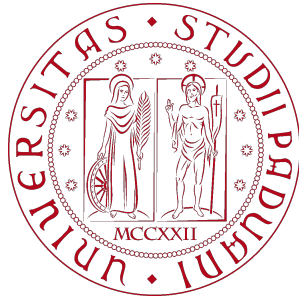


UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI INGEGNERIA

Corso di Laurea Magistrale in Ingegneria Informatica



Tesi di Laurea Magistrale in Ingegneria Informatica

**Development of a model-based algorithm for the
assessment of the Obsessive-Compulsive Disorder**

Laureando: Ivan Donadello

Relatore: Prof. Silvana Badaloni

Correlatore: Andrea Spoto PhD

Controrelatore: Prof. Carlo Ferrari

15 Ottobre 2012
Anno Accademico 2011-2012

Ai miei genitori

Contents

1	Introduction	3
2	The Clinical Assessment and Computerized Diagnostic systems	7
2.1	The clinical assessment	7
2.1.1	Semi-structured interviews and Cognitive Behavioural Assessment 2.0 battery	8
2.2	Adaptive assessment systems	9
2.3	State of the art on computerized systems for clinical assessment	10
2.3.1	Expert systems	10
2.3.2	Computerized adaptive testing	11
2.3.3	Other technologies	12
3	Formal Concept Analysis and Knowledge Space Theory	15
3.1	Formalizing the concepts: the Formal Concept Analysis	15
3.1.1	Basic notions of lattice theory	16
3.1.2	Fundamental concepts of FCA	17
3.2	Formalizing the knowledge: Knowledge Space Theory	21
3.2.1	Introduction and fundamental concepts	21
3.2.2	The skill map	24
3.2.3	The disjunctive and conjunctive model	25
3.2.4	The competency model	28
3.2.5	Probabilistic knowledge structures	29
3.3	Unify the concepts with the knowledge: the joint between FCA and KST	31
4	The Formal Psychological Assessment	35
4.1	The core idea behind FPA	35
4.2	The translation from knowledge context to clinical context	36
4.3	Formalizing the Obsessive-Compulsive Disorder	37
4.4	Testing the structure through real data	38

5	The adaptive assessment algorithm	41
5.1	Sketching a stochastic procedure	41
5.2	Basic concepts and their formalization	44
5.3	The algorithm in details	47
5.3.1	The questioning rule	47
5.3.2	The updating rule	48
5.3.3	The proof of correctness	50
5.3.4	The stop condition	50
5.3.5	Refining the assessment	50
6	Developing AAS-PD	53
6.1	The Python programming language	53
6.2	ASS-PD architecture	54
6.3	Considerations on the complexity algorithm	55
6.4	The implemented modules	59
6.4.1	The <code>AlgorithmFunctions</code> module	59
6.4.2	The <code>ImportExport</code> module	61
6.4.3	The <code>InitData</code> module	62
6.4.4	The <code>main</code> module	62
6.5	An example of assessment	62
6.6	Toward a standalone system	64
7	Testing AAS-PD	67
7.1	Method	67
7.2	Results	69
7.2.1	Condition 1, results about the assigned states	69
7.2.2	Condition 2, results about the response patterns that are states	71
7.2.3	Condition 3, results about the distances	72
7.2.4	Condition 4, results about response patterns with non minimum distances	73
7.2.5	Condition 5, results about the means	75
7.2.6	Condition 6, results about the number of questions posed	75
7.3	Discussions and future work	76
	Bibliography	79

Abstract

The aim of this thesis is the development of a new software system for the adaptive psychological assessment, called AAS-PD (Adaptive Assessment System for Psychological Disorders). AAS-PD will allow the clinician to carry out the correct diagnosis of patients affected by Obsessive-Compulsive Disorder (OCD). The goal is to introduce in the field of clinical psychology an adaptive diagnostic software based on a new formal model, the *Formal Psychological Assessment* (FPA). This work is framed in the context of mathematical psychology, the branch of psychology that deals to develop ideas and psychological problems with the mathematics formalism.

The developed system requires a mathematical structure, such as a knowledge structure (in our field clinical structure), and performs the assessment by making probabilistic inferences on that structure, using as a stop criterion the measure of the entropy of the structure. Finally, the assessment is refined considering the false negatives and false positives probabilities.

The original contribution of this thesis is the adaptation of the adaptive assessment algorithm designed by Doignon and Falmagne for the assessment of knowledge, in the clinical psychology context, following the formal representation of the OCD given by FPA. The software is inserted, in terms of psychological diagnosis, in the *Computerized Adaptive Testing* (CAT) context. Nevertheless, it presents some specific innovations, mostly referred to the use of FPA formal model in the psychological diagnosis context.

In addition, such application of FPA has a good degree of independence from the reference domain content and then, along with the implemented algorithm, we get an assessment system applicable in many different domains with object-attribute data type.

Results show that AAS-PD assigns response patterns to clinical states properly. Furthermore, such results point out the need of some improvements in the formal model. Future developments involve the implementation of a real software able to support the clinician in the assessment of the main psychological disorders.

Sommario

L'obiettivo del presente lavoro di tesi è la realizzazione di un nuovo sistema software di assessment psicologico adattivo, chiamato AAS-PD (Sistema di Assessment Adattivo per i disturbi psicologici). AAS-PD permetterà allo psicologo clinico di eseguire la diagnosi corretta di pazienti affetti da Disturbi Ossessivo-Compulsivi (DOC). Lo scopo è di introdurre nel campo della psicologia clinica un software di diagnosi adattiva basato su un nuovo modello formale, il *Formal Psychological Assessment* (FPA). Tale lavoro è inquadrabile nel contesto della psicologia matematica, quel ramo della psicologia che si occupa di elaborare idee e problemi psicologici con il formalismo della matematica.

Il sistema sviluppato prende una struttura matematica, quale una struttura di conoscenza (nel nostro ambito struttura clinica), ed esegue l'assessment facendo inferenze probabilistiche su tale struttura, usando come criterio di stop la misura dell'entropia della struttura. In fine l'assessment viene rifinito considerando le probabilità di falsi negativi e falsi positivi.

Il contributo originale della presente tesi è stato l'adattamento dell'algoritmo di assessment adattivo pensato da Doignon e Falmagne per la valutazione delle conoscenze, al contesto della psicologia clinica, seguendo una rappresentazione formale del DOC data dal FPA. Il software si inserisce, in chiave di diagnosi psicologica, nell'ambito dei test adattivi computerizzati (*Computerized Adaptive Testing*, CAT), ma a differenza loro utilizza il modello formale FPA, portando quindi un'innovazione nell'ambito della diagnosi psicologica.

Inoltre tale applicazione del FPA presenta un buon grado di indipendenza dal contenuto del dominio di riferimento e quindi, congiuntamente all'algoritmo implementato, si ottiene un sistema di valutazione applicabile a qualsiasi dominio con dati del tipo oggetto-attributo.

I risultati mostrano che AAS-PD assegna correttamente pattern di risposta a stati clinici, evidenziando inoltre alcuni miglioramenti del modello formale da fare. Sviluppi futuri comportano lo sviluppo di un vero e proprio software capace di supportare il clinico nell'assessment dei principali disturbi psicologici.

Chapter 1

Introduction

A psychological disorder is a pattern of behavioural or psychological symptoms that involves several life areas and/or creates distress for the person experiencing these symptoms. The set of this kind of disorders is very broad, and includes, for instance, obsessive-compulsive disorder, mood disorders, anorexia nervosa, bulimia nervosa, anxiety disorders, psychotic disorders, multiple personality disorder, antisocial personality, post-traumatic stress disorder, etc. A psychological disease can often damage the individual, his family and also the collectivity. Indeed, psychological disorders are identified among the main causes to worker removal from workplace. Moreover, they can influence the individual immune system, causing frequent diseases, such as sleep and behaviour disorders, infections and inflammations [1, 2]. In the most serious cases, some types of psychological disorders can lead to a degeneration of the nervous system of the individual. Another drastic consequences for a person who suffers from psychological disorders is the image of incapable made by the society in which he/she lives. Starting from this preamble, it is not difficult to conclude that mental health problems require time consuming investigations before a diagnosis is reached [3], thus an automated support tool can give an early and effective diagnosis that could benefit the individual's health and the collectivity's social costs.

This kind of problems is tackled by clinical psychology, the branch of psychology that integrating theory and practice, understands, predicts and alleviates maladjustment, disability, and discomfort as well as promoting human adaptation and personal development [4].

Diagnosis can be defined as the description of a health problem in terms of known diagnostic criteria, and it is the outcome of the diagnostic assessment defined as a set of actions needed to obtain the diagnosis [3, 5]. Diagnostic criteria, or symptoms, are features that distinguish a certain disorder. Diagnosis can be expressed in a variety of manners, for example, by means of medical indexes or verbal modalities.

Clinical psychologists often refer to the most widely accepted diagnostic manual when formulating a diagnosis, that is the Diagnostic and Statistical Manual of Mental Disorders (DSM), published by the American Psychiatric Association now in the fourth edition (DSM-IV-TR) [1]. It provides a common language and standard criteria for the classification of mental disorders. The DSM-IV-TR organizes each psychiatric diagnosis into five dimensions (axes) relating to different aspects of the disorder or disability. For example Obsessive-Compulsive Disorders (OCD) is described in the first axis, and for each of them it includes typical patterns of behaviour, thinking and emotion.

Another used diagnostic manual is the 10th revision of the International Statistical Classification of Diseases and Related Health Problems (ICD-10), a medical classification listed by the World Health Organization (WHO). It codes for diseases, signs and symptoms, abnormal findings, complaints, social circumstances, and external causes of injury or diseases [6].

Psychological assessment is not based on a simple questionnaire (interview or test) whose responses return immediately the diagnosis, but it is rather a decision making process, where the clinician guides the procedure discarding hypotheses on the basis of the patient's responses. This task can be affected by problems, for example, the assumptions inferred by the clinician can be erroneous, and often the psychological assessment takes up to four hours to identify a diagnosis [7]. The aim of our work is the construction of an adaptive assessment system in order to provide the clinician with a certainly correct inferences procedure based on logical implications able to improve the diagnostic accuracy. Furthermore, the procedure happens to be time saving in terms of questions formulated. We called the developed system AAS-PD (acronym for Adaptive Assessment Algorithm for psychological disorders), and it will help the psychologist in the assessment of OCD.

Traditional psychological assessment systems return only a simple numeric score which the clinician uses in identifying a "clinical label". Nevertheless, two response patterns items may score the same. Thus, this hidden information may lead the psychologist to give the same diagnosis to patients with different sets of symptoms but with the same score. Our proposed solution allows to avoid this problem since it does not return a number but a relation between a set of items of a questionnaire and a set of symptoms of the subject.

The developed system is the application in the clinical context of an algorithm designed for the assessment of students' knowledge. Indeed, the field of clinical psychology is very poor with respect to computerized adaptive assessment systems, and they are usually based on classical *Item Response Theory* (IRT) [8].

We started considering an algorithm designed for *Knowledge Space The-*

ory¹ (KST) [9] and whose goal was the knowledge assessment. The algorithm takes as input a particular mathematical structure, given by KST, called knowledge structure. The succeeding step was to consider a formal representation of the clinical assessment called *Formal Psychological Assessment*² (FPA) [7, 10, 11] that returns a knowledge structure (here renamed clinical structure) starting from a simple clinical questionnaire. The last step was to development the software system applying the algorithm for KST to the clinical structure of FPA.

The output of AAS-PD includes a set of diagnostic criteria of the subject that the psychologist will use, together with the informations coming from the behavioural observations [7], in order to formulate the diagnosis. Other informations are returned by FPA, namely, indexes assigned to every item of the questionnaire, representing the false negatives and the false positives. These indexes are useful in estimating the goodness of the formal representation of FPA and in refining the results of the algorithm.

The main advantage of AAS-PD is the possibility to perform logically correct inferences on the basis of the whole information engaged in the assessment. This task is carried out by an algorithm whose correctness is formally demonstrated. Thus, such psychological evaluation results more accurate and faster.

Another advantage is that our system provides the clinician with a much more complete and detailed information than the mere numeric score of a test, i.e. the diagnostic criteria on which the psychologist can formulate a diagnosis, using a manual or his knowledge. In this manner, differences among patients, otherwise hidden by the score, come out.

Moreover, this software represents an innovation in the field of psychological assessment. Indeed, is the only computerized adaptive system based on the formal representation of the assessment provided by FPA.

The system developed in this thesis is only the first step to a more ambitious project, the new version of the diagnostic battery CBA 2.0 (Cognitive Behavioural Assessment) [12, 13, 14], a wide spectrum tool for the assessment of the main psychological disorders such as depression, anxiety, phobias, obsessive-compulsive disorders, psycho-physiological disorders. CBA 2.0 represents the reference point for cognitive behavioural assessment in Italy [7]. The philosophy behind CBA 2.0 is falsificationist³, i.e. CBA 2.0 goes on by hypotheses falsification, eliminating the assumptions falsified by

¹KST is a mathematical theory developed with the aim of constructing an efficient system for the assessment of knowledge.

²FPA was developed in Padua University by Spoto et al. since 2006.

³Falsificazionism is an epistemological orientation which refers to the ideas of Austrian philosopher K. R. Popper, and affirms that scientific progress would proceed by of those theories that are falsified by results [15, 16].

the response pattern. In this manner the tool detects the areas worthy to be further examined. According in the spirit of the CBA 2.0 we do not want to provide the psychologists with an assessment machine replacing them, but rather supporting them in hypotheses formulation during the process of assessment.

The thesis is organized as follows, in the second chapter we define the problem of clinical assessment and how it is dealt with semi-structured interviews and psychological testing like the battery CBA 2.0, moreover we will see the concept of adaptive assessment and a brief state of the art of computerized assessment systems. The third chapter exposes the fundamental notions of KST and another mathematical theory, the *Formal Concept Analysis* (FCA) [17], and their union preliminary for understanding FPA, moreover we give our new version demonstration of some results. FPA is exposed in the fourth chapter. Specifically we will see how to obtain a knowledge structure (i.e. a clinical structure) from a formal context, that is considering the items of a questionnaire as objects and the diagnostic criteria investigated by each single item as attributes. In the two succeeding chapters we present in detail the adaptive assessment algorithm implemented and in the last chapter we report the obtained results together with conclusions and new proposal for future work.

Chapter 2

The Clinical Assessment and Computerized Diagnostic systems

Generally speaking the assessment is a process where an agent (man or machine) collects a certain amount of data and informations about an object (or person) in such a manner it could give a measure of it (him) about a certain quality of a domain of example. For example if the domain is wine deliciousness the assessment consist of a wine waiter tasting the wine, collecting information about its color, taste and smell and then giving his judgement. Assessment can be applied to the most part of human activities, also to psychology. Psychological assessment can be defined as as the continuous and active process carried out by a clinician in order to evaluate an individual [18, 19]. This kind of evaluation is very important nowadays because it explores several areas like attitudes, clinical disorders, personality traits etc, in order, for example, to investigate about forensic test, job selection, diagnosis formulation [7]. In this chapter we will focus on a kind of psychological assessment named clinical assessment, which its aim is to collect useful informations about a patient at the beginning of the therapeutic work in order to formulate a diagnosis and propose a therapeutic work. Moreover, the chapter exposes some techniques of clinical assessment as semi-structured interviews and CBA 2.0 battery that represents the basis for our assessment algorithm.

2.1 The clinical assessment

The first action a clinician has to carry out when working with a patient is collecting diagnostic elements necessary in programming his further intervention [7], that is the clinical assessment. The quality of the initial assessment represents one of the main predictors of a good therapeutic outcome, and an erroneous assessment could lead to ruinous therapeutic interventions and

patient disappointments [10].

Clinical assessment is composed by three main levels of information: the subjective level, the behavioural level, and the physiological level. Our focus is the subjective level that collects the informations through the verbal channel, that is clinical interviews, questionnaires, diaries and tests. The integration of these three sources is called *horizontal integration* while the *vertical integration* is the process of collecting informations through logical inferences of the clinician.

As introduced in chapter 1, psychological assessment is not a passive gathering of informations, but an active process similar to a problem solving and decision making process, thus the clinician has to move correctly in a framework made of logic reasoning, hypothetical and deductive logics making the right inferences. In other words, the clinician is asked to formulate hypotheses and then try to check if there are correspondences in the patient, so wrong inferences could mislead the clinician with the frustrating consequence of wasting time. It is not a simple task, because the clinician has to make right deductions including a lot of critical informations. In literature exist different diagnostic tools whose aim is helping the specialist to perform the assessment in a logically correct manner, we will see two of them.

2.1.1 Semi-structured interviews and Cognitive Behavioural Assessment 2.0 battery

Semi-structured interviews are questionnaires formulated like traditional interviews, that permit the clinician to assign a specific score to the set of the patient's answered questions. In fact, these tools present a set of items, but the clinician does not necessarily asks all of them, instead he explores a path of questions selecting the following one given the answer to the previous question. These instruments introduce the concept of adaptivity of the assessment (we will see it better in section 2.2) that is the main feature of our software. These interviews investigate the main disorders included in the DSM-IV-TR [1], such as depression, anxiety disorder, addictive disorders, etc, and has its international reference point in Structured Clinical Interview for DSM-IV-TR (SCID) [20].

The Cognitive Behavioural Assessment 2.0 (CBA 2.0) battery [13, 14] is a wide spectrum tool for the assessment of the main psychological disorders and it was developed at the beginning of the eighties with the aim of supply the clinician a more adequate assessment tool. It is included in the category of psychological testings and contains questionnaires the clinician can use on the basis of the phase of the assessment. For example at the beginning of the assessment wide spectrum tools are needed, then in a second phase when several diagnosis have been excluded, the clinician could be interested in going into the details of some specific disorders by using specifically focused tools. The main result of a questionnaire is a score (a number) that could

have been obtained through several response patterns. In the Italian context CBA 2.0 represents the most popular wide spectrum assessment battery [7].

The battery is organized in two scales, the *primary scales* contains ten sheets of questionnaires providing the clinician with a wide spectrum picture of the patient, and returning a score. The score identifies an area of possible disorders that will be investigated by specific tools included in the *secondary scales*. We will present only the ninth sheet because this questionnaire is the instrument used to achieve the formalization of FPA. This sheet, regarding the OCD, is the reduced form the Maudsley Obsessional-Compulsive Questionnaire (MOCQ-R) [21, 22], a 21 dichotomous items questionnaire investigating the three main specifications of the OCD through three different sub-scales, i.e. Checking, Cleaning and Doubting-Ruminating. These sub-scales are also important for the work in this thesis, indeed the first two were used for testing our adaptive algorithm. Moreover, the battery has a control system consisting of some indexes for evaluating the reliability of the answers, showing a possible low level of collaboration of the patient or a desire of giving a positive image of himself. Once the patient has responded to the items of the ten sheets, the response patterns are passed to a computer programme that calculates the score and then checks on a boolean matrix which aspects are better to investigate. On the basis of these indications the clinician can use the instruments of the secondary scales, conducting the vertical integration of the clinical assessment. For instance, if the patient under analysis presents a critical score at sheet number nine this datum should be further investigated with an instrument specialized in obsessions and compulsions.

Finally, CBA 2.0 does not provide only a mere and numeric score about a patient, like other computerized tools, but it returns a verbally descriptive report of the score; this characteristic was totally new in Italy [7].

Our developed system will work as a computerized semi-structured interview based on the items of sheet 9 of the CBA 2.0, our future aim is to enlarge this set of questions to all of the other sheets.

2.2 Adaptive assessment systems

For adaptive assessment system we mean an instrument like a computer software, or a more general test, where the question (or item) posed by an examiner (a teacher or also a clinician) at a given instant depends on the answers given by an examinee (or a patient) to previous questions, i.e. it is a function of them. So the items change to reflect the performance on the preceding questions, and the test constantly establishes the current knowledge, or clinical level, of the subject examined and tailors to it. The testing starts at a moderate level and, as the user answers questions, the procedure selects the more informative item to pose next, it can be more

difficult or easier, depending on the correctness of the responses. The loop stops when no more questions are needed to determine the subject's final level (i.e. there is enough information to terminate the test), and the score is not derived only from the number of correct answers but, rather, from the level of difficulty of the questions answered correctly. This procedure is the same performed by a clinician during an interview [7].

2.3 State of the art on computerized systems for clinical assessment

Clinical assessment is a very poor field regarding computerized systems, due to difficulties in formalizing such field and interpreting the results. The first computer programmes in the eighties performed only simple scoring of the questionnaires, but with the improvement of the technology there was a growing interest in this branch of psychology, with the realization of electronic questionnaires able to reproduce the standard pen-and-paper version and also to interpret them [23]. Nevertheless this interest, only few technologies form the state of the art of computerized diagnostic tools, but new and creative approaches are emerging, for example the assessment programme developed in this thesis is the only adaptive assessment instrument based on KST. The most important and implemented technology in diagnostic assessment are expert systems, but their are not adaptive in the sense of the definition given above, if we want to see the topic of adaptive assessment we have to move in the knowledge domain (and this is what we have done) where the most of computerized assessment systems are based on *Item Response Theory* (IRT) or *Classical Test Theory* (CTT). Computerized adaptive testings (CATs) are used in evaluation of a subject's knowledge, the same starting point of Doignon and Falmagne but with a different theoretic basis.

2.3.1 Expert systems

In artificial intelligence, an expert system is a software able to simulate professional practices of a human expert in a given domain. For example in the domain of blood infections the expert is the physician, and a possible professional practice would be diagnosing patients on the basis of reported symptoms and medical test results, this is what the expert system MYCIN [24] really do, and moreover, it recommends antibiotics with the dosage adjusted for patient's body weight. A more formal definition is provided by Jackson [25]: "an expert system is a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advices".

In our work the domain is clinical psychology, the human expert is the clinical psychologist and the professional practices are considering symptoms

to diagnose an underlying mental disorder. In this field there is a lot of approaches, Spiegel and Nenth in 2004 adopted the most direct and simple strategy, considering the relation symptoms-mental disorders [26], i.e. the psychologist enter the symptoms in the system and then it calculates possible symptom combinations and returns a feedback based on the classical if-then rules. The outcome is not deterministic, but fuzzy: it indicates all possible diagnoses and estimates the risk for each possible diagnosis individually. Following the classical approach based on if-then rules, DECES is an interactive self-help on-line expert system developed in 2007, that diagnoses patients depressive conditions and provide advices to lower their levels of depressions [27], while ESDAP is concerned with diagnosis in art psychology [28]. ESDAP provides a web-based architecture so non-experts such as parents and teachers can take under control their children's psychological problems, from an early stage, simply by posting their children's drawings.

But also hybrid approaches are exploited with interesting results, for example Nunes et al. in 2009 combined the technology of the classical expert systems with MACBETH (Measuring Attractiveness by a Categorical Based Evaluation Technique), a system for multi-criteria decision analysis [29], with the aim of diagnosing the OCD [2]. Another hybrid system is PsyDis [3], not properly an expert system but a decision support system that joins semantic technologies (specifically ontologies) with the classic logic inference mechanisms, aiming at giving decision support in psychological diseases diagnosis.

For a good state of the art about expert systems in practical psychology we point put the work of Kan et al. of 2010 where they depict also a framework to create expert systems in this field [30].

2.3.2 Computerized adaptive testing

Computerized adaptive testing (CAT) is a kind of computer-based test that adapts to the examinee's ability level. For this peculiarity they are also called tailored testing.

Normally a CAT works like this: the software chooses and displays questions, it records the examinee's answers, it updates the subject's latent trait level and then it selects a new item in function of the examinee's answers to previously administered questions, and of the specific statistical qualities of the items pool. It repeats this loop until a stopping rule has been satisfied. That is, if an examinee's responds correctly on an item of intermediate difficulty, the next question will be more difficult. On the other hand, if he responds incorrectly, next item will be simpler. The software repeats this loop until no new item identifies better the ability level. This procedure results in higher levels of test score precision and shorter test-lengths [31].

CAT systems are generally used for knowledge assessment but their behaviour is very general, no matter what domain they are applied, so the application to clinical psychology is direct, for this reason we consider them

in this thesis and shortly we will see some application. Studies have shown that CATs can save time and alleviate the burdens on both examinees and test administrators, as compared to traditional computer-based or pen-and-paper assessments [32]. For example Waller and Reise (1989) simulated a CAT version of the Absorption scale of the Multidimensional Personality Questionnaire and the item savings ranged from 50% to 75% depending on the stop criterion utilized [23].

Modern CAT algorithms are based on IRT [8], maximum likelihood and Bayesian methods [31, 33]. Even if the bulk of the research on adaptive testing uses IRT as the underlying model [34].

IRT [8], a psychometry theory, assigns an Item Characteristic Curve (ICC) to every item i in order to express the probability ($P_i(\theta)$) of a correct response for item i as a function of subject's latent trait level (or level ability) θ . This function equips item i with the ability of discriminate the trait level and, thanks to this feature, IRT provides methods for item selection (i.e. the item that maximizes the information given the currently estimated trait level) and subject's trait level estimation [35]. An example of IRT-based CAT is SIETTE [35], a web-based system to assist teachers and instructors in the assessment process in educational settings. Another example is the CAT proposed by Eggen and Straetmans of Cito [36], a combination of IRT with statistical computation procedures like sequential probability ratio test and weighted maximum likelihood, with the purpose of classifying examinees into three categories. Following IRT but moving in the field of psychological assessment, Chien et al. in 2011 developed a web-based CAT for efficiently collecting data regarding workers' perceptions of job satisfaction in hospital workplace [32]. Their system is based on Rasch model a simplification of the ICC. While Simms et al. in 2011 started a five steps project, the CAT-PD project, aiming to realize a computerize adaptive assessment of personality disorders IRT-based [23].

The other important (and alternative) model for adaptive testing is the Bayesian approach, for example EDUFORM is an adaptive questionnaire based on Bayesian statistical techniques [37], and Rudner in 2002 proposed a simple Bayesian decision theory [34] instead of the classic IRT, and his model is the basis of PARES, an adaptive system for the assessment of students' knowledge [38].

Lot of research work has been done over the past 40 years in CATs, but there is little literature providing a practical framework on the development of a CAT [39], a good guide performing this task can be found in the work of Thompson and Weiss [39].

2.3.3 Other technologies

Finally we present some alternative methodologies for achieving the psychological assessment, we start with the promising theme of virtual reality.

Nowadays virtual environments (VEs) offer a new human-computer interaction paradigm in which users actively act in a computer generated three dimensional virtual world, so they are no longer simply external observers of images on a computer screen. A user in a VE faces a variety of scenarios with a lot of controlled stimuli, his reactions can be measured and monitored in such a way the system could be tailored to the needs of the patient or to the therapeutic application [40, 41], all without leaving the therapist's office [41]. Taking into account this considerations, Riva, in 1999, thought and developed an adaptive assessment system virtual reality based [41]. Another approach is given by artificial neural networks, Suhasini et al. in 2010 proposed and developed a multi decision support system based on neural networks to diagnose psychiatric problems like depression and anxiety [42].

Chapter 3

Formal Concept Analysis and Knowledge Space Theory

In this chapter we present the *Formal Concept Analysis* (FCA), a mathematical theory that formalizes the classical, and philosophic, notion of “concept”, born for data analysis tasks, and the *Knowledge Space Theory* (KST), other mathematical theory that formalizes the notion of subject’s knowledge, born with the goal of constructing an adaptive knowledge assessment system. We will see only few (and basic) concepts about them, and their potential overlaps, indeed, the last theorem of the chapter allows us to move from one theory to the other, and this was crucial for the development of FPA. The joint between FCA and KST was proposed by Rush and Wille [43], by Doignon and Falmagne [9] and also by Spoto et al. [10], we followed the latter approach giving our proof of their results.

3.1 Formalizing the concepts: the Formal Concept Analysis

FCA [44] is a mathematical theory born around 1980, when a research group in Darmstadt, Germany, begun to systematically develop a framework for lattice theory applications in order to performing data analysis. It formalizes the classic philosophic notions of concept: a concept can be seen as a unit of thought consisting of two parts: the extension and the intension. The extension covers all objects (or entities) belonging to the concept, while the intension comprises all attributes (or properties) valid for all those objects. So the core of this theory is the organization of the data under analysis into units, which are formal abstractions of the notion of concept, such units are organized in a particular mathematical structure, a complete lattice.

A distinguishing feature of FCA is an inherent integration of three com-

ponents of conceptual processing of data and knowledge, namely, the discovery and reasoning with concepts in data, discovery and reasoning with dependencies in data, and visualization of data, concepts, and dependencies. Integration of these components makes FCA a powerful tool which has been applied to various problems. Examples include hierarchical organization of web search results into concepts based on common topics, gene expression, information retrieval, analysis and understanding of software code, debugging, data mining, design in software engineering, internet applications including analysis and organization of documents and e-mail collections, annotated taxonomies, and further various data analysis projects described in the literature [45].

The father of FCA is Rudolph Wille, and the reference book is [17].

3.1.1 Basic notions of lattice theory

FCA, as KST, is based on mathematical order theory, in particular on the theory of complete lattice. In this section we present only the most important notions of this theory in order to understand FCA, and also KST. For a complete study of these topics we point out the work of Birkhoff [46].

Definition 3.1 (Quasi (partial) orders, quasi (partial) ordered sets). *A quasi order on a set X is any relation \mathcal{P} which is transitive and reflexive on X . A quasi ordered set (X, \mathcal{P}) is a set equipped with a quasi order. A quasi order \mathcal{P} with the antisymmetry property is a partial order. A partially ordered set (poset) (X, \mathcal{P}) is a set equipped with a partial order.*

Every finite poset (X, \mathcal{P}) can be represented graphically by a *Hasse diagram*, or *covering relation*. The elements of X are depicted by small circles, and a line between two circles represents the covering relation: the element $x \in X$ is *covered* by element $y \in X$ when $x \mathcal{P} y$, $x \neq y$ and moreover $x \mathcal{P} t \mathcal{P} y$ implies $x = t$ or $t = y$.

Definition 3.2 (Lower bound, upper bound, infimum, supremum). *Let be (X, \mathcal{P}) a poset and A a subset of X . A lower bound of A is an element s of X with $s \leq a$ for all $a \in A$. An upper bound of A is defined dually. If there is a largest element in the set of all lower bounds of A , it is called the infimum of A and is denoted by $\inf A$ or $\bigwedge A$; dually, a least upper bound is called supremum and denoted by $\sup A$ or $\bigvee A$. If $A = \{x, y\}$, we also write $x \wedge y$ for $\inf A$ and $x \vee y$ for $\sup A$.*

Now the most important definition for the comprehension of FCA.

Definition 3.3 (Lattice, complete lattice). *A poset $\mathbf{X} = (X, \mathcal{P})$ is a lattice, if for any two elements x and y in X the supremum $x \vee y$ and the infimum $x \wedge y$ always exist. \mathbf{X} is a complete lattice, if the supremum $\bigvee A$ and the infimum $\bigwedge A$ exist for any subset A of X .*

Every complete lattice \mathbf{X} has a largest element, $\bigvee X$, the unit element of the lattice, denoted by $\mathbf{1}_X$. Dually, the smallest element $\mathbf{0}_X$ is called the zero element.

Definition 3.4 (Closure operator). *Let be (X, \mathcal{P}) a quasi ordered set, and h a mapping of X into itself. We say that h is a closure operator on (X, \mathcal{P}) if it satisfies, for each $x, y \in X$:*

1. $x \mathcal{P} y$ implies $h(x) \mathcal{P} h(y)$;
2. $x \mathcal{P} h(x)$;
3. $h(h(x)) = h(x)$.

Moreover, any $x \in X$ is closed if $h(x) = x$.

Definition 3.5 (Galois connection). *Let (Y, \mathcal{U}) and (Z, \mathcal{V}) be two quasi ordered sets and let be $f : Y \rightarrow Z$ and $g : Z \rightarrow Y$ two mappings. The pair (f, g) is Galois connection between (Y, \mathcal{U}) and (Z, \mathcal{V}) if the following six condition hold: for all $y, y' \in Y$ and all $z, z' \in Z$:*

1. $y \mathcal{U} y'$ and $y' \mathcal{U} y$ imply $f(y) = f(y')$;
2. $z \mathcal{V} z'$ and $z' \mathcal{V} z$ imply $g(z) = g(z')$;
3. $y \mathcal{U} y'$ implies $f(y) \mathcal{V}^{-1} f(y')$;
4. $z \mathcal{V} z'$ implies $g(z) \mathcal{U}^{-1} g(z')$;
5. $y \mathcal{U} (g \circ f)(y)$;
6. $z \mathcal{V} (f \circ g)(z)$.

3.1.2 Fundamental concepts of FCA

Definition 3.6 (Formal context). *A formal context is a triple (G, M, I) where G and M are sets, while I is a binary relation between G and M , namely $I \subseteq G \times M$. The elements of G and M are called objects (in German Gegenstände) and attributes (in German Merkmale) respectively, and gIm with $g \in G, m \in M$ is read: the object g has attribute m .*

Usually, a formal context is represented in the form of a table (called a cross-table) which describes a relationship between objects (represented by table rows) and attributes (represented by table columns). An example of such a table is the table 3.1, the entry containing the symbol \times indicates that the corresponding object has the corresponding attribute. For example, in table 3.1 the objects are animals (in rows), the attributes are animals characteristics (in columns) such as “has wings”, and the presence or not of the symbol \times indicates that a particular animal has a given attribute.

Every formal context (G, M, I) induces a pair of operators, so-called *concept-forming operators*.

Table 3.1: A cross table for a formal context about animals where, a: four legs, b: suckles its offspring, c: flies, d: spawns, e: two legs, f: lives in water, g: has wings.

I	a	b	c	d	e	f	g
dog	×	×					
salmon				×		×	
duck			×	×	×		×
bear	×	×					
man		×			×		
lizard	×			×			
penguin				×	×		×

Definition 3.7 (Concept-forming operators). *For an arbitrary set $A \subseteq G$ of objects we define*

$$A^I = \{m \in M \mid gIm \text{ for all } g \in A\}. \quad (3.1)$$

Correspondingly, for an arbitrary set $B \subseteq M$ we define

$$B^J = \{g \in G \mid gIm \text{ for all } m \in B\}. \quad (3.2)$$

Operator I assigns subsets of M to subsets of G . A^I is just the set of all attributes shared by all objects from A . Dually, operator J assigns subsets of G to subsets of M . B^J is the set of all objects sharing all attributes from B . The notion of formal concept is fundamental in FCA. Formal concepts are particular clusters in cross-tables, defined by means of attribute sharing.

Definition 3.8 (Formal concept). *A formal concept of a formal context (G, M, I) is a pair (A, B) with $A \subseteq G, B \subseteq M, A^I = B$, and $B^J = A$. We call A the extent and B the intent of the formal concept (A, B) respectively. The set of all formal concepts of the context (G, M, I) is denoted $\mathfrak{B}(G, M, I)$.*

This formal definition means that (A, B) is a formal concept if and only if A contains just objects sharing all attributes from B and B contains just attributes shared by all objects from A . For example in table 3.1 there is the concept of mammal = $(\{\text{dog, bear, man}\}, \{\text{suckle its offspring}\})$ because $\{\text{dog, bear, man}\}^I = \{\text{suckle its offspring}\}$ and $\{\text{suckle its offspring}\}^J = \{\text{dog, bear, man}\}$.

The notion of formal concept can be seen as a simple formalization of a well-known notion of a concept. A concept is determined by a collection of objects (extent) which fall under the concept, and by a collection of attributes (intent) covered by the concept. Concepts are naturally ordered

using a subconcept-superconcept relation. The subconcept-superconcept relation is based on inclusion relation on objects and attributes. Formally, the subconcept-superconcept relation is defined as follows.

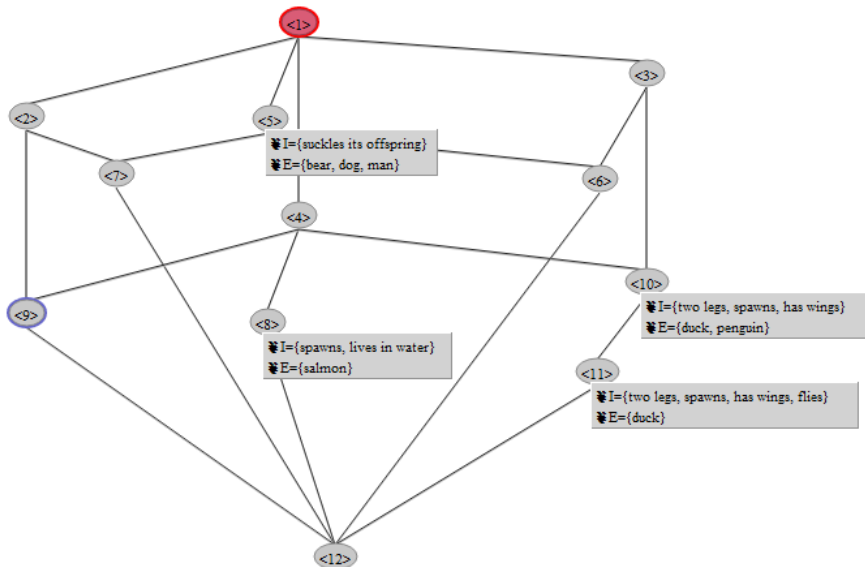
Definition 3.9 (Subconcept-superconcept ordering). *If (A_1, B_1) and (A_2, B_2) are concepts of a context, (A_1, B_1) is called a subconcept of (A_2, B_2) , if $A_1 \subseteq A_2$ (which is equivalent to $B_2 \subseteq B_1$). In this case, (A_2, B_2) is a superconcept of (A_1, B_1) , and we write $(A_1, B_1) \leq (A_2, B_2)$. The relation \leq is called the order of the concepts.*

With the definition above $(A_1, B_1) \leq (A_2, B_2)$ means that (A_1, B_1) is more specific than (A_2, B_2) , and consequently (A_2, B_2) is more general. The subconcept-superconcept ordering captures the intuition behind $\{\text{dog}\} \leq \{\text{mammal}\}$ (the concept of a dog is more specific than mammal one).

Definition 3.10 (Concept lattice). *The set of all concepts of (G, M, I) equipped with the subconcept-superconcept ordering is denoted by $\mathfrak{B}(G, M, I)$ and is called the concept lattice of the context (G, M, I) .*

The notion of concept lattice is very powerful because represents all (potentially interesting) clusters, and the relations between them, which are “hidden” in the simple data (G, M, I) . The Figure 3.1 represents the Hasse diagram of the concept lattice of table 3.1, notice the four concepts of mammal, fish, bird and duck with the relation $\{\text{duck}\} \leq \{\text{bird}\}$. The concept lattice presented in this thesis are derived with the software Galicia [47].

Figure 3.1: The concept lattice for table 3.1.



The concept-forming operators define a Galois connection between the

objects and the attributes of a formal context, this important property is formulated in the following theorem.

Theorem 3.1. *The pair of operators (I, J) induced by (G, M, I) forms a Galois connection between the ordered sets $(2^G, \subseteq)$ and $(2^M, \subseteq)$. The pairs (A, B) such that A is a closed set in 2^G and B is a closed set in 2^M with $B = A^I$ (and thus also $A = B^J$) are exactly the formal concepts of (G, M, I) .*

This results allows us to formulate a first method for deriving the formal concepts of a context, exploiting the properties of Galois connection (A^{IJ}, A^I) is always a concept, so for every $A \subseteq G$, A^I is an intent of some concept. Thus, it suffices for for every $A \subseteq G$ calculate the intent A^I and then the extent A^{IJ} . There are many algorithms for concept lattice construction, for a survey we point out the work of Kuznetsov and Obiedkov [48]; and there are many software systems with the purpose varying from formal context creation to formal concept mining and generation of the concept lattice and the corresponding association rules. Some examples of such systems are ToscanaJ [49], Galicia [47], Lattice Miner, etc.

Moreover, the extents (respectively intents) are closed under intersection. Further explanations can be found in [17]. The previous results enable us to formulate the following theorem characterizing the structure of concept lattices.

Theorem 3.2 (The Basic Theorem on Concept Lattices). *The concept lattice $\underline{\mathfrak{B}}(G, M, I)$ is a complete lattice in which infimum and supremum are given by:*

$$\begin{aligned} \bigwedge_{t \in T} (A_t, B_t) &= \left(\bigcap_{t \in T} A_t, \left(\bigcup_{t \in T} B_t \right)^{JI} \right) \\ \bigvee_{t \in T} (A_t, B_t) &= \left(\left(\bigcup_{t \in T} A_t \right)^{IJ}, \bigcap_{t \in T} B_t \right) \end{aligned}$$

where $A_t \subseteq G$ and $B_t \subseteq M$. A complete lattice \mathbf{X} is isomorphic to $\underline{\mathfrak{B}}(G, M, I)$ if and only if there are mappings $\gamma : G \rightarrow X$ and $\mu : M \rightarrow X$ such that $\gamma(G)$ is supremum-dense in \mathbf{X} , $\mu(M)$ is infimum-dense in \mathbf{X} and gIm is equivalent to $\gamma g \leq \mu m$ for all $g \in G$ and all $m \in M$. In particular, $\mathbf{X} \cong \underline{\mathfrak{B}}(G, M, I)$.

Concentrating only in the first part of the theorem, it unifies the concept lattice of FPA with the already existing mathematical structure of complete lattice. The fact that $\underline{\mathfrak{B}}(G, M, I)$ is a complete lattice is a “welcome” property. Namely, it says that for any collection $K \subseteq \underline{\mathfrak{B}}(G, M, I)$ of formal concepts, $\underline{\mathfrak{B}}(G, M, I)$ contains both the “direct generalization” $\bigvee K$ of concepts from K (supremum of K), and the “direct specialization” $\bigwedge K$ of concepts from K (infimum of K).

3.2 Formalizing the knowledge: Knowledge Space Theory

3.2.1 Introduction and fundamental concepts

Knowledge space theory (KST) [50] is a mathematical theory developed by Jean-Paul Doignon and Jean-Claude Falmagne starting from 1982, with the intent of building an efficient machine for the assessment of knowledge, for example the knowledge in an educational context. This is a very ambitious goal because it implies to formalize the knowledge of a human being, indeed the two key concepts are knowledge state and knowledge structure. So, such a machine will simulate an human examiner, for example, a common teacher asking a question to a student and then another chosen as a function of the student's response to the first one. After a few questions, a picture of the student's knowledge state will emerge, which will become increasingly more precise in the course of examination. By *knowledge state* it is meant the set of all problems the student is capable of solving in ideal conditions, and by *knowledge structure* a distinguished collection of knowledge states. In general, an individual's knowledge state is not directly observable, and has to be inferred from the responses of some questions. For ideal conditions it is assumed that careless errors and lucky guesses do not occur, they will be considered together in the connections between the knowledge state and the observed answers (section 3.2.5), where the probabilistic aspects of the theory are explored.

Such adaptive assessment system was implemented by Falmagne and collaborators from 1992 in a web-based, artificially intelligent assessment and learning system called ALEKS [51]. ALEKS (acronym for Assessment and LEarning in Knowledge Spaces) is able to determine quickly and accurately what a student knows and what doesn't know, then it instructs the student on the topics he is most ready to learn. The software covers various disciplines, ranging from mathematics and natural sciences, to selected topics in business and social sciences with approximately 10.000 knowledge states for only basic arithmetic [10]. In contrast to standardized tests, which typically return a numerical measure of achievement or "aptitude", the outcome of ALEKS consists in the precise and comprehensive delineation of an individual's competence in a subject, in the form of his knowledge state describing all the types of problems he is able to do, and a complete list of the topics he is ready to learn, in simple words the systems displays a pie chart showing the slice of knowledge he possesses and the slice to be learn. Technically it implements a Markov procedure analogue to the one described in chapter 5, and at the end of the assessment it delivers a report about the student's knowledge state and giving immediate access to a teaching module with recommendations for further learning.

It is worth noticing that the main authors of KST have focused their

attention on specific topics included in education of mathematics and statistics, but the formal structure of KST is very general and powerful so they did not exclude the possibility to apply it to different fields of human knowledge. Several examples are medical diagnosis, pattern recognition, axiomatic systems [9], psychological assessment [7, 10, 11], experimental, theoretical and applied cognitive psychology¹.

Another important point is that a knowledge structure needs to be built and the number of states tends to be very large. So, it was developed a first automated *query to experts* procedure called QUERY [9] to build the list of state. A second and more recent developed technique is based on *database query*, where the employed databases are those collected through the years at ALEKS corporation, and the query is based on the prerequisite relations between items. In our case this is not necessary, we will see in the next chapter how it is possible to extract the knowledge structure from a questionnaire using *Formal Concept Analysis*.

The whole KST is fully described in [9], this book doesn't expose only KST mathematical foundations but also more advanced topics such as learning paths, algorithms for constructing a knowledge space, assessment routines and more. But now let's dive in the topic with the first important definitions.

Let be Q the set (non-empty) of all the items (or questions) that it is possible to investigate about a subject, and it is supposed that it is large enough to give a fine-grained, representative coverage of the field. We refer to him as *knowledge domain*. Given the knowledge domain Q , a *knowledge state* $K \subseteq Q$ represents the subset of Q that a specific subject is able to solve.

Definition 3.11 (Knowledge structure). *A knowledge structure is a pair (Q, \mathcal{K}) in which Q is a knowledge domain, and \mathcal{K} is a family of knowledge states of Q , containing at least Q and the empty set \emptyset .*

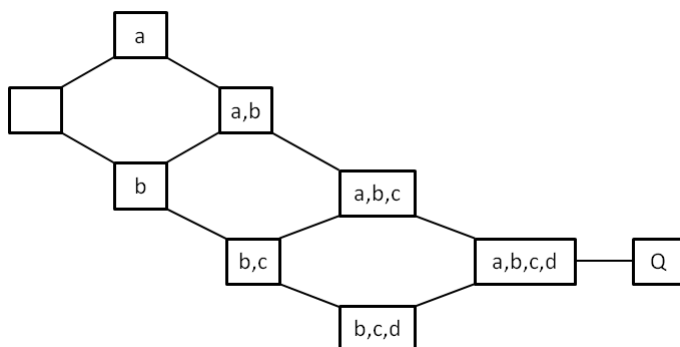
From now \mathcal{K} will be referred as a knowledge structure on a set Q to mean that (Q, \mathcal{K}) is a knowledge structure. The specification of the domain can be omitted without ambiguity since $\bigcup \mathcal{K} = Q$. An example may be useful to clarify these definitions.

Example 3.1. *Let's consider the following knowledge structure defined on domain $Q = \{a, b, c, d, e\}$:*

$$\mathcal{K} = \{\emptyset, \{a\}, \{b\}, \{a, b\}, \{b, c\}, \{a, b, c\}, \{b, c, d\}, \{a, b, c, d\}, Q\}.$$

This knowledge structure in figure contains nine states, the state Q represents the total knowledge about the domain, while \emptyset symbolizes complete ignorance. The graph in figure is the Hasse diagram of (\mathcal{K}, \subseteq) or the covering

¹See the CSS lab at Graz University, <http://wundt.uni-graz.at/>, where you can find a comprehensive bibliography on KST and research projects.



relation of set inclusion: an edge linking state K and a state K' located to its right means that $K \subset K'$, and there is no K'' such that $K \subset K'' \subset K'$. This graphical representation is helpful in understanding the relations among the items, the mastery of item b is a prerequisite for the item of mastery c , in fact there is no state in \mathcal{K} containing c and not containing b , that is any subject failing item c would necessary fail item b , in ideal conditions. Scanning the graph from left to right suggests a learning process: at first the student knows nothing about the field, and is thus in state \emptyset , but then he gradually progresses from state to state, following one path of the graph reaching a complete mastery of the topic (state Q), or stopping before.

Definition 3.12. Let \mathcal{K} be a knowledge structure, \mathcal{K}_q denotes the collection of all states containing item q .

Example 3.2. Following the example 3.1 we have:

$$\begin{aligned}\mathcal{K}_a &= \{\{a\}, \{a, b\}, \{a, b, c\}, \{a, b, c, d\}, Q\} \\ \mathcal{K}_d &= \{\{b, c, d\}, \{a, b, c, d\}, Q\}\end{aligned}$$

An important concept in KST is a special case of knowledge structure, when the family of state is closed under union. This is not essential property, in fact the formal psychological assessment described in chapter 4 uses a simple knowledge structure.

Definition 3.13 (Knowledge space). We say that (Q, \mathcal{K}) is a knowledge space, or \mathcal{K} is a knowledge space on Q , if the family \mathcal{K} is closed under union, that is when $\cup \mathcal{F} = \mathcal{K}$ whenever $\mathcal{F} \subseteq \mathcal{K}$.

It is interesting noticing that in a knowledge space a subject could reach a state following different learning paths, that is the same item can be solved using different solution strategies [50]. In the example a subject can reach the state $[a, b, c]$ mastering first a , then b and c or following the path $\emptyset - [b] - [b, c] - [a, b, c]$.

Definition 3.14 (Dual of a knowledge structure). *The dual of a knowledge structure \mathcal{K} on Q is the knowledge structure $\overline{\mathcal{K}}$ containing all the complements of the states of \mathcal{K} , that is*

$$\overline{\mathcal{K}} = \{K \in 2^Q \mid Q \setminus K \in \mathcal{K}\}.$$

Thus, $\overline{\mathcal{K}}$ and \mathcal{K} have the same domain.

Definition 3.15 (Collection, closure space). *A collection on Q is a collection \mathcal{K} of subsets of the domain Q . Thus, a knowledge structure (Q, \mathcal{K}) is a collection \mathcal{K} which contains both \emptyset and Q . Notice that a collection may be empty. A collection (Q, \mathcal{L}) is a closure space when the family \mathcal{L} contains Q and is closed under intersection. This closure space is simple when $\emptyset \in \mathcal{L}$.*

Next theorem (with our version proof) is of important application in KST, because it links the dual of a knowledge structure with the concept of knowledge space.

Theorem 3.3. *A collection \mathcal{K} of subsets of a domain Q forms a knowledge space on Q if and only if the dual structure $\overline{\mathcal{K}}$ is a simple closure space.*

Proof. In this proof it is necessary to recall *the Morgan's law* $\overline{A \cup B} = \overline{A} \cap \overline{B}$, applying to our domain we have $(Q \setminus K) \cup (Q \setminus K') = Q \setminus (K \cap K')$ with $K, K' \subseteq Q$.

(\Rightarrow) $\overline{\mathcal{K}}$ is a knowledge structure by definition 3.14, let's see if it is closed under intersection. Let be $K, K' \in \mathcal{K}$ and $\overline{K}, \overline{K'} \in \overline{\mathcal{K}}$ such as $K = Q \setminus \overline{K}$ and $K' = Q \setminus \overline{K'}$, by definition of knowledge space we have $K \cup K' \in \mathcal{K} \Leftrightarrow (Q \setminus \overline{K}) \cup (Q \setminus \overline{K'}) \in \mathcal{K} \Leftrightarrow$ (applying *the Morgan's law*) $Q \setminus (\overline{K} \cap \overline{K'}) \in \mathcal{K} \Leftrightarrow \overline{K} \cap \overline{K'} \in \overline{\mathcal{K}}$ by definition 3.14. So $\overline{\mathcal{K}}$ is a closure space, to prove it contains the empty set we simply notice that $Q \in \mathcal{K} \Rightarrow Q \setminus \emptyset \in \mathcal{K} \Rightarrow \emptyset \in \overline{\mathcal{K}}$.

(\Leftarrow) Let be $\overline{K}, \overline{K'} \in \overline{\mathcal{K}}$ thus $\overline{K} \cap \overline{K'} \in \overline{\mathcal{K}}$. By definition 3.14 \mathcal{K} is a knowledge structure and let be $K, K' \in \mathcal{K}$ such as $K = Q \setminus \overline{K}$ and $K' = Q \setminus \overline{K'}$. $\overline{K} \cap \overline{K'} \in \overline{\mathcal{K}} \Leftrightarrow Q \setminus (\overline{K} \cap \overline{K'}) \in \mathcal{K} \Leftrightarrow$ (applying *The Morgan's law*) $(Q \setminus \overline{K}) \cup (Q \setminus \overline{K'}) \in \mathcal{K} \Leftrightarrow K \cup K' \in \mathcal{K}$, so \mathcal{K} is a knowledge space. \square

Now in next three sections we will see some possible relationships between knowledge states and skills, formalizing them with the concept of *skill map* and knowledge state delineated by a skill map; we will proceed catching first the idea (using an example) and then formalizing it.

3.2.2 The skill map

Doignon and Falmagne in [9] assume the existence of some basic set S of *skills*, these skills may consist in methods, procedures, abilities or strategies which could be identified. The idea is to associate with each question q the skills in S , which a subject could follow in order to solve q , and to deduce from this association what the knowledge states are.

Example 3.3. *Question a: Given two points in the Cartesian plane $A(2, 1)$, $B(3, 1)$ what is the equation of the straight line passing through them?*

We explain two simple and fast ways to solve the problem:

1. starting from the equation of the straight line $y = mx + q$, it is possible to impose the passage through A and B obtaining two equations, and then solving the first order linear system associated;
2. knowing the geometric meaning of m in equation $y = mx + q$ we can calculate it, using the Pythagorean theorem, and then calculate q imposing the passage through a point.

These two methods suggest several possible types of associations between the skills and the question, and corresponding ways of constructing the knowledge states consistent with those skills. The simplest idea is to consider each one of the methods as a skill. So the complete set S of skills contains those two skills and some others relevant to different questions. The connection between the questions and skills is formalize by the concept of *skill map*, a mapping $\tau : Q \rightarrow 2^S \setminus \{\emptyset\}$ associating to each question q a subset $\tau(q)$ of skills. In the case of example we have:

$$\tau(a) = \{1, 2\}$$

Definition 3.16 (Skill map). *A skill map is a triple (Q, S, τ) , where Q is a non-empty set of items, S is a non-empty set of skills, and τ is a mapping from Q to $2^S \setminus \{\emptyset\}$.*

When the sets Q and S are specified by the context, we shall refer to the function τ itself as the skill map. For any q in Q , the subset $\tau(q)$ of S will be referred to as the set of skills assigned to q (by the skill map τ).

This important concept of KST was developed by the authors following the works of [52, 53, 54], in order to allow the theory to go beyond a mere formal-mathematical interpretation, where the involved cognitive aspects are limited to comprehensive and general notions. In our view this concept can be easily adapted and reinterpreted in a clinical context, see chapter 4.

3.2.3 The disjunctive and conjunctive model

Let's consider a subject possessing the skill number 2 of the example plus other skills, his skill set is:

$$T = \{2, s, s'\}.$$

We can notice that he is able to solve problem a because he possesses at least one right skill, that is $T \cap \tau(1) = \{2\} \neq \emptyset$, and generalizing we can say that the knowledge state of this subject contains all those questions that can be solved by at least one skill possessed by him, this suggest the following definition.

Definition 3.17 (knowledge state delineated by a subset via the disjunctive model). *Let be (Q, S, τ) a skill map and T a subset of S . We say that $K \subseteq Q$ is the knowledge state delineated by T via the disjunctive model if*

$$K = \{q \in Q \mid T \cap \tau(q) \neq \emptyset\}.$$

In this manner we are defining the so called disjunctive model: in order to master an item it is sufficient to have at least one of the required skills. Notice that the empty set of skills ($T = \emptyset$) delineates the empty knowledge state (because $\tau(q) \neq \emptyset$ for each item q), and that S delineates Q

Definition 3.18 (Knowledge structure delineated by a skill map via the disjunctive model). *A knowledge structure delineated by the skill map (Q, S, τ) via the disjunctive model is the set of all knowledge states delineated by all subsets of S via the disjunctive model.*

Theorem 3.4. *Any knowledge structure delineated via the disjunctive model by a skill map is a knowledge space. Conversely, any knowledge space is delineated by at least one skill map.*

Let's consider the following example:

Example 3.4. *Question b: given a triangle with base 12,3 cm and height 6,5 cm calculate its area A .*

Notice that there is only one and direct strategy to solve the problem, using the standard formula $A = (\text{base} \times \text{height})/2$ and it involves the following skills:

1. knowing the formula $A = (\text{base} \times \text{height})/2$;
2. knowing the multiplication between real numbers;
3. knowing the division between real numbers.

A person to master the problem has to own all the skills above, that is all the skills assigned to the question are required. Thus, K is a state if exists a set T of skills such that, for any item q , we have $q \in K$ exactly when $\tau(q) \subseteq T$. This suggest the following definition.

Definition 3.19 (knowledge state delineated by a subset via the conjunctive model). *Let be (Q, S, τ) a skill map and T a subset of S . We say that $K \subseteq Q$ is the knowledge state delineated by T via the conjunctive model if*

$$K = \{q \in Q \mid \tau(q) \subseteq T\}.$$

In this manner we are formalizing the so called conjunctive model: for any question q there is a unique solution method represented by the set $\tau(q)$, which gathers all the skills required. Notice that the empty set of skills delineates the empty knowledge state, and S delineates Q .

Definition 3.20 (Knowledge structure delineated by a skill map via the conjunctive model). *A knowledge structure delineated by the skill map (Q, S, τ) via the conjunctive model is the set of all knowledge states delineated by all subsets of S via the conjunctive model.*

In the following theorem (with our version proof) we will see the joint between the disjunctive and conjunctive model using the concept of dual of a knowledge structure.

Theorem 3.5. *The knowledge structures delineated via the disjunctive model and conjunctive by the same skill map are dual one to the other.*

Proof. The proof consists in showing that any state in the disjunctive knowledge structure has his complement state in the corresponding conjunctive knowledge structures and vice versa. Let be (Q, S, τ) a skill map and $\mathcal{K}^D, \mathcal{K}^C$ the corresponding disjunctive and conjunctive knowledge structures.

Let's consider $K^D \in \mathcal{K}^D$, thus by definition $K^D = \{q \in Q \mid \tau(q) \cap T \neq \emptyset\}$ and his complement state is composed by the items of Q that aren't in K^D , that is $\bar{K}^D = \{q \in Q \mid \tau(q) \cap T = \emptyset\} = \{q \in Q \mid \tau(q) \subseteq S \setminus T\} = \{q \in Q \mid \tau(q) \subseteq T'\} = K^C \in \mathcal{K}^C$ (by definition 3.19).

Now the proof follows showing the vice versa, that is for any $K^C \in \mathcal{K}^C$ then $\bar{K}^C \in \mathcal{K}^D$. $K^C = \{q \in Q \mid \tau(q) \subseteq T\}$ and his complement state is composed by the items q such as $\tau(q)$ has at least one skill s not in T , that is $\bar{K}^C = \{q \in Q \mid \tau(q) \cap S \setminus T \neq \emptyset\} = \{q \in Q \mid \tau(q) \cap T' \neq \emptyset\} = K^D \in \mathcal{K}^D$ (by definition 3.17). \square

As a consequence if we apply theorem 3.4 and then theorem 3.3 to the theorem above we have the following corollary.

Corollary 3.6. *A knowledge structure delineated via the conjunctive model is a closure space.*

Example 3.5. *Let be $Q = \{1, 2, 3\}$ the knowledge domain, $S = \{a, b, c\}$ the skills set and the mapping τ :*

$$\tau(1) = \{a\}, \tau(2) = \{b\}, \tau(3) = \{b, c\}$$

the disjunctive knowledge structure is

$$\mathcal{K}^D = \{\emptyset, \{1\}, \{3\}, \{1, 3\}, \{2, 3\}, Q\}$$

and the conjunctive one is

$$\mathcal{K}^C = \{\emptyset, \{1\}, \{2\}, \{1, 2\}, \{2, 3\}, Q\}$$

so it's easy to verify that their are dual one respect the other.

3.2.4 The competency model

In example 3.3 we have been a little coarse in our skills analysis, we considered the two methods themselves as skill, even though several knowledges are required for both of them:

1. knowledge of the canonical equation of the line: $y = mx + q$;
2. knowledge of geometric meaning of coefficient m ;
3. knowledge of geometric meaning of coefficient q ;
4. knowledge of Pythagoras' theorem;
5. knowledge of Euclidean's first postulate²;
6. knowledge of the resolution of a first grade equation;
7. solving a first order linear system;
8. imposing the passage of a line through a point.

So a possible complete set S of skill could be

$$S = \{1, 2, 3, 4, 5, 6, 7, 8, s_1, s_2, \dots, s_n\}$$

where s_1, \dots, s_n refer to skills relevant to other questions in the domain under consideration, the sets for solving question a with methods 1 and 2 are respectively

$$\begin{aligned} M_1 &= \{1, 5, 6, 7, 8\}, \\ M_2 &= \{1, 2, 3, 4, 5, 6, 8\}. \end{aligned}$$

Let's suppose that a subject under examination is equipped with a skills set

$$P = \{1, 5, 6, 7, 8, s_1, s_3\},$$

he would be able to master question a only with the first method because the knowledges belonging to method 1 are included in the subject's skills set:

$$M_1 \subseteq P.$$

This suggest a different type of association between questions and skills, a new function μ maps each item $q \in Q$ to the collection of all the subsets of skills corresponding to possible solutions of q ; this concept is formalized in the next definition.

²Given any two points there is a unique straight line passing between them.

Definition 3.21 (Skill multi-map). *A skill multi-map is a triple (Q, S, μ) , where Q is a non-empty set of items, S is a non-empty set of skills, and $\mu : Q \rightarrow (2^{2^S \setminus \{\emptyset\}}) \setminus \{\emptyset\}$ is a mapping that associates to any q a non-empty family $\mu(q)$ of non-empty subsets of S .*

Any subset C of skills in $\mu(q)$ is called competency and it can be viewed as a method, a strategy in order to master item q . This is the competency model, possessing just one of these competencies is sufficient to solve question q but all the skills contained in that strategy are necessary. This formulation, involves that each state K is composed by all those items with at least one of their competencies included in X .

Definition 3.22 (knowledge state delineated by a subset via the competency model). *Let be (Q, S, μ) a skill multi-map and T a subset of skill. We say that $K \subseteq Q$ is the knowledge state delineated by T via the competency model if K contains all the items having at least one competency included in T , formally*

$$q \in K \iff \exists C \in \mu(q) : C \subseteq T.$$

Definition 3.23 (Knowledge structure delineated by a skill multi-map via the competency model). *A knowledge structure delineated by the skill multi-map (Q, S, μ) via the competency model is the set of all knowledge states delineated by all subsets of S via the competency model.*

With the new model introduced it is possible to see the disjunctive and conjunctive models as particular cases of it. In fact, the knowledge structures corresponding to these two models are respectively closed under union and intersection but the structures delineated by a skill multi-map are neither closed under intersection nor under union.

3.2.5 Probabilistic knowledge structures

All the elements introduced so far are by definition deterministic. As such the concept of knowledge structure doesn't provide a realistic prediction of subject's response, indeed, sometimes the observed responses doesn't correspond to a knowledge state in the structure. Subjects could fail problems that they fully understand (careless errors), or provide correct responses to problems that they do not understand at all (lucky guesses). Moreover, it is reasonable to think that the knowledge states will occur with different frequencies in the population of reference. The usual way to face such characteristics is to enlarge our theoretical framework with a probabilistic approach.

Definition 3.24 (Probabilistic knowledge structure). *A probabilistic knowledge structure is a triple (Q, \mathcal{K}, p) in which*

1. (Q, \mathcal{K}) is a knowledge structure;

2. the mapping $p : \mathcal{K} \rightarrow [0, 1] : K \mapsto p(K)$ is a probability distribution on \mathcal{K} . Thus, for any $K \in \mathcal{K}$, we have $p(K) \geq 0$, and moreover, $\sum_{K \in \mathcal{K}} p(K) = 1$.

Let's recall the knowledge structure of example 3.1

$$\mathcal{K} = \{\emptyset, \{a\}, \{b\}, \{a, b\}, \{b, c\}, \{a, b, c\}, \{b, c, d\}, \{a, b, c, d\}, Q\}, \quad (3.3)$$

with $Q = \{a, b, c, d, e\}$; we suppose that any subject sampled from a given population of reference will be in one of these states, the probability $p(K)$, with $K \in \mathcal{K}$, attached to each state represents the likelihood that a subject is in that state. But the presence of careless errors and lucky guesses entails that all kinds of response patterns may be generated, this implies that the subject's knowledge state may not be directly observable, and explains why an observed response pattern may not correspond to a knowledge state. It needs to deal with a conditional probabilities of responses, given the state. Let's formalize the problem, let be $R \subseteq Q$ the response pattern (or simply pattern) containing all the items correctly solved by a subject, there are $2^{|Q|}$ possible patterns; and we indicate $r(R, K)$, for any $R \subseteq Q$ and $K \in \mathcal{K}$, as the conditional probability of response pattern R given state K .

Definition 3.25 (Response function). *A response function for a probabilistic knowledge structure (Q, \mathcal{K}, p) is a function $r : (R, K) \mapsto r(R, K)$, defined for all $R \subseteq Q$ and $K \in \mathcal{K}$, and specifying the probability of the response pattern R for a subject in state K .*

Thus, for any $R \in 2^Q$ and $K \in \mathcal{K}$, we have $r(R, K) \geq 0$; moreover $\sum_{R \subseteq Q} r(R, K) = 1$.

Definition 3.26 (Basic probabilistic model). *A basic probabilistic model is a quadruple (Q, \mathcal{K}, p, r) , in which (Q, \mathcal{K}, p) is a probabilistic knowledge structure and r its response function.*

Now let's consider a subject responding correctly to questions a, d of example 3.1, and fails to solve b, c and e , and if we write $R \mapsto \rho(R)$ for the probability of pattern $R = \{a, d\}$ (in this case not corresponding to a state in our knowledge structure), we obtain

$$\rho(a, d) = r(\{a, d\}, \{a, b, c, d\})p(\{a, b, c, d\}) + r(\{a, d\}, Q)p(Q) \quad (3.4)$$

indeed, the only two states that can generate R are the states including it, $\{a, b, c, d\}$ and Q . We introduce the *local independence* assumption, i.e. each response is independent to the other. For simplicity we assume, in this example, that a subject never guesses an item not in his state (no lucky guesses), and finally we indicate β_q as the careless error probability, the probability of an incorrect response to a question q in the subject's state. So

$$\begin{aligned} r(\{a, d\}, \{a, b, c, d\}) &= \beta_b \beta_c (1 - \beta_a)(1 - \beta_d), \\ r(\{a, d\}, Q) &= \beta_b \beta_c \beta_e (1 - \beta_a)(1 - \beta_d). \end{aligned}$$

3.3 Unify the concepts with the knowledge: the joint between FCA and KST

With this example it's possible to generalize equation 3.4 (including lucky guesses probability) and extend the probabilistic framework. For the total probability theorem we have, for any $R \subseteq Q$,

$$\rho(R) = \sum_{K \in \mathcal{K}} r(R, K)p(K) \quad (3.5)$$

and the local independence assumption can be translate in the formula,

$$r(R, K) = \left(\prod_{q \in K \setminus R} \beta_q \right) \left(\prod_{q \in K \cap R} (1 - \beta_q) \right) \left(\prod_{q \in R \setminus K} \eta_q \right) \left(\prod_{q \in \overline{R \cup K}} (1 - \eta_q) \right) \quad (3.6)$$

where $\beta_q, \eta_q \in [0, 1[$ are two constants indicating respectively the care-less error probability and the lucky guess probability for each $q \in Q$, and $\overline{R \cup K} = Q \setminus (R \cup K)$.

Definition 3.27 (BLIM). *The basic local independence model (BLIM) is the basic probabilistic model satisfying local independence assumption.*

The model has some parameters that must be estimated from the data (i.e. the patterns of a sample of population), that are the constant β_q, η_q (in number $2 \times |Q|$) and the probabilities $p(K)$, with $K \in \mathcal{K}$ (in number $|\mathcal{K} - 1|$), with a total number of $2 \times |Q| + |\mathcal{K} - 1|$ parameters. The Chi-square statistic or maximum likelihood estimates can give a measure of them and evaluate the goodness of fitting of the structure, while the expectation-maximization algorithm [55] can extract them, see [9]. So in order to have a good model (the knowledge structure), parameters β and η are in general expected to be low.

3.3 Unify the concepts with the knowledge: the joint between FCA and KST

The most used procedure in literature for constructing a knowledge space is consulting experts of a certain field, asking them questions on the items complexity and then building the knowledge structure on the basis of their responses. This method is known as query to expert procedure and it involves the possibility of sorting items from the simplest to the most complex; this method is formalized through the entail relation, see [9] for a better understanding. So, this task is quite simple in a mathematical-statistical context, but asking a therapist to sort the clinical manifestations of OCD from the most important to the least important is very difficult, for this reason is better exploring other simplest methodologies.

In this section we show the connection between FCA and KST, in order to derive a knowledge structure from a formal context (or skill map). Either

Falmagne and Doignon [9], Rush and Wille [43] and [10] demonstrated this connection but different approaches, here we propose our detailed proof.

Let be \mathcal{K} a knowledge structure delineated by the skill map (Q, S, τ) via the conjunctive model, with Q and S finite, and let's define the complementary relation R between Q and S : for every $q \in Q$ and $s \in S$ we have

$$qRs \iff s \notin \tau(q).$$

A generic knowledge state $K \in \mathcal{K}$ delineated by a subset T of S via the conjunctive model, has the form $K = \{q \in Q \mid \tau(q) \subseteq T\}$, considering the relation R and with few formal passages we have $K = \{q \in Q \mid \forall s \in S \setminus T : qRs\}$. Now if we consider Q as a set of objects and S as a set of attributes, we can regard (Q, S, R) as a formal context, and let be $\underline{\mathfrak{B}}(Q, S, R)$ the relative concept lattice. We have the following theorem.

Theorem 3.7. *Let be \mathcal{K} a knowledge structure delineated by the skill map (Q, S, τ) via the conjunctive model, with Q and S finite, its knowledge states are the extents of the concept lattice $\underline{\mathfrak{B}}(Q, S, R)$ of the formal context (Q, S, R) .*

Proof. Let's consider a generic knowledge state K of \mathcal{K} , for the considerations above we have $K = \{q \in Q \mid \forall s \in S \setminus T : qRs\}$, the proof will show that K is an extent of $\underline{\mathfrak{B}}(Q, S, R)$. With the definition of concept-forming operators in mind we notice that $K = (S \setminus T)^J$, now let's apply the operator I to K , if $K^I = (S \setminus T)^{JI} = S \setminus T$ the pair $(K, S \setminus T)$ is a formal concept, whose extent is exactly K . Applying I : $K^I = \{s \in S \mid \forall q \in K : qRs\} = S \setminus T$, because K is composed by the elements of the set $\tau(q) \subseteq T$, whose complementary, under relation R , is exactly $S \setminus T$.

The second part of the proof shows that every extent of $\underline{\mathfrak{B}}(Q, S, R)$ is a knowledge state of \mathcal{K} . Let's consider a generic formal concept (A, B) with $A \subseteq Q$ and $B \subseteq S$, by definition $B^J = \{q \in Q \mid qRs : \forall s \in B\}$, if we choose T such as $B = S \setminus T$ and, considering the definition of conjunctive model and of the relation R , we have that $B^J = \{q \in Q \mid qRs : \forall s \in S \setminus T\}$ is a knowledge state of \mathcal{K} delineated by T . \square

Thanks to this result it is possible using existing concept lattice construction systems (as those seen above) for deriving a knowledge structure by a given skill map (Q, S, τ) (a simple cross-table) via the conjunctive model.

Example 3.6. *Let's consider the skill map (or the cross-table) of table 3.2, where $Q = \{1, 2, 3, 4\}$ and $S = \{a, b, c, d\}$, and its complementary relation R in table 3.3.*

Now let's consider the concept lattice $\underline{\mathfrak{B}}(Q, S, R)$ of cross-table 3.3 in Figure 3.2, with few simple passages it is possible to derive the knowledge structure of table 3.2 via the conjunctive model and see that is the same mathematical structure of $\underline{\mathfrak{B}}(Q, S, R)$.

3.3 Unify the concepts with the knowledge: the joint between FCA and KST

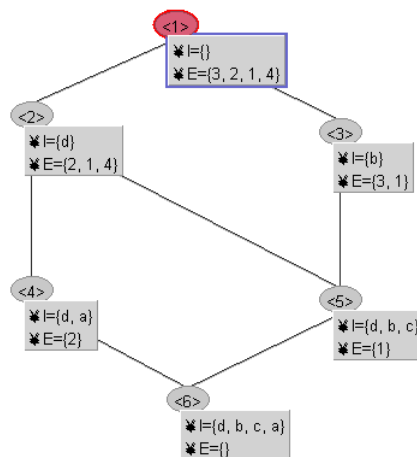
Table 3.2: The skill map τ .

τ	a	b	c	d
1	×			
2		×	×	
3	×		×	×
4	×	×	×	

Table 3.3: The complementary relation R of the skill map τ .

R	a	b	c	d
1		×	×	×
2	×			×
3		×		
4				×

Figure 3.2: The concept lattice for table 3.3.



Chapter 4

The Formal Psychological Assessment

In chapter 3 we have explored overlaps between KST and FCA, in this chapter we will see how their conjoint application in a clinical framework can result in a formal model (or formal representation) of the OCD. This union is the basis of FPA, indeed “FPA arises from the application in the clinical psychology context, and, more specifically, in the psycho-diagnostic assessment, of two mathematical psychology theories: *Knowledge Space Theory* and *Formal Concept Analysis*” [7]. Particularly FPA is a methodology for deriving a clinical structure (representing the relations between the items of a given clinical questionnaire) from a clinical context (consisting in a cross-table containing items in rows and diagnostic criteria inferred by the items in columns). This procedure represents an innovative way, and poorly used, to construct a knowledge structure; furthermore, it allows to consider β and η parameters estimates as diagnostic tools of models goodness of fit. The output of FPA, i.e. the formal model, will be a data structure used as input of the adaptive assessment algorithm of chapter 5.

4.1 The core idea behind FPA

As already seen in section 3.3 it is quite difficult for a psychologist sorting the item of a clinical questionnaire by importance order, for this reason Spoto et al. in developing FPA explored the join between FCA and KST.

The core idea is to depict each psychological disorder as a concept lattice whose intents are attributes (i.e. diagnostic elements) configurations that can approach or move away from the diagnosis of the disorder. This information is certainly more complete than the simple score of a test, but in performing such formalization it is necessary to build a formal context having the items of a questionnaire as objects, and the diagnostic criteria investigated by each

single item as attributes, this kind of relation can be interpreted as a skill map. The next step will be to derive a closure space from this lattice in order to obtain the probabilistic parameters of the BLIM, required for the assessment.

So it emerges the necessity to interpret the main concepts of KST and FCA in the perspective of clinical assessment: it has to be verified whether it is possible performing this “translation”, and if this formal model provide a good representation of the empirical data [11].

4.2 The translation from knowledge context to clinical context

Let's start by considering the concept of skill map and let's see how to adapt it to the psychological assessment context. Let be (Q, S, τ) a skill map, where Q is the set of items of a clinical questionnaire investigating a certain disorder, S is a set of diagnostic criteria (clinical symptoms or diagnostic attributes) defining the given disorder and τ a function associating every item q the set of attributes it infers. In this translation the concept of skill map is changed, that is, a skill is no more a set of abilities needed to solve an item, but rather a set of diagnostic criteria that a patient answering “True” to an item possesses. Another important remark concerns the answers “True”, does the patient possess all the diagnostic elements inferred by the item or only a subset? Namely, is it better to consider a conjunctive model or a disjunctive one? In the first case a positive answer indicates that the patient possesses all the criteria inferred by the item, so a positive answer is more informative than a negative one. In the disjunctive case, on the contrary, negative answers are more informative, because they indicates that the patient have no one of the attributes implied by the item [7]. But in a psychological context the clinician evaluates directly the score obtained by a patient assuming that the patient shows all the symptoms investigated by a specific item, so the conjunctive model reflects better the psychologist's approach.

The next step is to consider the triple (Q, S, R) as a clinical context, interpreting Q as the set of objects, S as the collections of attributes, and defining the binary relation $R \subseteq Q \times S$, such as for every $q \in Q$ and $s \in S$ we have

$$qRs \iff s \notin \tau(q).$$

Now, from the clinical context (Q, S, R) is possible to derive the concept lattice $\underline{\mathfrak{B}}(Q, S, R)$, whose objects are composed by a collection of items (the extent) and by the set of all those attributes that no one of the item in the extent infers (the intent).

The last step consists in applying theorem 3.7 to our lattice $\underline{\mathfrak{B}}(Q, S, R)$ that guarantee us that its extents are the knowledge states of the knowledge

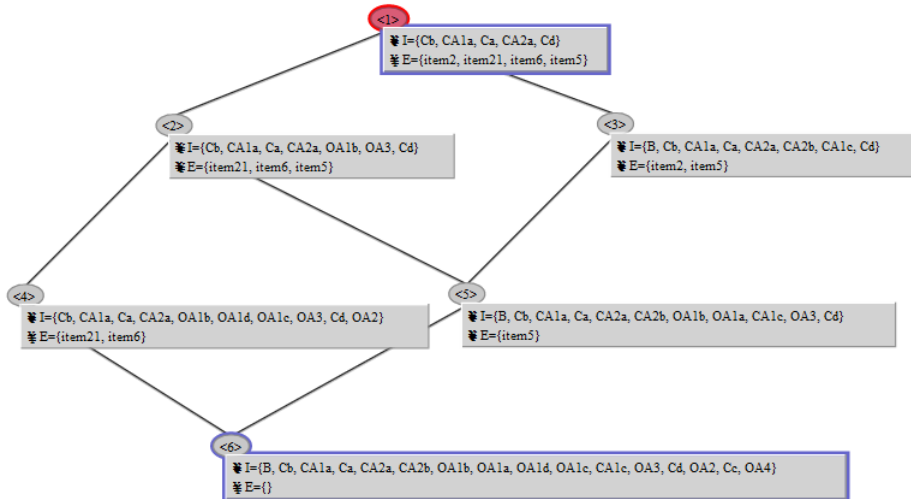
structure \mathcal{K} delineated by the skill map (Q, S, τ) via the conjunctive model; moreover, \mathcal{K} is a closed under intersection by corollary 3.6.

A final remark concerns with the conjunctive model. In this case it reflects better the reality, but nothing excludes a possible use of disjunctive model, indeed, from our knowledge structure \mathcal{K} delineated via the conjunctive model it is possible to derive its dual knowledge structure $\overline{\mathcal{K}}$, that is (theorem 3.5), the knowledge structure delineated via the disjunctive model by the same skill map (Q, S, τ) .

4.3 Formalizing the Obsessive-Compulsive Disorder

Now we have all the formal instruments for deriving a representation of the OCD, we need only the formal context. The objects are the items of the MOCQ in its reduced version presented by Sanavio and Vidotto (MOCQ-R) [22], and the attributes are the diagnostic criteria for OCD taken from the DSM-IV-TR [1]. As in CBA 2.0 the formalization is divided into three scales, Checking, Cleaning and Doubting-Ruminating. For example the doubting-ruminating sub-scale investigates the presence of intrusive and disagreeable thoughts. A typical item of this sub-scale is “I frequently have disagreeable thoughts and I cannot get rid of them”, and peculiar diagnostic criteria (or symptoms) can be “Recurrent and persistent thoughts, impulses, or images” or “Recognition of the excessiveness and unreasonableness of obsessions or compulsions”. In Figure 4.1 is shown the concept lattice of the doubting-ruminating sub-scale.

Figure 4.1: The concept lattice of the Doubting-Ruminating sub-scale.



Going into details of the figure, every node is a formal object, the extents are sets of items, the intents set of diagnostic attributes, and the lines be-

tween nodes represent the ordering relation between concepts, see definition 3.9. Concentrating on the extents set, we can notice that item 2 appears always in presence of number 5, but item 5 appears also alone (is a singleton), so in KST we say that item 5 is a *prerequisite* for item 2, that is, the mastery of item 2 entails the mastery of item 5. This fact has important consequences from practical point of view, indeed the assessment algorithm will not ask item 5 if the patient has response yes to item 2, it automatically infers item 5. Let's consider now a particular intent I , we don't have to consider its attributes because they are derived using the relation $R = \{(q, s) \mid s \notin \tau(q)\}$, thus for representing the attributes that every item infers, in the corresponding extent E , we have to calculate the complementary set I' of I :

$$I' = S \setminus I. \quad (4.1)$$

This is totally equivalent of calculating the closure under union of the diagnostic attributes inferred by every item by the skill map (Q, S, τ) :

$$I' = \bigcup_{q \in I} \tau(q). \quad (4.2)$$

For example, if the assessment algorithm assigned a patient the knowledge state $K = \{\text{item 5, item 2}\}$, his attributes would be the union set of the attributes of item 5 and 2.

It is noteworthy to observe that this procedure takes the form of an initial application relating to a very narrow field (the OCD). But the method provided by FPA presents an high degree of generality, and may be applied to contexts much broader and ever-increasing complexity, for example the other questionnaires of CBA 2.0.

4.4 Testing the structure through real data

As emerged in the previous chapter, probability can arise in different ways in a deterministic knowledge structure, for example a person can give a response pattern not included in the knowledge structure, this concerns with the fitting of the model to the reality. Furthermore, the states of a knowledge structure could spread out with a certain probability in a sample of population, for example in a sample of random individuals the state with greatest number of occurrences, almost surely, will be the empty set. We also have introduced the careless errors and lucky guesses parameters, in clinical context we will rename them in a more appropriate manner, respectively, false negatives and false positives.

From one side we have the model, the three clinical structures given by FPA, and on the other side we have the reality, the response patterns to the sheet number nine of CBA 2.0, along with the respective frequencies. The goal of Spoto et al. was to validate the three structures and obtain

the BLIM parameters, that is the distribution probability of each state and the β, η parameters. They used a sample of 33 patients from north-east of Italy, all with an OCD diagnosis. The BLIM parameters were estimated by a specific version of the expectation-maximization algorithm [55] for Matlab, i.e. CEMBLIM [7]. The fit of the model was tested by chi-square technique and by p-value for χ^2 obtained by parametric bootstrap. In table 4.1 we report the parameter β, η estimates.

Table 4.1: Estimated parameters β, η for each item of the three structures.

Checking			Cleaning			Doubting-Ruminating		
Item	β	η	Item	β	η	Item	β	η
4	0.00	0.00	3	0.27	0.07	2	0.00	0.40
7	0.24	0.23	8	0.21	0.00	5	0.06	0.00
9	0.06	0.13	10	0.00	0.21	6	0.22	0.00
11	0.00	0.26	13	0.23	0.27	21	0.10	0.09
12	0.00	0.40	16	0.05	0.00			
14	0.00	0.00	17	0.00	0.00			
15	0.16	0.00	18	0.24	0.00			
19	0.00	0.20	20	0.16	0.00			

These parameters are conceptually very important, they represent false negatives and false positive probabilities, and the difference respect KST is that a patient could intentionally fake the answer (e.g. he would show social desirability), or could be affected by poor introspection capabilities [7]. High values of this parameters are an indicator of bad specification of the model or bad wording of the items, so it emerges the possibility of using these probabilities on order to improve the structures or the items [7]. The authors of FPA provided also a methodology to cope with high values of β, η (deflating them to 0), but this issue exceeds the goal of our work. We will use β, η as a refinement of the output of the algorithm.

In table 4.2 are listed the fit indexes of the structures. It is important to notice that for large data matrices, as those used in this contest, the asymptotic distribution of χ^2 is not reliable, for this reason the it has been

Table 4.2: Fit indexes of the three structures.

Structure	χ^2	bootstrap p
Checking	127.39	0.2186
Cleaning	141.65	0.1003
Doubting-Ruminating	2.30	0.8056

used the parametric bootstrap technique. From the values in the table we can see that the three models present a good degree of fitting with the data.

Chapter 5

The adaptive assessment algorithm

As seen in the previous chapters the aim of KST authors was the construction of an efficient system for the assessment of knowledge. They developed and adjust and a generic adaptive assessment algorithm to KST, i.e. the input of the of the assessment procedure is a knowledge structure. For this reason their work can be regarded as the already mentioned CAT [31], with the critical difference that the outcome of the assessment here is a knowledge state, rather than a numerical estimate of a student's competence in a certain topic. The goal is to focus, as fast as possible, on some knowledge state capable of explaining all the responses.

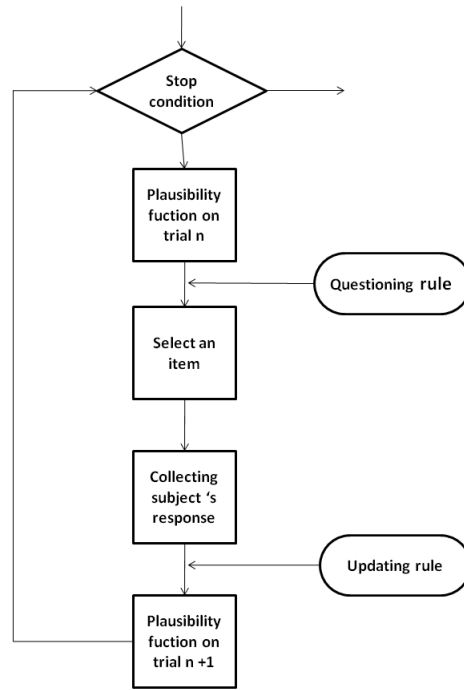
In this chapter we present in details such Markovian procedure that was crucial for our OCD assessment system, indeed our work was to adapt this algorithm to the clinical structure given by FPA.

5.1 Sketching a stochastic procedure

A typical assessment procedure includes three basic elements: a procedure for selecting items from the pool (the questioning rule), a procedure for estimating the subject's latent trait level (the updating rule), and a termination rule to determine when testing may be discontinued (the stop condition). All the assessment procedures available pertain to the scheme outlined in Figure 5.1.

At the beginning of the assessment (step or trial 1), the procedure requires a probabilistic knowledge structure (Q, \mathcal{K}, L) , where for every state $K \in \mathcal{K}$, $L(K)$ represents the probability of the state in the population of reference. The function L can be consider a sort of a priori likelihood, which may depend on statistical data of the population of reference, and the sum of every $L(K)$ is equal to 1. If no useful information is available, then all the states are assigned the same likelihood, namely L is an uniform distribution. At the step n the algorithm considers as plausibility function of every state

Figure 5.1: Diagram of transitions in an assessment procedure.



its current likelihood, based on all the information accumulated so far. This function is used to select the next question to ask, a task performed by the procedure called “questioning rule”. It requires as input the plausibility function and chooses the question, q , “maximally informative”, that is, the sum of the likelihoods of all the states containing q has to be as close as possible to 0.5. If several items are equally informative, one of them is chosen at random.

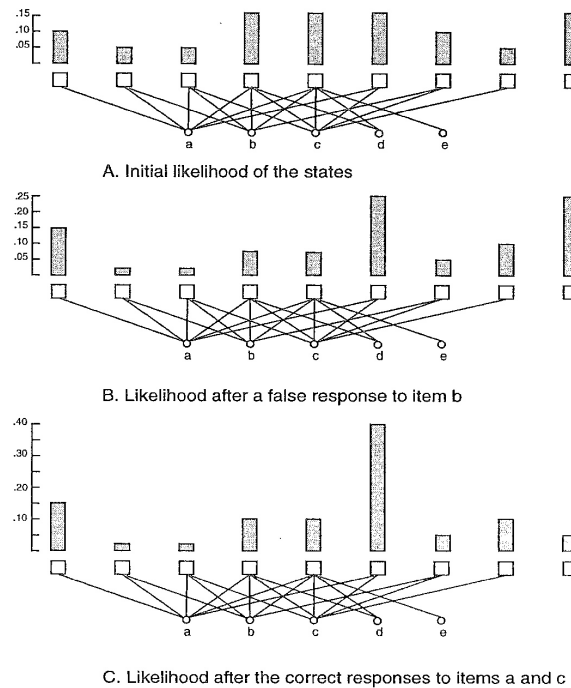
The subject’s response is then observed and collected by the system in a data structure, careless error (false negatives) and lucky guesses (false positives) can arise in this moment, but the procedure will consider them at the end of the assessment. Now the procedure can update the likelihood of every state according to the following updating rule procedure. If subject’s answer to q is correct, the likelihoods of every state containing q are increased and, correspondingly, the likelihoods of every state not containing q are decreased (so that the overall likelihood, summed over all the states, remains equal to 1). A wrong response, on the contrary, has the opposite effect: the likelihoods of every state not containing q are increased, and that of the remaining states decreased.

The assessment procedure stops when a stop condition is satisfied, there are many criteria useful for the stop condition, but in the general, the system

stops when no other questions can specify better the final knowledge state. For example the entropy of the likelihood distribution, which measures the uncertainty of the assessment system regarding the subject's state, can reach a critical low level and thus stopping the procedure. Now the system is ready to select the most likely state.

In Figure 5.2 we have an example, taken from [9], of the assessment procedure above. In this case $Q = \{a, b, c, d, e\}$, the knowledge states are represented by squares, and the items by circles. A link between a square and a circle means that the corresponding state contains the corresponding item. The a priori likelihood of the states, L , is presented in the histogram

Figure 5.2: Successive updating of the likelihood induced by the events: (item b , false), (item c , correct), (item a , correct).



of Figure 5.2A, then, the system asks question b , collects a wrong answer and update the likelihood, Figure 5.2B shows this case. Next, items a and c are presented successively, eliciting two correct responses, and updating again the likelihood. In Figure 5.2C it is possible to see that the probability of state $\{a, c\}$ is much higher than that of any other state.

5.2 Basic concepts and their formalization

In this section we present the mathematical formalization of the procedure seen above. We start with a probabilistic knowledge structure (Q, \mathcal{K}, L) , where L is the initial likelihood of every state. The set of all probability distributions on \mathcal{K} is denoted by Λ , thus $L \in \Lambda$. A subject under examination lies in some knowledge state K_0 , called latent state, the goal of the procedure is to uncover this state. The algorithm proceeds in many steps, or trial, numbered with the letter n , and the plausibility function at the beginning of step n will be denoted by the vector \mathbf{L}_n , thus the initial step has $n = 1$, and $\mathbf{L}_1 = L$. In the framework of the stochastic process, \mathbf{L}_n is a random vector in Λ , moreover for any state $K \in \mathcal{K}$, we denote by $\mathbf{L}_n(K)$ the likelihood that the subject under examination is in the state K at the beginning of the trial n . The likelihood $\mathbf{L}_n(K)$ can be defined also for a set of knowledge states, for every $\mathcal{F} \subseteq \mathcal{K}$:

$$\mathbf{L}_n(\mathcal{F}) = \sum_{K \in \mathcal{F}} \mathbf{L}_n(K). \quad (5.1)$$

The question asked to the subject, at step n , is formalized as a random variable \mathbf{Q}_n taking its values in Q , while the questioning rule is a function $\Psi : Q \times \Lambda \rightarrow [0, 1]$, such as

$$(q, \mathbf{L}_n) \mapsto \Psi(q, \mathbf{L}_n),$$

that specify the probability that $\mathbf{Q}_n = q$.

The observed response on trial n is denoted by the random variable \mathbf{R}_n , that assumes the value 1 in the case of correct response and 0 in case of wrong response.

At the core of this stochastic procedure (Markovian in this case) there is the updating rule that updates the likelihood \mathbf{L}_n at the beginning of step n , on the basis of the question asked q and of the response collected r . This transition rule is formalized as a function $u : \{0, 1\} \times Q \times \Lambda \rightarrow \Lambda$:

$$\mathbf{L}_{n+1} = u(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n).$$

With this formalization, each step of the algorithm (or better, stochastic process) is characterized by the transitions

$$(\mathbf{L}_n \rightarrow \mathbf{Q}_n \rightarrow \mathbf{R}_n) \rightarrow \mathbf{L}_{n+1},$$

and the complete history of the process from trial 1 to trial n is denoted by

$$\mathbf{W}_n = ((\mathbf{R}_1, \mathbf{Q}_1, \mathbf{L}_1), \dots, (\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n)),$$

where \mathbf{W}_0 stands for the empty history.

As already said, the assessment problem consists in uncovering the latent state K_0 , this fact can be formalized with the condition

$$\mathbf{L}_n(K_0) \rightarrow 1,$$

when this condition holds, we say that K_0 is uncoverable by the procedure.

Finally we recall the indicator function of a set, for every subset A of a given set S let be $\iota_A : S \rightarrow \{0, 1\}$ such as:

$$\iota_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \in S \setminus A. \end{cases}$$

A summary of the notation introduced so far is presented in table 5.1.

Table 5.1: Notation involved in the formalization of the assessment procedure.

(Q, \mathcal{K}, L)	a finite probabilistic knowledge structure;
Λ	the set of all probability distribution on \mathcal{K} ;
K_0	the latent knowledge state of the subject;
$\mathbf{L}_1 = L$	the initial likelihood of the states, $0 < L < 1$;
$\mathbf{L}_n(K)$	a random variable representing the likelihood of state K on trial n ;
\mathbf{Q}_n	a random variable representing the question asked on trial n ;
\mathbf{R}_n	a random variable representing the response given on trial n ;
Ψ	$(q, \mathbf{L}_n) \mapsto \Psi(q, \mathbf{L}_n)$, the questioning rule;
u	$(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n) \mapsto u(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n)$, the updating rule;
\mathbf{W}_n	random history of the process from trial 1 to trial n ;
ι_A	the indicator function of set A .

Before giving an important definition about the stochastic procedure seen, we have to introduce some axioms, as done in [9]. These axioms concern the probabilistic knowledge structure (Q, \mathcal{K}, L) , the latent state K_0 of the subject, and the triple of random vectors $(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n)$.

Axiom [U]. We have $\mathbb{P}(\mathbf{L}_1 = L) = 1$, and for any positive integer n and all measurable sets $B \subseteq \Lambda$,

$$\mathbb{P}(\mathbf{L}_{n+1} \in B \mid \mathbf{W}_n) = \iota_B(u(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n)).$$

Writing u_k for the coordinate of u associated with the knowledge state K , we have thus

$$\mathbf{L}_{n+1}(K) = u_K(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n).$$

Furthermore, the function u satisfies the following condition:

$$u_K(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n) = \begin{cases} > \mathbf{L}_n(K) & \text{if } \iota_K(\mathbf{Q}_n) = \mathbf{R}_n \\ < \mathbf{L}_n(K) & \text{if } \iota_K(\mathbf{Q}_n) \neq \mathbf{R}_n. \end{cases}$$

This axiom concern the reallocation of the mass of \mathbf{L}_n on trial $n + 1$ depending on the values of \mathbf{Q}_n and \mathbf{R}_n . This axiom ensures that no knowledge state will ever have a likelihood of zero, and that the likelihood of any state K will increase every time we observe either a correct response to a question $q \in K$, or an incorrect response to a question $q \notin K$, and decrease in the two remaining cases.

Axiom [Q]. For all $q \in Q$ and all positive integers n ,

$$\mathbb{P}(\mathbf{Q}_n = q \mid \mathbf{L}_n, \mathbf{W}_{n-1}) = \Psi(q, \mathbf{L}_n).$$

This axiom govern the choice of an item \mathbf{Q}_n to be asked.

Axiom [R]. For all positive integers n ,

$$\mathbb{P}(\mathbf{R}_n = \iota_{K_0}(q) \mid \mathbf{Q}_n = q, \mathbf{L}_n, \mathbf{W}_{n-1}) = 1.$$

Definition 5.1 (Stochastic assessment procedure). *A stochastic assessment procedure for (Q, \mathcal{K}, L) parametrized by u , Ψ and K_0 is a process as $(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n)$ satisfying the Axioms [U], [Q] and [R].*

Now we have the formal language to state a very important result [9], that is, the stochastic assessment process $(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n)$ is a Markovian process.

Theorem 5.1. *The process (\mathbf{L}_n) is Markovian. That is, for any positive integer n and any measurable set $B \subseteq \Lambda$:*

$$\mathbb{P}(\mathbf{L}_n \in B \mid \mathbf{L}_n, \dots, \mathbf{L}_1) = \mathbb{P}(\mathbf{L}_{n+1} \in B \mid \mathbf{L}_n).$$

A similar property holds for the processes $(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n)$ and $(\mathbf{Q}_n, \mathbf{L}_n)$.

This property is very important because it assures us that \mathbf{L}_{n+1} depends only on \mathbf{L}_n (besides \mathbf{R}_n and \mathbf{Q}_n), and not on the whole history \mathbf{W}_n . Thus, from the implementing point of view, it suffices to store only a vector containing \mathbf{L}_n , with a saving in terms of computer memory.

A final remark concerns the initial likelihood L , it can be estimated by testing a representative sample of subjects from the population, using, for example the expectation-maximization algorithm. In absence of information on that initial likelihood, we can use the uniform distribution setting:

$$\mathbf{L}_1(K) = \frac{1}{|\mathcal{K}|}.$$

5.3 The algorithm in details

In this section we will see in details the several parts of the stochastic assessment algorithm of Figure 1, but with the formalism seen above, we can already write the pseudocode. The algorithm requires a probabilistic knowledge structure, it uses a parameter ζ that we will see in subsection 5.3.2, and a data structure D for storing the question chosen on trial n , and relative answer.

Algorithm 1 Adaptive assessment algorithm.

Require: $Q, \mathcal{K}, \mathbf{L}_1$
 $n \leftarrow 1$
while $stopCondition(\mathbf{L}_n)$ **do**
 $question \leftarrow questioningRule(\mathcal{K}, \mathbf{L}_n)$
 ask $question$
 append in D $question$ and user's answer
 $\mathbf{L}_{n+1} \leftarrow updatingRule(D, \mathbf{L}_n, \mathcal{K}, \zeta)$
 $n \leftarrow n + 1$
end while
return the states k_i with the highest likelihood \mathbf{L}_n , their likelihood $\mathbf{L}_n(k_i)$ and D

The algorithm above is the core of the assessment procedure, but its result can be refined by a procedure of Bayesian update, whose input is composed by the output of algorithm 1 and by the vectors $\vec{\beta}, \vec{\eta}$, containing false negatives and false positives probabilities.

Algorithm 2 Adaptive assessment algorithm refined.

Require: $Q, \mathcal{K}, \mathbf{L}_1, \vec{\beta}, \vec{\eta}$
 $(A, \mathbf{L}, D) \leftarrow AdaptiveAssessment(Q, \mathcal{K}, \mathbf{L}_1)$
 $K_0 \leftarrow bayesianUpdate(A, \mathbf{L}, D, \vec{\beta}, \vec{\eta})$
return K_0

5.3.1 The questioning rule

The most simple idea for the item selection rule (or questioning rule) is to select, on any step n , a question q that partitions the set \mathcal{K} of all the states into two subsets \mathcal{K}_q and $\mathcal{K}_{\bar{q}}$ with a mass as equal as possible; that is, such that $\mathbf{L}_n(\mathcal{K}_q)$ is as close as possible to $\mathbf{L}_n(\mathcal{K}_{\bar{q}})$, so it needs to minimize the function $|\mathbf{L}_n(\mathcal{K}_q) - \mathbf{L}_n(\mathcal{K}_{\bar{q}})|$. But $\mathbf{L}_n(\mathcal{K}_{\bar{q}}) = 1 - \mathbf{L}_n(\mathcal{K}_q)$, it follows that any likelihood \mathbf{L}_n defines a set $S(\mathbf{L}_n) \subseteq Q$ containing all those questions q minimizing

$$|2\mathbf{L}_n(\mathcal{K}_q) - 1|,$$

or better

$$S(\mathbf{L}_n) = \underset{q}{\operatorname{argmin}}(|2\mathbf{L}_n(\mathcal{K}_q) - 1|). \quad (5.2)$$

The question in the set $S(\mathbf{L}_n)$ are then chosen with equal probability.

Definition 5.2 (half-split questioning rule). *This particular form of questioning rule defined as*

$$\Psi(q, \mathbf{L}_n) = \frac{\iota_{S(\mathbf{L}_n)}(q)}{|S(\mathbf{L}_n)|}$$

is called *half-split*.

Another method may be used [9], and is based on the evaluation of the entropy of the likelihood on a certain step n , that is

$$H(\mathbf{L}_n) = - \sum_{k \in \mathcal{K}} \mathbf{L}_n(k) \log_2 \mathbf{L}_n(k);$$

the goal is to select an item so as to reduce the expected entropy, on trial $n + 1$, as much as possible, that is to minimize the quantity

$$H'(q, \mathbf{L}_n) = \mathbf{L}_n(\mathcal{K}_q)H(u(1, q, \mathbf{L}_n)) + \mathbf{L}_n(\mathcal{K}_{\bar{q}})H(u(0, q, \mathbf{L}_n)),$$

over all possible $q \in Q$. Thus, the question asked a next step will be chosen in the set $J(\mathbf{L}_n) \subseteq Q$ that minimizes H' . The questioning rule specified by the equation

$$\Psi(q, \mathbf{L}_n) = \frac{\iota_{J(\mathbf{L}_n)}(q)}{|J(\mathbf{L}_n)|}$$

is called *informative*. We do not use this kind of item selection rule because is computationally more demanding. Algorithm 3 expresses the half-split questioning rule procedure.

Algorithm 3 questioningRule algorithm.

Require: $\mathcal{K}, \mathbf{L}_n$

$$S(\mathbf{L}_n) = \underset{q}{\operatorname{argmin}}(|2\mathbf{L}_n(\mathcal{K}_q) - 1|)$$

select randomly a question q in $S(\mathbf{L}_n)$

return q

5.3.2 The updating rule

We have make some considerations about the procedure of collecting the subject's responses, for example a correct answer to a multiple choice question may be due to a lucky guess, so it should not be given the same weight of a numerical response resulting from a computation. Moreover, a correct numerical response may signify the mastery of a question, but an error does

not necessarily imply complete ignorance [9]. Thus, the reallocation of the mass \mathbf{L}_n on trial $n + 1$ has to consider a parameter depending upon the question asked q and the corresponding response r . This considerations are necessary in knowledge context, but in psychological context the items are dichotomous, and there isn't any item more difficult of another, so in our case the parameter will be a constant.

Definition 5.3. *The updating rule is called multiplicative with parameters $\zeta_{q,r}$, where $1 < \zeta_{q,r}$ for $q \in Q, r = 0, 1$, if the function u of Axiom [U] satisfies the condition: with $\mathbf{Q}_n = q, \mathbf{R}_n = r$ and*

$$\zeta_{q,r}^K = \begin{cases} 1 & \text{if } \iota_K(q) \neq r \\ \zeta_{q,r} & \text{if } \iota_K(q) = r \end{cases} \quad (5.3)$$

we have

$$u_K(r, q, \mathbf{L}_n) = \frac{\zeta_{q,r}^K \mathbf{L}_n(K)}{\sum_{K' \in \mathcal{K}} \zeta_{q,r}^{K'} \mathbf{L}_n(K')}. \quad (5.4)$$

This rule has the property to be permutable, that is, given the pairs

- (1) $(\mathbf{Q}_{n-1} = q, \mathbf{R}_{n-1} = r), (\mathbf{Q}_n = q', \mathbf{R}_n = r')$,
- (2) $(\mathbf{Q}_{n-1} = q', \mathbf{R}_{n-1} = r'), (\mathbf{Q}_n = q, \mathbf{R}_n = r)$,

on trials $n - 1$ and n , and the same likelihood \mathbf{L}_{n-1} , the likelihood \mathbf{L}_{n-1} has to be the same in these two cases because the two pairs convey the same information.

Another kind of updating rule is the convex with parameters $\theta_{q,r}$ updating rule, where $0 < \theta_{q,r} < 1$, and

$$u_k(r, q, \mathbf{L}_n) = (1 - \theta_{q,r}) \mathbf{L}_n(K) + \theta_{q,r} g_K(r, q, \mathbf{L}_n),$$

with

$$g_k(r, q, \mathbf{L}_n) = \begin{cases} r \frac{\mathbf{L}_n(K)}{\mathbf{L}_n(\mathcal{K}_q)}, & \text{if } K \in \mathcal{K}_q \\ (1 - r) \frac{\mathbf{L}_n(K)}{\mathbf{L}_n(\mathcal{K}_{\bar{q}})}, & \text{if } K \in \mathcal{K}_{\bar{q}}. \end{cases}$$

But this rule is not permutable [9], and moreover, more computationally demanding than the multiplicative one. For these reasons we implemented the multiplicative questioning rule, see algorithm 4. Other kinds of updating rules exist in literature, similar situations are reviewed in [56, 57, 58].

Multiplicative updating rule can be seen as a sort of Bayesian update, for the proof and further details see [9].

Algorithm 4 updatingRule algorithm.

Require: $D, \mathbf{L}_n, \mathcal{K}, \zeta$

extract the last response r from D

for all K in \mathcal{K} **do**

 compute $\mathbf{L}_{n+1}(K) = u_K(r, q, \mathbf{L}_n)$ using equation 5.4

end for

return \mathbf{L}_{n+1}

5.3.3 The proof of correctness

All the theoretic framework is provided with a proof of correctness of the algorithm 1, that assures us that the stochastic assessment procedure can uncover the latent state K_0 . That is, this result prevents the possibility of infinite loops, without the convergence of $\mathbf{L}_n(K_0)$ to 1.

Theorem 5.2. *Let be $(\mathbf{R}_n, \mathbf{Q}_n, \mathbf{L}_n)$ a stochastic assessment procedure parametrized by u , Ψ and K_0 , with u either convex or multiplicative, and Ψ half-split. Then, K_0 is uncoverable in the sense that*

$$\mathbf{L}_n(K_0) \rightarrow 1.$$

Notice that this theorem does not involve a stop criterion, i.e. it does not give any information about the number of steps n , it simply states $n \not\rightarrow \infty$, and furthermore, the refinement of the assessment with $\vec{\beta}$ and $\vec{\eta}$, is not considered in the proof. Moreover, a similar result also exists for the informative questioning rule.

5.3.4 The stop condition

The stop criterion is based on the evaluation of the likelihood $\mathbf{L}_n(K)$ for every $K \in \mathcal{K}$. For example, the system can return the states with the associated likelihood bigger than a certain threshold, or the system can measure the entropy of the knowledge structure at every step, and when the entropy is smaller than a certain values, it returns the state with higher likelihood. In our case we based on the experimental data, we noticed that as soon as a certain likelihood $\mathbf{L}_n(K)$ gets over the value 0.7, there was only a state with that likelihood, K , and the other likelihoods were all under the value 0.1, thus not very informative.

5.3.5 Refining the assessment

Algorithm 1 returns a set of states A having the bulk of probability mass \mathbf{L}_n , and a data structure D containing pairs of question and relative response. From D it is possible to extract all the questions whose response is correct, that is, the response pattern R . The pattern R , along with the vectors $\vec{\beta}, \vec{\eta}$,

permits to refine the result of the assessment using a Bayesian heuristic, notice that such a computation makes sense only when good estimates for β_q, η_q are available, that is, if the model has a good fit with the data.

The procedure proceeds as follows, the first step is to compute the a posteriori probability of having pattern R considering the subject in state $K_i \in A$:

$$P(R | K_i) = r(R, K_i)$$

where $r(R, K_i)$ is the response function of a probabilistic knowledge structure, whose computation is given by the BLIM with the formula 3.6. The second step consists of a Bayesian rule to recompute the a posteriori probabilities of the states according to the response pattern R :

$$P(K_i | R) = \frac{P(R | K_i) \mathbf{L}_n(K_i)}{\sum_{j=1}^{|A|} P(R | K_j) \mathbf{L}_n(K_j)} \quad (5.5)$$

Algorithm 5 shows the procedure of this Bayesian update.

Algorithm 5 bayesianUpdate algorithm.

Require: $A, \mathbf{L}, D, \vec{\beta}, \vec{\eta}$

extract R from D

for all $K_i \in A$ **do**

 compute $P(R | K_i) = r(R, K_i)$ using equation 3.6

end for

for all $K_i \in A$ **do**

 compute $P(K_i | R)$ using equation 5.5

end for

return the state K with the highest a posteriori likelihood $P(K | R)$

Chapter 6

Developing AAS-PD

In the previous chapter we saw the mathematical formalization of an adaptive assessment algorithm based on KST. From such algorithm we developed our assessment system, this is the topic of the present chapter. Our aim was to realize a first working prototype with the purpose of studying its behaviour and making decision about next interventions. That is, such a system was totally new in clinical psychology, so we preferred to draw conclusions from the system behaviour observation, thus adopting the *bottom up* approach. For this reason we chose Python 2.7 as programming language, its powerful libraries and built-in objects allows a rapid prototype development, moreover if any efficiency problems arose it would be possible to include faster C++ extensions [59]. Before describing the system architecture and first implementative considerations we spend few words about Python.

6.1 The Python programming language

Python [60, 61] was born in the late eighties and its implementation started in December 1989 by Guido van Rossum at CWI in the Netherlands. It is a general-purpose, powerful dynamic, interpreted high-level programming language whose design philosophy emphasizes code readability. The language presents a syntax very clear and readable, strong introspection capabilities on its objects, very high level dynamic data types, and an automatic memory management. Moreover, Python is a multi-paradigm programming language, it does not force programmers to adopt a particular style of programming, but it permits a variety of styles: from object-oriented programming to imperative programming passing through functional programming and logic programming.

One of Python's greatest strength is its large standard library, providing pre-written tools for many tasks, i.e. from asynchronous processing to zip files. This large library has found the expression "batteries included" to

describe such Python philosophy [60]. The modules of the standard library can be augmented with custom modules written in either C, C++ or Python.

Python is available for all major operating systems: Windows, Linux/Unix, OS/2 and Mac, among others. We developed AAS-PD on Windows7 o.s., using PyDev. PyDev is a plugin for the development environment Eclipse [62], that enables the latter to be used as a Python IDE.

6.2 ASS-PD architecture

After choosing the proper programming language we started by analysing algorithms 1 and 2 of the previous chapter, and asking how to implement them. Let's recall the algorithms:

Algorithm 6 Algorithm 1: the adaptive assessment algorithm.

Require: $Q, \mathcal{K}, \mathbf{L}_1$

$n \leftarrow 1$

while $stopCondition(\mathbf{L}_n)$ **do**

$question \leftarrow questioningRule(\mathcal{K}, \mathbf{L}_n)$

 ask $question$

 append in D $question$ and user's answer

$\mathbf{L}_{n+1} \leftarrow updatingRule(D, \mathbf{L}_n, \mathcal{K}, \zeta)$

$n \leftarrow n + 1$

end while

return the states K_i with the highest likelihood \mathbf{L}_n , their likelihood $\mathbf{L}_n(K_i)$ and D

Algorithm 7 Algorithm 2: the adaptive assessment algorithm refined.

Require: $Q, \mathcal{K}, \mathbf{L}_1, \vec{\beta}, \vec{\eta}$

$A, \mathbf{L}, D \leftarrow AdaptiveAssessment(Q, \mathcal{K}, \mathbf{L}_1)$

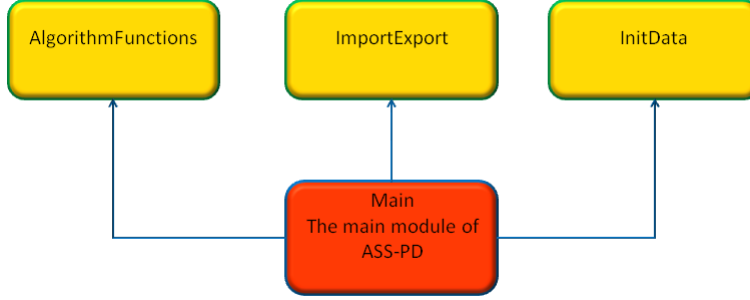
$K_0 \leftarrow bayesianUpdate(A, \mathbf{L}, D, \vec{\beta}, \vec{\eta})$

return K_0

We saw that they are composed by four important functions, the *questioningRule* function, the *updatingRule*, the *stopCondition*, and the *bayesianUpdate*. It is also possible, for simplicity, to define a function that collects the user's responses, called *responseRule* [9]. Moreover, the algorithm requires as input a probabilistic knowledge structure and the vectors of the false negatives and false positives probabilities. Thus, it is reasonable to build a software module, in the specific a Python module, managing the algorithm functions, a module dedicated for the import/export functions and a main module that using the previous modules, imports the data, executes the algorithm, and displays/exports the data. Moreover, it is necessary a

module for initializing the data structures with the imported data. Figure 6.1 shows AAS-PD architecture, an arrow from a bloc A to a bloc B means that module A calls some functions of module B .

Figure 6.1: AAS-PD architecture.



6.3 Considerations on the complexity algorithm

After the analysis of the system architecture, the second step consists in the evaluation of the complexity of algorithm. We were interested in the running time of the algorithm, so we introduced a *cost model* that allowed us to analyse each function running time and then summing up all the several times.

Formally a computational problem Π is a relation between a set \mathcal{I} (the set of instances) and a set \mathcal{S} (the set of solutions) [63], in our case a typical instance $i \in \mathcal{I}$ is the quintuple $(Q, \mathcal{K}, \mathbf{L}_1, \vec{\beta}, \vec{\eta})$, and a typical solution $s \in \mathcal{S}$ is the set K_0 . We considered as the size of an instance the cardinality of the clinical structure: $|\mathcal{K}|$. The crucial operations, of our cost model, are the arithmetic sums and the belonging of an element to a set, i.e. given a set A and an element x , if $x \in A$. These operations have cost 1, while all other operations have cost 0.

The questioning rule

The function *questioningRule* handles the computation of equation 5.2:

$$S(\mathbf{L}_n) = \underset{q}{\operatorname{argmin}}(|2\mathbf{L}_n(\mathcal{K}_q) - 1|),$$

and the simplest algorithm performing this task is the following:

Algorithm 8 Simple argmin algorithm

Require: $Q, \mathcal{K}, \mathbf{L}_n$
for all q in Q **do**
 $\mathbf{L}_n(\mathcal{K}_q) \leftarrow 0$
 for all K in \mathcal{K} **do**
 if $q \in K$ **then**
 $\mathbf{L}_n(\mathcal{K}_q) \leftarrow \mathbf{L}_n(\mathcal{K}_q) + \mathbf{L}_n(K)$
 end if
 end for
 $list[q] \leftarrow |2\mathbf{L}_n(\mathcal{K}_q) - 1|$
end for
return $S = \underset{q}{\operatorname{argmin}} list$

Thus, for every $q \in Q$, and for every $K \in \mathcal{K}$ we have to test if $q \in K$, totally we have $|\mathcal{K}| - 1$ operations (we do not test the empty set). If $q \in \mathcal{K}$, we have to sum all the likelihoods of the states containing q , we do not know a priori how many states contain q , so we consider the quantity $m_q = |\mathcal{K}_q|$, that represents the number of states containing item q . So, the number of sums, for every q , is $m_q - 1$. Hence the total number of operations become:

$$\sum_{q \in Q} (|\mathcal{K}| - 1 + m_q - 1) = |Q|(|\mathcal{K}| - 2) + \sum_{q \in Q} m_q.$$

Let's introduce a new parameter, l , that is a sort of "length" of a knowledge structure: $l = \sum_{K \in \mathcal{K}} |K|$. The term "length" derives from the fact that if we write every element of every knowledge state $K \in \mathcal{K}$ consequently, the total number of elements written is l . Now, $\sum_{q \in Q} m_q = l$, the proof is very simple, so we can rewrite the number of performed operations as $T_{que}(|\mathcal{K}|) = |Q|(|\mathcal{K}| - 2) + l$ for every step n . This function will be executed at every trial n , so we are interested in speeding up this operation. We can use an auxiliary data structure K_q , a dictionary with pairs containing keys and values. For every pair the key is an item $q \in Q$, and the value is a list containing all the states K such as $q \in K$. This new data structure can be initialized once with a total cost of $T_{split}(|\mathcal{K}|) = l$ with a procedure called *split* and showed in algorithm 9.

Algorithm 9 Splitting algorithm

Require: \mathcal{K}
for all state in \mathcal{K} **do**
 for all item in state **do**
 append state in $Kq[item]$
 end for
end for

So the new *argmin* procedure can be computed summing up, for all $q \in Q$, $\mathbf{L}_n(K)$ for all $K \in Kq[q]$:

Algorithm 10 New *argmin* algorithm

Require: $Q, \mathcal{K}, \mathbf{L}_n, Kq$
for all q **in** Q **do**
 $\mathbf{L}_n(\mathcal{K}_q) \leftarrow 0$
 for all K **in** $Kq[q]$ **do**
 $\mathbf{L}_n(\mathcal{K}_q) \leftarrow \mathbf{L}_n(\mathcal{K}_q) + \mathbf{L}_n(K)$
 end for
 $list[q] \leftarrow |2\mathbf{L}_n(\mathcal{K}_q) - 1|$
end for
return $S = \underset{q}{\operatorname{argmin}list}$

With a total amount of operations $T'_{que}(|\mathcal{K}|) = \sum_{q \in Q} (m_q - 1) = l - |Q|$ for every step n . We do not have to test if $q \in K$ for every $K \in \mathcal{K}$, and thus the new data structure lowers the number of operations.

Our analysis does not include the last operation $S = \underset{q}{\operatorname{argmin}list}$, because the operations as comparisons between real numbers are not included in our cost model, we do not consider as crucial operations, and they do not affect the running time.

The updating rule

The *updatingRule* function simply computes the functions 5.4, 5.3:

$$u_K(r, q, \mathbf{L}_n) = \frac{\zeta_{q,r}^K \mathbf{L}_n(K)}{\sum_{K' \in \mathcal{K}} \zeta_{q,r}^{K'} \mathbf{L}_n(K')}.$$

where

$$\zeta_{q,r}^K = \begin{cases} 1 & \text{if } \iota_K(q) \neq r \\ \zeta_{q,r} & \text{if } \iota_K(q) = r. \end{cases}$$

Thus, for a given question q , we have to test if $q \in K$ (one crucial operation) for every $K \in \mathcal{K}$, and then compute the simple summation of equation 5.4 ($|\mathcal{K}| - 1$ crucial operations). The total number of operations is then $T_{upd}(|\mathcal{K}|) = (1 + |\mathcal{K}| - 1) = |\mathcal{K}|$ for every step n .

Instead, if we use the Kq data structure we do not have to test if a given $q \in K$ for all $K \in \mathcal{K}$, leading to a running time $T'_{upd}(|\mathcal{K}|) = |\mathcal{K}| - 1$ for every step n . In this case we have a slight reduction of the number of operations performed.

The stop criterion

The *stopCondition* function stops the assessment when no further information can specify better the latent state K_0 . It is possible to choose the states whose likelihood is greater of a certain threshold, or it is possible to use the measure of the entropy:

$$H(\mathbf{L}_n) = - \sum_{k \in \mathcal{K}} \mathbf{L}_n(K) \log_2 \mathbf{L}_n,$$

which requires a running time $T_{stop}(|\mathcal{K}|) = |\mathcal{K}| - 1$ of computation.

The Bayesian update

The *bayesianUpdate* function refines the assessment at the end of the while loop of algorithm 1. It is composed by the formula 3.6 and the summation of equation 5.5. The summation has $|A| - 1$ sum operations. Let's recall formula 3.6:

$$r(R, K) = \left(\prod_{q \in K \setminus R} \beta_q \right) \left(\prod_{q \in K \cap R} (1 - \beta_q) \right) \left(\prod_{q \in R \setminus K} \eta_q \right) \left(\prod_{q \in \overline{R \cup K}} (1 - \eta_q) \right),$$

it is not possible to compute the running time of the computation of this formula, because we do not know a priori the subject's response pattern, moreover the multiplications are not included in our cost model, so this running time is 0. This assumption about the cost model is not too strong, because the core of the computation is made of summations in the while loop of algorithm 1, and the Bayesian update occurs only once. Furthermore, the the number of states to be refined, $|A|$, can be very low, it depends on the stop criterion, so the quantity $|A| - 1$ of sum operations, does not affect very much the overall algorithm complexity. For these reasons, we can leave out the running time of *bayesianUpdate* function from the overall running time.

The whole algorithm

Now we simply sum up all the running times found up to now, and we obtain the overall running time of the algorithm. We consider the running times using the data structure Kq because they are smaller:

$$\begin{aligned} T_{tot}(|\mathcal{K}|) &= N(T_{stop}(|\mathcal{K}|) + T'_{que}(|\mathcal{K}|) + T'_{upd}(|\mathcal{K}|)) + T_{split}(|\mathcal{K}|) = \\ &= N(|\mathcal{K}| - 1 + l - |Q| + |\mathcal{K}| - 1) + l = N(2|\mathcal{K}| - |Q| + l - 2) + l = \\ &= O(N(l + |\mathcal{K}|)), \end{aligned}$$

where N is the last step of the algorithm, i.e. the total number of question posed.

6.4 The implemented modules

Next step of AAS-PD development consists in implementing the modules of the system architecture of Figure 6.1 with Python modules. We performed this “translation” directly, an architecture module has been implemented by a Python module. Python modules are not simply classes, but files containing data structures, functions, classes and whatever necessary Python object. Nevertheless this difference between modules and classes we present, only for simplicity, Python modules as UML class diagrams [64].

6.4.1 The AlgorithmFunctions module

This is the most important Python module of AAS-PD, it is the core of the whole system, because it contains the data structures and the functions on which algorithm 2 is grounded, see Figure 6.2.

Figure 6.2: The AlgorithmFunctions module.

AlgorithmFunctions
+Q: dictionary
+K: list
+Kq: dictionary
+L: dictionary
+D: list
+beta: list
+eta: list
+Att: dictionary
+attMap: dictionary
+stopCondition(L)
+questioningRule(K, L)
+responseRule(item, D)
+updatingRule(D, L, K, zeta)
+bayesUpd(A, L, D, beta, eta)
+blim(responsePatt, state)
+returnAtt(latentState, attMap)

A direct translation occurs also between the algorithm formalization and its implementation, the data structure of this module are exactly the same of algorithm 2, plus two new data structures needful in our psychological context:

- Q is a simple Python dictionary representing the domain Q . A pair key-value contains the index of an item and its text in string format.
- K represents the clinical structure \mathcal{K} . It is a Python list of clinical states, each of them is a list containing all the item indexes forming that state. Every clinical state has an index, i.e. its position in the list.
- Kq represents the auxiliary data structure Kq . Its a Python dictionary, A pair key-value contains an item index, i , and the values are lists containing the indexes of the states including i as element.

- \mathbf{L} , for all states $K \in \mathcal{K}$ represents the vector of likelihoods $\mathbf{L}(K)$ of the whole history of the process: from step 1 to step n . It is a Python dictionary, a key represents the index of a state and the corresponding value is a list containing $\mathbf{L}_i(K)$, for $i = 1 \dots n$. With this Python data structure it is possible to performing operations on the likelihood $\mathbf{L}(K)$, as a summation for the updating or the entropy evaluation at trial n , with only a few lines code:

```
for state in L.keys():
    #execute an operation on L[state][n]
```

- \mathbf{D} represents the data collected D . It is a simple Python list of pairs (tuples). Each pair contains an item index and the subject's response to the item. As we did not need to perform operations on this structure we chose a simple list instead of a more complex dictionary.
- \mathbf{beta} , \mathbf{eta} are two Python lists containing the false negatives and false positives probabilities.
- \mathbf{Att} is a Python dictionary representing the diagnostic attributes. A pair key-value contains the index of an attribute and its description in string format.
- \mathbf{attMap} is a Python dictionary representing the cross-table (or skill map) between items and attributes inferred. Keys contain the indexes of the items, while the values are lists of the attributes inferred by the items.

A remark here is needed, in theorem 5.1 we affirmed that the stochastic process (\mathbf{L}_n) is Markovian with the consequent advantage to store only the vector \mathbf{L}_n , to compute \mathbf{L}_{n+1} . But our data structure \mathbf{L} contains all the history \mathbf{L}_i , for $i = 1 \dots n$. This decision is taken for studying purposes, in this phase of the work we were interested in analysing the system behaviour through the whole likelihood history of every state, during the trials $i = 1 \dots n$. Future developments of AAS-PD may delete the history of (\mathbf{L}_n) , keeping only one list.

After the presentation of the data structures we show the Python functions of Figure 6.2. As in precedence, the translation between the formal model and the implementation is simply and direct. We will do not enter into details, they implement the core algorithm functions already seen, with the algorithms outlined in section 6.3:

- $\mathbf{stopCondition}(\mathbf{L})$ implemented the the criterion of the likelihood greater of a certain threshold, this at the beginning of our work. This threshold was detected by experimental data, we noticed that a likelihood of 0.7 was enough to identify correctly the latent state K_0 . In

a second stage we adopted the entropy criterion, noticing that an entropy smaller than 1 does not carry further information for uncovering K_0 .

- `questioningRule(K, L)` is the Python function that, using data structure `Kq`, implements the *questioningRule* mathematical function, according algorithm 10.
- `responseRule(item, D)` is the Python function that asks an item and then stores, in data structure `D`, the item asked and the subject's response.
- `updatingRule(D, L, K, zeta)` updates the likelihoods `L[state][n]` for next trial $n + 1$, implementing the *updatingRule* mathematical function. A remark here is necessary, in an OCD questionnaire the possible answers are only "True" or "False", and the questions have the same weight, so the parameter $\zeta_{q,r}$ does not depend from q and r , we considered it a constant. Furthermore, AAS-PD gives the user the possibility to enter a value for ζ .
- `bayesUpd(A, L, D, beta, eta)` is the Python function that refines the assessment according to equation 5.5, and using the formula 3.6.
- `blim(responsePatt, state)` is the Python function that implements formula 3.6. It is called by `bayesUpd(A, L, D, beta, eta)` function.
- `returnAtt(latentState, attMap)` is the function that, given a latent state K_0 , computes (and returns) all the attributes inferred by all the items of K_0 . It performs a closure under union of the values of `attMap[item]` for every `item` in K_0 , see equation 4.2.

A final comment regards the initial likelihood \mathbf{L}_1 , we chose the uniform probability distribution, $\mathbf{L}_1 = 1/|\mathcal{K}|$, because it formulates less a priori assumptions, it excludes nothing, indeed, it gives us the maximum entropy at the start of the assessment.

6.4.2 The ImportExport module

This Python module displayed in Figure 6.3 contains functions for importing the data, displaying to terminal output and exporting in a file.

The data to import are the probabilistic clinical structure $(Q, \mathcal{K}, \mathbf{L}_1)$, the vectors $\vec{\beta}$, $\vec{\eta}$, and the attributes map between items and attributes. They are in csv format file, and the procedure `importData(Q, K, beta, eta, Att, attMap)` imports them in the system initializing the data structures `Q, K, beta, eta, Att, attMap`. Moreover, it implements procedures for displaying the data structure to terminal output, for debugging or studying the system

Figure 6.3: The `ImportExport` module.

ImportExport
<pre>+importData (Q,K,beta,eta,Att,attMap) +printQ(Q) +printK(K) +printKq(Kq) +printL(L) +exportL(L)</pre>

behaviour purposes. Finally, the `exportL(L)` procedure exports the whole dictionary `L` in a `csv` format file for analysing the whole likelihood history.

6.4.3 The `InitData` module

This module, shown in Figure 6.4, simply initializes dictionary `L` to a uniform distribution probability, using function `initL(L, sizeK)`. Moreover, it provides function `split(K, Kq)` that implements algorithm 9, for initializing the data structure `Kq`.

Figure 6.4: The `InitData` module.

InitData
<pre>+initL(L, sizeK) +split(K, Kq)</pre>

6.4.4 The main module

This module starts the execution of AAS-PD, providing a textual interface which the users can interact with the system. After importing the data and initializing the data structures, this module implements an alternative version of algorithm 2, calling the functions of the modules seen above. The alternative version uses the Bayesian update also for the stop criterion. After updating the likelihoods \mathbf{L}_n in \mathbf{L}_{n+1} the system performs the Bayesian update on them (in this case $\mathbf{A}=\mathbf{K}$), creating a new temporary list `tmpL`. If such likelihoods list satisfies the stop criterion the system stops and returns the latent state. Otherwise, the system follows to selecting a new question based on \mathbf{L}_{n+1} . In such a manner the false negative and false positives probabilities concurs to stop the assessment.

Finally, the system displays the uncovered latent state, in the form of a list of items, and its corresponding items.

6.5 An example of assessment

Let's show an example of the assessment performed by AAS-PD. We use the Doubting-Ruminating clinical structure of Figure 6.5 because it is very

simple to follow the passages.

Figure 6.5: The concept lattice of the Doubting-Ruminating sub-scale.

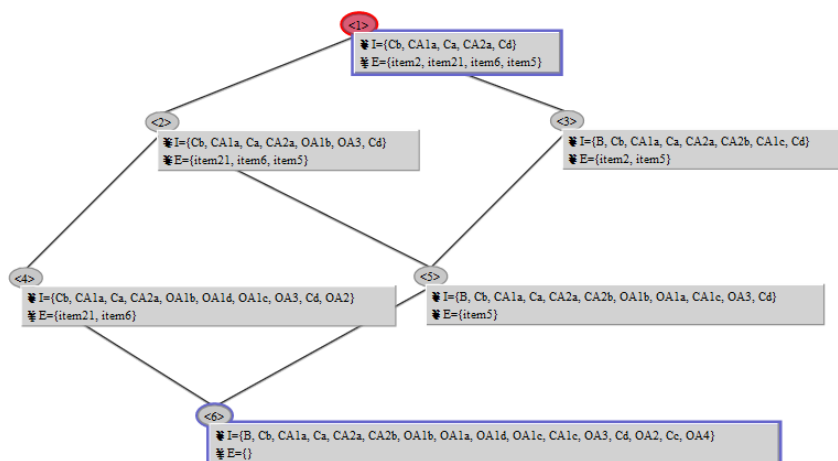


Table 6.1 shows initial likelihood of every state at step 1, it is an uniform distribution probability. The system selects the more discriminative items, that partition the clinical structure in two subset of states having the mass probability as close as possible, i.e. see equation 5.2. In this case they are the set {6, 21}.

Table 6.1: Initial likelihood of the states.

L	[]	[5]	[2, 5]	[6, 21]	[5, 6, 21]	[2, 5, 6, 21]
L ₁	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667

Let’s suppose the system randomly chooses item 6: “I frequently have disagreeable thoughts and I cannot get rid of them”, and suppose that the users answers “True”, the system now increases the likelihood of the states containing item 6 and decreases the others, see Table 6.2. As we can see the likelihoods do not satisfy the threshold stop criterion, and if the system performs the Bayesian update with the response pattern observed, [6], it does not satisfy the entropy stop criterion. So the next more discriminative items are item 2 and item 5.

AAS-PD is now at trial 2 and it selects randomly item 2: “Usually, I have serious doubts about simple things I do every day”, and let’s suppose that the users answers again “True”, the system now increases the likelihood of the states containing item 2 and decreases the others, see Table 6.3.

Table 6.2: Likelihoods of the states at trial 2.

L	[]	[5]	[2, 5]	[6, 21]	[5, 6, 21]	[2, 5, 6, 21]
L₂	0.0278	0.0278	0.0278	0.3056	0.3056	0.3056

Table 6.3: Likelihoods of the states at trial 3.

L	[]	[5]	[2, 5]	[6, 21]	[5, 6, 21]	[2, 5, 6, 21]
L₃	0.0064	0.0064	0.0705	0.0705	0.0705	0.7756

We can see that the likelihood of state [2, 5, 6, 21] is over 0.7 satisfying the first stop criterion. Moreover, if the system performs the Bayesian update, on pattern [6, 2] and then calculates the entropy it obtains a value of 0.1883. Thus, the system returns as output the state [2, 5, 6, 21], with attributes {Cb, CA1a, Ca, CA2a, Cd}, where:

- CA1a: repetitive behaviours or constrained mental acts;
- CA2a: behaviours are designed to reduce or prevent discomfort;
- Ca: marked discomfort;
- Cb: waste of time;
- Cd: interference with social and working life.

Let's analyse the output, the users responded only to questions 6 and 2 and the system inferred, in a probabilistic way, that the clinical state is [2, 5, 6, 21]. That is, every one that respond to item 6 and 2 automatically will respond positively to the other questions. This strong inference is the core of assessment algorithms based on clinical structures.

6.6 Toward a standalone system

A psychologist interested in AAS-PD should download Python 2.7, install it on his personal computer, copying the Python modules of AAS-PD on a computer folder and then execute the program. This task is too difficult, and could be frustrating for a normal computer user. For this reason we aim

at building a system installable on any personal computer. This can be done using third-party tools, indeed Python code can be packaged into standalone executable programs.

We used py2exe [65], a Python extension which converts Python scripts into executable Windows programs, able to run without a Python installation. It needs to know if the program uses any external DLLs or if does dynamic imports of modules, in these cases it is necessary to explicitly declare them to py2exe.

Furthermore, we combined py2exe with EclipseNSIS, an Eclipse plugin, in order to have a self-installing Windows program.

Testing AAS-PD

In the previous chapters we presented the theoretical background and the developing process of AAS-PD. The next step will be to test whether our system respects the requisites, namely, if it converges to the correct latent clinical state for every response pattern, and if it is affected by any possible bug or bias. Moreover, we will discuss the developed system and suggest some future proposals.

7.1 Method

In order to evaluate the developed system we conducted some simulations of assessment. The simulations were based on a sample of 4324 subjects taken from normal population. The subjects answered to all questions of sheet number nine of CBA 2.0 in a non adaptive way, that is, they responded to all items sequentially. The data collected were in the form of a database where each row represented a subject and each column represented one item of Cleaning and Checking sub-scale. An entry with the number 1 corresponds to a positive answer of a given subject to a certain item, an entry with number 0 corresponds to the opposite situation.

Before going into the details we give some specifications about the language used in this section. A *response pattern* is a list of items, for example [1, 4, 7], containing the indexes of items that a subject responded “True”. Different subjects can have the same response pattern to a certain sub-scale. The set \mathcal{R} includes all the different response patterns, i.e. it lists once each pattern independently from its frequency in the sample. We say that a response pattern R is *assigned* to a state K if our system outputs K with input R . Moreover, we need a method to measure the similarity between response patterns and states, thus we assign to every response pattern a *distance* $d(R, K)$ to a state K , this measure is their symmetric distance:

$$d(R, K) = |(R \setminus K) \cup (K \setminus R)|,$$

and $d(R, K) = 0$ means that R corresponds to the knowledge state K . We say that a response pattern R has a *minimum distance* to a state K , if does not exist a state K' such as $d(R, K') < d(R, K)$.

The goal of the simulations was to test whether the system worked properly, i.e. if the subjects with a response pattern identical to a clinical state in the structure, are assigned to that state. Moreover, we wanted to study the behaviour of the system in presence of patterns that do not correspond to a state. In this step of the work we were much more interested in the system efficacy rather than in its efficiency, so we tested the system with small sized clinical structures, and we obtained an obvious real time system response between the subject's answer and the next question posed.

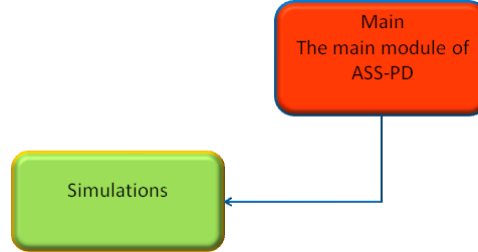
In this perspective our aim was to gather the following information:

1. for every state $K \in \mathcal{K}$:
 - (a) the number of response patterns assigned to K ;
 - (b) the response patterns assigned to K ;
 - (c) for every response pattern R assigned to K , their symmetric distance $d(R, K)$;
2. whether a response pattern R identical to a state K is assigned to that state;
3. for every distance $d(R, K)$ the number of response patterns having that distance from the assigned state K ;
4. whether $d(R, K)$ is minimum for every R with $d(R, K) > 1$. In the opposite case:
 - (a) for every distance $d(R, K)$ the number of response pattern with non minimum distance.
5. the arithmetic and weighted means of the distances of every response pattern from its assigned K ;
6. the arithmetic mean of the number of questions posed for every response pattern R .

These requests concern with system efficacy. Condition 2 is a sort of correctness verification of our system, that is, if the implemented system effectively respects theorem 5.2. Condition 4 concerns with the distribution of response patterns that are not states.

In order to perform the simulations and obtain the requested informations we introduced in our system architecture a new Python module managing this task, the `simulation` module. It includes functions for the data import of the subjects, a class representing a simulation step, a class representing

Figure 7.1: Inclusion of the new Python module for simulations.



the subject and functions aimed to extract the listed informations, see Figure 7.1.

The simulations proceed as follows, the system imports the data, stored in a `csv` format file, with an import procedure that organizes the data in a list of objects, every object represents a subject with its code, its response pattern for every sub-scale and others informations to be filled. The user selects the sub-scale, and the system loads the corresponding clinical structure. In this case we used a uniform initial likelihood, $\mathbf{L}_1 = 1/|\mathcal{K}|$, because it formulates less a priori assumptions, so we preferred using this setting in testing phase. For every item of the subjects list the `main` module performs a simulation, extracting the response pattern R , asking the maximally informative question, answering automatically, updating the likelihood till it uncovers the latent state K_0 . Now the system has completed the simulation, then it calculates the distance $d(R, K_0)$ and stores an object representing this simulation in a list. At the end of the loop the `main` module uses some functions of the `simulation` module in order to extract the informations listed above.

7.2 Results

Table 7.1 lists some simulations results for the Cleaning and Checking sub-scales. We can immediately notice that the considered sub-scales contain the same number of items, so they are both a collection of 256 possible total states. Thus, it is simpler to make comparisons between them. We present, and discuss, the results in subsections according to the six conditions listed above, and for every subsection, comparing the considered sub-scales.

7.2.1 Condition 1, results about the assigned states

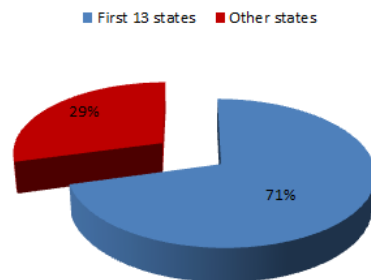
We start presenting the states with the higher number of response patterns assigned. It is not possible to list all the states and their frequencies for every sub-scale, so, we concentrate on a subset of them, the subset which covers about the 70% of the response patterns.

Table 7.1: Simulations results of the Cleaning and Checking sub-scale.

	Cleaning	Checking
Number of items	8	8
Number of states	92	20
$ \mathcal{R} $	220	185
Response patterns not satisfying condition 2	0	0
Number of response patterns with non minimum $d(R, K)$	0	392
$d(R, K) = 0$	3223	1581
$d(R, K) = 1$	1023	1517
$d(R, K) = 2$	78	861
$d(R, K) = 3$	0	337
$d(R, K) = 4$	0	28
Arithmetic mean of $d(R, K)$	0.27	1.01
Weighted mean of $d(R, K)$	0.72	1.69
Arithmetic mean of the number of questions posed	6.90	4.37

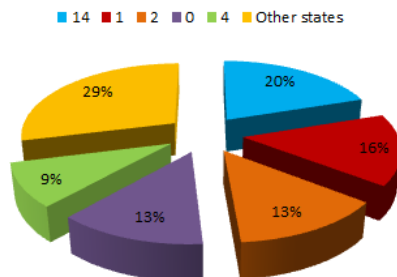
In the Cleaning case we have that the first 13 states with higher number of assigned response patterns covers the 71% of all the response patterns, see Figure 7.2. This number represents the 14% of the states of the whole structure.

Figure 7.2: Cleaning sub-scale: response patterns distribution according to the states, every slice represents a set of states.



In the Checking case we have that the 5 most frequently assigned states cover the 71% of all the response patterns, see Figure 7.3. This number represents the 25% of the states of the whole structure. Thus, a smaller number of states of the Cleaning structure, gives the same percentage of response patterns of the Checking structure. This datum needs further investigations.

Figure 7.3: Cleaning sub-scale: response patterns distribution according to the states, every slice represents a state or a set of states. The number assigned to each color is the index of a state.



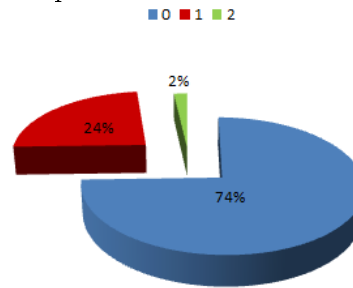
7.2.2 Condition 2, results about the response patterns that are states

As we expected, for every sub-scale all the response patterns corresponding to a state K are assigned to K . This is a good verification of the system correctness.

7.2.3 Condition 3, results about the distances

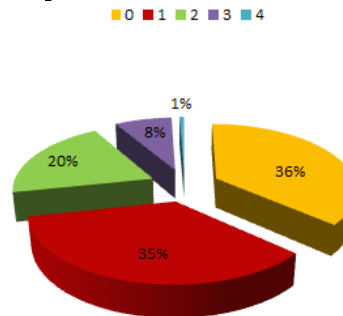
In the Cleaning sub-scale there are some response patterns not included in the clinical structure, but their distances $d(R, K)$ are at most 2. The 78 subjects with $d(R, K) = 2$ can be considered critical subjects (who could have intentionally faked the answers) and represent only the 2% of the sample, as we can see in the pie chart of Figure 7.4. While the bulk of the subjects, the 74%, is gathered at distance $d(R, K) = 0$.

Figure 7.4: Cleaning sub-scale: response patterns distribution according to the distances, every slice represents a distance.



Let's consider the situation for Checking sub-scale, it presents a critical situation, the number of response patterns with $d(R, K) > 1$ is high, about the 64%, and the distances measure is up to 4. Figure 7.5 depicts such a situation, we can notice that the patterns with $d(R, K) = 0$ are only the 36%.

Figure 7.5: Checking sub-scale: response patterns distribution according to the distances, every slice represents a distance.

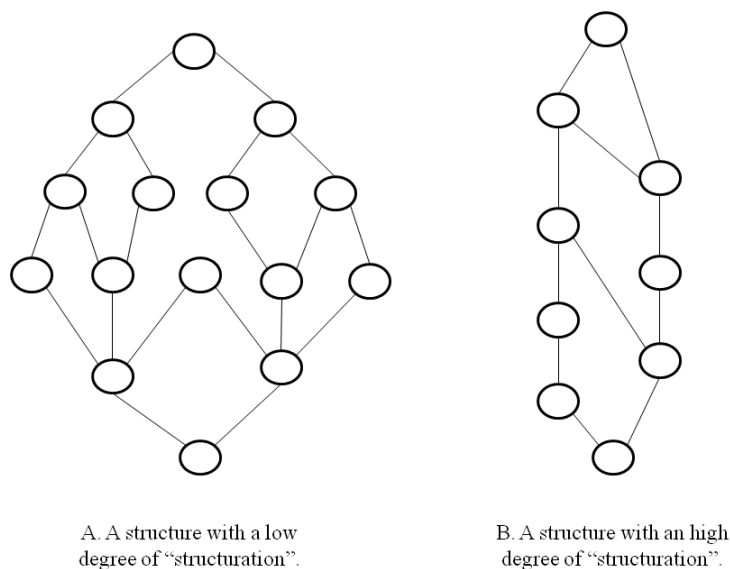


This fact can be explained looking at the number of states of the structure, there are only 20 clinical state and 185 patterns (the cardinality of \mathcal{R}), so the states cover only the 11% of the patterns. This structure could be affected by an over-fitting problem: it is a good representation of the structure concerning subjects affected by OCD, but it does not have enough states to fit normal population. It could be interesting to reformulate the structure

using alternative methods, such as the extraction of the structure from the response patterns [66].

A greater or smaller cardinality of the states is directly linked to a property of such structures, the “structuration” of a structure. With abuse of language we say that a structure (or ordered set) is “more structured” if less states are in the structure. i.e. if it is more similar to the total order. In our case, the Cleaning structure has 92 states over 256 total possible states, it is similar to figure 7.6A, while the Checking structure has only 20 state over 256, see Figure 7.6B. Thus, our Checking structure is more structured than the Cleaning one, this fact explains the higher percentage of response patterns with distance $d(R, K) > 0$.

Figure 7.6: Diverse kinds of “structuration” of the ordered sets



At first glance, the Cleaning structure seems better performing, but if we consider the response patterns with $d(R, K) = 0$, in the Cleaning case we have 35 response patterns per state, while in the Checking case we have 79 response patterns per state. The latter structure is more performing for the response patterns corresponding to clinical states. So a possible reformulation should enlarge the structure keeping the states corresponding to response patterns.

7.2.4 Condition 4, results about response patterns with non minimum distances

An important result of Cleaning sub-scale, is that every response pattern with $d(R, K) > 1$ is assigned to its nearest state, that is the state with

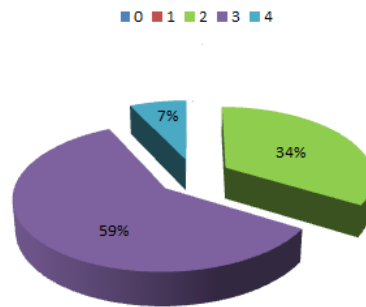
minimum distance. Since the system cannot end in a state not included in the clinical structure, it is supposed to assign such pattern to the nearest state in terms of distance.

Passing to Checking case, we notice that 392 response patterns are assigned to a state with non minimum distance, table 7.2 shows the distribution of these 392 response patterns according to the distances, while figure 7.7 gives a graphic summary of the matter. This result could be explicated as the combination of smaller cardinality of the Checking structure and the algorithm behaviour. Indeed, the system asks question in an adaptive way, an finally it infers a clinical state containing also questions inferred and not observed, see section 6.5. So, AAS-PD makes inferences based on a small number of states, and this behaviour could be a plausible explanation of the different results between the sub-scale considered.

Table 7.2: Number of subjects with non minimum distance in Checking sub-scale.

Checking sub-scale	
Distance	Number of subjects
$d(R, K) = 2$	132
$d(R, K) = 3$	232
$d(R, K) = 4$	28

Figure 7.7: Checking sub-scale: distribution of the response patterns with non minimum $d(R, K)$, every slice represents a distance.



7.2.5 Condition 5, results about the means

Table 7.3 is an extract of Table 7.1, and presents the means of the distances. In the Cleaning case both the arithmetic mean and the weighted mean of the distances show good results. The Checking case, nevertheless his high “structuration”, presents encouraging results about means of the distances.

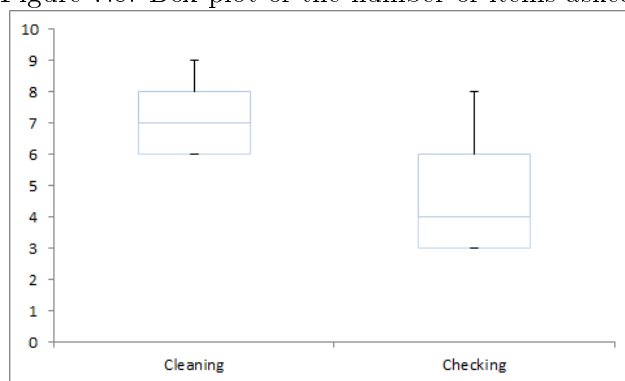
Table 7.3: Means of the distances.

	Cleaning	Checking
Arithmetic mean of $d(R, K)$	0.27	1.01
Weighted mean of $d(R, K)$	0.72	1.69

7.2.6 Condition 6, results about the number of questions posed

As already said, we were not really concerned about efficiency, but it was important to check whether the algorithm converged to the latent state K_0 asking a number of question smaller than the cardinality of Q . Recalling table 7.1, we have a mean of 6.9 asked items for the Cleaning structure with a saving of 14%, and a mean of 4.4 asked items in the other case, with a greater saving of 45%. Figure 7.8 illustrates the box-plot of the arithmetic mean.

Figure 7.8: Box-plot of the number of items asked.



Also this fact can be explained by the topology of the clinical structure, a more “structured” order will converge faster than a less “structured” one, see Figure 7.6. Our cases present only few items, so the saving is not perceptible;

we expect the higher the number of items, the higher the saving, for example the administration of only 70 questions of a 100 item questionnaire.

7.3 Discussions and future work

Clinical psychology has few computerized diagnostic systems, and the concept of adaptivity of assessment based on logical inferences of the clinician is not exploited properly. Our aim was to fill this gap with a system, called AAS-PD, able to conduct the process of clinical assessment in an adaptive, and logically correct, manner, in order to assist the clinician in the diagnosis formulation. For this reason, AAS-PD can be classified in the category of *Computerized Adaptive Testing* (CAT) for psychological disorders. AAS-PD's innovation lies in its theoretical framework, indeed it is based on the clinical structure given by FPA, i.e. a rigorous formal representation of the psychological disorder. At our knowledge this is the only computerized system based on such a model. The clinical structure returned by FPA is a knowledge structure, a typical mathematical structure of the *Knowledge Space Theory*. We applied an adaptive assessment algorithm conceived for this theory to our clinical structure.

The results fit our expectations. The system converges correctly to the latent state of every response pattern. In the Cleaning sub-scale the response patterns with $d(R, K) > 0$ are a small percentage: this portion of the sample can be explained as critical subjects through false positives and false negatives. On the contrary, in the Checking case, the response patterns with $d(R, K) > 0$ are much more, and this is due to the low cardinality of the structure, the is too small and it cannot cover the subjects as the Cleaning structure does. So, the tests can also be interpreted as diagnostic tools about the goodness of the structure.

Finally, results of analysis of the number of items saved show a saving ranging from 14% to 45%. Future tests are needed with bigger structures, initial not uniform probability distribution and with a Checking sub-scale modified. In conclusion, we can affirm that ASS-PD works properly with both structure as input. Nevertheless, further studies, for example the extraction of the structure from the patterns [66], can be conducted.

The main future proposal is to develop a real software able to assist the clinician in the assessment of the main psychological disorders, a sort of modern CBA, totally computerized and based on a solid formal representation of the disorders. We think in a scenario where the patient, with a tablet for example, fills in a questionnaire in an adaptive manner, and in a second moment the clinician controls the response of the system. The response could be a pie chart, where every slice represents a psychological disorder (OCD, anxiety disorder, mood disorder, etc) and has a probability indicating how likely the patient can fall in that clinical diagnosis. In order

to achieve this goal it is necessary to calculate the attributes probabilities [7], unify the three sub-scale of the OCD, and construct the clinical structure for the other questionnaires of CBA 2.0. This last task goes beyond the classic dichotomous logic on which our structures are based, indeed, many clinical questionnaire present questions with many values answers, a possible solution is given by many-valued formal contexts and fuzzy concept lattice [67, 68]. Last but not least a simple graphical user interface will provide the clinician an helpful way to interact with the system.

Bibliography

- [1] American Psychiatric Association [APA]. *DSM-IV-TR. Diagnostic and Statistical Manual of Mental Disorders*. American Psychiatric Association, Washington D. C., fourth edition, text revision edition, 2000.
- [2] L.C. Nunes, P.R. Pinheiro, and T.C. Pequeno. An expert system applied to the diagnosis of psychological disorders. In *IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009. ICIS 2009.*, volume 3, pages 363 –367, 2009.
- [3] C. Casado-Lumbreras, A. Rodríguez-González, J. M. Álvarez Rodríguez, and R. Colomo-Palacios. Psydis: Towards a diagnosis support system for psychological disorders. *Expert Systems with Applications*, 39(13):11391 – 11403, 2012.
- [4] American Psychological Association. About clinical psychology. <http://www.apa.org/divisions/div12/aboutcp.html>, 2012.
- [5] J. H. Van Bommel and M. A. Musen. *Handbook of medical informatics*. Berlin: Springer-Verlag, 1997.
- [6] World Health Organization. *The ICD-10 classification of mental and behavioural disorders : diagnostic criteria for research*. Geneva: World Health Organization, 1993.
- [7] A. Spoto. *Formal Psychological Assessment Theoretical and Mathematical Foundations*. PhD thesis, Università degli studi di Padova, 2010.
- [8] F. M. Lord. *Applications of item response theory to practical testing problems*. Erlbaum Associates, 1980.
- [9] J. P. Doignon and J. C. Falmagne. *Knowledge Spaces*. Springer-Verlag, Berlin and Heidelberg, 1999.

- [10] A. Spoto, G. Vidotto, D. Postal, and E. Pendoni. Rappresentazione formale dell'assessment cognitivo-comportamentale per il disturbo ossessivo-compulsivo. *Giornale Italiano di Medicina del Lavoro ed Ergonomia: Supplemento B. Psicologia*, 2008.
- [11] A. Spoto, L. Stefanutti, and G. Vidotto. Knowledge space theory, formal concept analysis and computerized psychological assessment. *Behaviour Research Methods*, 2010.
- [12] E. Sanavio. *Le Scale CBA, Cognitive Behavioural Assessment: un modello di indagine psicologica multidimensionale*. Raffaello Cortina Editore, 2002.
- [13] E. Sanavio, G. Bertolotti, P. Michielin, G. Vidotto, and A. M. Zotti. *CBA-2.0 Scale Primarie: Manuale. Una batteria ad ampio spettro per l'assessment psicologico*. Firenze: Organizzazioni Speciali, 1997.
- [14] G. Bertolotti, A. M. Zotti, P. Michielin, G. Vidotto, and E. Sanavio. A computerized approach to cognitive behavioural assessment: An introduction to cba-2.0 primary scales. *Journal of Behaviour Therapy and Experimental Psychiatry*, 21(1):21–27, 1990.
- [15] Enciclopedia Treccani. Falsificazionismo. <http://www.treccani.it/vocabolario/falsificazionismo/>, 2012.
- [16] PhilosophyofScience.info. Falsificationism. <http://www.philosophyofscience.info/falsificationism.html>.
- [17] B. Ganter and R. Wille. *Formal concept analysis - mathematical foundations*. Springer, 1999.
- [18] J. Wolpe and I. D. Turkat. Behavioural formulation of clinical cases. In I. D. Turkat, editor, *Behavioural Case Formulation*. New York: Plenum, 1985.
- [19] E. Sanavio and C. Cornoldi. *Psicologia Clinica*. Bologna: Il Mulino, 2001.
- [20] M. B. First, R. L. Spitzer, M. Gibbon, and J. B. W. Williams. *Structured Clinical Interview for DSM-IV Axis I Disorders, Clinician Version (SCID-CV)*. American Psychiatric Press, Washington D. C., 1996.
- [21] R. J. Hodgson and S. J. Rachman. Obsessional-compulsive complaints. *Behaviour Research and Therapy*, 15:389–395, 1977.
- [22] E. Sanavio and G. Vidotto. The components of the Maudsley Obsessional-Compulsive Questionnaire. *Behaviour Research and Therapy*, 23:659–662, 1985.

- [23] L.J. Simms, L.R. Goldberg, J.E. Roberts, D. Watson, J. Welte, and J.H. Rotterman. Computerized adaptive assessment of personality disorder: Introducing the cat-pd project. *J Pers Assess*, 93(4):380–9, 2011.
- [24] E. H. Shortliffe and B. G. Buchanan. A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23(3–4):351 – 379, 1975.
- [25] P. Jackson. *Introduction to Expert Systems, 3rd Edition*. Addison-Wesley, 1999.
- [26] R. Spiegel and Y. P. Nenh. An expert system supporting diagnosis in clinical psychology. *Human Perspectives in the Internet Society: Culture, Psychology and Gender*, 2004.
- [27] S.P. Yong, A. Rambli, D. Rohaya, and N.T.M Anh. Depression consultant expert system. In *6th Annual Seminar on Science and Technology*, October 2007.
- [28] K. Seong-in, R. Hyun-Jung, H. Jun-Oh, and K. Seong-Hak. An expert system approach to art psychotherapy. *The Arts in Psychotherapy*, 33(1):59–75, 2006.
- [29] C.A. Bana e Costa and J.C. Vansnick. The MACBETH approach: Basic ideas, software and an application. In N. Meskens and M. Roubens, editors, *Advances in Decision Analysis*. Dordrecht: Kluwer Academic Publishers, 1999.
- [30] L. V. Kan, Yu. M. Kuznetsova, and N. V. Chudova. On the methodological foundations for the creation of expert systems for practical psychology. *Sci. Tech. Inf. Process.*, 38(6):420–427, 2011.
- [31] D. O. Segall. Computerized adaptive testing. Defence Manpower Data Center, United States Department of Defence Encyclopaedia of Social Measurement, in press.
- [32] T. W. Chien, W. P. Lai, C. W. Lu, W. C. Wang, S. C. Chen, H. Y. Wang, and S. B. Su. Web-based computer adaptive assessment of individual perceptions of job satisfaction for hospital workplace employees. *BMC Med Res Methodol*, 11(1):47, 2011.
- [33] O. Sitthisak, L. Gilbert, and H. C. Davis. Towards a competency model for adaptive assessment to support lifelong learning. In *TENCompetence Workshop on Service Oriented Approaches and Lifelong Competence Development Infrastructures*, 2007.
- [34] L. M. Rudner. An examination of decision-theory adaptive testing procedures. In *Annual meeting of the American Educational Research Association*, 2002.

- [35] R. Conejo, E. Guzmán, E. Millán, M. Trella, J. L. Pérez-De-La-Cruz, and A. Ríos. Siette: A web-based tool for adaptive testing. *J. Artif. Intell. Ed.*, 14(1):29–61, 2004.
- [36] T.J.H.M. Eggen and G.J.J.M. Straetmans. Computerized adaptive testing for classifying examinees into three categories. *Educational and Psychological Measurement*, 60(5):730–734, 2000.
- [37] P. Nokelainen, M. Niemivirta, H. Tirri, M. Miettinen, J. Kurhila, and T. Silander. Bayesian modelling approach to implement an adaptive questionnaire. In C. Montgomerie and J. Viteli, editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 1412–1413, Norfolk, VA, 2001. AACE.
- [38] C.C. Marinagi, V.G. Kaburlasos, and V.T. Tsoukalas. An architecture for an adaptive assessment tool. In *Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE '07. 37th Annual*, 2007.
- [39] N. A. Thompson and D. J. Weiss. A framework for the development of computerized adaptive tests. *Practical Assessment Research & Evaluation*, 16(1), 2011.
- [40] K. Glantz, N. I. Durlach, R. C. Barnett, and W. A. Aviles. Virtual Reality (VR) and Psychotherapy: Opportunities and Challenges. *Teleoperators and Virtual Environments*, 6:87–105, 1997.
- [41] Riva G. Virtual reality as assessment tool in psychology. *Stud Health Technol Inform*, 44:71–9, 1997.
- [42] A. Suhasini, S. Palanivel, and V. Ramalingam. Article: Multi decision support model for psychiatry problem. *International Journal of Computer Applications*, 1(25):61–69, 2010.
- [43] A. Rusch and R. Wille. Knowledge spaces and formal concept analysis. In H. H. Bock and W. Polasek, editors, *Data analysis and information systems: Statistical and conceptual approaches*. Berlin: Springer, 1996.
- [44] R. Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In I. Rival, editor, *Ordered Sets*, pages 445–470. Dordrecht: Reidel, 1982.
- [45] B. Ganter, G. Stumme, and R. Wille. *Formal Concept Analysis: Foundations and Applications*. Springer, 2005.
- [46] G. Birkhoff. *Lattice Theory*. American Mathematical Society, 1995.

- [47] P. Valtchev, D. Grosser, C. Roume, and M. R. Hacene. Galicia: An open platform for lattices. In *Using Conceptual Structures: Contributions to the 11th Intl. Conference on Conceptual Structures*, pages 241–254. Shaker Verlag, 2003.
- [48] S. Kuznetsov and S. Obiedkov. Algorithms for the construction of concept lattices and their diagram graphs. In L. De Raedt and A. Siebes, editors, *Principles of Data Mining and Knowledge Discovery*. Berlin: Springer/Heidelberg, 2001.
- [49] P. Becker and J. H. Correia. The Toscanaj suite for implementing conceptual information systems. In *Formal Concept Analysis – State of the Art*. Berlin – Heidelberg: Springer, 2004.
- [50] J. P. Doignon and J. C. Falmagne. Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies*, 23:175–196, 1985.
- [51] J.C. Falmagne, J.P. Doignon, E. Cosyn, and N. Thiéry. The assessment of knowledge, in theory and in practice. <http://www.aleks.com>.
- [52] S.P Marshall. Sequential item selection: optimal and heuristic policies. *Journal of Mathematical Psychology*, 1981.
- [53] J. C. Falmagne, M. Koppen, M. Villano, J. P. Doignon, and L. Johannesen. Introduction to knowledge spaces: how to build, test and search them. *Psychological Review*, 97:201–224, 1990.
- [54] J. Lukas and D. Albert. Knowledge assessment based on skill assignment and psychological task analysis. In G. Strube and K.F. Wender, editors, *The Cognitive Psychology of Knowledge*. Amsterdam: Elsevier, 1993.
- [55] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.
- [56] R. D. Luce. *Individual choice behaviour : a theoretical analysis*. N.Y. : Wiley, 1959.
- [57] R.R. Bush and F. Mosteller. *Stochastic models for learning*. Wiley, 1955.
- [58] S. Landy, R.A. Hummel, and Courant Institute of Mathematical Sciences. Computer Science Division. Robotics Research. *A Brief Survey of Knowledge Aggregation Methods*. Defense Technical Information Center, 1985.
- [59] Python Software Foundation. Python v2.7.3 documentation - extending Python with C or C++. <http://docs.python.org/extending/extending.html>, 2012.

- [60] Python Software Foundation. Python Programming Language – Official Website. <http://www.python.org/>, 1990–2012.
- [61] M. Lutz and D. Ascher. *Programmare con Python*. Hoepli, 2004.
- [62] The Eclipse Foundation. <http://www.eclipse.org/>, 2012.
- [63] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2 edition, 2001.
- [64] M. Fowler. *UML Distilled: A Brief Guide to the Standard Object Modelling Language*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3 edition, 2003.
- [65] py2exe. <http://www.py2exe.org/>, 2011.
- [66] M. Schrepp. Extracting knowledge structures from observed data. *British Journal of Mathematical and Statistical Psychology*, 52(2):213–224, 1999.
- [67] A. Burusco and R. Fuentes-Gonzalez. The study of the L-fuzzy concept lattice. *Mathware & Soft Computing*, 3(3):209–218, 1994.
- [68] A. Burusco and R. Fuentes-González. Construction of the L-fuzzy concept lattice. *Fuzzy Sets and Systems*, 97(1):109–114, 1998.

Ringraziamenti

Il primo e più importante ringraziamento che voglio fare va ai miei genitori. Questa tesi non è solo frutto del lavoro degli ultimi sei mesi, ma è il traguardo di un percorso di studi iniziato molti, troppi, anni fa. In questi anni, dove sono emerse difficoltà, ripensamenti e frustrazioni, loro c'erano credendoci più di me. Con ingenuità, parole umili e semplici, mi sono sempre stati accanto nei momenti difficili, pur non sempre capendo le motivazioni di un ambiente così complicato. Del resto non le ho capite nemmeno io. A loro voglio dire un semplice, ma pieno d'affetto, "grazie di cuore". Ringrazio anche mia sorella Barbara e mio nipote Marvin per farmi tornare ogni tanto alla realtà dopo varie ore di studio.

In secondo luogo voglio ringraziare la mia relatrice, professoressa Silvana Badaloni, per la possibilità offerta con questa tesi, per aver creduto in me e per la pazienza nei confronti del mio cattivo rapporto con le scadenze. Ringrazio molto Andrea Spoto, il mio correlatore, per aver lavorato al mio fianco in questi mesi ed essermi sempre stato di supporto ogni volta che ne avevo bisogno. Un ringraziamento anche al professor Giulio Vidotto per avermi dato la possibilità di lavorare su questi argomenti e per i consigli ai momenti giusti, e a Francesco Sambo per la gentilezza e i puntuali consigli informatici quando ne avevo bisogno.

Ringrazio i miei amici di corso, Luca, Stefano, Marco, Riccardo, Ale, Nicola, che hanno reso più piacevoli e divertenti questi anni di studio, con risate, sketch, pranzi in Forcellini (con viaggio annesso).

Ma tra tante difficoltà scolastiche ci sono stati anche bellissimi periodi, l'erasmus. Voglio ringraziare tutti gli amici della Spagna per aver colorato il mio erasmus, in particolar modo Elena e Irene per avermi aiutato con arguti consigli in tanti momenti (anche dopo l'erasmus), e José per il costante clima di buon umore e lo "spirito mediterraneo" infusi.

Ringrazio pure gli amici di una vita Alessandro, Andrea, Riccardo, Samuele, Nicolò per le risate di un tempo, e con alcuni anche di adesso, e le loro ragazze, Paola, Giulia, Chiara, Marianna che piano piano si sono unite alla compagnia, diventandone parte, e contribuendo a piacevoli serate.

Ringrazio in particolar modo Riccardo per le innumerevoli birrette assieme, riflessioni filosofiche/matematiche varie, accesi dibattiti sui massimi sistemi che alla fine si risolvono in una gran bella risata. Va ricordato e ringraziato anche Andrea il mio storico, e unico, compagno di stanza con cui ho condiviso questa avventura Patavina e anche molti momenti divertenti. E come non ringraziare le mie tre amiche di una vita, Chiara, Silvia e Martina per ricordarmi che anche la mia terra natale ha qualcosa di prezioso da offrire. Ringrazio inoltre Nadia, per tutte le altre, e tante, cose.