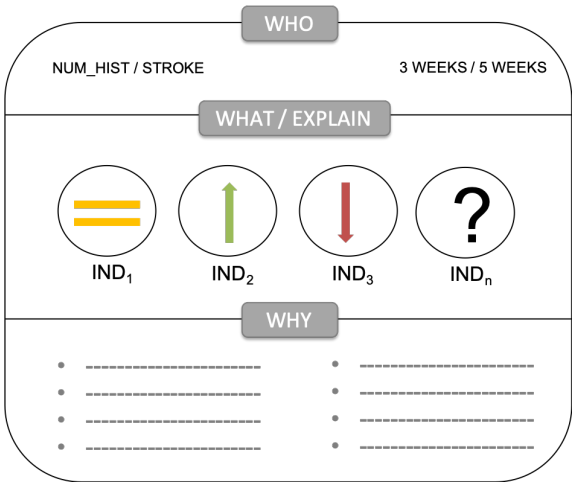

Figure 2.4: Design of monitoring dashboard.

2.4 CDSS Warnings on Monitoring phase

The monitoring phase is a continuous process where professionals revise the evolution of patients and decide how to adjust their interventions based on the intervention's performance. Our aim was to incorporate notifications (either positive or negative) when an unexpected behavior occurs. For instance, if a patient is improving faster than expected, the system notifies such unexpected behavior. Alternatively, if a patient is performing less than expected, the system also notifies to experts of an unexpected performance that probably will require a revision of the interventions.

Specifically, the monitoring system estimates four patient conditions:

- **Expected**: the patient is behaving as expected.
- **Better than expected**: the patient is progressing faster than expected.
- **Worse than expected**: the performance of the patient points to a slow progress or to a stagnation.
- **Unexpected**: the performance of the patient is not the appropriate.

Note that the monitoring phase implies that the CDSS system will be periodically re-evaluating each patient. The normal re-evaluation of patients should be performed once a week, being the minimum time period established by professionals to assess significant changes.

After a week, all patients may be re-evaluated incorporating new information resulting from latest interventions or any clinical test results. The monitoring of patients is performed by a CBR system and starts by retrieving most similar patients. After the retrieval step, two situations may occur: either the rank of the retrieved cases remains unchanged with respect of the previous week or new cases are appearing at the top of the list (new nearest neighbors). If the first scenario, no alarm (notification) is thrown. If a change

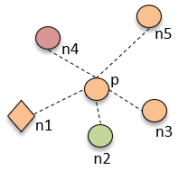

Figure 2.5: Initial neighborhood for the retrieval step of problem p.

occurs, the CBR reuse stage compares the prediction performed with these new cases with the previous prediction.

Figure 2.6: Possible new retrieval scenarios generated after one week of treatment for patient p.

An example is introduced to illustrate the generation of alarms using Figure 2.5 and Figure 2.6, where shapes represent the patient complexity class and colors the patient evolution. In the initial prediction for problem p (Figure 2.5), the CDSS retrieved 5 neighbors (n_1,n_2,n_3,n_4,n_5). According to the neighborhood, for the complexity class (5 circles and 1 rhombus), the problem p was categorized as circle. Additionally, since the predicted evolution of the majority of neighbors is orange (n_1, n_5, n_3 have this color), the estimated evolution was orange. Figure 2.6 shows four possible future scenarios as an evolution of the initial retrieved scenario (Figure 2.5). *Expected* scenario is a scenario where, although the neighbors changed, the estimated evolution remains unchanged. *Better than expected* is a scenario where neighbors changed and, accordingly to the new neighbors, the estimation for the evolution has improved (it changed from orange to green). *Worse than expected* is the opposite scenario. Neighbors have changed and, with them, the new estimation has worsened from orange to red. Finally, the *Unexpected* scenario illustrates a radical change in the neighborhood. The problem is not only a change of neighbors but a change in the class of the neighbors, i.e. a scenario where the complexity class of the neighbors is different.

Note that, the information related to each problem increases over the weeks, therefore,

the CDSS system has to deal with partial and incomplete data as the full patient picture will only be available at the end of the hospitalisation process.

2.4.1 Experiments

As described previously, unbalanced datasets are usually composed by two classes in which one of the classes has a large volume of examples. This uncompensated size within the classes produces that classification algorithms tend to skew to the dominant class, penalizing the minority class with less accuracy. In some domains, such as clinical problems, classification tasks are performed each time a new information comes or changes over time. Moreover, despite a classification algorithm may by skewed to the dominant class, new features may modify previous results changing the opinion concerning the previous classification class.

```
Dominant class -> Minority class
Minority class -> Dominant class
```

In two-class unbalanced datasets, there are two possible directions to modify previous classifications. The first direction is, given a new information, a classification change from dominant class to minority class. This opinion change has a huge impact because it implies evidence of the presence of a problem. The impact is similar when the change happens in the opposite direction, from minority class to dominant class, as it may involve a radical change in treatments. For instance, in the mortality unbalanced dataset, any classification change from dominant to minority class may require the prescription of more aggressive treatments while a change from minority to dominant class means a significant improvement of the patient.

Figure 2.7 illustrates the behavior on the blood-transfusion-service-center dataset. This dataset has two classes with 570 (dominant) and 178 (minority) instances. As percentage, 76% and 24%. This means that 3/4 of the instances in blood-transfusion-service-center dataset belong to the dominant class. The three series in Figure 2.7 show the accuracy to the minority class (green line), classification changes to the dominant class (blue line), and classification changes to the minority class (orange line). For instance, when problems have two features, around 20% of them previously categorized as dominant class become classified to the minority class (orange line). On the opposite direction, the change from minority class to dominant class only occurs in 5%, i.e. 9 instances. These results reflect that changes from dominant to minority class are more frequent than changes from minority to dominant class.

Exploring these classification changes over all the two-class datasets, the mean and std results are showed in Table 2.4. Results confirm the stated above: the most frequent opinion change is in the direction from dominant class to minority class. Taking into account that the accuracy of the minority class is in general low, the utility of generating notifications/alarms is clearly high.

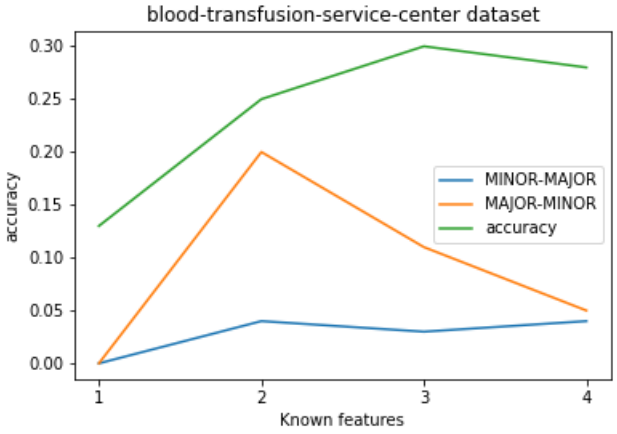

Figure 2.7: Changing the classification result when more data is available.

	MINOR-MAJOR	MAJOR-MINOR	accuracy
blood-transfusion-service-center	0.03 ± 0.02	0.09 ± 0.09	0.24 ± 0.08
contraceptive	0.02 ± 0.01	0.06 ± 0.03	0.1 ± 0.03
phoneme	0.09 ± 0.05	0.16 ± 0.134	0.63 ± 0.14
pima-diabetes	0.13 ± 0.05	0.19 ± 0.08	0.47 ± 0.07
mortality_eicu_1000	0.0 ± 0.01	0.08 ± 0.03	0.11 ± 0.04
mortality_eicu_10000	0.02 ± 0.01	0.08 ± 0.03	0.13 ± 0.04

Table 2.4: Mean and std results to opinion changes in two-class classification problems.

Chapter 3

Inference with Partial Information

With or without missing data, the main goal of any system is to provide valid and efficient inferences given a dataset [67]. However, many data algorithms were designed to work with complete datasets, i.e. where all feature values are known [67]. Some authors justify to skip datasets with missing values, arguing that the control over the data in the dataset is lost, introducing uncertainty in the inferences [6]. Nevertheless, in some domains, such as clinical problems, where resources are limited or information may experience some delays, reasoning with partial information becomes mandatory.

3.1 Inference and Missing data

Missing data is a common scientific issue present in multiple domains such as mortality [77], climate [68], DNA microarrays [75], ovarian cancer [16], etc. For instance, more than 40 % of datasets from UCI Machine Learning Repository [57], widely used for comparison of machine learning algorithms contain missing data. The causes behind a missing information are diverse and usually not reported (e.g. confidential information removed, non-response items, participants that left a study) [64]. In clinical domains the reasons behind missing information may indicate that a given information was considered non essential but this implicit hypothesis may not be the real cause.

Missing data can be produced by a deficient sensor, which may lose the signal in some periods. Missing data even may be caused because the methodology used to collect the data changes (e.g. by recollecting new variables and skipping to recollect others). Additionally, missing data may have the origin on the combination of several similar but not identical data sets [20].

On top of that, missing values may be generated by an *item nonresponse* where the entire data collection procedures fail or by a *unit response* where partial data is available [67]. For instance, in survey studies, some participants may refuse to answer some questions, answering that they don't know something, or skip questions that are addressed to another type of participants. Errors can ever be caused by an interviewer error [3]. Moreover, it may be possible that, depending on the answer on a specific question, some subset of following questions may become irrelevant [67].

3.1.1 Characterization of missing data

A classification of the missing mechanisms was presented by Rubin [65] in 1986 and is still applied to characterize the type of missing values [67]:

- **Missing at Random (MAR):** The missing data depends only on the observed data.
$$P(R|X) = P(R|X_o)$$
- **Missing Completely at Random (MCAR):** The missing data does not depend on either the known values or the missing data.
$$P(R|X) = P(R)$$
- **Not missing at Random (NMAR):** The missing data is not random and depends only on the missing values.
$$P(R|X) = P(R|X_m)$$

where X denotes the complete data, X_o denotes the observed data, X_m denotes the missing data, and R denotes the *probability of the distribution of the missingness* [67]. Note, that for a statistician, random suggests a probabilistic process rather than deterministic. Thus, R is treated as a set of random variables having a joint probability distribution [67].

In MAR, R is the same when we have the complete data and the observable data [67]. In the opposite case of NMAR, the complete data has the same probability of the distribution of R in the missing data. For MCAR, the probability of R does not either depend on the observable and missing data [67].

Sometimes MAR and NMAR mechanisms are impossible to verify as they depend on the unobserved data [76] which is normally controlled only by the data providers [64]. When missingness is beyond the researcher's control, its distribution is unknown [67]. Due to this cause, some authors design experiments modifying the real dataset creating new ones following *MAR* and NMAR mechanisms [81]. On the opposite side, another alternative is to make some assumptions about the missing mechanisms and then apply the different missing data treatments based on the assumptions [24].

Frequently the different missing mechanisms are confused or are not correctly understood. For instance, people very similar in terms of demographics and medical history may have missing values in the Body Mass Indicator (BMI). It is easy to imagine that *BMI* is more likely to be registered in overweight patients [67]. Thus, non reported *BMI* will have a lower distribution with respect to the reported data [10].

A similar explanation may be behind missing blood pressure. Blood pressure in elderly or in people with cardiovascular issues tend to be more frequently registered than in healthy young people. Consequently, people with missing blood pressure are likely to have a lower blood pressure on average than those with recorded blood pressure recorded, i.e. blood pressure is not randomly missing [10].

Finally, for the MCAR case, randomness exists at the highest level and any missing data treatment method can be employed without risk of including bias on the data [67, 34]. An example of a completely random process could be a coin flip [34], if it's not a trick coin.

3.1.2 Strategies to deal with missing data

Several strategies have been proposed to deal with missing data. Some of them try to maximize the number of examples while other try to maximize the number of features. The different strategies may be grouped into four categories:

- **Deletion of Instances**: Deletion of instances is a common strategy to discard cases that contain missing feature values. Its fundamental virtue is its simplicity when a bit portion of instances has missing data [67]. Otherwise, should not be used [20]. Case deletion is only valid under MCAR, where the whole dataset is representative of the incomplete instances [67].

  There are two main strategies for case deletion: *listwise deletion* that omits all instances that have missing values on any of the variables; and *pairwise deletion* that generates different sub-sets without missing information.

  Listwise deletion usually works reasonably well if values are MCAR and the dataset is huge [3]. The advantage of pairwise deletion is that it uses all known information. However, pairwise deletion may introduce artificial correlations.

- **Substitution of Instances**: Substitution of instances is a common strategy when additional instances are available. However, this approach is usually not feasible either because the cost of acquiring new data or because data providers are not the same that data analysts [6].

- **Reduce-feature models**: Reduce-Feature (RF) models generate data models that exploit only the features that are known for a given new problem. For each combination of missing features, a different model is pre-computed for prediction [66]. The strategy proposed by different authors is to use ensemble classifiers [66, 28, 74] An alternative strategy is to compute online the model when a new problem has to be solved. However, this alternative has a huge computational cost.

- **Imputation**: Imputation is a process where the missing data is filled by values generated from observed data [65]. There are three main types of imputation processes [66]: *Predictive or Data-driven value imputation* that uses a model to estimate missing data generating a complete dataset [66]; *Model or Distribution-based imputation* that estimates the parameters able to mimic the data distribution assuming that the data can be generated by a model governed by unknown parameters [25]; and *Unique-value imputation* that replaces unknown data by an arbitrary unique value. The Predictive value imputation methods are the most used [74]. Imputation methods have demonstrated some ability to fit to unanticipated properties of the data such as interactions, non-linearities and complex distributions [53].

A figure representing the different strategies is summarized in Figure 3.1 in which the impact on resultant data is shown. Instance Deletion penalizes small datasets. Reduce-feature models penalizes the features producing that important features may disappear. Case substitution is costly as it requires to add new instances. Finally, imputation methods are one of the most popular approaches because resultant data keeps the number of instances and features.

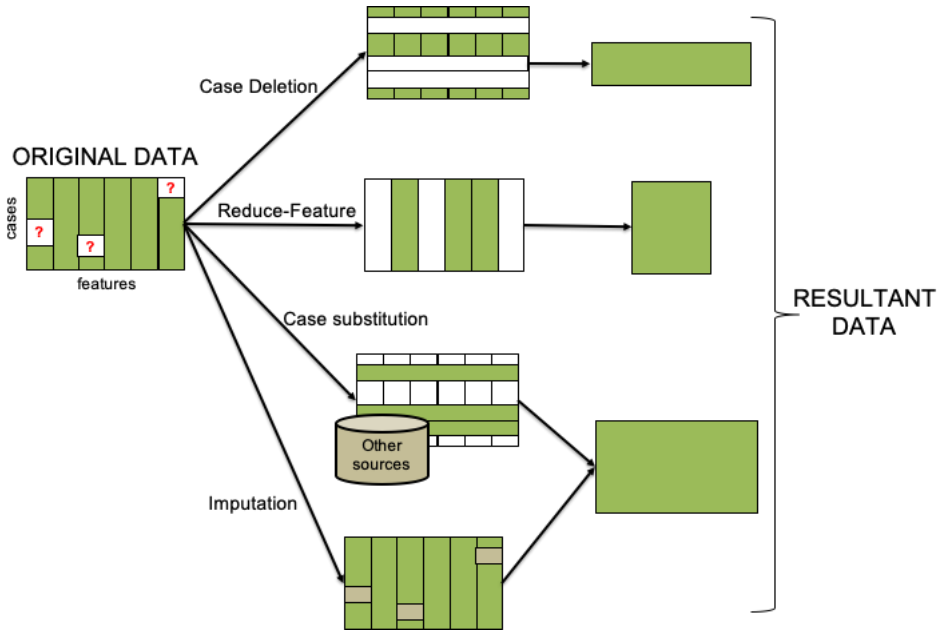

Figure 3.1: Strategies to deal with missing data.

3.2 Imputation

There are two main methodologies to exploit imputation methods: *single imputation* and *multiple imputation*. Single imputation is a methodology that assigns values locally, i.e. taking into account each missing value individually. Some authors suggest that single imputation tends to underestimate the standard errors and to overestimate the level of precision because it tends to not add some uncertainty in the missing values inputted because it omits possible differences between multiple imputations [3]. In the same line, other authors report that it is very complicated to represent any uncertainty facing the problem locally because one imputed value cannot itself represent any uncertainty about the value it imputes [65]. Some authors remark that imputed values should maintain the data structure, including the missing data uncertainty [78]. To cover these issues about uncertainty, the Multiple Imputation strategy was defined [65].

Multiple Imputation analyses several versions of the dataset with different imputation strategies. Additionally, it also incorporates an adjustment of standard errors and other statistics to add some uncertainty in the imputation process [6]. The ability to estimate the uncertainty of a parameter estimation in missing data when merging the results of the analysis of multiple imputed datasets is called "Rubin's rules" [35]. Adding the uncertainty by the standard error correction to the estimated values makes multiple imputations a powerful methodology [35].

Additionally to adding uncertainty, the basic idea behind multiple imputation is to define several imputation models and to execute them in several repetitions. Each repetition has a certain random component to obtain a different version of the complete dataset. Then, each missing value is replaced by a list of simulated values [67]. Finally, the dataset repetitions are analyzed and combined to provide a unique complete dataset [65]. During this analysis part, the difference between the estimation of the assigned missing values and

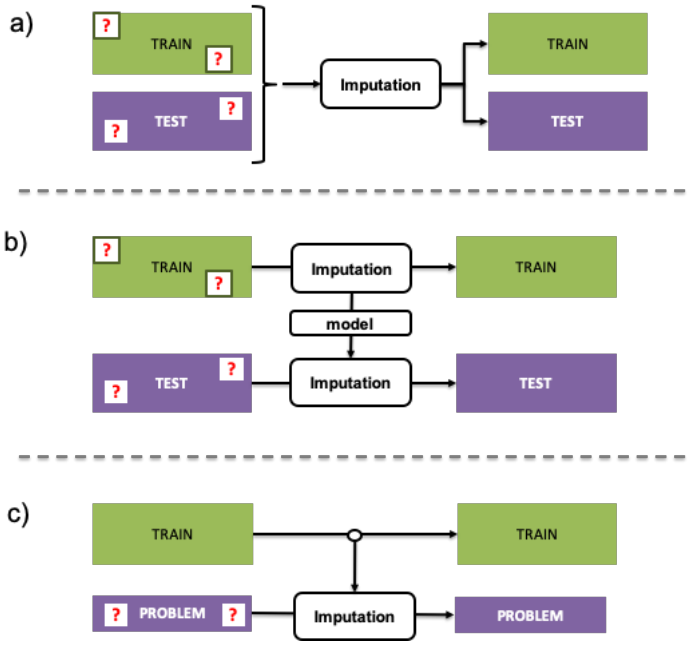

Figure 3.2: Missing scenarios.

the variance are computed. Each estimated parameter is simply the mean of the estimated replications. The standard error incorporates the uncertainty by adding to the mean of the error the variance between the solutions [3].

As some authors report, multiple imputation may propagate the uncertainty about the missing values via some stochastic mechanism, i.e, by adding a randomly generated residual to the regression prediction [53]. Moreover, multiple imputation theory suggests that three to five complete imputed datasets are enough to obtain desired results [35].

Independently of the imputation methodology selected, different methods from statistics and machine learning fields have been proposed to estimate unknown feature values. In the next section, we first describe the different imputation scenarios. Then, Section 3.2.2 introduces the different imputation methods proposed. Finally, in Section 3.2.3 the strengths and weaknesses of existing imputation methods are summarized.

3.2.1 Imputation Scenarios

Imputation methods have been applied to several scenarios. Historically, imputation methods were used as a pre-processing stage in datasets with missing data. In this first scenario, before applying machine learning algorithms, all missing values are estimated (see scenario (a) in Figure 3.2). In scenario (a) it does not matter if the missing values are in the train or in the test data because imputation is performed on all data. In this scenario, missing values are in all the data (train and test). A second possible scenario (see scenario (b) in Figure 3.2) is to perform imputation only in the training set. This second scenario generates an imputation model from training data. Then, the imputation model is used to estimate missing values from test data. Nonetheless, scenario (b) requires the test set to have the same missing features as the training set. This assumption may only work in

environments where missing values are known in advance, i.e. can be determined in training. Finally, in the third scenario (see scenario (c) in Figure 3.2) training data is complete and missing information is only present in test set. In this last scenario, training data is exploited to impute missing information in test data. The counterpart is that imputation models can only be generated when a test problem is known because different problems may have different features with missing information.

3.2.2 Imputation methods

Existing proposals of imputation methods have been organized in two different groups: those based on statistics and those based on Machine Learning.

Statistical methods

The **zero imputation** method assigns zero to each missing value [15]. It is a quick and useful method when computing similarities. Applying zero imputation, instances sharing many missing features will be considered as similar.

Nevertheless, one of the most used imputation methods in the **mean imputation**. The mean imputation assigns to each missing feature value the mean of observable feature values [6]. However, the mean is a global measure and may not be representative for all instances. Several extensions of the mean imputation have been proposed to allow the calculation of local means.

The first extension is the mean imputation by subgroups. Mean imputation by subgroups selects an additional feature to divide the dataset in different subsets given the value of this additional feature. For instance, imagine that income feature has missing values. Teenagers are very likely to earn less than adults because they depend on their parents and are still students. Thus, instead of assigning a global mean for the income feature, the age feature might be used to divide the dataset in two subgroups. Then, if the missing income value corresponds to a teenager instance, then the value assigned will be the mean of all teenager's incomes [3]. This method attenuates the variance and preserves more variance than giving everyone with a missing value the overall mean [3]. A second extension of the mean imputation is the use of clustering techniques, such as k-medoid clustering, to determine cluster prototypes. Then, mean imputation is calculated for each prototype [11].

Although the mean strategy is very popular, some authors suggest that mean imputation may accurately predict missing data but distort the estimated variances and correlations [67]. Additionally, some authors comment that people who are in the middle of distribution in most of the questions tend to be the most likely to answer them. People at the extremes most often refuse to answer questions. For instance, people suffering depression tend to skip items that seem to measure depression i.e "Have you ever feel sad?" or "How often do you go with your friends?" [3]. Then, it would make no sense of substituting a mean value for these cases [3]. Moreover, not all features have the same number of missing values. This fact can produce inconsistent biases when there is great inequality because it attenuates variances [3]. The mean imputation may assign a value that does not exist in the observable features. To avoid this situation, median imputation is also

proposed. Finally, some authors suggest that mean imputation provides improvements only when is used for a substantial (50%) amount of missing data [24].

The **mode imputation** method is another imputation strategy that imputes the most appeared value in the observable data [6]. This method may be used either in continuous features and in categorical features because, as a difference between the mean and median methods, the value used in the mode strategy is an existing value.

Imputation methods based on **regression** have been extensively used. Depending on the type of the feature, different regression algorithm can be used [16]: Linear Regression for continuous data; Logistic Regression for binary data; or Polytomous Logistic Regression for categorical data.

Machine Learning methods

Expectation Maximization (EM) was proposed as an imputation method. Expectation Maximization may find a model of unknown parameters describing missing data. Using a maximum likelihood method, values are imputed iteratively until some stopping criterion is obeyed. Parameters are re-estimated at each step using the observed and filled data. This approach injects some uncertainty adding some random error between the iterations [19, 3].

More refined EM versions have been proposed. *Regularized EM* [68] performs an iterative analysis of ridge regression (linear regression) between missing features and known features to regularize the EM algorithm. In *Structural equation modeling* [3] all the observable information is used to generate the maximum likelihood estimation of parameters instead of imputing directly values. *Patter-mixture models* [67] classify cases by their missingness and describe the observed data within each missingness group trying to characterize the missigness. *Likelihood-based Hybrid* solutions [75] induce imputation models for each new combination of missing features. To reduce the computational cost, previously induced models are stored. Another clustering approach is *Class Center Based Missing Value Imputation* (CCMVI). CCMVI measures the class center and standard deviation in each solution class to calculate imputation values [76].

Bayesian methods have also been used as imputation methods [41]. Bayesian methods cluster data to obtain the probability distribution of cluster classes instead of giving directly a value for the missing data. For instance, *Naïve Bayes (NB)* may be used by considering the solution feature to divide the dataset into training and test sets. Training set includes all the instances with observable data. Then, the probabilities are computed and used to impute values to missing data in testing set. [25] presents a framework that combines Naïve Bayes, Hold deck and boosting for imputing missing values. First a mean pre-imputation is performed to obtain an initial dataset. Then, Naïve Bayes and hold deck are used to refine the imputation. Next, boosting is used to generate several modifications of the imputed dataset and a voting scheme is incorporated at the end.

Decision trees is another ML algorithm exploited to impute missing values using the *C4.5* supervised algorithm. Imputation with decision trees follow a probabilistic approach with train and test data based on an information-based measure, usually gain ratio, as a splitting criterion. Several partitions are obtained with the splitting criterion until a stopping rule is obeyed [41]. Some authors suggest that using decision trees the bias may increase because decision trees ignore redundant features [66].

K-Nearest Neighbors (KNN) constitutes an alternative to calculate imputation values locally. The known features of a problem are used to find k-closest training instances by computing a similarity measure. From k-closest instances, the mean values of each unknown feature in the problem are calculated. Extensions of the basic KNN algorithm have been considered for imputation [20]: computing distances and then *normalizing* to compensate missing values; establishing the distance for a missing feature as the *average* for this feature; considering the distance for a missing feature as *zero*; or randomly selecting the value from a nearest neighbour. Some results suggest that normalization and average may provide good results although depends a lot on the data characteristics [20]. Other results suggest that KNN outperforms the *Reduce-Feature* and *Mean-Mode* methods in several datasets [6]. Other authors extended KNN approach taking into account the importance of features using Mutual Information [34, 57]. In [34] Mutual Information is exploited to obtain the feature weights by measuring the correlations between imputed features and the target class. In [57] a *Grey Relation Analysis* is proposed to measure the similarity between cases taking into account mutual information.

In general, KNN is a good option even when data has a high percentage of missing information [20]. Furthermore, the imputation process is independent of the classification/prediction process allowing to add more variables after the imputation process [6]. Nonetheless, one of the major problems of KNN is the computation cost of the similarity calculus [6].

The basic idea of using **Ensemble Classifiers** for imputation is to cover several combinations of missing features [74] by building multiple classifiers. For instance, *Cage-MetaCombiner* [28] defines a meta ensemble architecture that generates a set of datasets with different possible data partitions considering different number of features. Then, it removes all the instances having missing values at each generated dataset and some classifiers are built usually using part of the dataset. In [74] feature selection is used to select the most relevant features with the aim to reduce the number of instances with missing information.

Multivariate Imputation by Chained Equations (MICE) uses different regression methods to estimate missing values[79]. Each missing value is replaced by a random value in the same feature. Then, a regressor is used for each missing feature using the known features. As multiple imputations, several repetitions are performed to obtain several datasets. Finally, the average of all datasets is performed to obtain a unique dataset solution [74]. For each type of missing feature a different regressor is used. MICE requires a lot of computational effort as it involves several regressors [74].

Neural Networks have also been used as imputation methods. *Group Method of Data Handling* [82] (GMDH) builds a multi layer network structure to identify the data relationship between input and output. *Extreme Learning Machines* combines Gaussian Mixture models to generate multiple imputations with extreme learning machines (NN) to generate a complete dataset. *Auto-Encoders* have been used to impute values by exploiting their capabilities to generate missing data [15]. One point to remark in NN approaches is that missing feature values appear only in the training set [15].

Proposal	MV > 50%	Train data	Test data	All data
RF (Luego et al [46])				
NB (Farhangfar et al [24])		X		
NN (Zhu et al [82])		X		
Ensemble (Folino et al [28])	X	X		
RF (Saar-Tsechansky et al [66])			X	
KNN (Batista et al [6])	X			X
NN (Sovilj et al [72])				X
Clustering (Tsai et al [76])				X

Table 3.1: Summary of state of the art proposals.

3.2.3 Discussion

As described in previous Section, there is an extensive catalog of strategies and algorithms to deal with missing values. Table 3.1 summarizes most recent proposals. Only two of them deal with missing data higher than 50% (see column "MV > 50%"). Different approaches have as hypothesis that missing information may appear either in the training set, in the test set or in both sets. This characteristic is important as it determines the type of problem domains each proposal is addressing. For instance, dealing with missing data only in the training set is a hypothesis which usually does not correspond to clinical problems. As Table 3.1 reports, only one approach deals with the scenario where missing information is focused on the test data. Table 3.2 summarizes the most common baselines used to compare imputation methods. Note that Reduce-Feature (RF) strategy, Mean imputation, and KNN-based imputation are those most commonly used.

Batista et al [6] proposes a KNN approach for imputation. Using different missing rates from 0% to 60%, KNN is compared with Reduce-Feature and Mean imputation methods. Authors highlight that in some datasets KNN does not work well due to it depends largely on the relationship between the unknown and known feature values and on whether these features are correlated. Another aspect to consider, and reported in [6], is that strategies work differently depending on which missing feature is involved and on the percentage of instances with missing data. [6] compares different imputation strategies with two prediction models (C4.5, CN2), a classifier based on decision-trees and a classifier based on rules, achieving different results. This relationship is also explored in [46] using Multiple imputation to check how the relationship between each input attribute and the target class is maintained using different imputation methods. Results show that the performance of MI changes depending on the imputation method used [46]. Both results suggest that it might be necessary to use different imputation methods for each feature with missing data.

Saar-Tsechansky and Provos [66] reports that imputation methods achieve different performances depending on the classifier algorithm that is used. Luengo et al [46] analyzes the impact of several imputation methods using different classification algorithms. Reported results suggest that no single imputation method may reach highest performance for all classifiers. However, they remark that imputation methods are a better option than Reduce-Feature and Instance-Deletion strategies.

Saar-Tsechansky and Provos [66] suggest that performance of Reduce-Feature (RF) strat-

Proposal	Beselines
RF (Luego et al [46])	RF, Mean, KNN
NB (Farhangfar et al [24])	Mean, NB, Regression
NN (Zhu et al [82])	Clustering, Regression
Ensemble (Folino et al [28])	Boosting
RF (Saar-Tsechansky et al [66])	Mean, DTree
KNN (Batista et al [6])	RF, Mean
NN (Sovilj et al [72])	
Clustering (Tsai et al [76])	Mean, KNN

Table 3.2: Imputation methods Compared.

egy is superior to Decision-Tree and Mean imputation methods. Although, the results are in the opposite direction to the results reported in [6, 46], it seems that in some domains RF is a good option as it does not introduce additional noise. Nevertheless, RF imputation models can only be generated when missing features are known. [66] proposes an hybrid version of RF, where the trade-off between computational/memory cost is debated and reports that increasing the size of the training set improves the accuracy percentage. Additionally, Luengo et al [46] report that the Mean imputation method seems to only work with less than 50% of missing data.

The relationship between the imputation accuracy and noise level is analyzed in [82]. According to [82], a low noise level sometimes even improves the results when using some imputation methods. However, a high level of noise level causes a deterioration in imputation results. [82] proposes an imputation method and demonstrates that is a good option in noisy environments. As a counter part, the proposed method does not perform the imputation methods when noise level is low. Nonetheless, authors considered a high-noise scenario with only 20% of missing data, which is not a sufficient noise level some real domains. Finally, most of reported results in the literature do not prove how their proposals deal with a high level of missing data (i.e. higher than the 50%).

3.3 Dynamic Multiple Imputation

As discussed in previous Section, there is more than a unique winning method to fill unknown features. The most appropriate method depends (1) on the characteristics of each domain; (2) on the subset of features known for each incoming problem; and (3) on the importance of the features with respect to the task to be solved. Guided by our clinical problems, and following [66] approach, our focus is on problems where missing values occur only in problems to be solved (test instances in ML terminology). A new multiple imputation methodology is proposed, *Dynamic Multiple Imputation* (DMI), that for each new problem so be solved estimates the best model for each unknown feature from a collection of imputation methods.

DMI performs multiple imputation feature by feature in two stages. In a first stage, given a partially described problem, for each feature with a missing value a specific imputation method is selected from the dataset (training set using ML terminology, case base using CBR terminology) following a cross-fold evaluation strategy. In a second stage, applying

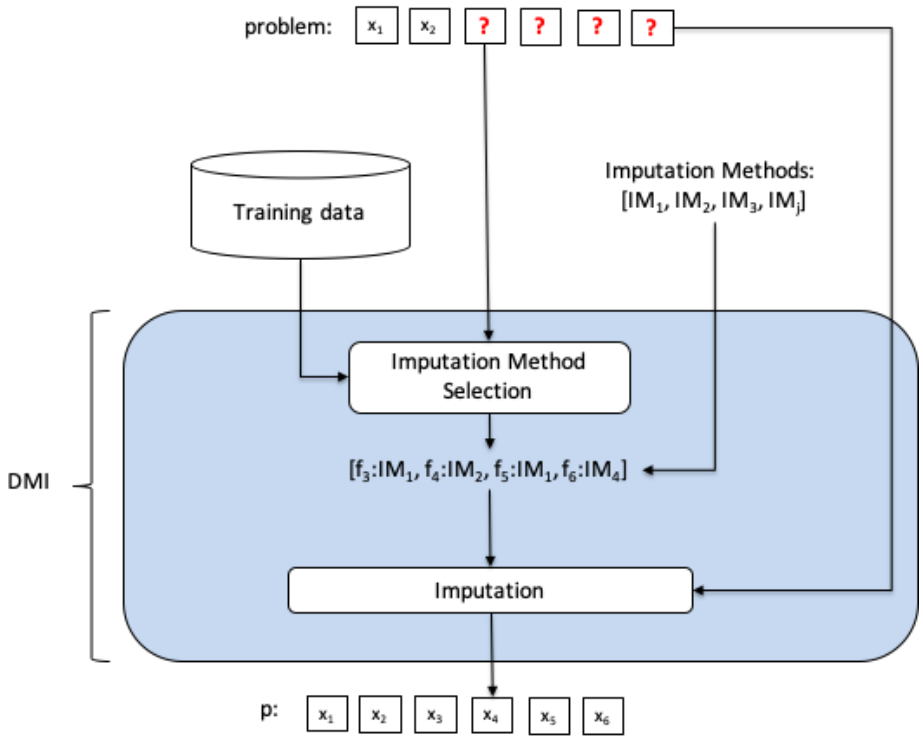

Figure 3.3: DMI methodology.

each selected method over the dataset, imputation models are generated and missing values are imputed. Figure 3.3 summarizes *DMI* stages and Algorithm 1 introduces the pseudo-code where $D[X]$ represents the projection of a dataset over features X.

Determining the most promising imputation method

Given a problem p with a set of observable features X_o and a set of missing features X_m, the first DMI step is to determine the most promising imputation method for each missing feature. The approach to determine most promising imputation methods is to exploit historical data, using a cross-fold methodology. That is, accuracy of all imputation methods is evaluated in each fold (ten times) using Mean Squared Error (MSE). With MSE the difference between imputed (predicted) values and true values are measured. For each feature, the imputation method with lowest MSE is selected.

Notice that in this step only historical data is used to select imputation methods. Current problem p is only needed to determine the subset of features, i.e. observable features, to be used to learn imputation models. See Algorithm 2 for a detailed pseudo-code.

Feature Imputation

After the list of most promising imputation methods have been selected, the second step in DMI is to use the whole historical data to learn imputation models and to apply them

Algorithm 1: DMI main loop

Input:
    p as a problem with features $X = X_o \cup X_m$,
    D as the Historical Data, and
    M as the set of imputation methods
foreach f_i *in* X_m **do**
    $m_i = DMI\_Selection(D[X_o],\ D[f_i],\ M)$
    $p[f_i] \leftarrow DMI\_Imputation(D[X_o],\ D[f_i],\ m_i,\ X_o)$
end
return p

Algorithm 2: DMI_Selection

Input:
    D as training data,
    T as the target feature, and
    M a the set of imputation methods
$errors = \{m_i : 0$ **for** m_i **in** $M\}$
foreach X_test, y_test *in gen_10_folds(D, T)* **do**
    $X\_training = D \setminus X\_test$
    $y\_training = T \setminus y\_test$
    foreach m_i *in* M **do**
        $m_i.fit(X\_training, y\_training)$
        $y\_predict = m_i.predict(X\_test)$
        $errors[m_i] += MSE(y\_predict, y\_test)$
    end
end
return m_i **such that** $errors[m_i]$ **is minimal**

Algorithm 3: DMI_Imputation

Input:
    D as training data,
    T as the target feature,
    m_i as the imputation method, and
    X_o as the observable features of problem p
$m_i.fit(D, T)$
$v = m_i.predict(X_o)$
return v

to the current problem p. Note that, to maximize the representativeness of historical instances, imputation models are built again in this stage.

	FMean	KNNMean	KNNReg	BRidge
f0	0.053352	0.047619	0.047619	**0.043694**
f3	**0.063590**	0.065331	0.065331	0.063775
f4	0.144901	**0.004733**	0.004733	0.024320
f5	0.036989	**0.007242**	0.007242	0.016550
f6	0.082051	**0.034044**	0.034044	0.046386
f8	0.053163	0.011019	0.011019	0.038916
f9	0.063716	0.005548	0.005548	0.024491
f10	0.056983	0.012174	0.012174	0.025812
f11	0.018542	0.014026	**0.014026**	0.016006
f12	0.039835	0.019055	0.019055	0.024810

Table 3.3: Example of MSE results from one DMI_Selection fold execution.

3.4 Illustration Example

To illustrate the DMI algorithm, the Boston dataset will be used (see Section 2.3 for a detailed description):

```
Giving the Boston dataset defined by a group of 13 features:
[f0,f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12]

Suppose that at a specific time the system only has a partial view of a problem
    with observable features:
[f1,f2,f7]

Thus, the rest of feature values are missing:
[f0,f3,f4,f5,f6,f8,f9,f10,f11,f12]
```

Following this example, DMI algorithm is executed to determine which is the best imputation method for each unknown value. Instead of using the same method for all the features, DMI tries to estimate which is the best method for each feature from a list of possible imputation methods. In this context, the list of possible imputation methods is:

```
Imputation methods: ['FMean', 'KNNMean', 'KNNReg', 'BRidge']
```

This illustrative example will show the comparative between the imputation methods in four different scenarios: 1) if all the features are available, 2) if the Reduce-Feature strategy is used, which does not consider the unknown information, 3) if the the same imputation method is used for all the features, and 4) if DMI is applied.

Table 3.3 shows the results of an internal DMI iteration fold. Only using the training data, the system evaluates individually each feature with the list of possible imputation methods. The algorithm works in the way that every unknown value is grouped with the known values to estimate this concrete missing value. For instance, [f0] + [f1,f2,f7], [f3] + [f1,f2,f7], [f4] + [f1,f2,f7] over all the possible imputation methods. Results are expressed as MSE, i.e by the difference between the real and the imputed values. The methods achieving better results for each feature are highlighted. Less error means that

the imputed value is closest to the real value.

For instance, in this example, DMI proposal is to use the following configuration to impute the missing values:

```
Estimated methods: {f0:'BRidge', f3:'FMean', f4:'KNNMean', f5:'KNNMean',
    f6:'KNNMean', f8:'KNNMean', f9:'KNNMean', f10:'KNNMean', f11:'KNNReg',
    f12:'KNNMean'}
```

As the list shows, not all the features obtain the best results with the same imputed method. In this concrete case, KNNMean is the most popular choice. Also, it is important to remark that treating each feature with the most convenient method guarantees better results. Once the algorithm finds the list of the imputation methods, then each method is applied to generate values for the missing data.

The last step is to evaluate how it works the problem with the imputed values. Results reported in Table 3.4 suggest that our method is better than Reduce Feature. Additionally, as was mentioned before, in this case, it seems that the KNNMean and KNNReg work similarly.

Method	MSE
RF	65.969 %
FMean	69.529 %
KNNMean	52.103 %
KNNReg	52.103 %
BRidge	66.947 %
DMI	50.597 %

Table 3.4: Total MSE errors.

Finally, Figure 3.4 shows, as bar plots, errors for all the possible alternatives. The algorithm is better than the Reduce-Feature which is the solution that discards the missing features. Moreover, although the DMI works better than the Reduce-Feature, results suggest that there is also a gap between the algorithm's respect to the best case (when all the features are known). The best case, when all the information is available (without missing data) is represented as a dotted black line called 'all'.

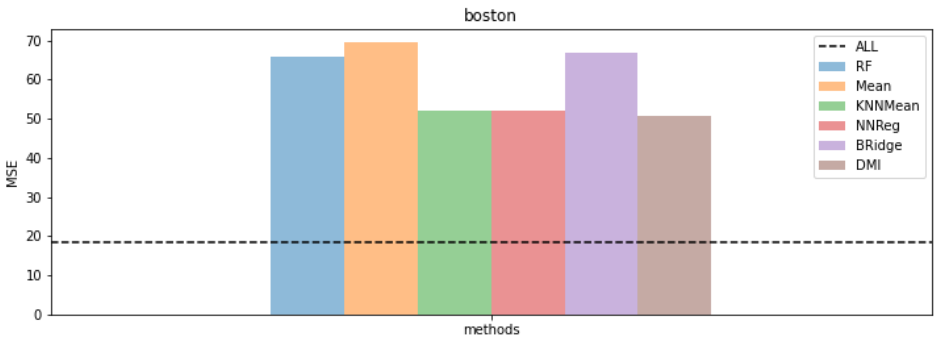

Figure 3.4: Adding f12 to the group of f1, f2, f3.

3.5 Experimental Settings

The DMI methodology was evaluated in regression and classification datasets introduced in Section 2.3. Specifically, the proposal was evaluated in ten regression datasets from public repositories, on a regression problem from eICU database, on five unbalanced datasets from public repositories, and two unbalanced problems from eICU database.

The imputation methodology has been compared with widely used baselines in the literature: Reduce-Feature (RF) model, Mean imputation, K-NN imputation, and MICE (a regression-based methodology). The measures selected to compare the errors were the mean squared error for regression datasets and the sensitivity for classification datasets.

Mean Squared Error (MSE): Measure the difference between the predicted value and the correct value.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)$$

Sensitivity: Measures the percentage of true positives.

$$Sensitivity = \frac{TP}{TP + FP}$$

where TP are true positives and FP false positives.

The aim of the experiments is to compare the imputation methodologies with several percentages of missing values. For each dataset combinations of known features have been generated, starting with only one known feature and increasing the number of known features to all features minus one. At each step 50 different sets of known features have been randomly generated (see Figure 3.5).

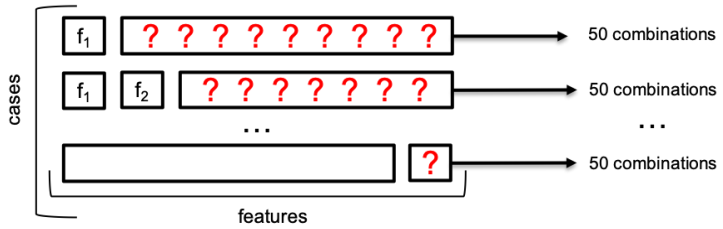

Figure 3.5: Experimental setup.

For instance, given a dataset with 9 features, the options at step 2 (two known features) are exemplified in Figure 3.6. As Figure 3.6 shows, combinations are $X_o : [f_1, f_2]$ and $X_m : [f_3, f_4, f_5, f_6, f_7, f_8, f_9]$, $X_o : [f_8, f_9]$ and $X_m : [f_3, f_4, f_5, f_6, f_7, f_1, f_2]$ ... For a given number of known features, although the level of noise (missing values) is the same, because the features affected are different, the impact on the solution will vary.

For each generated combination of known features a 10 cross-fold validation has been conducted to evaluate the performance of each imputation strategy. Note that training subsets contained the information regarding all features. Only test instances were presented to

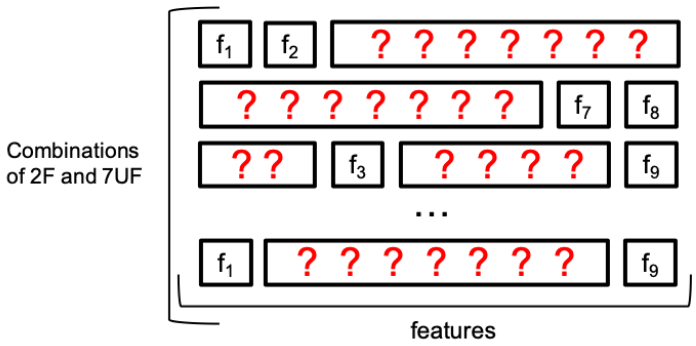

Figure 3.6: Possible level 2 combinations.

	RF	Mean	KNNMean	MICE	DMI
Auto_price	18.55	36.13	16.33	22.15	**12.08**
Bodyfat	35.41	34.63	27.54	31.16	**25.00**
Boston	37.86	47.92	27.82	35.96	**26.89**
California	53.01	70.88	45.87	56.13	**38.04**
CPS_85_wages	42.39	26.76	43.75	**15.80**	26.54
CPU_small	41.90	47.79	30.29	41.95	**27.27**
Diabetes	38.93	61.03	38.62	50.00	**31.71**
LOS_eICU	**45.39**	61.21	48.33	53.71	47.93
ICU	55.62	54.06	48.31	**48.28**	49.86
Plasma_retinol	44.52	49.51	45.05	45.68	**44.28**
Wine_quality	45.52	57.93	45.23	54.73	**41.18**
Av. Rank	3,27	4,55	2,45	3,27	**1.36**

Table 3.5: MSE for Regression datasets.

each imputation strategy with the appropriate combination of known features. After the imputation stage, training subsets were used by a KNN inference (either a classifier or a regressor) to determine the outcome value for test instances.

3.6 Results

3.6.1 Regression datasets

Results reported in Table 3.5 shows MSE errors each imputation method reaches in regression problems (lower values better results). The last row reports the average rank, i.e the average position, of each method across all datasets. DMI achieves the lowest MSE, i.e best rank, in eight of the datasets, the second rank in two datasets, and the third rank in one dataset. Thus, DMI is clearly competitive.

Focusing on scenarios with high percentage of missing information (see Table 3.6), we can confirm that DMI is very competitive. Specifically, DMI achieves lowest MSE in most of the datasets when missing information is between 100 and 75 percent and when missing information is between 75 and 50 percent.

Dataset	>75%					75-50%				
	RF	Mean	KNN	MICE	DMI	RF	Mean	KNN	MICE	DMI
Auto_price	61.72	78.47	43.82	68.55	**40.72**	13.02	46.25	16.86	21.27	**8.91**
Bodyfat	80.84	85.6	75.93	86.52	**69.56**	40.84	45.11	31.11	38.58	**28.13**
Boston	79.9	87.18	64.41	79.46	**60.53**	43.08	61.82	29.79	41.59	**29.38**
California	100.0	93.99	94.85	100.0	**88.73**	78.21	97.04	51.49	85.05	**50.27**
CPS_85_wages	**0.0**	3.05	4.51	**0.0**	26.9	**0.0**	66.27	44.0	1.7	23.59
CPU_small	91.73	93.27	74.07	90.82	**67.62**	55.87	63.34	35.81	54.81	**31.36**
Diabetes	87.14	94.26	84.1	96.04	**77.47**	49.91	77.7	44.4	79.14	**39.43**
ICU	88.19	92.16	**87.61**	87.85	89.03	64.86	72.31	64.49	**63.71**	65.55
LOS_eICU	**91.44**	96.6	93.47	96.44	93.48	**58.17**	80.41	65.46	75.72	64.17
Plasma_retinol	**6.21**	11.39	9.32	9.09	8.79	32.73	38.34	32.61	33.5	**30.81**
wine_quality	92.17	94.55	86.1	96.76	**82.79**	59.36	70.37	53.27	70.76	**51.84**
Av. Rank	2.72	4.18	2.36	3.09	**1.66**	2.82	4.72	2.18	2.9	**1.72**

Table 3.6: MSE in scenarios with high missing data percentage.

Dataset	50-25%					<25%				
	RF	Mean	KNN	MICE	DMI	RF	Mean	KNN	MICE	DMI
auto_price	5.5	20.96	7.58	5.45	**3.07**	1.49	5.64	1.73	0.75	**0.52**
Bodyfat	23.9	15.64	9.82	**8.24**	8.52	5.91	2.78	1.85	**1.29**	1.6
Boston	20.82	30.49	**12.95**	17.53	13.38	7.63	12.19	**4.12**	5.25	4.28
California	42.65	72.72	34.87	49.68	**27.22**	14.67	31.35	26.77	11.71	**11.29**
CPS_85_wages	**0.0**	3.05	4.51	**0.0**	26.9	**0.0**	66.27	44.0	1.7	23.59
CPU_small	91.73	93.27	74.07	90.82	**67.62**	55.87	63.34	35.81	54.81	**31.36**
Diabetes	87.14	94.26	84.1	96.04	**77.47**	49.91	77.7	44.4	79.14	**39.43**
ICU	88.19	92.16	**87.61**	87.85	89.03	64.86	72.31	64.49	**63.71**	65.55
LOS_eICU	**91.44**	96.6	93.47	96.44	93.48	**58.17**	80.41	65.46	75.72	64.17
Plasma_retinol	**6.21**	11.39	9.32	9.09	8.79	32.73	38.34	32.61	33.5	**30.81**
Wine_quality	92.17	94.55	86.1	96.76	**82.79**	59.36	70.37	53.27	70.76	**51.84**
Av. Rank	2.82	4.55	2.45	3.1	**2.1**	3.0	4.72	2.64	2.91	**1.73**

Table 3.7: MSE in scenarios with low missing data percentage.

	RF	Mean	KNNMean	MICE	DMI
blood-transfusion-service-center	0.23	**0.36**	0.22	0.26	0.26
contraceptive	**0.09**	0.03	0.07	0.06	0.06
phoneme	**0.6**	0.41	0.52	0.48	0.51
pima-diabetes	**0.46**	0.28	0.41	0.34	0.37
mortality_eicu_1000	**0.1**	0.05	0.07	0.07	0.06
mortality_eicu_10000	**0.12**	0.06	0.11	0.1	0.1
Av. Rank	1.6	4.2	2.6	2.8	3

Table 3.8: Sensitivity in Unbalanced domains.

In medium and low percentage of missing information (see Table 3.7), DMI is also the best competitive approach. Notice that where RF is quite competitive is in the range between 25% to 50% of missing information. The range reported in the literature.

3.6.2 Unbalanced problems

DMI has been also tested on classification problems. Experiments have been focused on assessing the sensitivity, i.e. accuracy of the minority class. Results from Table 3.8 shows that, as literature pointed, RF is the best strategy in most of the datasets. An explanation of this result is that the classification methods tested are not able to reach high sensitivity scores. This behavior affects a lot when, additionally, the percentage of missing information is important. Thus, further research has to be conducted to try to improve these results that are affecting all imputation methods.

Chapter 4

Confidence measures with partial information

Confidence measures intend to assess the certainty on the solution provided by an AI system. In the context of CBR, confidences are estimated from the set of k neighbors selected to calculate the solutions. There is no unique and best way to estimate confidences. Existing literature [14, 18, 37] propose different confidence measures that exploit different properties such as the similarities between a problem and its neighbors and/or the diversity of solutions assigned to these neighbors. For instance, in a classification domain if the solution for almost all neighbors is the same, confidence values will tend to be high. Contrarily, if similarities are low confidence values will tend to decrease.

The working principle of existing proposals is that problems to be solved are either completely described or has, at most, only a few unknown features. Therefore, these particular hypotheses may become false in clinical domains, specially at early stages of a patient treatment. To improve the reliability of confidence measures in such stages, two different strategies have been proposed.

4.1 Confidence measures

Before introducing the proposed solutions, the third most common confidence measures in the CBR community are presented: the *Classical Confidence*; the *First versus Second Confidence*; and the *Distance Based Confidence*.

The *Classical Confidence* (CC) measure analyzes the diversity of solutions proposed by the nearest neighbors (NN). Specifically, the confidence on a given solution is calculated as the percentage of retrieved cases with this solution. Being s the solution with higher support and NN_s the subset of nearest neighbors with solution s:

$$CC(p) = \frac{|NN_s|}{k} \tag{4.1}$$

To illustrate this measure, Figure 4.1 is used. The new problem is represented as a green circle while neighbors are represented by randomly allocated geometric shapes depending

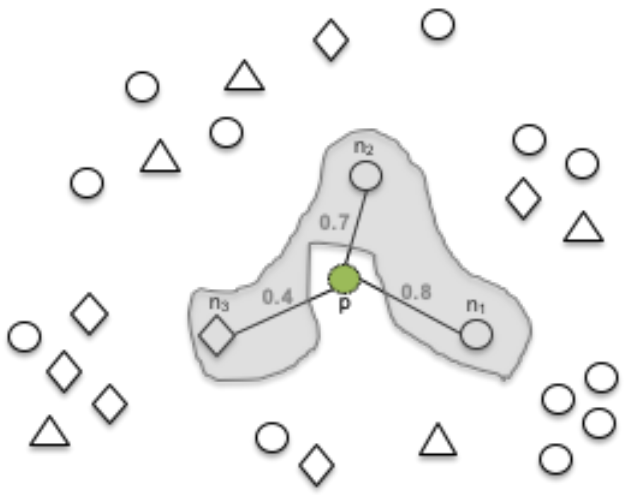

Figure 4.1: Illustration example for exemplifying confidence measures.

on their class type. Given a new problem p and $k = 3$, the closest neighbors to p are n_1, n_2, and n_3. As the dominant class (shape) for these neighbors is a circle, the solution proposed for p will be circle class. Regarding confidence, using CC, the confidence will be 2/3 as the dominant class appears two times with respect to the total neighbors.

The *First versus Second Confidence* (FSC) measure is an extension of the CC measure and is usually selected when there are more than two solution classes. Specifically, FSC calculates the difference between the most popular solution and the second most popular solution. Being s_1 the solution with higher support, NN_{s_1} the subset of nearest neighbors with solution s_1, s_2 the solution with the second higher support, and NN_{s_2} the subset of nearest neighbors with solution s_2:

$$FSC(p) = \frac{|NN_{s_1}| - |NN_{s_2}|}{|NN_{s_1}| + |NN_{s_2}|} \tag{4.2}$$

For instance, following with Figure 4.1, p has three neighbors (n_1, n_2, n_3). The NN_{s_1} (dominant solution) has two neighbors represented by circles while NN_{s_2} has one neighbor represented by a rhombus. Therefore, FSC is computed as (2 - 1) / (2 + 1) = 1/3.

The *Distance Based Confidence* (DBC) measure calculates confidences by comparing similarities to neighbors of the two solutions with higher support. The reasoning behind DBC is that, although the proposed solution is the solution with higher support, the distances among nearest neighbors shall interfere and, therefore, have to be taken into account when calculating the confidence of the solution. Being s_1 the solution with higher support for a problem p, NN_{s_1} the subset of nearest neighbors with solution s_1, s_2 the solution with the second higher support, and NN_{s_2} the subset of nearest neighbors with solution s_2, DBC

is calculated as follows:

$$DBC(p) = \frac{\sum_{i \in NN_{s_1}} Sim(NN_i) - \sum_{j \in NN_{s_2}} Sim(NN_j)}{\sum_{i \in NN_{s_1}} Sim(NN_i) + \sum_{j \in NN_{s_2}} Sim(NN_j)} \tag{4.3}$$

whenever $DBC(p) > 0$. Otherwise $DBC(p) = 0$.

Using a reference the example on Figure 4.1, the similarities between p and n_1, n_2, n_3 are, respectively, 0.8, 0.7, and 0.4 . Then, $DBC(p) = ((0.8 + 0.7) - (0.4))/((0.8 + 0.7) - (0.4))$, i.e. 0.58.

4.2 Mutual Information Based Confidence

The problem with SoA confidence measures is that they do not distinguish between cases when only few information is available and when all the relevant information is present. In the clinical domain this distinction is critical as in early stages of treatment the information available regarding a patient is most probably partial or incomplete.

To overcome this issue, we propose to introduce a τ factor capturing the uncertainty generated when only partial information is available. Given a new problem, the application of the τ factor to any SoA confidence measure it will adjust the confidence in the proposed solution by an estimation of the uncertainty introduced by features with missing values in this problem.

As it was explored in previous chapter, the impact of a specific missing feature value depends on the contribution of this specific feature in the estimation of the solution. We propose to use *Mutual Information* [63] as a measure to estimate the contribution of features. Mutual Information is a measurement used to rank features by importance in feature selection.

Although mutual information measures individually the importance of each input feature with respect to the target feature, we can exploit this information to design a global estimation. The intuition behind the proposed measure is that the risk of a prediction failure increases proportionally to the significance of unknown information. A normalized mutual information measure has been used, i.e. each feature has associated a mutual information score between 0 and 1, and the sum of all values is 1. Normalized mutual information is calculated by dividing each individual mutual information score by the sum of total scores. Then, we can define τ as follows:

$$\tau = \sum_{j \in X_o} MI_j \tag{4.4}$$

where MI is the list of normalized mutual information scores, and X_o the list of known features.

To illustrate the impact of τ in a specific inference, lets take two examples from the mortality_eicu_1000 dataset. Figure 4.2 summarizes the list of MI for this dataset. In a

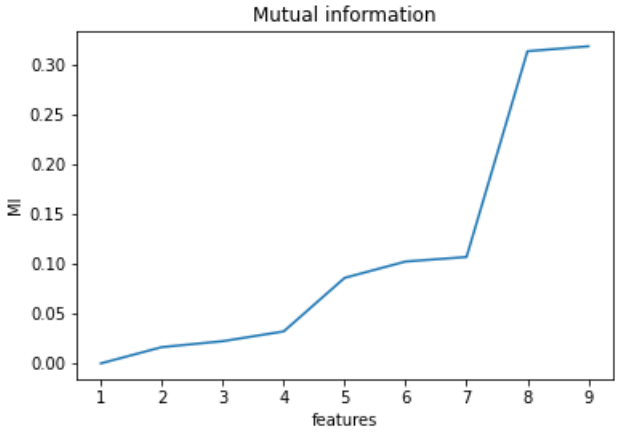

Figure 4.2: Mutual Information of features in dataset mortality_eicu_1000.

first hypothetical problem, only features f_2 and f_7 are available with a classical confidence of $CC = 0.8$. Given that MI for f_2 and f_7 are, respectively, 0.02 and 0.11, we will obtain a $\tau = 0.13$. Then, the confidence of the solution will be adjusted obtaining a final confidence $\tau CC = 0.13 * 0.8 = 0.104$. Analogously, if in another problem the two known features are f_8 and f_9, the τ factor to be applied to the confidence will vary to $0.31 + 0.32 = 0.64$. Again, the confidence of the solution will be adjusted, obtaining a final confidence $\tau CC = 0.64 * 0.8 = 0.512$.

4.3 Experimental setup

Classification datasets introduced in Section 2.3 have been used to evaluate the contribution of τ factor in unbalanced classification. Specifically, we have conducted experiments in five publically available classification datasets and on two different samples of mortality information from eICU database.

Experiments have been performed to evaluate the improvement in the confidence quality of three state of the art confidence measures: Classical Confidence (CC), First versus Second Confidence (FSC), and Distance-Based Confidence (DBC). To assess the quality of a confidence, the Brier Score [12] was used. The Brier Score is an error measure that calculates the relation between the confidence and the accuracy. Specifically, the hypothesis behind the Brier Score is that whenever the predicted class is correct, the confidence should be close to 1 and whenever the prediction is incorrect the confidence should tend to 0. In particular, the Brier Score is calculated as:

$$BS = \frac{1}{N} * \sum_{t=1}^{N} (pred - conf)^2 \tag{4.5}$$

where *pred* is 1 if the predicted value is correct and 0 otherwise and *conf* is the confidence measure.

	CC	τCC	FSC	τFSC	DBC	τDBC
blood-transfusion-service-center	0.52	0.30	0.44	0.29	0.42	0.29
contraceptive	0.70	0.30	0.59	0.26	0.54	0.25
online_intention	0.56	0.24	0.48	0.24	0.47	0.26
phoneme	0.25	0.25	0.28	0.30	0.38	0.39
pima-diabetes	0.35	0.28	0.34	0.32	0.43	0.39
mortality_eicu_1000	0.74	0.32	0.68	0.30	0.67	0.31
mortality_eicu_10000	0.71	0.31	0.64	0.29	0.60	0.29

Table 4.1: Brier score of the minority class.

Analogously to experiments conducted with Dynamic Multiple Imputation, the aim of the experiments was to evaluate the error in confidence with several percentages of missing values. Like in previous Chapter (see Section 3.5), for each dataset combinations of known features have been randomly generated and a 10 cross-fold validation have been performed.

Finally, we evaluated the contribution of τ factor when Dynamic Multiple Imputation (DMI) is performed before classification. That is, DMI was applied to unknown features and τ factor was calculated taking into account only real known features.

4.4 Results

Table 4.1 reports the different Brier scores of confidence measures allowing to compare original confidences with respect to the same measure corrected by the proposal. Such proposal is indicated as a τ prior the confidence is modified (for instance, τCC is the modified version of CC).

In general, in all datasets, proposed τ versions obtained better results with respect to their corresponding SoA confidence measures. There is no clear dominant τ confidence measure, although, there is one dataset in which the proposal has not accomplished any improvement on the Brier score. In the phoneme dataset, results are quite similar between the SoA and τ versions. This suggests that this concrete dataset's characteristics do not fit with the proposal's property. Future research will need to explore which are the reasons that in the phoneme dataset it does not work well.

Additionally, we conducted some experiments applying Dynamic Multiple Imputation (DMI). The application of DMI produces changes in neighbors and in classification decisions. Thus, confidence values will also change. As expected from experiments conducted in previous chapter, confidences present worst Brier results when DMI is applied. However, the introduction of the τ factor allows to improve confidences.

	CC	τCC	FSC	τFSC	DBC	τDBC
blood-transfusion-service-center	0.56	0.36	0.56	0.38	0.52	0.36
contraceptive	0.88	0.35	0.81	0.32	0.84	0.33
phoneme	0.47	0.26	0.56	0.35	0.48	0.29
pima-diabetes	0.57	0.29	0.48	0.28	0.5	0.27
mortality_eicu_1000	0.89	0.39	0.83	0.37	0.86	0.37
mortality_eicu_10000	0.85	0.36	0.77	0.33	0.8	0.34

Table 4.2: Brier score of the minority class when applying DMI.

Chapter 5

Decision Support for non explicit relationships

In health domains, deciding which are the most appropriate interventions is not an easy task when patients simultaneously present several impairments, multiple diagnoses, and require complex interdisciplinary approaches. When interventions are guided by a variety of therapeutic goals, not all the goals may be addressed at the same time. Clinicians have to prioritize some therapeutic goals over others, affecting the schedule of interventions. However, the relationship between patient impairments, therapeutic goals and interventions is entwined and usually not explicitly reported in EHR. Thus, in the context of patients with multiple-impairments, it is still a challenge to design a CDSS to assist in the task of analyzing these relationships and on proposing the most appropriate personalized interventions.

An interesting characteristic of clinical data, is that it is usually organized in taxonomies. For instance, diseases are represented using ICD (either ICD-9 or the latest ICD-10), and ICF (International Classification of Functioning, disability and health) is also a widely established taxonomy promoted by the WHO. As it will be shown in the proposal, this hierarchical characterization of the knowledge brings the opportunity to incorporate an additional perspective when analyzing the relationships between patient impairments, therapeutic goals and interventions.

The relationships between patient profiles, therapeutic goals, and interventions can be modeled as graphs. The collection of techniques proposed in the literature to study groups in graphs is known as community detection [29]. Community detection techniques can be oriented to data properties (nodes), to network structures (edges) or to a mix of them (nodes and edges). Two main algorithms are widely used: the Girvan and Newman (GN) algorithm and the Clique Percolation Method (CPM) algorithm. GN is orientated to find communities based on a hierarchical strategy [32] by using metrics to measure the modularity of the communities and can deal with weighted graphs [55]. CPM searches overlaps between communities in complex networks [56]. CPM is not the most suitable algorithm for our problem due its poor performance on dense graphs.

5.1 Community detection algorithms

The Girvan-Newman (GN) algorithm is a community detection algorithm to detect communities in graphs [32]. The algorithm follows an iterative strategy removing at each step an edge of the graph, focusing on the edges that are most "between" communities. In the basic GN algorithm, edge centrality is calculated using the betweenness of an edge, i.e. calculating the number of shortest paths between pairs of other edges that pass through it. Edge centrality can be determined using different measures.

The GN algorithm iteratively (1) calculates the centrality for all edges in the network and (2) removes the edge with the highest rank until no edges remain. Each time an edge is removed from the network, the remaining connected sub-networks are considered as communities. Thus, storing the resulting communities in a hierarchical structure, the strongest node communities are obtained in the leafs and, traversing the tree bottom-up, at each level nodes are merged in weaker related communities.

5.2 Community Detection to highlight non explicit relations

The proposed methodology is based on graph methods and decomposed in three main stages: a pre-processing stage where data is represented as multiple co-occurrence graphs; a second stage where from each co-occurrence graph and using GN algorithm communities are determined; and a third stage where inter-relations between hierarchies of communities are established. Finally, and not less important, multiple ways to present the results to the clinicians are explored.

5.2.1 Graph pre-processing

The first process is the construction of the co-occurrence graphs. A different co-occurrence graph for each feature of interest is constructed. That is, a first graph capturing information related to patient profiles, a second graph catching information related to therapeutic goals, and a third graph grouping patient interventions.

For each feature of interest, a set of representative variables that will constitute the nodes of the graph have to be selected first. As it will be shown later in the experiments, a decision whether to include all features (variables) or only those that meet specific properties (e.g. a minimum and maximum occurrence) will may be taken. A weight is assigned to each node representing the occurrence of the variable in the set of patients. Then, an edge between two nodes of the graph is established if at least for a given patient both variables represented by the nodes are present. Finally, edges are weighted by representing the occurrences of two variables (two nodes) in the set of patients.

For instance, the graph of co-occurrences for patient profiles is built as follows: (1) the impairments in patient profiles are modeled as the nodes of the graph; (2) nodes are weighted by the number of patients with each impairment; (3) edges between graph nodes model a co-occurrence between two impairments in patient profiles; (4) edges are weighted by the number of patients with a specific joint impairment (see Algorithm 4 and Figure 5.1).

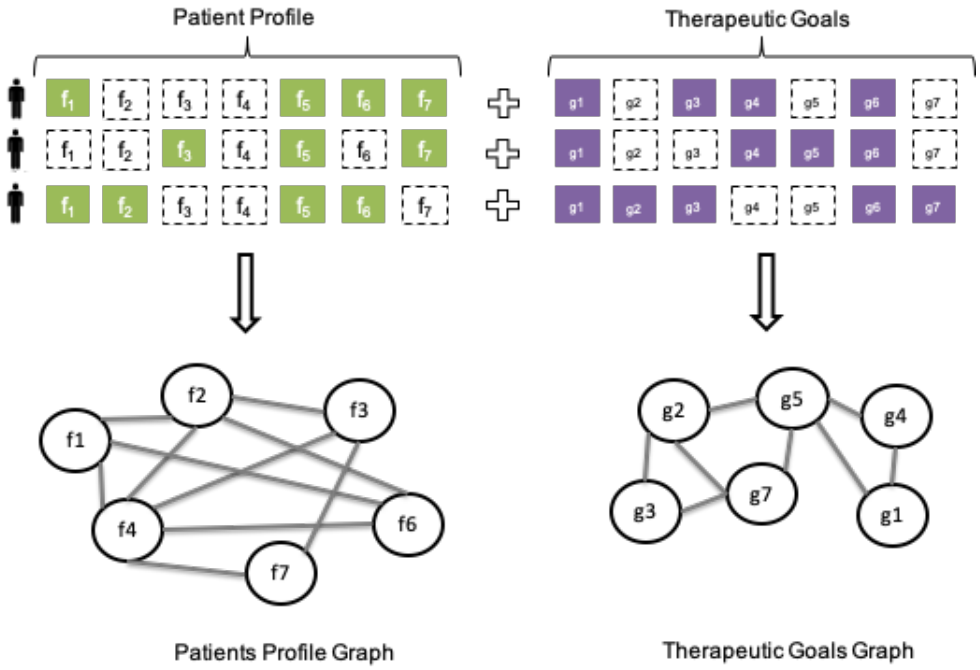

Figure 5.1: Graph construction process.

An analogous procedure is performed for therapeutic goals: (1) therapeutic goals are modeled as the nodes of the graph; (2) nodes are weighted by the number of patients with a specific therapeutic goal; (3) edges between graph nodes model a co-occurrence between two specific therapeutic goals, i.e. two goals shared by the same patient; (4) edges are weighted by the number of patients with joint goals (see Algorithm 4 and Figure 5.1).

In some clinical domains, depending on patient characteristics and on the methodology to determine therapeutic goals, highly sparse or highly dense graphs may be generated. For instance, if patient profiles are very homogeneous, it may cause patients with very similar therapeutic goals. On the other hand, in the scenario of highly detailed patient profiles, therapeutic goals may tend to be very specific, generating unique goals for each patient.

These extremes are easy to identify and report numerically and graphically. Since the nodes and edges are weighted taking into account their occurrence in patients, occurrences may be expressed as percentages. Then, one of the possible solutions is to have lower and upper filters. For instance, 20% and 80% respectively. Consequently, graph elements with a weight lower than 20% or greater than 80% may be reported and removed. In clinical domains theses elements represent either relations univocally determined (e.g. a goal only related to an specific impairment) or preventive actions that are prescribed to all patients (e.g. heparin prescription). The normalization formula used for edges is the following:

$$w^n_{i,j} = \frac{w_{i,j}}{\min(w_i, w_j)} \times 100 \tag{5.1}$$

where given two nodes i and j with weights w_i and w_j; and $w_{i,j}$ the absolute weight of the edge between i and j.

Algorithm 4: Graph construction process

```
Input:
    P as a set of patients,
    itemList as a set of features
Graph = [ ]
foreach patient in P do
    // Creating the nodes
    foreach item in itemList do
        node = create_node(item)
        if node is not in Graph then
            Graph.addNode(node)
        else
            Graph.nodes[node].count++
        end
    end
    // Creating the edges
    combinationPairs = generate_combinations(itemList)
    foreach c1, c2 in combinationPairs do
        edge = create_node(c1, c2)
        if edge is not in Graph then
            Graph.addEdge(edge)
        else
            Graph.edges[edge].count++
        end
    end
end
return Graph
```

5.2.2 Detection of Communities

Once the pre-processing step is completed and co-occurence graphs are build, the next step is to hierarchically cluster each graph according to co-occurrences. For instance, to establish the relationships among patient impairments.

To determine the hierarchical relationships in a co-occurrence graph Girvan and Newman (GN) algorithm is used. As it has been presented before, GN algorithm perfectly fits the goal. The analysis of each resulted hierarchy provides three main clues: (1) items belonging to the same community that are also part of the same domain taxonomy; (2) item communities where items come from different domain taxonomy regions; and (3) clustering degree of items by analyzing their position in the hierarchy of communities (see Figure 5.2).

The analysis of communities is exploited in two perspectives. First, an aggregated view presents those items that are strengthened although they are not part of the same domain group and, conversely, those items belonging to the same domain group and that are less correlated than expected. The second perspective is the patient perspective. Comparing a specific patient with the hierarchies of communities, it may be determined if a patient

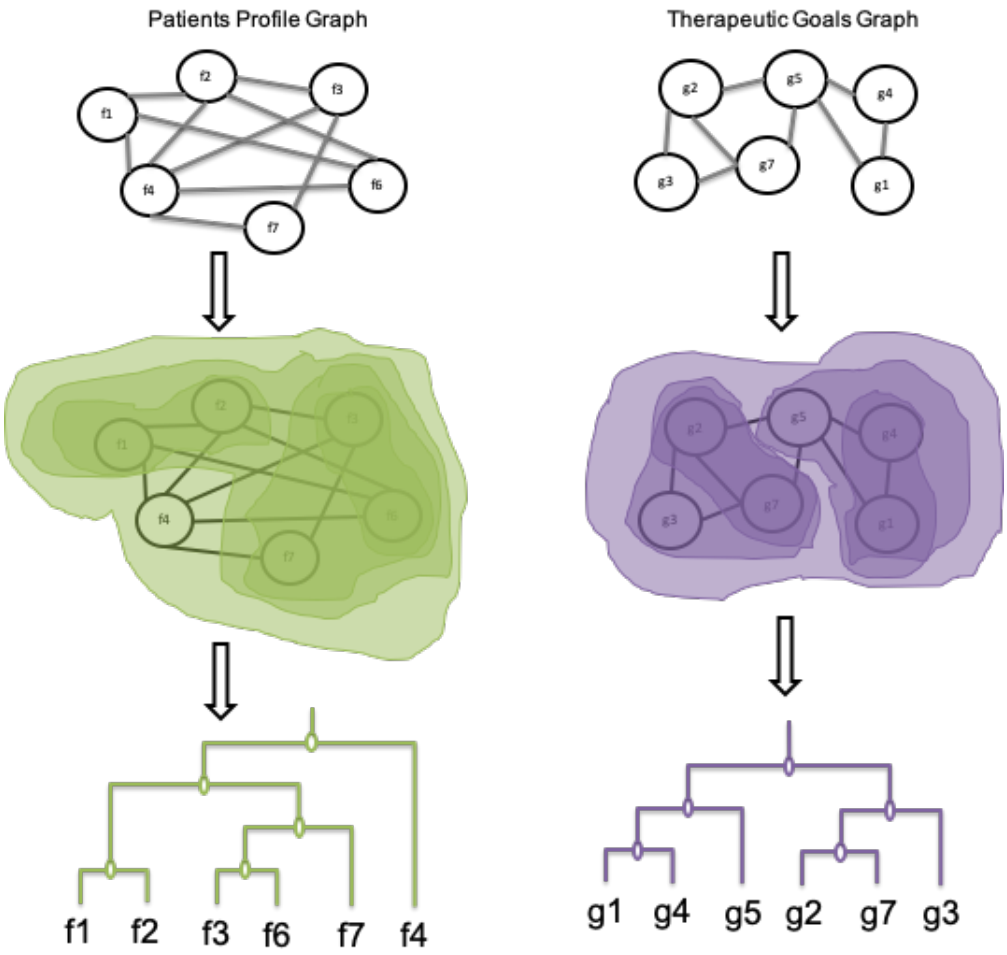

Figure 5.2: Clustering the graphs.

is more or less distant from prototypical patients.

5.2.3 Inter-relations between communities

Finally, given two hierarchical communities discovered in the previous step, the aim of the last analysis step is to study the relationships between them (see Figure 5.3). The main focus of this stage is to bring out non explicit relations between two features of interest. For instance, to find the relationships among groups of impairments with groups of therapeutic goals.

The final aim in this stage is to improve clinical strategies by increasing the evidences associated to the treatments. For instance, showing which sets of treatments are most successful for specific sets of patient impairments and linked to which specific therapeutic goals.

To analyze the inter-relations between the two hierarchical communities, several measures have to be defined. First, the percentage of patients sharing the same communities

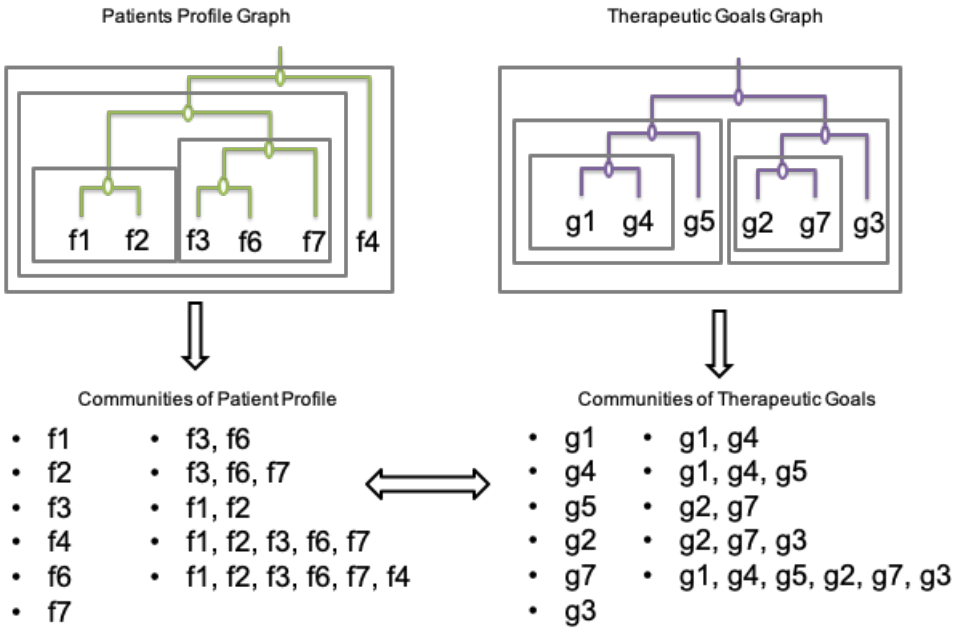

Figure 5.3: Relationship between communities.

are measured, based on an inclusion measure, and normalized by community cardinalities. Thus, given a source community C_i^S, a target community C_j^T, and the function $patientsWith$ returning the patients sharing a community, the proposed measure is the following:

$$I_{i,j} = \frac{|patientsWith(C_i^S) \cap patientsWith(C_j^T)|}{|patientsWith(C_I^S) \cup patientsWith(C_j^T)|} \tag{5.2}$$

To assess the inclusion taking into account the directionality, just one of the elements in the denominator has to be removed. That is, to measure the agreement with respect of the source community only the denominator $patientsWith(C_i^S)$ should be considered.

An example of inter-relationship $I_{i,j}$ may be calculated by the fraction of patients sharing both communities of impairments and therapeutic goals with respect of patients sharing either impairments or therapeutic goals.

5.2.4 Decision Support Tools

One of the key aspects to consider is how to present the results so that they are useful and intelligible for clinicians. With the idea to simplify the interpretation of the results, several visualizations have been explored. First, a graphical tool has been designed to summarize and explore hierarchical relations between communities and, at the same time to stress the inter-relations between communities of different features of interests (e.g. between impairments and therapeutic goals). Figure 5.4 illustrates an example of the proposed tool. Communities discovered from the analysis of patient profiles are plotted on the left vertical axis while communities discovered from the analysis of therapeutic goals

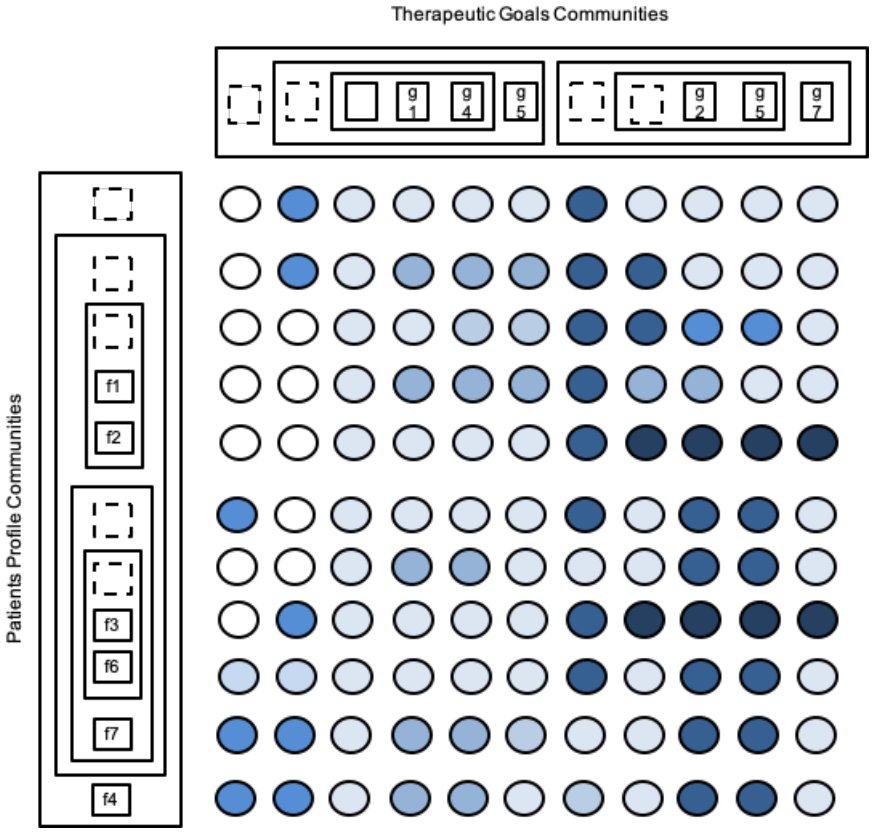

Figure 5.4: Graphical summary of community inter-relations.

are plotted on the top horizontal axis. In the center area of the figure, the relationship is showed, where dark blue colors mean more relationship.

When a new patient enters with a list of impairments, then the impairments are highlighted in the left axis as Figure 5.5 shows. In this example, the patient has impairments in f1, f6 and f7. To determine which are the best targets to assign, their rows are highlighted allowing to identify which are the most relevant therapeutic goals. In this patient, the group composed by g2, g5, g7 presents more relationship with f1. Regarding f6 and f7, the highest relationships are with g2 and g5. Thus, the system will propose the two alternatives for this patient: to assign g2,g5,g7 together or to assign g2 and g5 without g7.

But in problems where the number of variables (nodes) is huge, previously to exploit this detailed view, we propose to use a summary view of the total relationships (see Figure 5.7) where hierarchical representations of communities are removed.

5.3 Experiments

We conducted two types of experiments to evaluate the proposed methodology. First, we performed experiments using a random sample of 2000 patients extracted from the eICU database. Specifically, we exploited as patient profiles the information related to patient

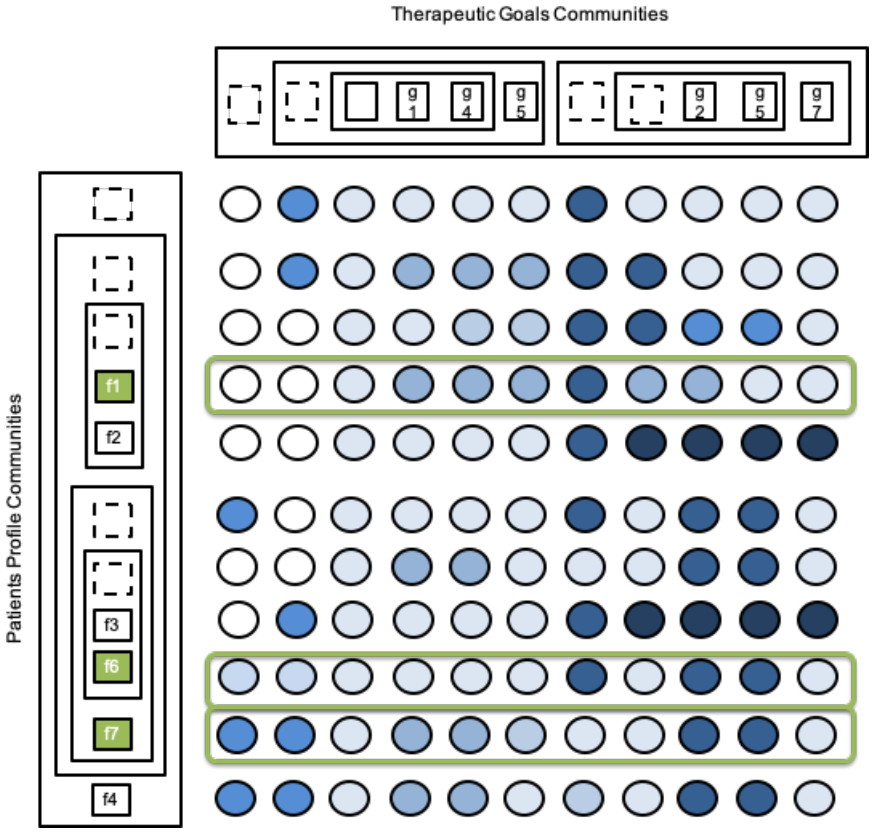

Figure 5.5: Graphical summary of community inter-relations with a concrete patient.

diagnoses and the therapeutic goals clinicians assigned to the patients. The results of these experiments are reported below. Next, we performed experiments using clinical data from Guttmann Neuro-rehabilitation EHR. These second experiments are reported in Chapter 6 where Guttmann's use case is introduced and explained.

5.3.1 Graph pre-processing

In a first stage we generated the co-occurrence graphs of patient profiles and therapeutic goals. Diagnoses in eICU dataset are codified using ICD-9 (International Classification of Diseases), a standard hierarchical classification of diseases. This codification allowed us to explore the relationships between types of diseases besides of exploring the relationships between specific diseases. The sample of 2000 patients includes 1046 different diagnoses.

Therapeutic goals are defined in a customized hierarchical structure. The number of therapeutic goals is lower than the number of diagnoses. Specifically, the sample of 2000 patients includes 83 different therapeutic goals.

To build the co-occurrence graphs we used the Networkx 2.4 library [36], a specific library for graph creation and manipulation. Without any filtering process, from the 2000 patients we obtained a graph for patient profiles (PP graph) composed of 1046 nodes and 3908 edges. Figure 5.6 shows the relationship between nodes and edges. The majority of the nodes (diseases) have an occurrence lower than 20%, i.e. it is a non condensed graph. In

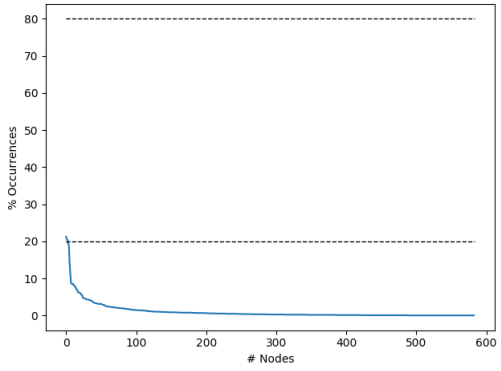

Figure 5.6: Graph diagnose properties.

other words, there is a huge volume of diseases that are not shared for many the patients. There are two possible interpretations: or the diseases are very detailed or the diversity of patient impairments is huge.

Regarding the graph of therapeutic goals (TG graph), without any filter, the graph is composed of 83 nodes and 3182 edges. Here the number of edges with respect to the number of nodes is huge producing a highly connected graph. Two reasons are behind of this graph structure. First, the number of therapeutic goals is low (83 goals over 1000 diagnoses). Next, the number of therapeutic goals per patient is lower generating a graph with similar edges than the PP graph.

5.3.2 Community detection and Inter-relations between communities

To determine the communities inside every graph, the Girvan-Newmann algorithm implementation from the Networkx 2.4 library was used [36]. For PP graph, a hierarchy of 2085 communities was generated. Regarding TG graph, a hierarchy of 62 communities was generated.

Finally, from two obtained community groups, the last step is to find the inter-relations between the communities of diagnoses and communities of therapeutic goals. Figure 5.7 summarizes graphically the inter-relations between the two types of communities. The figure shows the results without any filter. However, one point that should be remarked is the fact that there is a huge white zone on the image top. In other words, this white zone from around 500 impairments confirms that there are a lot of impairments that historically have never been assigned any goal. Specially, with the huge volume of impairments and goals, a useful strategy that could be performed is to filter this white zone. Another important point to remark is that Figure 5.7 shows the relationship of the complete historical data. However, an important goal is when a new patient enters the system and the CDSS needs to propose which are the most relevant goals for this patient. Given a concrete patient, a zoom in Figure 5.7 is mandatory.

Figure 5.8 shows the visualization of a concrete patient. Only the rows where the patient has an impairment are showed in the figure. Although the global vision of all the possibilities between impairments and goals is lost, for a clinical expert starting with a

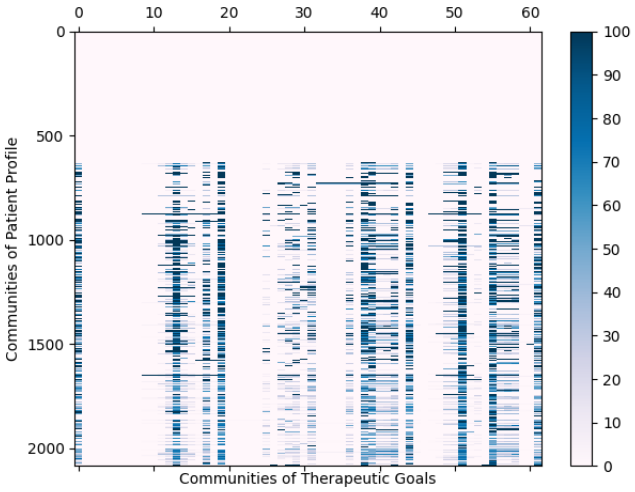

Figure 5.7: EICU communities.

simpler visualization may help to focus on specific patient issues. For instance, in this concrete patient, communities of therapeutic goals gc_0, gc_1, gc_2, gc_4, and gc_9 could be then presented to the experts. Then, highlighting which specific therapeutic goals below to several communities, experts may decide which are the final goals to prescribe.

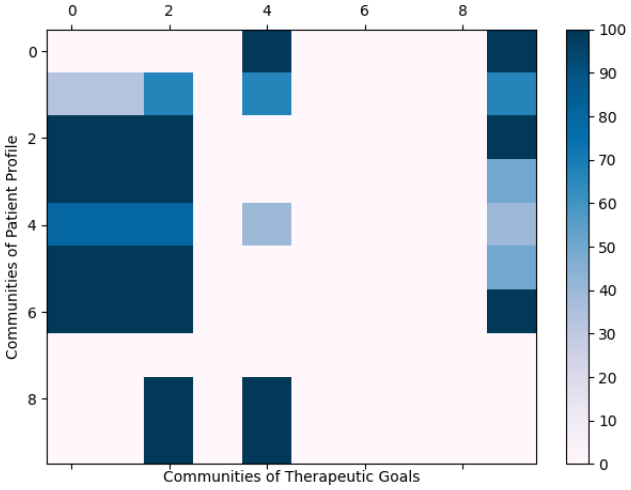

Figure 5.8: eICU communities for a specific patient.

Chapter 6

Use cases

This PhD has been developed in the context of two different clinical problems. The first part of the research in collaboration with the "Institut Guttmann Hospital de Neurorehabilitació". The second part of the research in the context of the Play&Sign project funded by the foundation "la Marató de TV3" and collaborating with IDIBELL (the project coordinators) and the "Hospital de l'Esperança" .

6.1 Institut Guttmann Hospital de Neurorehabilitació

The first use case is related to the Institut Guttmann Hospital de Neurorehabilitació The Institut Guttmann is a hospital specialized in the spinal cord, traumatic brain injury, stroke, and minority illnesses such as polio or Guilliam Barré injuries. It is a hospital focused on neurorehabilitation. The importance of neurorehabilitation has increased in the last years as a consequence of new society demands not only based on mortality or morbidity conditions, but also on improving the quality of life and chronic conditions of the population [73].

Neurorehabilitation is a complex process where patients, concurrently presenting diverse impairments and several diagnoses, receive multiple and complex treatments. High complexity patients, like patients who have suffered a traumatic brain injury, stroke, or spinal cord injury need long periods to regain or readjust to their loss of functioning in cognitive and physical abilities. These multiple impairments need to be treated by multidisciplinary teams composed by different professionals, such as rehabilitation doctors, nurses, physiotherapists, occupational therapists, neuropsychologists, and social workers. Patients in neurorehabilitation units typically spend from three to six months.

6.1.1 Clinical context

Within the clinical workflow presented in Section 2.1 (see also Figure 2.1), the Institut Guttmann helps patients when acute problems are stabilized and the mortality risk is reduced in intensive care units, a previous stage. Patients are survivors but they carry clinical sequelae that must be mitigated or solved.

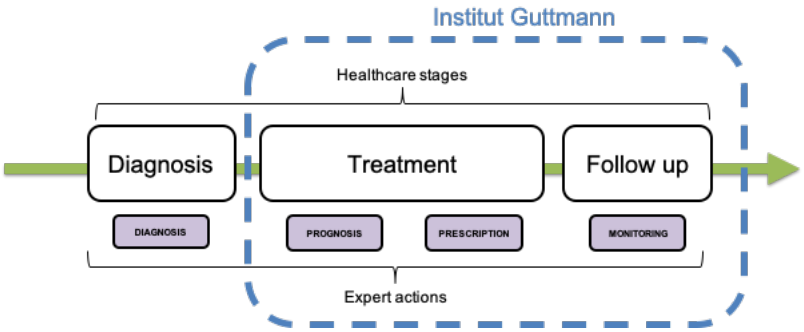

Figure 6.1: Clinical workflow at the Institut Guttmann.

Patients arrive at Institut Guttmann with the problem ethology diagnosed, e.g. an stroke diagnosis and a set of associated diagnoses. Moreover, information related to how the problem occurred is also collected as some additional physiological problems may be present.

In terms of the clinical workflow, the Institut Guttmann focuses specially on the treatment and the follow up stages (see Figure 6.1). Moreover, the treatment is separated in two phases: hospitalization and outpatient.

Hospitalization

During hospitalization patients spend all the days at the hospital, including sleeping. A hospitalization day is organized through an agenda of activities where several professionals have assigned different hours to work with the patient. It is considered an intensive treatment because patients perform activities all day. However, sometimes there are patients who are not able to perform some activities and spend more hours in the hospital room. The hospitalization period is critical to achieve the best recovery, i.e. to establish the foundations of the future quality of life. However, as it was mentioned previously, not all the patients follow the same progression and the continuous adaptation of treatments to the patient progress is not an easy task in the context of a multi-disciplinary intervention.

Outpatient

The outpatient is a phase where patients conduct their everyday lives at their home and only attend the hospital some regular hours a week. Outpatient is considered a transition phase where the intensity of neurorehabilitation treatments is periodically decreasing. Sometimes, some treatments that started in the hospitalization phase continue in the outpatient phase. However, many of them are new treatments and oriented to aspects related to personal autonomy and life in the community.

Follow up

Finally, there is a point where patients finish the clinical treatment phase and where the contact with the hospital becomes more sporadic. Then, all new acquired abilities are

performed autonomously by patients and their relatives. Institut Guttmann has a patient follow-up program to monitor their evolution over the years.

6.1.2 Knowledge and data sources exploited

One of the major initial efforts before starting the design of any Clinical Decision Support System is to identify the data resources and its availability throughout the clinical process. Specially, because not all information is available in the early stages of the patient treatment.

The data used to perform the experiments was gathered from EHR of the Institut Guttmann from 2007-2016. A sample of 1960 patients that suffered an Acquired Brain Injury caused either by a Traumatic Brain Injury (TBI) or a Stroke have been collected. From the 1960 patients, 792 are patients that suffered a Traumatic Brain Injury and 1168 a Stroke. From EHR different types of information have been extracted.

Demographic and Clinical data

The first data collected from the patient is related to the personal and social context. Examples of this data are gender, age, level of studies, type of work, and social support. This information allows to establish the personal and social context of the patient. The second type of information is related to existing diagnoses (e.g. general and specific etiologies) and to physiological and neuropsychological assessments performed from a battery of tests, such as the Fugl-Meyer Assessment of Motor Recovery (FMA), the Functional Independence Measure (FIM), the Stroop Color and Word Test (SCWT), or the Wechsler Adult Intelligence Scale (WAIS).

Patient Profile

The WHO has been working during the last years in the definition of an International Classification of Functioning, Disability and Health (ICF) to describe in a holistic vision the patients. At Institut Guttmann, ICF is used as the core element to describe patient profiles. ICF introduces different core-sets for different etiologies which are organized as taxonomies. At a first level of the taxonomy there are the four main chapters: Function chapter, Activity and Participation chapter, Structure chapter, and Environmental chapter (see Table 6.1 for an example). The second level of each taxonomy defines the scopes of intervention. Finally, taxonomy leafs contain specific items.

The chapter on Function factors groups items such as *Orientation*, *Attention*, or *Higher-level cognitive functions*. The chapter on Activity and Participation factors contains items such as *Speaking*, *Walking*, *Toileting*, *Dressing*, *Eating*, *Family relationships*, *Remunerative employment*, or *Recreation and leisure factors*. The chapter on Structure factors includes items such as *Structure of the brain* or *upper extremity indicators*. Finally, the chapter on Environmental factors includes items such as *Products and technology for personal use in daily living*, *Products and technology for personal indoor and outdoor mobility and transportation*, or *Health services, systems and policies*.

Chapter	ID	ICF name
Function	b114	Orientation
	b140	Attention functions
	b164	Higher-level cognitive functions
	b167	Mental functions of language
Activity and Participation	d330	Speaking
	d450	Walking
	d510	Washing oneself
	d530	Toileting
	d540	Dressing
	d550	Eating
Structure	s110	Structure of brain
Environmental		

Table 6.1: Example of the ICF taxonomy.

ICF Patient Profiles (ICF-PP) contain around 30 items from ICF brief coreset versions for TBI and Stroke. Values of ICF items range from 0 to 4, where 0 indicates normality and 4 complete impairment. We will consider that a patient has an impaired item when its value is 3 o 4.

Therapeutic Goals

Therapeutic Goals (TG), are goals that professionals assign to patients to establish the priorities in the Neurorehabilitation process. Specifically, the Institut Guttmann has a list of possible goals to achieve during the treatment process organized in a taxonomy. The first level of the taxonomy describes the areas of intervention such as Nursery, Functional Rehabilitation, speech therapy, or Neurophysiology. At a second level, therapeutic goals are grouped in intervention concepts such as *Bladder/Bowel removal, Skin care, Sanitary Educational, Joint Balance, Bipedestation, Transfers, Dressing, Patient Psychological Intervention, Familiar Psychological Intervention.* Finally, at a third level, the concrete goals are described by a name and an id such as *Vesicular sphincter control (1024), Bed-chair-bed (1235), Establishing cooperation pact (1659)* (see Table 6.2 for an example).

At Institut Guttmann there is an initial joint meeting where all experts involved in the neurorehabilitation of a specific patient reach a consensus regarding the list of initial therapeutic goals. The initial list of therapeutic goals incorporates, implicitly, the long-term prognosis. Afterwards, the goals are periodically revised taking into account the progress and the patient opinion. Patients are specially empowered to perform an active

Goal Group	ID	Goal name
Bladder /Bowel removal	1024	Vesicular sphincter control
	1039	Anal sphincter control
	1041	WC evacuation (3 person)
Skin cure	1051	Shower (3 person)
Sanitary Educational	1066	Prevention urinary system
	1067	Prevention digestive system
Joint Balance	1214	Upper right extremities
	1215	Upper left extremities
	1216	Lower right extremities
	1217	Lower left extremities
Bipedestation	1218	Autonomous
Transfers	1235	Bed - Chair - Bed
	1240	Chair - WC/Shower- Chair
	1245	Chair - Car - Chair
Dressing	1269	Upper body Autonomous
	1274	Lower body Autonomous
Patient Psy.Int.	1605	Sustained Attention
Familiar Psycho-logical Intervention	1659	Establishing coop. pact
	1662	Knowing how to act
	1665	Aim planning
	1666	Training technical aids
	1667	Training rehab. activities
	1670	Establish priorities

Table 6.2: Example of a portion of the taxonomy of therapeutic goals.

role in short-term goals.

In the data used to perform experiments and the proof of concept, 292 therapeutic goals were represented. Patients have a mean of 25 therapeutic goals. Those with less therapeutic goals have usually 15 goals and there are some that may reach 40 goals.

Interventions

Interventions are all the prescriptions and activities patients perform during the rehabilitation process. Some of them are pharmacological while others focuses on physical or cognitive aspects.

Based on the patient profile and therapeutic goals, experts prescribe a set of interventions. For instance, if the patient presents problems to walk, one goal will be to walk 100 meters without help (without crutches). In this context, one intervention will be to gain more muscle power on affected legs. Then, the expert may typically prescribe two-week sessions of one hour with the static bicycle.

Interventions are periodically assessed to determine the progress of the patient. The way the progress is reported may vary from a direct data acquisition from clinical facilities to a textual description incorporated into the EHR. The exploitation of the information generated during the activities performed by the patients is a key issue as it provides measures of the patient progress and also of the possible positive or negative interactions between different parallel interventions. Incorporating CDSS in this stage may become a high contribution for current clinical trend of personalized medicine.

6.1.3 Data enhancement

As mentioned previously, missing data is common in clinical domains. In the context of Institut Guttmann use case, missing data is specially frequent in early stages of the treatment. The first reason is that many patients begin the rehabilitation process when their clinical condition is still critical. Some of them require some weeks to be able to perform specific tests. The second reason is that some results may require clinical cultures that take several days. Moreover, since many of them are specialized assessments that require a patient effort, the prioritization of them may vary patient to patient.

An experiment has been conducted to determine if the dynamic imputation of some missing items in a specific questionnaire may improve the prediction of the global score. Specifically, the Functional Independence Measure (FIM) is used to prove this. This evaluation scale is composed of 18 items, which evaluate aspects of the physical, psychological and social function of the individual. FIM is administered by physiotherapists. FIM is used to assess a patient's level of disability as well as the change in patient's condition in response to rehabilitation [44].

Experiments have been conducted as described in Section 3.5. That is, randomly selecting different combinations of known features and performing a cross-fold validation. As Figure 6.2 shows, depending on the number of known features the imputation method with lowest MSE varies. RF strategy, i.e. no imputation, is more competitive when the number of known features is high enough (from 10 to 12 known features). However, when the

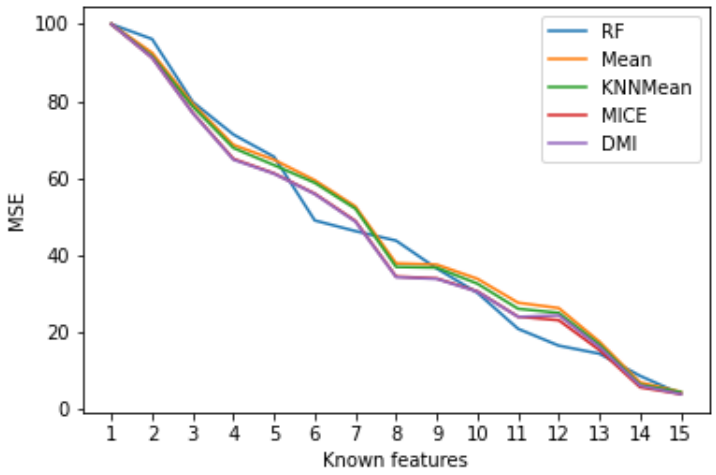

Figure 6.2: FIM with several imputation strategies.

number of known features is low, imputation methods reach lower errors. Globally, the most competitive measures are MICE and DMI (see Table 6.3). Taking into account that, as expected, RF is the most competitive strategy when the number of known features is high, MICE and DMI become even more competitive when the number of known features is low.

RF	Mean	KNNMean	MICE	DMI
45.6	47.37	46.56	**44.77**	**44.85**

Table 6.3: MSE when imputing missing features for FIM score.

6.1.4 Discovering Non-explicit relationships

Proposed community detection techniques have been applied to discover non-explicit relationships between patient profiles and therapeutic goals. That is, between ICF items and therapeutic goals (TG). As it was described in Chapter 5, the proposed methodology is based on graph methods and three main stages: a pre-processing stage, the identification of intra-communities, and the identification of inter-relations.

Pre-processing

The process started with the construction of the co-occurrence graphs from the information gathered from 1960 patients. As for ICF-PP graph, it was initially composed of 18 nodes, i.e. 18 ICF items, and 120 edges (see Table 6.4). After removing ICF items either common in most patients or anecdotal, the resulting graph holded 11 nodes. Next, reporting and removing the edges with highest weights, i.e. those ICF items clearly co-related, the remaining number of edges was 39 and none of the nodes was removed. Table 6.1

Name	Nodes	Edges	Average Degree	Density
ICF-PP	11 (18)	39 (120)	7.09 (11.78)	0.70 (0.78)
TG	23 (292)	28 (30587)	2.52 (209.50)	0.11 (0.72)

Table 6.4: Statistics of co-occurrence graphs.

summarizes the resulting ICF sub-taxonomy considered by the ICF-PP graph and co-occurrences are shown in the graph on the left in Figure 6.3. Colors in nodes represent the different ICF chapters.

Regarding the TG graph, it was initially composed of 292 nodes, i.e. 292 therapeutic goals, and 30587 edges (see Table 6.4). When reporting and removing anecdotal goals, the number of nodes dropped to 50. Specifically, the pre-processing over TG showed that there were around 100 TGs with an occurrence lower than 10% and another 100 with an occurrence lower than 20%. This result showed that the core of therapeutic goals is lower than expected. Finally, after reporting and removing common goals and highly weighted edges, the resulting number of nodes dropped to 23. Table 6.2 summarizes the resulting TG sub-taxonomy considered by the TG graph and co-occurrences are shown in the graph to the right in Figure 6.3. Colors in nodes represent different groups of therapeutic goals.

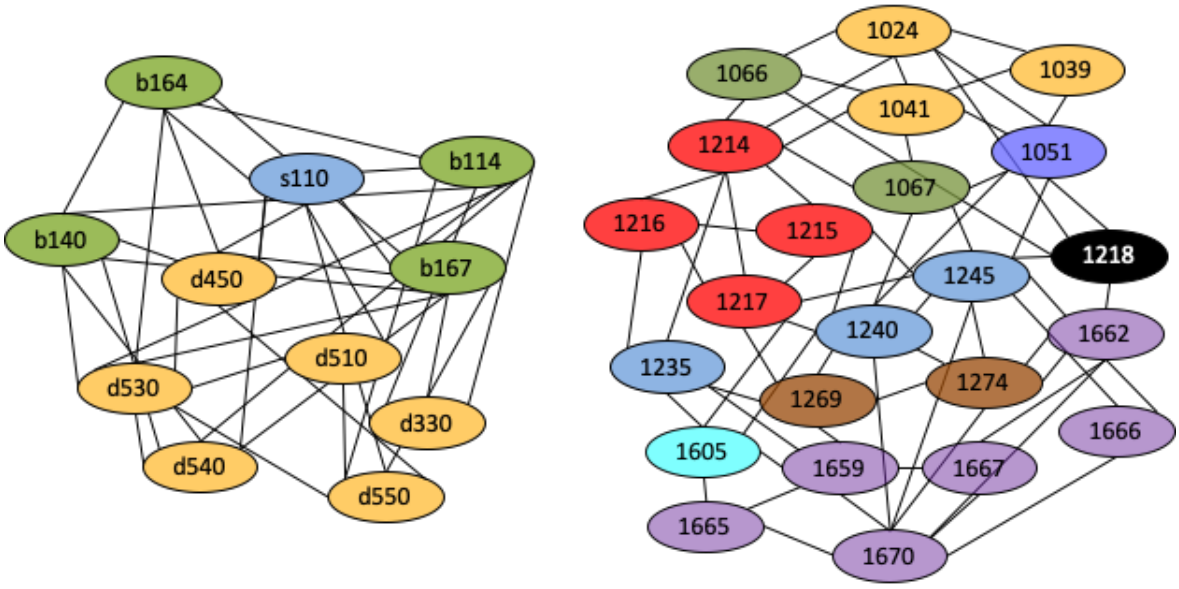

Figure 6.3: Co-occurence graphs for Guttmann's use case.

Community detection

From the two co-ocurrence graphs shown in Figure 6.3, the GN algorithm has been applied. GN algorithm provides a hierarchical structure of communities where leafs are individual items and tree nodes define the communities of the items included in the sub-tree. Lower nodes in the tree model imply stronger communities. In turn, as far as two items are connected in the tree, less co-related are these two items.

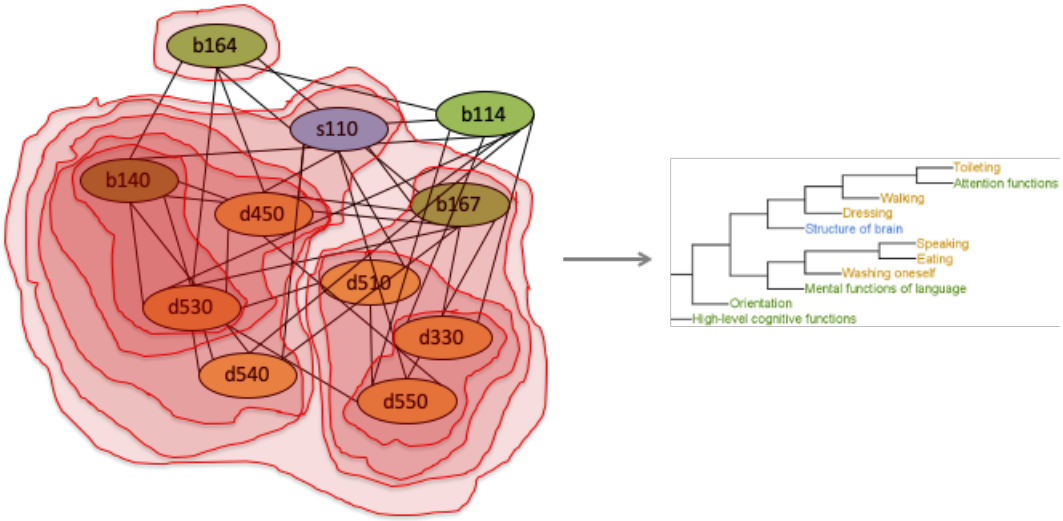

Figure 6.4: Communities for ICF Profile.

Note that leafs and tree nodes are vertically aligned to make explicit the result of the GN algorithm. For instance, the strongest community arisen from the GN algorithm in the ICF-PP graph (see Figure 6.4) is the community composed by *Toileting* (d530), and *Attention functions* (b140) items. Next, the second strongest community is composed by *Speaking* (d330), and *Eating* (d550) items. Then, the *Walking* (d450) item is added into the first community.

The hierarchy of ICF-PP communities was initially surprising. The arisen community joining *Toileting* and *Attention* was not expected as they are related to different domains (physical and cognitive, respectively). Nonetheless, the Toileting process needs abundant patient attention. The later inclusion of the Walking item into the community is coherent as it is a function that is necessary for the toileting process and also in other daily activities. Following the generalization of this community, *Dressing* (d540), and *Structure of brain* (s110) were added.

The second main branch in ICF-PP communities starts with *Speaking* (d330), and *Eating* (d550) items, two concepts that are related with the mouse/neck. Next *Washing oneself* (d510) joins the community. This strong connection was not expected as this item is apparently closer to *Toileting* (d530). Finally, *Mental functions of language* (b167), which it is an indicator closely related to the Speaking and Eating is added. Interestingly, *High-level cognitive functions* (b164) item is only included in the most general community.

Regarding the communities generated from the TG graph, it is organized as a wide and shallow tree structure. The reason behind this organization of communities is that therapeutic goals are less clustered and less co-related. The GN algorithm starts by creating small communities of 2-3 indicators. Most of these small communities never join other communities. These small communities are clearly related to the same intervention concepts. For instance, the community composed of *Prevention urinary system* (1066) and *Prevention digestive system* (1067) is related to educational aspects or the two communities related to the extremities. Interestingly, these two communities of goals are clustered by the body size (left and right). That is, the first community groups *Lower right extrem-*

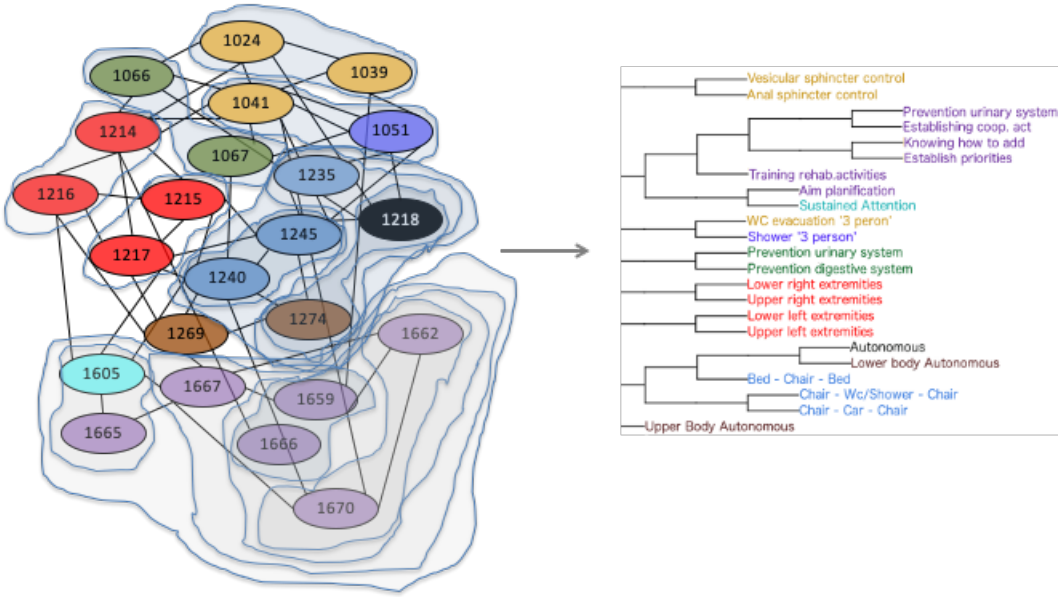

Figure 6.5: Communities for therapeutic goals.

ity (1216) and *Upper right extremity* (1214) while the second community groups *Lower left extremity* (1217) and *Upper left extremity* (1215). An example of community linking goals of different intervention concepts is the community composed of *WC Evacuation* (1041) and *Shower* (1051), related to personal hygiene and self-care. Another interesting example is the community composed of *Standing* (1218) and *Dressing lower body* (1274). These goals belong to different intervention concepts but are both related to the legs.

Two hierarchical aggregations of communities arise in the TG graph. One tree branch groups items related to family issues. The first sub-communities group items such as *training technical aids* (1666), *establishing cooperation pact* (1659), *knowing how to act* (1662), and *establish priorities* (1670). Then, other items related to planning issues (1667 and 1665) are incorporated. Interestingly, *Sustained Attention* (1605), an item devoted to patient psychological intervention arises as related to this community. The second branch groups items devoted to transfers and dressing. Specifically, transfers from/to chair (1240, 1245, and 1235) with dressing autonomy items (1274 and 1269).

Inter-relations between communities

To analyze the inter-dependencies between ICF-PP communities and TG communities, the $I_{i,j}$ measure (see Section 5.2.3) has been introduced. Applying the $I_{i,j}$ measure, a ranked list scoring the relationships between communities of the two clinical aspects (ICF-PP and TG) have been obtained. For instance, Figure 6.6 shows two communities highly ranked. Due to there are 19 different ICF-PP communities and 37 different TG communities the ranked list of scores has a length higher than 700. Of course, focus can be laied only on the top of the list, but then interesting relationships could be lost.

Thus, since the amount of relationships to explore is very high, all the information is summarized in a colored matrix (see Figure 6.8). In a colored matrix rows represent

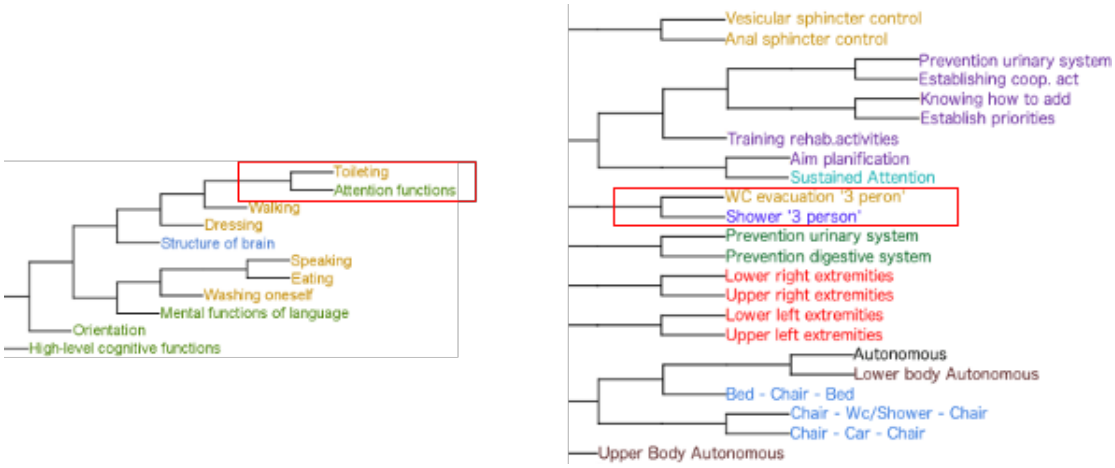

Figure 6.6: Detecting inter-communities.

ICF-PP communities while columns represent TG communities. Communities are sorted to preserve the hierarchical (tree) structure. Specifically, squares represent single items while rectangles encompass communities. Darkness level of cells corresponds to their interdependence value, i.e. the greater a value is, the darker its corresponding cell will be.

Summarizing the information in a sorted matrix provides several advantages. First of all, exploiting the darkness level of cells allows a qualitative comparison between communities. Next, sorting the communities preserving the hierarchical structure, provides an intuitive way to enhance clusters and patterns. For instance, from Figure 6.7 one highly interrelated region on left and another more sparse relation at right can be clearly observed. More interestingly, a high correlation is established between one ICF-PP tree branch (top branch in Figure 6.4) and one TG tree branch (deeper branch in Figure 6.5). From a clinical perspective, this result stresses that *Activity and participation* impairments are addressed by priorizing therapeutic goals involving family intervention.

An additional opportunity that the matrix representation is providing, is the analysis of a specific patient. In Figure 6.8 a specific patient profile (filled squares on the left) and the therapeutic goals associated to this patient (filled squares on top) have been highlighted. Additionally, squares/rectangles in the matrix are drawn to highlight how this specific patient stands with respect of characteristic profiles. For instance, the highlighted patient has only ICF impairments on the top ICF-PP tree branch described above, but presents therapeutic goals (communities) not commonly correlated with the ICF impairments.

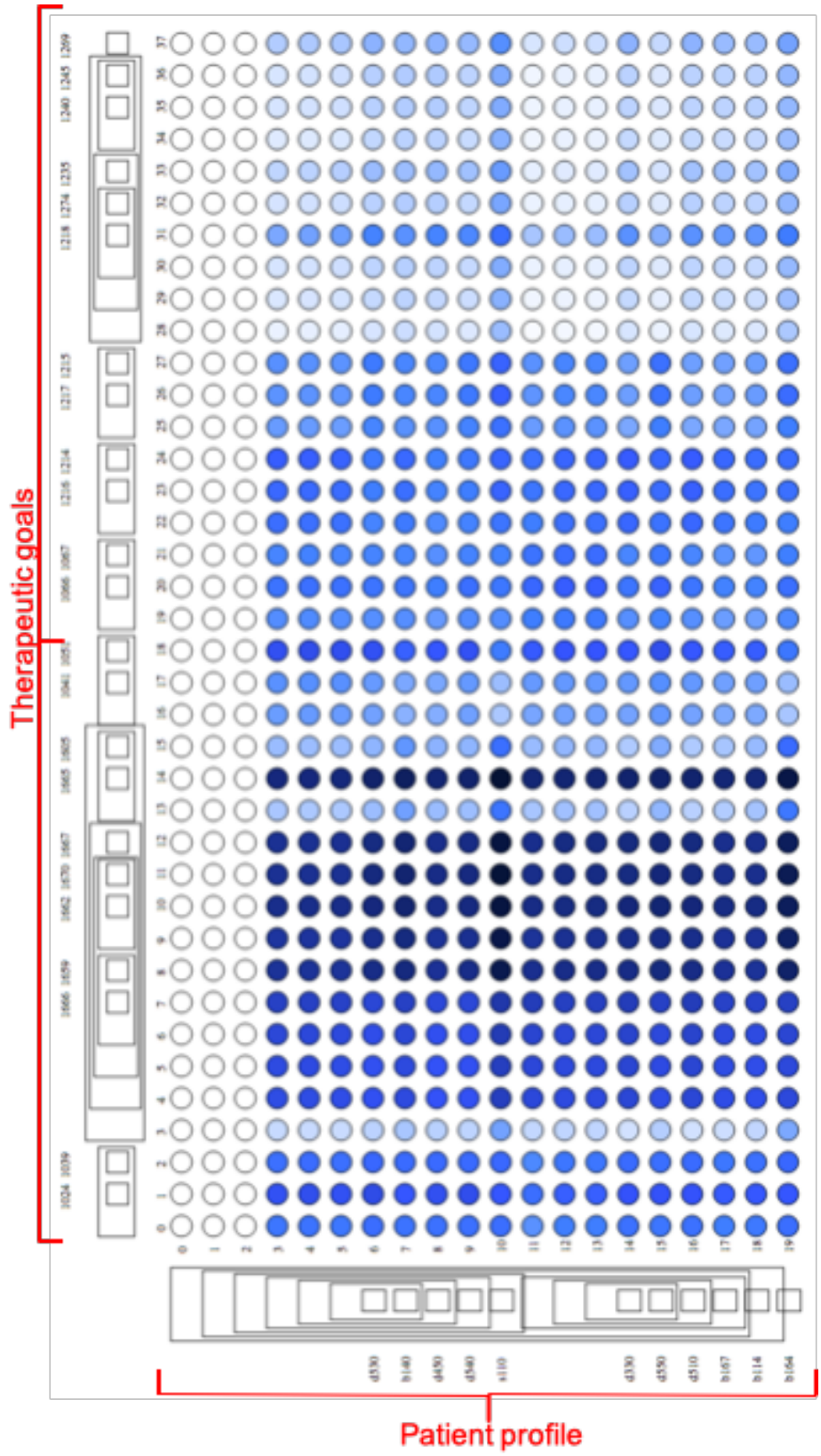

Figure 6.7: Inter-dependencies between ICF-PP and TG communities.

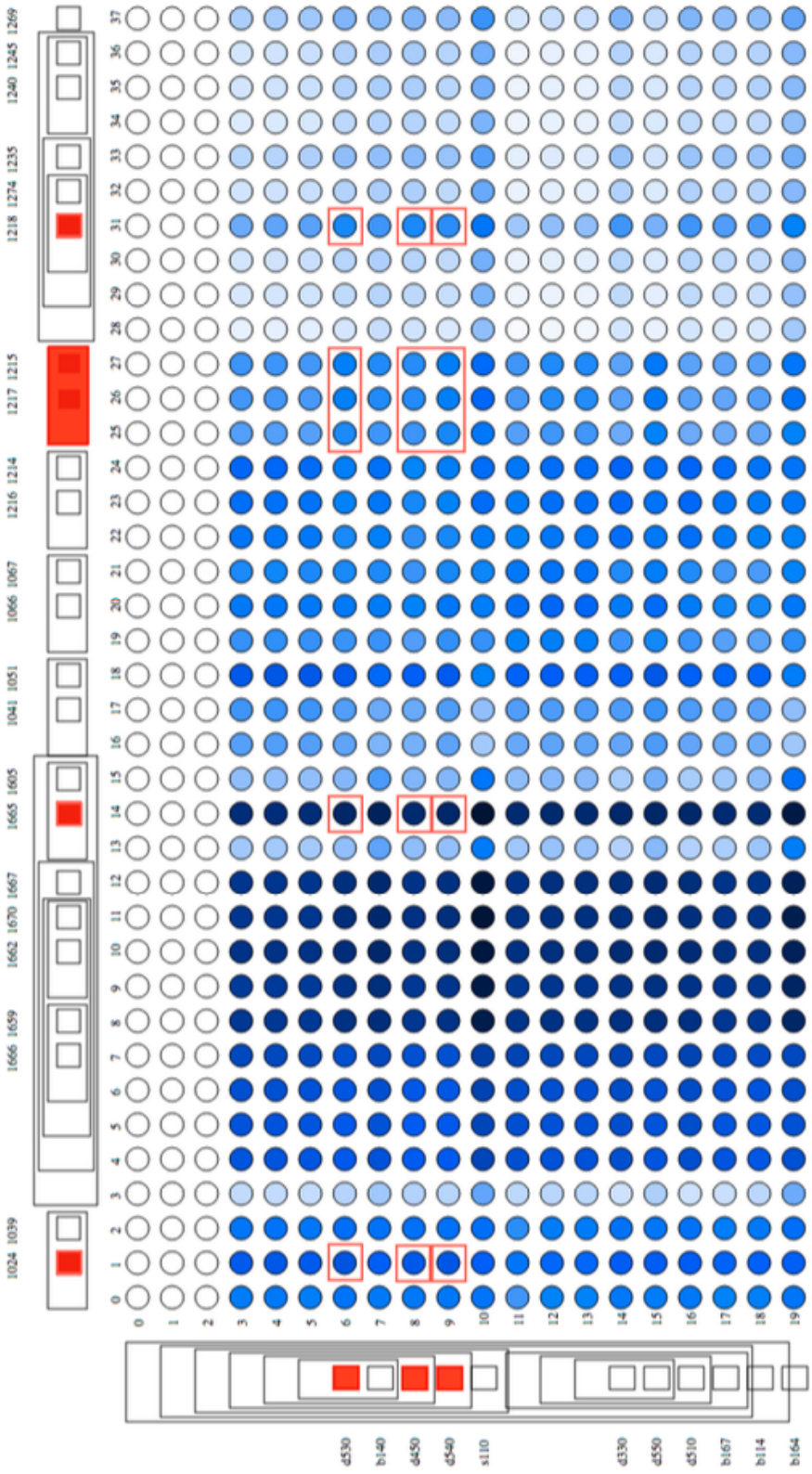

Figure 6.8: Inter-dependencies between ICF-PP and TG communities for a concrete patient.

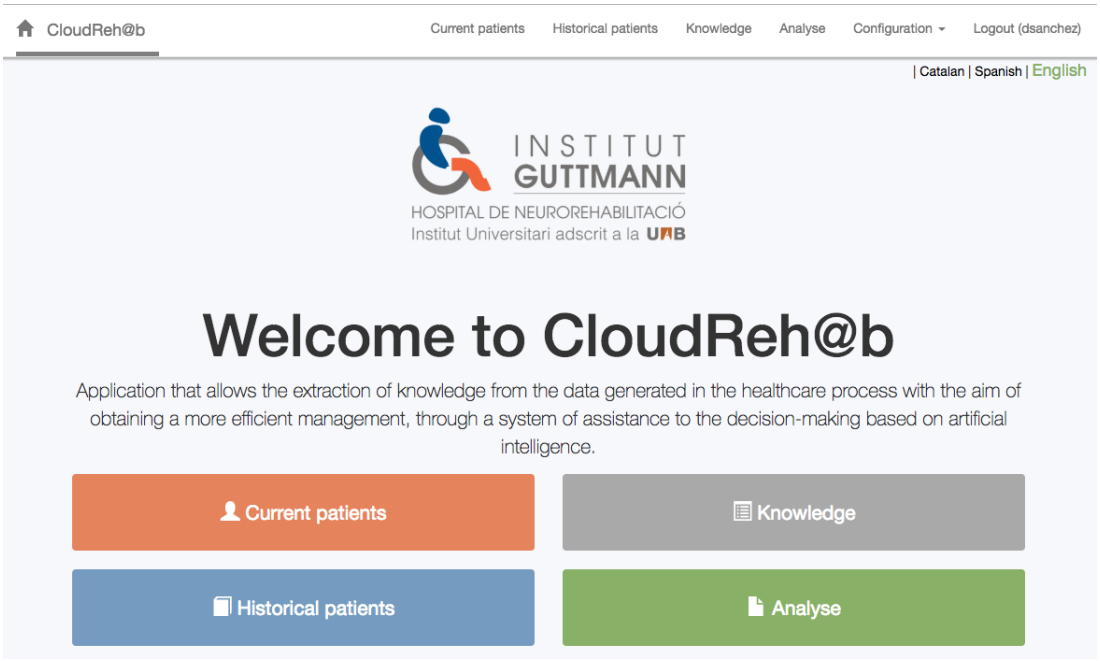

Figure 6.9: Main page.

6.1.5 Proposed CDSS

The proposed CDSS was designed as a web application. The CDSS is defined as a complementary tool because the EHR is more oriented to data storage rather to provide a whole picture of the patient. Several clinical experts are involved in the patient rehabilitation process and data is stored from different points of view. In the current Guttmann EHR sometimes is very difficult to have a whole vision of the patient because the information tends to be organized and presented by areas, generating the effect of different information silos.

The proposed CDSS is oriented to clinical experts and with the aim of providing a whole vision of the clinical workflow. The proposed CDSS, CloudReh@b, is divided into four main components (see Figure 6.9).

The first component is oriented to facilitate the follow up in all the activities a patient is involved during the rehabilitation. Initially, a list of patients is showed allowing to select a specific patient. In this table some basic patient features such as the age, origin and the medical unit are showed and may be used to filter patients. For instance, if the clinical experts would like to visualize all the patients with stroke, they can filter patients by the medical unit (see Figure 6.10). Once a patient is selected, all the current and past treatments are showed. Different colors used help to easily focus the expert on when the treatment starts, finalizes, or which are the durations (see Figure 6.11 for an example). Additionally, sometimes the treatment categorization is not well structured as it responds more to non clinical terminology. For instance, in the example showed in Figure 6.11, a 'First visit' appears when the patient has already started the treatment process. Therefore, such visit should not be labeled as the first.

Then, the next step is to show the treatments by weeks and days. Figure 6.12 and Fig-

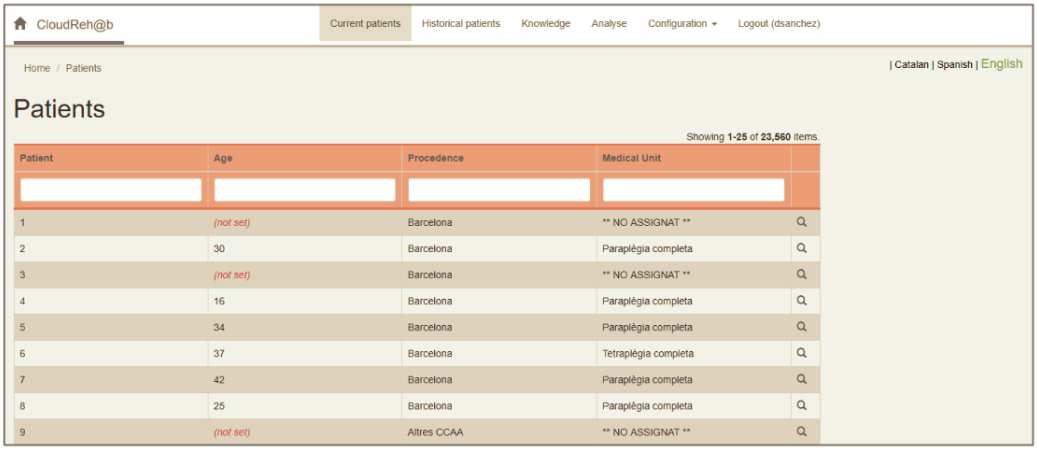

Figure 6.10: List of patients.

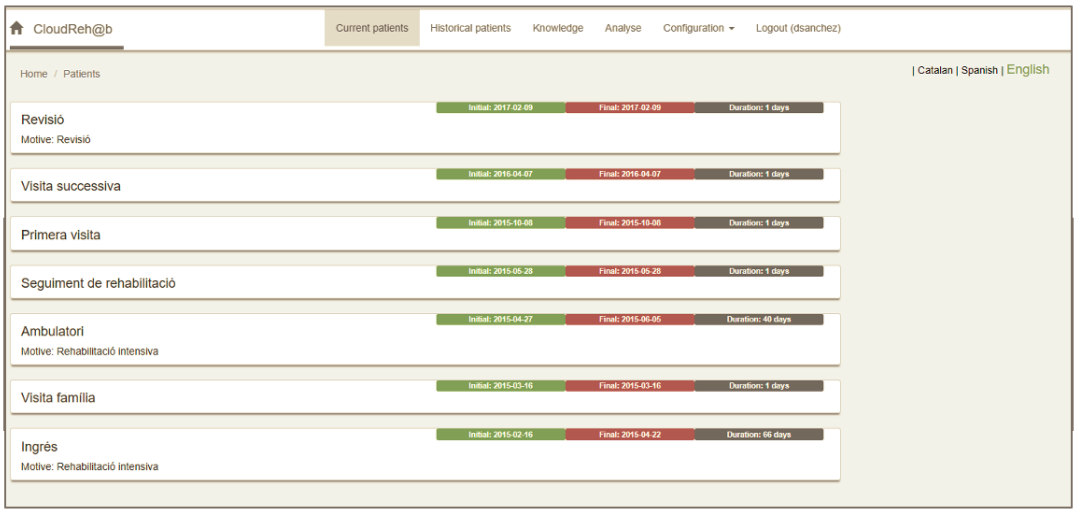

Figure 6.11: List of patient treatments.

ure 6.13 show how CloudReh@b organizes this information. This patient view is quite difficult to obtain with the current EHR as information generated by different experts is not summarized in EHR. For instance, as Figure 6.13 shows, only the scale "bateria" is administered in the first week. During the second week, where the rest of scales and questionnaires are passed, the experts may decide the therapeutic goals. Moreover, entering into the details showed in Figure 6.12, it may be noticed that the scale bateria is reported in the EHR on Friday. Thus, during initial days important features are still missing. Clinical experts require a lot of effort and time to complete the incorporation of the scales and questionnaires into the EHR. Therefore, DMI imputation method together with the proposed τ factor for confidence measures provide support to organize the first weeks of rehabilitation activities.

The second component is based on a historical view of the clinical workflow. Once the patient leaves the hospital, the support of the CDSS is focused on providing a summarized view. Usually, the interest of clinical experts with historical patients is to compare them

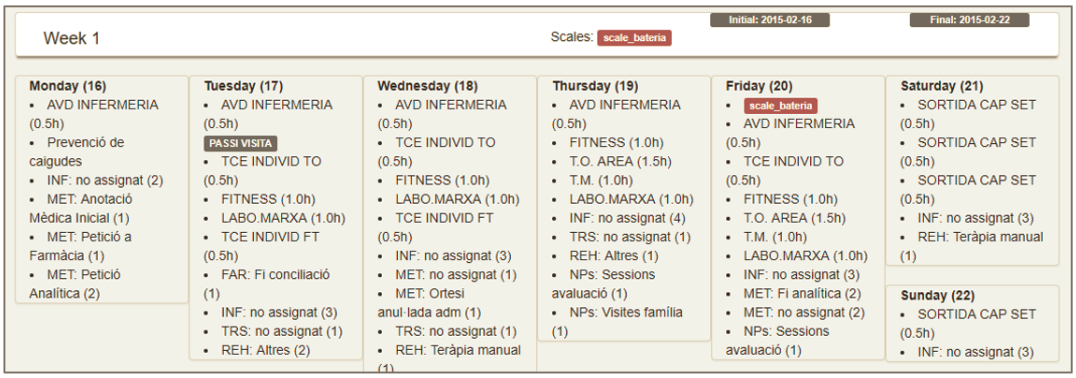

Figure 6.12: Daily treatment view of a patient.

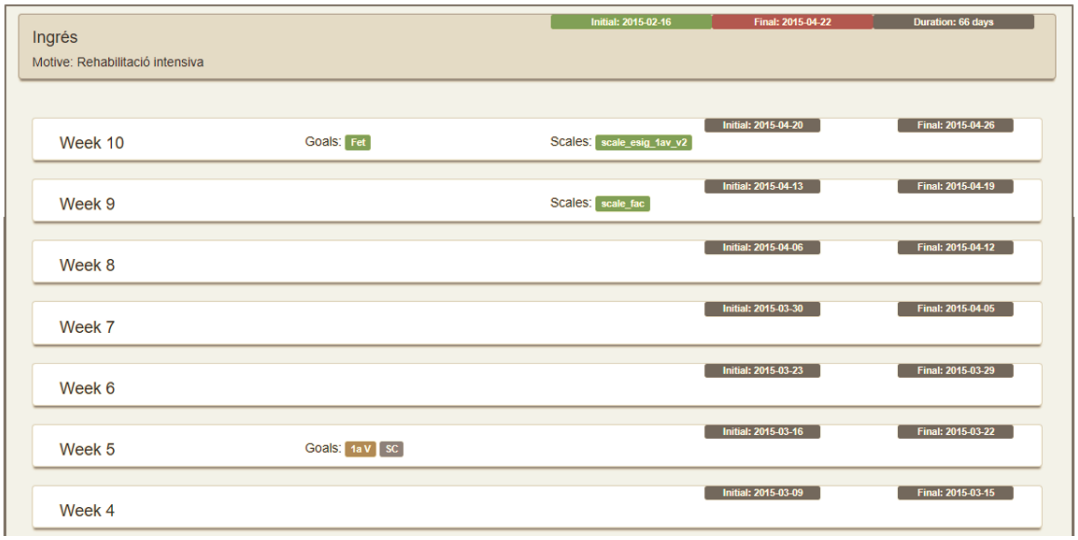

Figure 6.13: Weekly treatment view of a patient.

with a current patient. They have in mind similar past patients, and it is important to provide a resume of the whole clinical process.

The third component is devoted to explain the information that is exploited by the CDSS. The knowledge component tends to have less attention but is utterly important as the data that the system is using may be explored. Figure 6.11 shows details related to therapeutic goals where goals are initially displayed by areas. For instance, clinicians have few goals of respect for nurses.

Finally, the last component is related to the analysis of patients. Although this component may be mainly of interest for the visualization of current patients, the proposal has been generalized to allow any patient (current or historical). Specifically, Figure 6.14 shows the last ranking of the most similar cases. The first column highlights the changes from the previous estimation (last week). Note that the top five cases were also previously close neighbors but that there are three cases (patients 6386, 507, and 18967) that now become far.

Relevant Cases

	actual order	patient	distance
↑1	0	30347	1.0
↑1	1	32344	1.0
↑3	2	30816	1.41421356237
↑3	3	2558	1.41421356237
↑3	4	8196	1.41421356237
↓301	305	6386	5.83095189485
↓1160	1163	507	15.2970585408
↓1606	1606	18987	23.4733891886

Figure 6.14: Ranking of nearest neighbors.

Solution distributions

Days

Data	Max	Mean	Median	Min	Std
days	131	87.4	86	53	26.1885471151

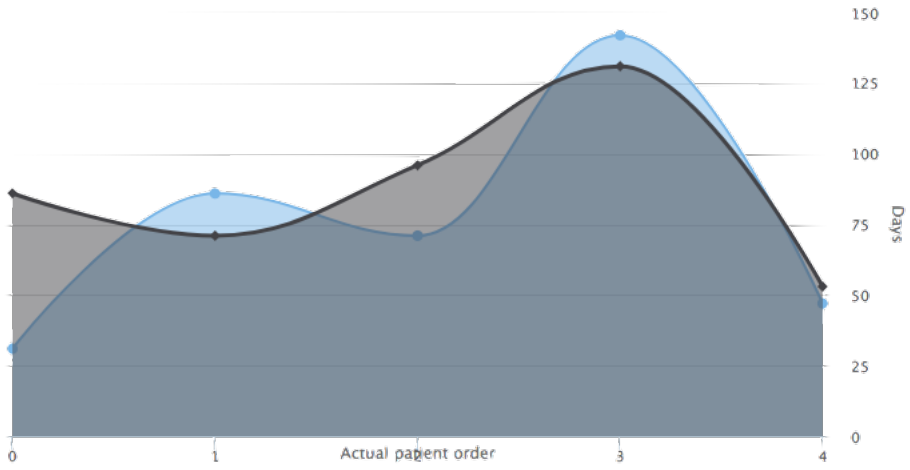

Figure 6.15: Exploring the distributions of solutions.

Figure 6.15 allows to compare the evolution of predictions regarding the expected hospitalization days. Specifically, quite similar predictions can be observed, i.e. we are in an *Expected* scenario according to Figure 2.6. Additionally, based on the same strategy of finding the closest patients, CloudReh@b may show how a patient could evolve in some scale or questionnaire (see Figure 6.15).

PostScales

Data	Value
scale_drs_apperance_cares	1 (=0)
scale_drs_empleability	3 (=0)
scale_drs_feeding	0 (=0)
scale_drs_functionality_level	3 (=0)
scale_drs_hygiene	1 (=0)
scale_drs_motion_response	0 (=0)
scale_drs_open_eyes	0 (=0)
scale_fac_category	1 (↓1)
scale_fim_appearance_care	4 (=0)
scale_fim_bath_shower	4 (=0)
scale_fim_bath_use	4 (=0)
scale_fim_bed_chair_wheelchair	4 (↓1)

Figure 6.16: Expected scale outcomes.

Figure 6.17: MST setup.

6.2 Music Supported Therapy

The second use case is related to Music Supported Therapy (MST), in the framework of the Play&Sign project, a project funded by La Marato de TV3 and performed in collaboration with IDIBELL, and "Hospital de l'Esperança".

The goal of the Play&Sing project, is to develop an AI platform to support home-based self-training interventions for chronic stroke patients. Stroke currently ranks as the second most common cause of death [45] and the second most common cause of disability-adjusted life-years worldwide [31]. Globally, the prevalence of stroke grew by more than 80% from 1990 to 2010 [26], and the ageing of the population continues to increase the individual-level, societal and economic burden caused by stroke. Motor deficits of the upper extremity (hemiparesis) are the most common and debilitating consequences of stroke, affecting around 80% of patients [62]. These deficits limit the accomplishment of daily activities affecting social participation and causing profound detrimental effects on quality of life [42]. The project is proposing and testing a new home-based self-training music supported therapy to induce upper limb motor recovery using several musical instruments and a tablet-based application in a Randomized Control Trial (RCT) study (see Figure 6.17).

6.2.1 Clinical context

The project is oriented to stroke chronic patients. Within the clinical context, chronic patients are involved in the follow-up phase where the hospitalization resources are finalized and the patients are returned to their homes (see Figure 6.18).

Chronic patients perform periodical revisions to measure how the patients are maintaining their recovery level. The Play&Sign project aims to provide an additional rehabilitation treatment for chronic patients able to be scaled to a high number of patients as it is based on exploiting digital capabilities on a self-training of patients. Additionally, patient

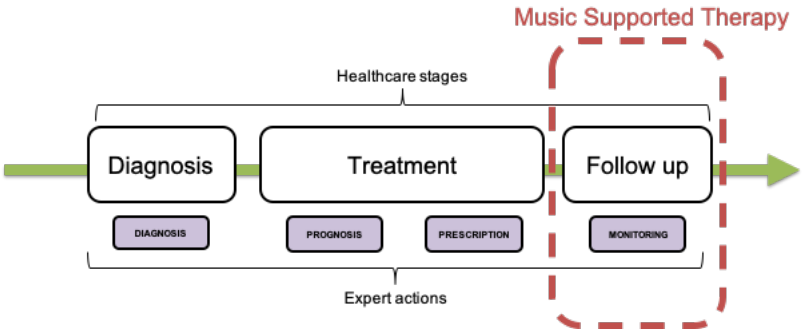

Figure 6.18: Clinical workflow at the Play&Sing project.

activities can be deeply monitored and personalized with the support of AI techniques, i.e. with the design of a CDSS.

6.2.2 Knowledge and data sources exploited

Although patients come from the long-term treatment at the hospital, the initial data from the MST is related to personal and social context. Data previously captured during hospital treatments is not incorporated.

Demographic and Clinical data

The initial information incorporated in MST is related to personal demographic data and social context. Analogously to the Institut Guttmann use case, information such as gender, age, or studies is collected.

Due to the aim of Play&Sign is to induce upper limb motor recovery, clinical information is focused on features related with upper limbs. Specifically, five different clinical motor tests are incorporated as clinical information: the Fugl-Meyer Assessment of Motor Recovery (FMA) [30], the Nine Hole Pegboard Test (NHPT) [58], the Box and Blocks Test (BBT) [49], the Action Research Arm Test (ARAT) [47], and the Chedoke Arm and Hand Activity Inventory [33]. Moreover, since cognitive well-being and quality of life aspects are utmost important at any rehabilitation process, some information such as Wechsler Adult Intelligence Scale (WAIS) or the Barcelona Music Reward Questionnaire have been also incorporated.

Patient Profile

Patient profiles summarize the scores obtained from motor and cognitive tests modulated by demographic information.

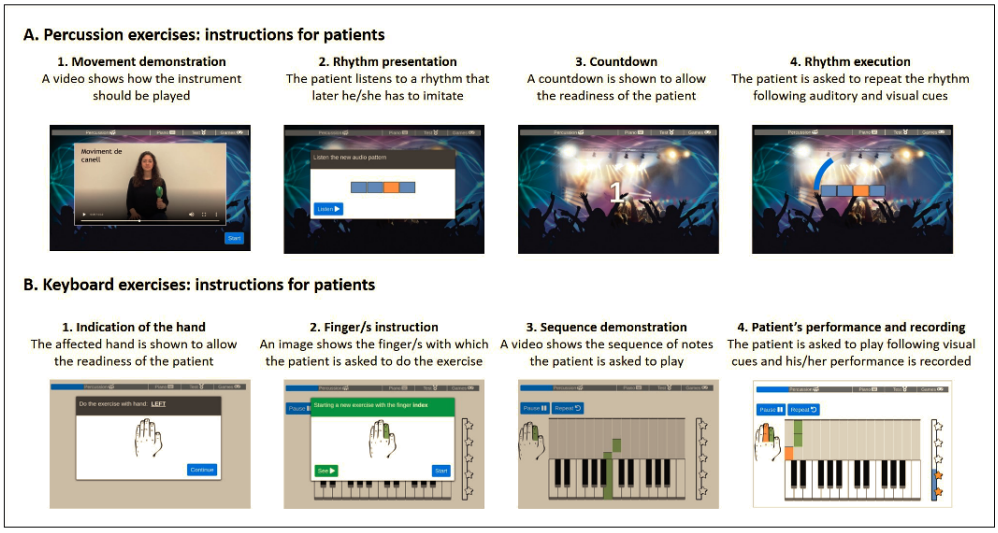

Figure 6.19: Individual home-based self-training sessions.

Therapeutic goals

The performance of patients in a set of selected piano exercises played during the first session is compared to control subjects and to historical patients to determine therapeutic goals. That is, the expected improvement in the execution of piano exercises.

Interventions

The interventions consist of 4 weekly one-hour sessions (total program duration: 40h). The training program comprises three individual home-based self-training sessions and one group session per week. MST is focused only on self-training sessions.

Two types of interventions are defined in MST. The first is oriented to more gross mobility. Nine different percussion instruments are given to the participants to perform certain movements following specific rhythms. The goal is to use percussion instruments, which provide sound feedback, to introduce some movements such as arm rotation to 90^{o} and so on. However, due to these instruments are not connected to any device, they cannot be monitored and do not generate any performance feedback.

The second type of intervention is focused on the use of an electronic piano keyboard. The aim of the piano is to perform fine movements with the fingers through short melodic exercises. The keyboard is connected to a tablet showing how to perform each exercise. Moreover, it provides visual feedback to the patient allowing to guide the progress of each exercise. Additionally, all the performance is captured by the tablet and information such as how many errors participants commit, which is the pressure associated to each key, and how much is the time the participants need to finish an exercise are sent to the MST server and later used by the CDSS. Some examples of interventions are shown in Figure 6.19.

With the aim to socialize the patients with others with similar impairments, once per week, a one-hour virtual group session of music therapy is conducted using a video com-

munication platform that is also installed in the electronic tablet provided to participants. Although participants perform several exercises with the percussion instruments, the performance in these sessions is not captured by the platform. However, some notes from the therapist will be used in the future to provide more feedback to the platform.

6.2.3 Methods and algorithms incorporated

During each session, patients perform rhythmic exercises with percussion instruments and play some musical sequences with a keyboard connected to a tablet-based application (APP). The APP assigns a different color to each finger (orange and green in the figure are associated to index and ring fingers) and highlights the fingers pattern on top of the displayed keyboard. The APP interacts with an AI Platform by means of an https connection. The AI platform also supports the therapists in the design of training sessions and analysis of patient's performance. In addition to monitoring and analytic components, the AI platform incorporates prediction and prescription components.

The role of the prediction component is to determine the effect of the intervention. To assess such effect, additionally to the patient profile generated from the different motor and cognitive tests, three different indicators are used to assess the performance of a patient during the training sessions: the execution speed, the key pressure, and the errors. These indicators are calculated for each finger and fingers pattern.

The prediction component calculates a first estimation from the initial exploration of the patient. After each patient session, the prediction component revises the prediction, by incorporating the performance information of the last session and stresses the changes to the therapists. The predictive component has to deal with partial information that, moreover, is incrementally updated. We incorporated DMI and τ factor as a proposal to improve prediction and to provide confidence measures that take into account missing information.

The role of the prescription component is to propose to each patient the most pertinent activities combining the therapeutic goals prioritized by therapists and the preferences of the patient. The task is modeled as an optimization problem where multiple activities (melodic sequences played with a specific fingering pattern) have to be distributed along several training sessions trying to maximize the value of the intervention, where the value depends on the estimated contribution of each activity. The contribution of each activity is continuously estimated from the analysis of patient performance, the therapeutic goals, the preferences, and the experience from other patients accumulated in the platform.

The prescription component exploits three sources of information: intra-patient information, inter-patient information, and information from control subjects (performance of healthy people). The first step is to select a subset of appropriate exercises, i.e. exercises estimated as feasible and not too trivial (e.g. which have been easily overcome previously). This selection is performed by using a CBR system that, measuring clinical and performance similarities, ranks activities according to past experience. The second step is to weight each exercise according to its effectiveness, by comparing with the effectiveness in similar patients. Next, activity weights are refined according to patient preferences and therapist goals. Finally, the list of activities, their associated weights, and intervention

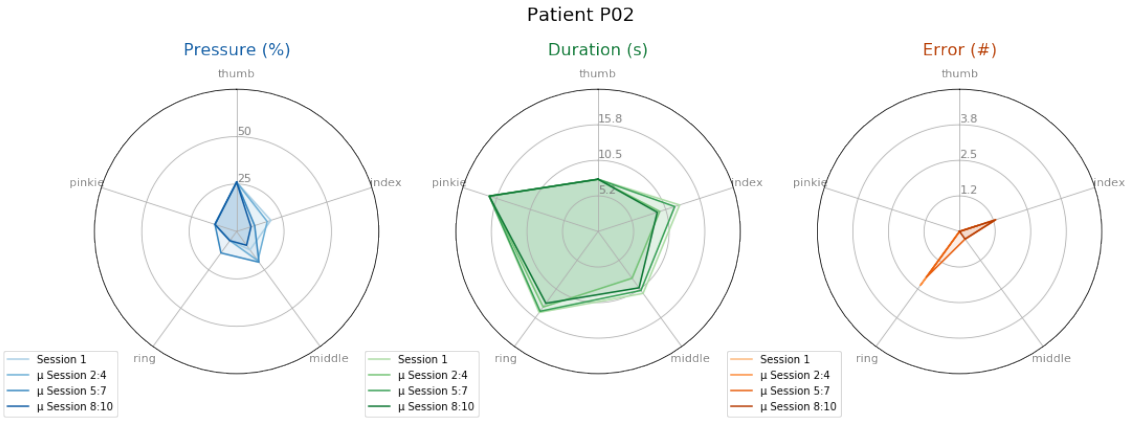

Figure 6.20: Patient performance evolution.

constraints (such as the maximum number or repetitions per activity) are sent to the optimizer.

The methodology proposed is also exploited to find non-explicitly relationships to help therapists to visualize exercises with higher impact giving specific patient profiles. But, due to the restrictions imposed as a consequence of COVID-19 pandemic with fragile people, such as people that suffered a stroke, the recruitment process for the study was stopped. Without patient information evaluating patients at the beginning of the treatment and at the end of the treatment it has been not possible to evaluate proposed methods and algorithms. However, during the development phase, a preliminary study was conducted with a few real patients (performing 30 sessions each patient). Although the initial aim was to prove that the proposed platform worked well in a home environment, all the patient interventions were also captured.

6.2.4 Proposed CDSS

The proposed CDSS is conceived as an independent tool not connected to any EHR. Proposed CDSS is based on a web app that provides the front-end with patients through a tablet and a web dashboard for the therapists to follow up the patient progress.

The CDSS impacts in the clinical dashboard (AI platform). The AI platform has two main goals. First goal is to provide monitoring capabilities to show all the data related to any patient. The second goal is to provide alarms an recommendations when patients are deviating from the expected intervention plan.

Several views are designed to provide quality information related to the intervention. Figure 6.20 shows the patient's evolution in terms of errors, pressure and time. For instance, summarized performance of patient P02 shows that it has a very low average of pressure level for all fingers, and that pressure decreases when the complexity of activities increases throughout sessions. Duration indicator shows that activities involving pinkie and ring fingers present more difficulty for P02. Finally, although some errors appear on ring and index fingers, their frequency remains within expected values.

Chapter 7

Conclusions

In long-term clinical treatments, the data available at different times is different, the information may arrive in several orders, or it may never arrive. This lack of information availability represents still a challenge to implement effective Clinical Decision Support Systems (CDSS). He have analyzed the main clinical workflow and proposed a general framework for long-term clinical treatments. To mitigate the challenges around the lack of information, three new components to improve CDSS are introduced: data enhancement, confidence measures, and community detection.

The first component proposed focuses on data enhancement when missing data is meaningful. In clinical domains is quite frequent that not all information is available when a clinician has to make decisions. To mitigate this issue, the DMI algorithm was presented. DMI is capable to adapt to different scenarios with a low or high percentage of missing data. Moreover, although DMI has been tested with a specific set of imputation methods, additional imputation methods may be incorporated easily. Reported results show that DMI performs well on regression datasets, while requiring some improvements when applied to classification.

Existing confidence measures do not take available data into account to determine how reliable an outcome is. To tackle this issue, a second component based on a mutual information measure is proposed. Instead of creating a new confidence measure, we proposed a metric to correct current confidence measures. Results indicate that proposed solution is appropriate for unbalanced classification datasets, where the aim is to improve sensitivity (accuracy in the minority class).

In long-term clinical treatments, there is a large number of clinical actions performed as a consequence of other previously prescribed clinical actions and that these relationships are not explicitly well reported in electronic health records. For instance, the relationship between patient impairments and therapeutic goals. To support clinicians in providing evidence regarding not well known interactions, a methodology to find a non-explicit relationships between two data taxonomies was proposed. The proposed methodology focuses on identifying which are the best therapeutic goals for patients with multiple impairments. Due to the importance of how CDSS may present the information to clinicians, and taking into account the amount of available information, several visualizations have been proposed.

To conclude the research and illustrate its potential, two real clinical use cases have been presented. First, a CDSS for the Institut Guttmann, a hospital for patient neurorehabilitation illustrates how the different proposed solutions may support clinical decisions in a hospital environment where patients expend several months in rehabilitation, i.e. long-term treatments. But, the rehabilitation in some cases does not finish when the patient leaves the hospital. Specially, when impairments become chronic. A second use case focused on home-based therapies with chronic patients has been also presented. Although the CDSS is fully implemented for this second use case, patient recruitment has been paralyzed as an unfortunate consequence of COVID-19 pandemic. Unfortunately, without that data, the proposed CDSS and methods presented in this research may not have been tested yet for this second use case.

To sum up, the main results achieved in this research have been:

- define a general architecture of Clinical Decision Support System (CDSS) for long-term healthcare processes.
- develop an algorithm, called DMI, to improve prediction results when data has missing information.
- improve confidence measures by incorporating the uncertainty related to missing information.
- develop a methodology to find non-explicit relationships between different groups of data
- illustrate CDSS and the proposed components in two real environments: Hospital Institut Guttmann and for a Home-based Music Supported Therapy.

7.1 Publications

- **David Sanchez-Pinsach**, Josep Lluís Arcos, Sara Laxe, Montserrat Bernabeu, and Josep Maria Tormos. **Using community detection techniques to discover non-explicit relationships in neurorehabilitation treatments**. *Frontiers in Artificial Intelligence and Applications*, volume 300, pages 26–35, 2017.
- **David Sanchez-Pinsach**, Mehmet Oguz Mulayim, Jennifer Grau-Sánchez, Emma Segura, Berta Juan-Corbella, Josep Lluís Arcos, Jesús Cerquides, Monique MessaggiSartor, Esther Duarte, and Antoni Rodriguez-Fornells. **Design of an AI Platform to Support Home-Based Self-Training Music Interventions for Chronic Stroke Patients.** *Frontiers in Artificial Intelligence and Applications*, 319:170–175, 2019.
- **David Sánchez-Pinsach**, Josep Lluis Arcos. **On the importance of data enhancement in early clinical treatment stages** (*submitted*)
- Jennifer Grau-Sánchez, Emma Segura, **David Sánchez-Pinsach**, Preeti Raghavan, Thomas F. Münte, Anna Marie Palumbo, Alan Turry, Esther Duarte, Teppo Särkämö, Jesus Cerquides, Josep Lluis Arcos, Antoni Rodríguez-Fornells. **Enriched Music-Supported Therapy for Chronic Stroke Patients: A Study Protocol of Randomised Controlled Trial** (*submitted*)

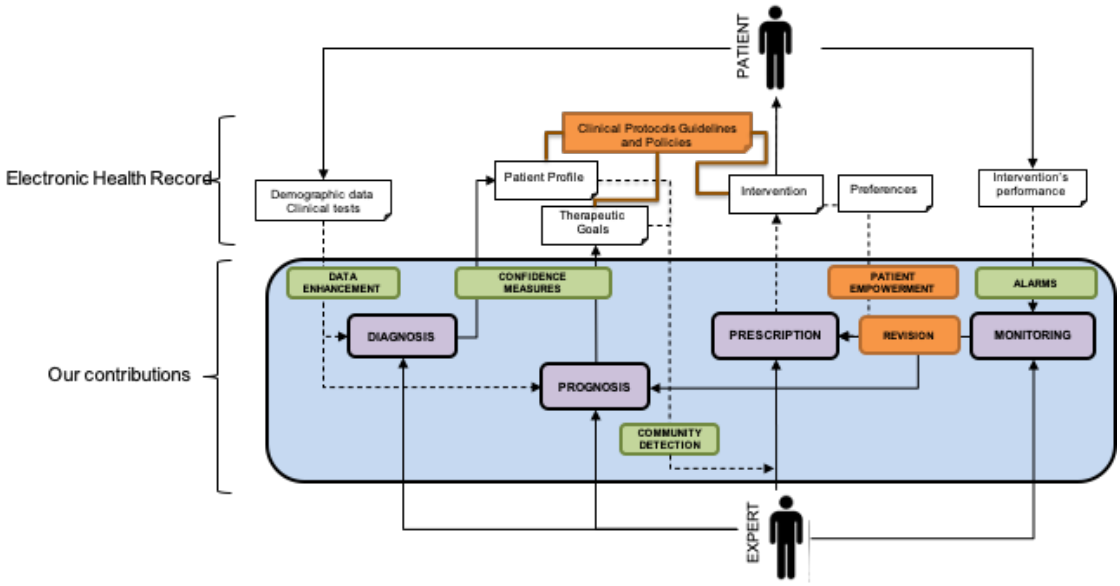

Figure 7.1: Future CDSS components.

7.2 Future Work

As a part of the research, some of the issues related to the clinical decision support systems for long-term treatments have been addressed. However, after conducting the experiments during this research, new issues and concerns have emerged. To tackle them, some future lines of work will be presented below, following the order in which they have appeared throughout the thesis.

Clinical decision support systems for long-term treatments Although the proposed CDSS framework with the new components may improve the CDSS adoption, there is room for improvement. For instance, in the CDSS alarm component, although the experiments were conducted with two-class datasets, the same idea may be applied for classification problems with multiple-class outputs. In all the cases, it is important to provide CDSS with alerts or warnings to notify the user of the existence of behavior changes (class change), especially in sensitive problems such as clinical domains.

Despite there is still work to be done on each proposed component, future research will be addressed to include three additional components (see Figure 7.1). The first component is addressed to incorporate clinical protocol guidelines and policies as part of EHR data model. This type of data is usually stored in Word documents hindering their integration into clinical workflows and tends to be used as a reference manual for clinicians. All these implicit actions will be incorporated explicitly, facilitating the interpretation of decisions taken by clinicians. The second component should address patient empowerment. That is, to offer the possibility to patients to be actively involved in their treatments. Patients' preferences are not usually incorporated into CDSS, and when their priorities become inconsistent with clinical recommendations, finding effective trade-offs is challenging because predisposition of patients is key to achieving better rehabilitation results. Finally,

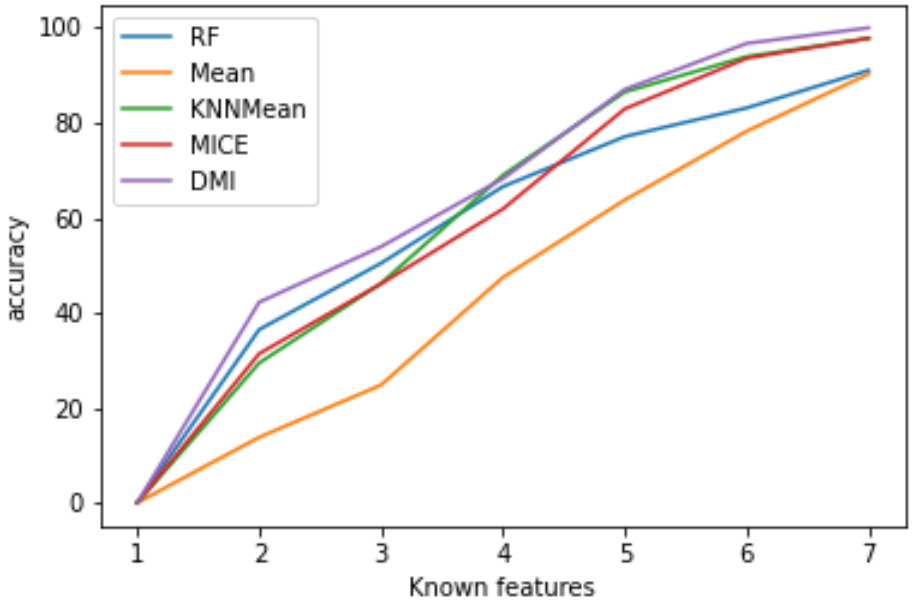

Figure 7.2: Pima diabetes dataset.

the third new component should be devoted to revise historical prescriptions. Comparing current with past prescriptions may help CDSS predict possible changes in the future. As not all patients evolve in the same way, adapting their interventions to their rhythm is essential.

Imputation. Although the results of DMI in regression problems are generally satisfactory, it is true that its performance is not enough with classification problems, especially in unbalanced datasets. Some authors [66] suggest that the better alternative is to use the Reduce-Feature strategy to deal with missing values in classification problems. Conducted experiments seem to support this hypothesis. However, in some settings other imputation methods perform better. This argument is well represented in Figure 7.2, where RF is not the best option. As expected, with only few features available, the best method is RF, but when more features become available, other alternatives enhance RF. Going deeply into our DMI proposal, DMI tries to impute always with the best imputation method from a list of possible imputation candidates. This proposal is penalized when there is no good imputation method and the best option is to discard missing features.

A future research line to explore could be to determine when imputation methods may become counterproductive for a specific missing feature. Additionally, instead of measuring the error between the real value and the imputed one, an alternative could be to measure the error only with respect to the target feature.

Confidence measures. Although the proposed confidence measure works properly on unbalanced datasets, we have identified two different scenarios that may bias the results:

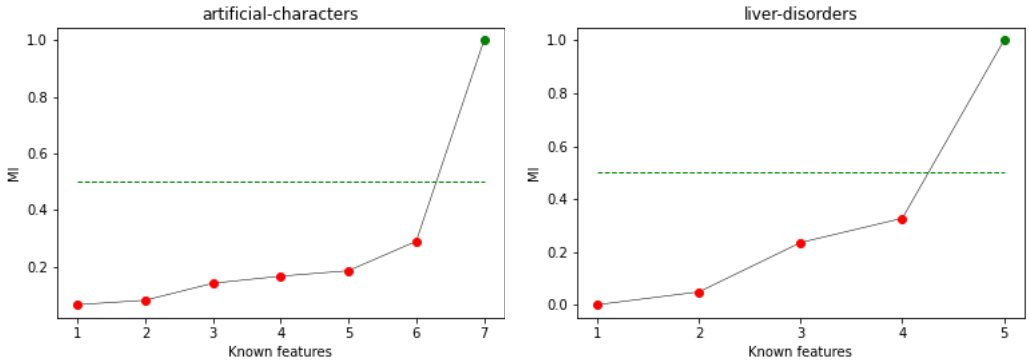

Figure 7.3: Examples of FFI scenario.

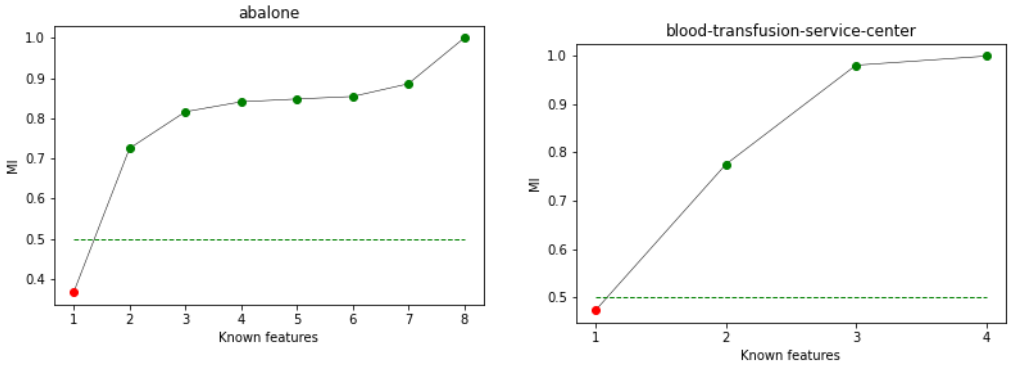

Figure 7.4: Examples of MFI scenario.

(1) when only one feature is important and the rest has no impact (FFI scenario); and (2) when most features are important and getting some of them is enough to achieve good prediction results (MFI scenario). Examples of the first scenario can be found in artificial-characters and liver-disorders datasets (see Figure 7.3). Examples of the second scenario are abalone and blood-transfusion-service-center datasets (see Figure 7.4).

To characterize datasets, we could use a threshold, based on mutual information, to determine if a feature is important or not to predict the target feature. Using this threshold, features could be divided into two groups: low impact and high impact features. Preliminary experiments suggest that the threshold should be established around 0.50. For instance, in Figure 7.3 and Figure 7.4, the threshold is represented with a green dotted line; red dots denote mutual information values below the threshold while green dots denote mutual information value above the threshold. Then, when any of these anomalous scenarios is detected, we may reconsider the strategy adopted to better estimate confidence measures. For instance, in the second scenario instead of using the sum we may apply the maximum. Table 7.1 summarizes the characteristics of datasets used in experiments. Column p measures the percentage of features above the threshold. Additionally, Shannon entropy could also help to measure how balanced a dataset is. As Table 7.1 summarizes, most datasets are unbalanced.

	p	scenario	Shannon entropy
abalone	0.87	MFI	0.75
artificial-characters	0.14	FFI	0.99
blood-transfusion	0.75		0.79
car	0.33		0.60
chess	0.67		0.84
contraceptive	0.25		0.38
ecoli	0.57		0.73
iris	0.75		1.00
letter	0.57		0.73
liver-disorders	0.20	FFI	0.77
online_intention	0.059	FFI	0.62
phoneme	0.80	MFI	0.87
pima-diabetes	0.37		0.93
yeast	0.37		0.75

Table 7.1: Mutual Information Based confidence results.

Music Supported Therapy. Undoubtedly, the most proximate future work is related to the evaluation of the CDSS developed for the Music Supported Therapy project. As mentioned previously, the whole project schedule has been delayed. Although, the proposed therapy is home-based, an initial and final physical contact with participants is required to perform pre- and post- assessments through scales and questionnaires.

Bibliography

[1] Agnar Aamodt. Knowledge acquisition and learning by experience–The role of case-specific knowledge. *Machine learning and knowledge acquisition*, 8:197–245, 1995.

[2] Plaza E Aamodt A. Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches. *AICom- Artificial Intelligence Communications*, 7(1)(IOS Press):39–59, 1994.

[3] Alan C. Acock. Working with missing values. *Journal of Marriage and Family*, 67(4):1012–1028, 2005.

[4] David W. Aha, Leonard A. Breslow, and Héctor Muñoz-Avila. Conversational Case-Based Reasoning. *Applied Intelligence*, 14(1):9–32, 2001.

[5] Fábio Alexandrini, Kerstin Maximini, Dirk Krechel, Aldo von Wangenheim, Kerstin Maximini, Aldo von Wangenheim, Dirk Krechel, Aldo von Wangenheim, Kerstin Maximini, and Aldo von Wangenheim. Integrating CBR into the Health Care Organization. *16th IEEE Symposium Computer-Based Medical Systems, 2003. Proceedings.*, pages 130–135, 2003.

[6] Gustavo E.A.P.A. Batista and Maria Carolina Monard. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17(5-6):519–533, may 2003.

[7] S Benferhat and S Benferhat. Hybrid Possibilistic Networks. *International Journal of Appoximate Reasoning*, pages 224–243, 2007.

[8] Casey C. Bennett and Kris Hauser. Artificial intelligence framework for simulating clinical decision-making: A Markov decision process approach. *Artificial Intelligence in Medicine*, 57(1):9–19, 2013.

[9] E.S. Berner. Clinical decision support systems: state of the art. *Agency for Healthcare Research and Quality*, (09):4–20, 2009.

[10] Krishnan Bhaskaran and Liam Smeeth. What is the difference between missing completely at random and missing at random? *International journal of epidemiology*, 43(4):1336–9, aug 2014.

[11] Steven Bogaerts and David Leake. Facilitating CBR for Incompletely-Described Cases: Distance Metrics for Partial Problem Descriptions. In *Advances in Case-Based Reasoning*, volume Lecture No, pages 62–76. 2010.

[12] GLENN W. BRIER. VERIFICATION OF FORECASTS EXPRESSED IN TERMS OF PROBABILITY. *Monthly Weather Review*, 78(1):1–3, jan 1950.

[13] Tiffani J. Bright, Anthony Wong, Ravi Dhurjati, Erin Bristow, Lori Bastian, Remy R Coeytaux, Gregory Samsa, Vic Hasselblad, John w. Williams, Michael D. Musty, Liz Wing, Amy S. Kendrick, Gillian D. Sanders, and David Lobach. Effect of Clinical Decision-Support Systems. *Annals of Internal Medicine*, 157(1):29–43, 2012.

[14] William Cheetham. Case-Based Reasoning with Confidence. *Proc 5th European Workshop on CBR EWCBR00*, (518):15–25, 2000.

[15] Suvra Jyoti Choudhury and Nikhil R. Pal. Imputation of missing data with neural networks for classification. *Knowledge-Based Systems*, 182, oct 2019.

[16] Taane G. Clark and Douglas G. Altman. Developing a prognostic model in the presence of missing data: An ovarian cancer case study. *Journal of Clinical Epidemiology*, 56(1):28–37, jan 2003.

[17] Ramon Lopez De Mantaras, David Mcsherry, Derek Bridge, David Leake, Barry Smyth, Susan Craw, Boi Faltings, Mary Lou Maher, Michael T. Cox, Kenneth Forbus, Mark Keane, Agnar Aamodt, and Ian Watsoni. Retrieval, reuse, revision and retention in case-based reasoning. *Knowledge Engineering Review*, 20(3):215–240, 2005.

[18] Sarah Jane Delany. Generating Estimates of Classification Confidence for a Case-Based Spam Filter. *In Proceedings of the 6th International Conference on Case-based Reasoning*, pages 177—-190, 2005.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin. *Maximum Likelihood from Incomplete Data Via the EM Algorithm*, volume 39. 1977.

[20] John K. Dixon. Pattern Recognition with Partly Missing Data. *IEEE Transactions on Systems, Man and Cybernetics*, 9(10):617–621, 1979.

[21] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[22] P Dellunde E. Armengol and C Ratto. Lazy Learning Methods for Quality of Life Assessment in people with intellectual disabilities. *Artificial Intelligence Research and Development, CCIA'11: IOS Press*, pages 41–50, 2011.

[23] R S Evans. Electronic Health Records : Then , Now , and in the Future. *Yearbook of medical informatics*, 25(Suppl. 1):1–14, may 2016.

[24] Alireza Farhangfar, Lukasz Kurgan, and Jennifer Dy. Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition*, 41(12):3692–3705, dec 2008.

[25] Alireza Farhangfar, Lukasz A. Kurgan, and Witold Pedrycz. A novel framework for imputation of missing values in databases. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, 37(5):692–709, sep 2007.

[26] Valery L Feigin, Mohammad H Forouzanfar, Rita Krishnamurthi, George A Mensah, Myles Connor, Derrick A Bennett, Andrew E Moran, Ralph L Sacco, Laurie Anderson, Thomas Truelsen, Martin O'Donnell, Narayanaswamy Venketasubramanian, Suzanne Barker-Collo, Carlene M M Lawes, Wenzhi Wang, Yukito Shinohara, Emma Witt, Majid Ezzati, Mohsen Naghavi, Christopher Murray, and Global Burden of Diseases, Injuries, and Risk Factors Study 2010 (GBD 2010) and the GBD Stroke Experts Group. Global and regional burden of stroke during 1990-2010: findings from the Global Burden of Disease Study 2010. *Lancet (London, England)*, 383(9913):245–54, jan 2014.

[27] Darren Flynn, Gary A Ford, Lynne Stobbart, Helen Rodgers, Madeleine J Murtagh, and Richard G Thomson. A review of decision support, risk communication and patient information tools for thrombolytic treatment in acute stroke: lessons for tool developers. *BMC health services research*, 13(1):225–252, 2013.

[28] G. Folino and F.S. S. Pisani. Evolving meta-ensemble of classifiers for handling incomplete and unbalanced datasets in the cyber security domain. 47:179–190, oct 2016.

[29] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

[30] A R Fugl-Meyer, L Jääskö, I Leyman, S Olsson, and S Steglind. The post-stroke hemiplegic patient. 1. a method for evaluation of physical performance. *Scandinavian journal of rehabilitation medicine*, 7(1):13–31, 1975.

[31] Simon I GBD 2016 DALYs and HALE Collaborators, Amanuel Alemu Abajobir, Kalkidan Hassen Abate, Cristiana Abbafati, Kaja M Abbas, Foad Abd-Allah, Rizwan Suliankatchi Abdulkader, Abdishakur M Abdulle, Teshome Abuka Abebo, et al. Global, regional, and national disability-adjusted life-years (DALYs) for 333 diseases and injuries and healthy life expectancy (HALE) for 195 countries and territories, 1990-2016: a systematic analysis for the Global Burden of Disease Study 2016. *Lancet (London, England)*, 390(10100):1260–1344, sep 2017.

[32] M Girvan and M E J Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826, 2002.

[33] C Gowland, P Stratford, M Ward, J Moreland, W Torresin, S Van Hullenaar, J Sanford, S Barreca, B Vanspall, and N Plews. Measuring physical impairment and disability with the Chedoke-McMaster Stroke Assessment. *Stroke*, 24(1):58–63, jan 1993.

[34] John W. Graham. Missing Data Analysis: Making It Work in the Real World. *Annual Review of Psychology*, 60(1):549–576, 2009.

[35] John W. Graham, Allison E. Olchowski, and Tamika D. Gilreath. How many imputations are really needed? Some practical clarifications of multiple imputation theory. *Prevention Science*, 8(3):206–213, sep 2007.

[36] Aric A Hagberg, Daniel A Schult, and Pieter J Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy 2008)*, number SciPy, pages 11–15, 2008.

[37] Rong Hu, Sarah Jane Delany, Brian Macnamee, Brian Mac Namee, Brian Macnamee, and Brian Mac Namee. Sampling with confidence: Using k-nn confidence measures in active learning. *Proceedings of the UKDS Workshop at 8th International Conference on Casebased Reasoning ICCBR 09*, 6176(Iccbr 09):181–192, 2009.

[38] E Hüllermeier. Case-Based Approximate Reasoning. *Springer-Verlag*, page 2007, 2007.

[39] C H K. Kawamoto. Improving clinical practise using clinical desicion support systems: a systematic review of trials to identify features critical to sucess. *BMJ*, pages 765–773, 2005.

[40] D Koller and N Friedman. Probabilistic Graphical Models. *MIT-Press*, 2009.

[41] Kamakshi Lakshminarayan, Steven A Harp, Robert Goldman, Tariq Samad, and Others. Imputation of missing data using machine learning techniques. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 140–145, 1996.

[42] Peter Langhorne, Julie Bernhardt, and Gert Kwakkel. Stroke rehabilitation. *The Lancet*, 377(9778):1693–1702, may 2011.

[43] J L Arcos; O Mulayim; D Leake. Introspective Reasoning to Improve CBR System Performance, in Metareasoning: Thinking about Thinking. *MIT Press*, pages 167–182, 2011.

[44] John Michael Linacre, Allen W. Heinemann, Benjamin D. Wright, Carl V. Granger, and Byron B. Hamilton. The structure and stability of the functional independence measure. *Archives of Physical Medicine and Rehabilitation*, 75(2):127–132, 1994.

[45] Rafael Lozano, Mohsen Naghavi, Kyle Foreman, Stephen Lim, Kenji Shibuya, Victor Aboyans, Jerry Abraham, Timothy Adair, Rakesh Aggarwal, Stephanie Y Ahn, AlMazroa, et al. Global and regional mortality from 235 causes of death for 20 age groups in 1990 and 2010: a systematic analysis for the Global Burden of Disease Study 2010. *The Lancet*, 380(9859):2095–2128, dec 2012.

[46] Julián Luengo, Salvador García, and Francisco Herrera. On the choice of the best imputation methods for missing values considering three groups of classification methods. *Knowledge and Information Systems*, 32(1):77–108, jul 2012.

[47] R C Lyle. A performance test for assessment of upper limb function in physical rehabilitation treatment and research. *International journal of rehabilitation research. Internationale Zeitschrift fur Rehabilitationsforschung. Revue internationale de recherches de readaptation*, 4(4):483–92, 1981.

[48] B M M.A. Musen. Clinical decision-support systems. *In Biomedical informatics. Stringer London*, pages 643–674, 2014.

[49] V. Mathiowetz, G. Volland, N. Kashman, and K. Weber. Adult Norms for the Box and Block Test of Manual Dexterity. *American Journal of Occupational Therapy*, 39(6):386–391, jun 1985.

[50] Patricia Matui, Jeremy C Wyatt, Hilary Pinnock, Aziz Sheikh, Susannah McLean, and J Beyene. Computer decision support systems for asthma: a systematic review. *npj Primary Care Respiratory Medicine*, 24(1):14005, nov 2014.

[51] Mark P McGlinchey and Sally Davenport. Exploring the decision-making process in the delivery of physiotherapy in a stroke unit. *Disability and rehabilitation*, 37(14):1277–1284, 2015.

[52] Lorenzo Moja, Koren H. Kwag, Theodore Lytras, Lorenzo Bertizzolo, Linn Brandt, Valentina Pecoraro, Giulio Rigon, Alberto Vaona, Francesca Ruggiero, Massimo Mangia, Alfonso Iorio, Ilkka Kunnamo, and Stefanos Bonovas. Effectiveness of computerized decision support systems linked to electronic health records: A systematic review and meta-analysis. *American Journal of Public Health*, 104(12):e12–e22, 2014.

[53] Jared S. Murray. Multiple Imputation: A Review of Practical and Theoretical Findings. jan 2018.

[54] M J Carman G I N. A. Zaidi J. Cerquides. Alleviating Naive Bayes Attribute Independence Assumption by Attribute Weighting. *Journal of Machine Learning Research*, pages 1947–1988, 2013.

[55] M. E J Newman. Analysis of weighted networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 70(5 2), 2004.

[56] Gergely Palla, Gergely Palla, Imre Derényi, Imre Derényi, Illés Farkas, Illés Farkas, Tamás Vicsek, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–8, jun 2005.

[57] Ruilin Pan, Tingsheng Yang, Jianhua Cao, Ke Lu, and Zhanchao Zhang. Missing data imputation by K nearest neighbours based on grey relational structure and mutual information. *Applied Intelligence*, 43(3):614–632, oct 2015.

[58] V M Parker, D T Wade, and R Langton Hewer. Loss of arm function after stroke: measurement, frequency, and recovery. *International rehabilitation medicine*, 8(2):69–73, 1986.

[59] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[60] Albert Pla, Beatriz López, Pablo Gay, and Carles Pous. EXiT*CBR.v2: Distributed case-based reasoning tool for medical prognosis. *Decision Support Systems*, 54(3):1499–1510, 2013.

[61] 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):180178, dec 2018.

[62] Saif S Rathore, Albert R Hinn, Lawton S Cooper, Herman A Tyroler, and Wayne D Rosamond. Characterization of incident stroke signs and symptoms: findings from the atherosclerosis risk in communities study. *Stroke*, 33(11):2718–21, nov 2002.

[63] Brian C. Ross. Mutual information between discrete and continuous data sets. *PLoS ONE*, 9(2), 2014.

[64] Donald B Rubin. Multiple Imputation after 18+ Years. *Journal of the American Statistical Association*, 91(434):473–489, 1996.

[65] Donald Bruce Rubin. Basic Ideas of Multiple Imputation for Nonresponse, 1986.

[66] Maytal Saar-Tsechansky and Foster Provost. Handling Missing Values when Applying Classification Models. *Journal of Machine Learning Research*, 8(Jul):1625–1657, 2007.

[67] Josepn L Schafer and John W Graham. Missing data: Our view of the state of the art. *Psychological Methods*, 7(2):147–177, 2002.

[68] Tapio Schneider. Analysis of incomplete climate data: Estimation of Mean Values and covariance matrices and imputation of Missing values. *Journal of Climate*, 14(5):853–871, 2001.

[69] Joan Serrà, Josep Lluis Arcos, Alejandro Garcia-rudolph, Alberto Garc, Teresa Roig Rovira, and Josep M Tormos. Cognitive Prognosis of Acquired Brain Injury Patients Using Machine Learning Techniques. *Conference on Advanced Cognitive Technologies and Applications*, pages 108–113, 2013.

[70] I Sim, P Gorman, R A Greenes, R B Haynes, B Kaplan, H Lehmann, and P C Tang. Clinical decision support systems for the practice of evidence-based medicine. *J Am Med Inform Assoc*, 8(6):527–534, 2001.

[71] Ida Sim and Amy Berlin. A framework for classifying decision support systems. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, (Figure 1):599–603, 2003.

[72] Dušan Sovilj, Emil Eirola, Yoan Miche, Kaj Mikael Björk, Rui Nian, Anton Akusok, and Amaury Lendasse. Extreme learning machine for missing data using multiple imputations. *Neurocomputing*, 174:220–231, jan 2016.

[73] G Stucki, J Bickenbach, C Gutenbrunner, and J Melvin. Rehabilitation: The health strategy of the 21st century. *Journal of Rehabilitation Medicine*, page 0, jan 2017.

[74] Cao Truong Tran, Mengjie Zhang, Peter Andreae, Bing Xue, and Lam Thu Bui. An effective and efficient approach to classification with incomplete data. *Knowledge-Based Systems*, 154:1–16, aug 2018.

[75] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[76] Chih Fong Tsai, Miao Ling Li, and Wei Chao Lin. A class center based approach for missing value imputation. *Knowledge-Based Systems*, 151:124–135, jul 2018.

[77] S. Van Buuren, H. C. Boshuizen, and D. L. Knook. Multiple imputation of missing blood pressure covariates in survival analysis. *Statistics in Medicine*, 18(6):681–694, 1999.

[78] Stef van Buuren. Multiple imputation of discrete and continuous data by fully conditional specification. *Statistical Methods in Medical Research*, 16(3):219–242, 2007.

[79] Stef van Buuren, Karin Groothuis-Oudshoorn, Stef van Buuren, and Karin Oudshoorn. MICE: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3):1–67, dec 2011.

[80] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: networked science in machine learning. 2014.

[81] Madan Lal Yadav and Basav Roychoudhury. Handling missing values: A study of popular imputation packages in R. *Knowledge-Based Systems*, 160:104–118, nov 2018.

[82] Bing Zhu, Changzheng He, and Panos Liatsis. A robust missing value imputation method for noisy data. *Applied Intelligence*, 36(1):61–74, jan 2012.