UM SISTEMA DE RECOMENDAÇÃO INTELIGENTE BASEADO EM VÍDIO AULAS PARA EDUCAÇÃO A DISTÂNCIA

Gaspare Giuliano Elias Bruno

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientadores: Edmundo Albuquerque de Souza e Silva
Rosa Maria Meri Leão

Rio de Janeiro
Janeiro de 2016

UM SISTEMA DE RECOMENDAÇÃO INTELIGENTE BASEADO EM VÍDIO AULAS PARA EDUCAÇÃO A DISTÂNCIA

Gaspare Giuliano Elias Bruno

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Edmundo Albuquerque de Souza e Silva, Ph.D.

Prof. Rosa Maria Meri Leão, Dr.

Prof. Felipe Maia Galvão França, Ph.D.

Prof. Daniel Sadoc Menasche, Ph.D.

Prof. Berthier Ribeiro de Araujo Neto, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
JANEIRO DE 2016

Bruno, Gaspare Giuliano Elias

Um Sistema de Recomendação Inteligente baseado em Vídio Aulas para Educação a Distância/Gaspare Giuliano Elias Bruno. – Rio de Janeiro: UFRJ/COPPE, 2016.

XIII, 107 p.: il.; 29, 7cm.

Orientadores: Edmundo Albuquerque de Souza e Silva
Rosa Maria Meri Leão

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2016.

Referências Bibliográficas: p. 98 – 107.

1. Intelligent Tutoring Systems, Adaptive Learning, Affective Computing. I. Silva, Edmundo Albuquerque de Souza e *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Para minha esposa Erica,
e minhas filhas Helena e Beatriz,
obrigado pela paciência durante
todos estes anos.

Agradecimentos

Gostaria de agradecer ao apoio de meus colegas de laboratório durante a realização desta tese. Agradeço também aos meus orientadores, professor Edmundo e professora Rosa, pelo tempo e auxílio empregados nesta pesquisa. Seus conhecimentos e ensinamentos foram de fundamental importância para a realização desta tese.

Este trabalho não poderia ter sido realizado sem o suporte financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e do GOOGLE Inc. Meus agradecimentos vão também ao curso de Tecnologia em Sistemas de Computação do CEDERJ, cuja abertura para pesquisa foi de grande valia para o desenvolvimento desta tese.

Um agradecimento especial a minha família, que me suportou firme e forte durante os longos anos deste trabalho.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

UM SISTEMA DE RECOMENDAÇÃO INTELIGENTE BASEADO EM VÍDIO AULAS PARA EDUCAÇÃO A DISTÂNCIA

Gaspare Giuliano Elias Bruno

Janeiro/2016

Orientadores: Edmundo Albuquerque de Souza e Silva
Rosa Maria Meri Leão

Programa: Engenharia de Sistemas e Computação

O aprendizado utilizando vídeo aulas vem se tornando o método de ensino a distância mais utilizado nos últimos anos. No entanto, diferentemente dos sistemas de tutoria inteligente, as vídeo aulas não têm a capacidade de adaptar seu conteúdo de acordo com as dificuldades de cada aluno, principalmente porque os alunos são passivos em relação à vídeo aula. Nesta tese é apresentado um sistema de recomendação inteligente baseado em vídeo aulas para o uso em ensino a distância. Tal sistema é baseado em modelos computacionais para detecção do engajamento do aluno enquanto assiste uma vídeo aula. Para tratar o problema da falta de interatividade do aluno, tais modelos utilizam sensores que monitoram o comportamento do aluno durante o aprendizado. Nós desenvolvemos um sensor baseado em uma câmera com infra-vermelho que captura características dos olhos como o diâmetro da pupila e a taxa de piscada dos olhos. Os modelos desenvolvidos atingem até 80% de precisão quando comparado a informação semelhante provida por professores. Também é apresentado um modelo para medir o engajamento do aluno baseado no tempo que o aluno assiste cada vídeo aula.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

AN INTELLIGENT RECOMMENDATION SYSTEM BASED ON VIDEO LECTURES FOR DISTANCE EDUCATION (REVELATION)

Gaspare Giuliano Elias Bruno

January/2016

Advisors: Edmundo Albuquerque de Souza e Silva
Rosa Maria Meri Leão

Department: Systems Engineering and Computer Science

Video lectures have become the most used method of distance learning. But, differently from modern Intelligent Tutoring Systems, video lectures lack the capability to adapt its contents to the student needs. Basically because video lectures are passive to the student. In this thesis, we present an intelligent recommendation system based on video lectures for distance learning. We address the problem of lack of interactivity between the student and the video lecture by passively monitoring the student with sensors. We developed a novel student model that can output the engagement of the student towards the system using the information provided by these sensors. Our system is capable of adapting the flow of the video lecture based on the reported output of our student model. To be able to monitor the engagement of a student, we developed a sensor based on a infrared webcam that can capture eye features, like pupil diameter and blink rate. Using this sensor and our model, we can achieve an accuracy higher than 80% for the engagement of students, compared to the feedback provided by teachers. We also present a model to measure the engagement of students in a video lecture based on how much time the student spent in the lecture.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Distance learning, or e-learning, has become a strategic educational model, particularly in developing countries. It is a promising solution to conciliate the career and life responsibilities with the needs to improve the skills of those with difficulties to attend a regular classroom environment. In Brazil, only a very small fraction of the young population has access to high level education. There are several reasons that may explain this fact, such as: insufficient number of openings at public universities, large distances from major cultural centers, lack of the infrastructure to facilitate mobility of students and lack of financial resources to support students with no means to maintain themselves while attending college.

In the past ten years, the Brazilian government has made efforts to increase the number of openings at the federal university system. However, a significant increase in openings requires a proportional increase in faculty members. Unfortunately, the training for graduating good quality teachers for high level education is a long term process, easily involving more than twenty years. Distance learning has been proposed as a solution to mitigate the gap between the country needs in terms of highly qualified workers and the number of graduates the universities can deliver. Recently, the Open University of Brazil (CAPES, 2015) was created inspired by the CEDERJ Consortium, a distance learning initiative of the state of Rio de Janeiro, involving public universities located in Rio (CEDERJ, 2015) .

The methodology used in distance learning can be synchronous or asynchronous (Hrastinski, 2008). In synchronous distance learning, teacher and students make use of technologies that enable both sides to meet at distance, like video-conferencing. This method partially solves the distance problem, but the rigid schedule remains an issue. Asynchronous distance learning systems allow students to have ubiquitous access to classes. This solves both problems — schedule and location — by allowing

students to attend classes anytime, anywhere. In this methodology, students use material previously prepared, like a recorded video of the lecture. As a consequence of initiatives like Coursera (2015), edX (2015) and Khan Academy (2015), this methodology became the most used in distance learning initiatives.

In most distance learning initiatives, a support environment is also available to the registered students. In the Computer Science course of the CEDERJ consortium, for instance, students have access to additional class material besides video lectures. This material includes homework assignments, links to relevant text in the Web, and access to tutors (usually graduate students from the signatories consortium universities), via synchronous technologies, like video-conferencing. However, although tutoring is certainly an important part of the educational process, it is difficult to scale.

In a traditional lecture, the teacher may infer when a group of students attending the class does not understand a particular explanation and then, she can adapt the lecture accordingly to the students' reaction. Feedback from questions asked during classes and the facial expressions of students are important clues that teachers use to evaluate if the topic being explained is being absorbed or not by students. This learning process is an active area of study in cognitive science (Graesser and Person, 1994). The feedback from students is the key to maintaining their motivation and engagement. In the United States, the National Survey of Student Engagement (NSSE, 2015) collects information to evaluate the engagement of students in educational activities. Unfortunately, in a distance learning environment, the feedback from students is not available during a video-lecture. Consequently, the class material cannot be adapted in real time to the student's needs.

Recent work on Adaptive Learning Systems (ALS)(Paramythis and Loidl-Reisinger, 2003), like the Adaptive Hypermedia Systems (AHS)(Brusilovsky and Millán, 2007; De Bra et al., 2003) and Intelligent Tutoring Systems (ITS)(Cooper, 2011; Graesser et al., 2005a; Sottilare and Holden, 2013) have addressed a number of issues targeted at providing help to students during the learning process. In fact, research on building personalized educational systems have raised a lot of attention in the last few years, perhaps boosted by a speech from President Obama in 2009 in which he exhorted the scientific community to develop "learning software as effective as a personal tutor".

ITS is a system that interacts with the student in a task-based environment. The system presents the student with a task, which can be a question on a specific topic the student is learning, or a simulated environment, like surgeries and first-person shooter games. As the student solves the problem, the system tracks her work and collect information about her performance. This information can be answers to questions, for example. The system uses the collected information to make infer-

ences about the development of the student on the learning topic, and may suggest additional work to help the student. For example, if the student provides a wrong answer, the system can, in turn, offers a few hints. On the other hand, if the answer is correct, the system may increase the difficulty of the exercises that would follow.

The interface used by ITSs to interact with students can be a simple point-and-click multiple answers, or a more advanced interface, which uses natural language for questions and answers. Advanced ITSs try to predict the mental state of a student while executing a task (D'Mello et al., 2007). Predicting the mental state is important because it can provide useful information about the student even if the student is not interacting with the system. When the system detects that the student got confused while executing a task, the system is capable of providing helpful hints. For detection, ITSs usually employ sensors of different kinds, such as motion or pressure-based(Cooper, 2011), Electroencephalography (EEG) (Crowley et al., 2010), or camera-based(el Kaliouby, 2005), which tries to predict emotions based on facial expressions. Engagement is one of the most important mental states(D'Mello and Graesser, 2012; Nakamura and Csikszentmihalyi, 2002; Picard, 1997; Risko et al., 2013), as it is an indication that the student is motivated in a task.

ITSs are effective in helping students on problem-solving and question-answering studies. They are a good approach to exercise a topic. Usually the student needs prior skills and theories about the topic in order to use the system properly. On the other hand, on the AHSs the student can learn from previously recorded content, like text, images, books, videos, etc. These systems adapt the content to the learner's goals, interests, knowledge, etc. Usually, these systems are HTML based and use some kind of inference to discover a priori information concerning the student. A few systems discover useful information just by tracking the student while she is interacting with the system. For example, after the student browses through a few pages, the system is capable of inferring that she has acquired the necessary knowledge for a topic and then presents additional links to more advanced material (Brusilovsky, 2012). The system may also split a page content into sections and conditionally present each section to the student, providing to each individual a distinct experience of the same content (De Bra et al., 2003). These systems can also act as a *recommendation system*, by showing additional information on a topic based on users' prior preferences (Romero et al., 2009; Wu et al., 2001). To our knowledge, no AHS has used sensors to model user's behavior.

In this thesis, we developed an *intelligent recommendation system* based on video lectures for distance learning. We followed the basic approach of Adaptive Educational Hypermedia Systems (AEHSs) that can recommend new material to the student and/or adapt the current content based on the student's behavior. We address

the problem of absence of interactivity present in distance learning systems based on video lectures. Different from existing AEHS, we overcome the problem of low interactivity by using sensors to monitor the student. To our knowledge, this is the first Educational Hypermedia System targeted to video lectures which is adaptive and employ sensors that are used to create a student model. Our Adaptive Learning System uses concepts of both AHSs and ITSs. Similar to AHSs, the student learns from previously recorded materials (the video lectures), and adapt the content to the student's needs. Our system may also present exercises and task-based problems, like an ITS, and is capable of inferring the mental state of the a student, but our focus is on the cognitive engagement state. For example, if the student is distracted from the lecture, the system may motivate the student to regain focus on the lecture by recommending additional examples or exercises about the topic. The system can also try to predict the best moment to interact with the student. For example, if a student is focused on a given activity, a sudden change of context may be harmful to the learning process. On the other hand, a change of context may be useful depending on the student *level of distraction* (D'Mello and Graesser, 2012). One additional goal is to give the lecturer useful information to aid in the development of a lecture that would improve the student's engagement.

Our research targeted the video-lecture environment of the Computer Systems Technology undergraduate program from the CEDERJ consortium. This is a three-years program that differs from the others offered by the Consortium because it is totally based on video lectures. The lectures are taught by faculty members both from the Federal University of Rio de Janeiro and the Fluminense Federal University. The program started in 2005 and currently offers more than 500 video lectures specially prepared for it. Nearly 2000 students are presently registered and have access to the lectures material through the CEDERJ website. In addition, since 2011, all lectures were made available in the Internet by the National Research and Educational Network (RNP), via the Videoaula@RNP site. RNP manages the video servers and the associated streaming software. In addition, RNP offers technical support to those institutions willing to produce and make available video lectures on any subject.

Our laboratory has developed all the software used by the video lectures of CEDERJ and at the RNP site. During this thesis, we modified this software to include new features resultant from our research, such as the ability to adapt the flow of a lecture according to its users "engagement state". We tested the system on students from UFRJ and CEDERJ and collected information to design a probabilistic classifier applying unsupervised learning techniques. The goal of the classifier is to predict the cognitive engagement state of the student with time. Our classifier can achieve an accuracy of over 70% when compared to classification done by profes-

sors. Judging from similar experiences throughout the world, it is easy to foresee the potential social impact of initiatives as CEDERJ. We hope that our work can collaborate to improve its potentials.

We also analyzed a database containing more than two years of data collected from students using the Videoaula@RNP system. This database contains the interactions of students (mostly CEDERJ students) with video lectures. Each access to the system is logged as a new session and recorded in the database. Over a million student sessions are available in this database. We extracted and analyzed several features, such as *watch*, *play* and *session* times, and the popularity of each segment of a lecture. We selected the *watch time*, defined as the fraction of the total lecture duration that was viewed during each session, as an "engagement metric". The *watch time* metric is employed in our analysis and shown to provide important feedback to evaluate the quality of a video lecture.

1.2 Goals, Challenges and Contributions

One of the goals of an Adaptive Learning System is the ability to adapt the presentation of the teaching material according to individual student's needs. Advanced ALSs can infer the "student mental state" during the different learning stages of a topic. The "mental state" is estimated from inputs captured by the system while the student interacts with it. This is particularly difficult in learning environments with low interactivity, like those that employ video lectures. In this thesis, inputs from sensors are combined to predict the level of student's cognitive engagement while watching a video-lecture. The output of the predictor is used to aid in the learning process.

From the results of our research we expect to improve the quality of the learning experience students get from distance learning systems. We also expect to give feedback to lecturers concerning the adequacy of the learning material. By detecting individual pieces in the lecture where students present greater difficulties, lecturers may improve the learning material.

In traditional classrooms, teachers are usually able to detect when students are not paying attention or focused on an explanation. Consequently, they may immediately adapt the presentation to motivate the students. Our goal is to emulate this ability for a distance learning system. The challenge is to develop a reasonably accurate model able to detect the student *cognitive engagement state* using a passive monitoring system, like sensors.

The interaction between teachers and students in classrooms occurs not only via verbal communications. Teachers also rely on nonverbal clues gathered from their past experience. To automate this process, it is necessary to address a basic question:

What are the clues that students unconsciously provide that allow teachers to guess when they are not paying attention to an explanation? It is an open research problem to recognize automatically such clues. To address this issue, we employ sensors such as the EEG and EDA to monitor the student. In addition, previous works (Beatty, 1982; Conati et al., 2013) have shown that eye metrics, like pupil diameter, may be a good indicator for sustained attention and, as a result, provide evidence on the engagement state of a subject. Motivated by those works, we develop the necessary hardware and software to capture eye metrics and use as a sensor input. We first try to determine which features are most relevant to describe an cognitive engagement state, and then develop an algorithm to automate detection.

The main objectives of this thesis are:

1. Advance the state of the art in machine inference for cognitive engagement. Based on past work in the area, we propose a new model to automatically capture the cognitive engagement level of students. Other researches define and capture *engagement* using methods which are based on the *active attitudes* of students while interacting with educational tools (Risko et al., 2013; Rotgans and Schmidt, 2011). On the other hand, most video lectures do not require that students interact with the lecture, that is, students watch passively video lectures. Motivated by the *passive student attitudes* with respect to the video lectures, we use *attention* to characterize the students engagement with the lecture.

2. Build an operational system that can adapt the lecture according to the student's level of engagement. To our knowledge, this is the first system to have a module that can adapt the video-lecture based on the output signals from passive monitors. Our module may change the flow of the video-lecture without relying on students interactions and, as such, provides each student with a different experience, tailored to each individual.

3. Analyze the engagement of students based on their past behavior while watching a video-lecture. After analyzing more than two years of student's recorded sessions of the videoaula@RNP service[1], we propose a model that can be used to classify lectures using the total time a student watched a video lecture. Our results can also be used by teachers to improve their video lectures.

Our main contributions can be summarized as follows:

[1]We helped design the new tool that collects the logs of users of the videoaula@RNP service. This tool is currently implemented in the videoaula@RNP service.

- A new architecture for adaptive learning systems based on video lectures.
- A working prototype of an adaptive learning system based on the proposed architecture.
- A new low-cost sensor, built with off-the-shelf components, for analyzing attention based on the eye metrics.
- An *unsupervised machine learning based classification module* that is able to report a student state of attention in real time. Our classifier can achieve an accuracy of over 70% when compared to classification done by professors.
- A database that includes sensor information from the experiments we performed with students and another that includes students logs during a video-lecture session. Both databases include novel information not available elsewhere. Databases like this are very useful for researchers in general and will be made publicly available.
- A set of tools that we developed to support experimentation and that can be used to perform additional studies and data gathering. One of the tools is a simple environment for experts to judge and classify the attention of students. The other is a sophisticated module that outputs a video from student while watching a lecture, the output of several sensors, the results from the automated classifier and also from the classification as given by experts. All outputs are synchronized in time.
- A model that can be used to classify video lectures based on the "watch time" (defined in chapter 7), which we use as a metric of engagement. Although we have not studied regular video in our work, our model should be immediately applicable to this area, with little effort.

1.3 Dissertation Overview

In this thesis, we present a novel framework for Intelligent Distance Learning Systems. One of our main contributions is the development of a system that is able to automatically output a measure of student cognitive engagement, in real time, from the measured data obtained from a set of sensors attached to the student while watching a video-lecture. We use *attention* as an indication of cognitive engagement, based on previous work in the literature. As such, the system automatically gives the probability that a student watching a video-lecture segment is paying attention to that segment. This system is founded on a probabilistic classifier based on unsupervised learning techniques. This classifier uses as input features extracted from

sensors. The automated classification can be used in many ways. We explore one of the several usages of the classifier and developed an application which automatically adapts a video lecture from the output of the classifier module. In addition, we analyze data collected over two years of usage of the CEDERJ video lectures and propose a novel model to classify video lectures that we use as an engagement metric.

The thesis is organized as follows:

Chapter 2: Background

We begin by reviewing the literature on the subject area of this thesis. This chapter first describes the theories of mental states in the learning process and the importance of the *engagement state*. We then provide an overview of recent studies on how to measure the engagement using sensors. The chapter concludes by presenting the work in the literature that is related to this thesis, i.e., a brief survey on adaptive learning systems.

Chapter 3: Framework

In this chapter, we present the architecture of our adaptive learning system. We present the additional features which will be added to the video-lecture system of CEDERJ. (Recall that these video lectures are made available through the videaula@RNP service.) We also describe the modules that comprise the system's architecture and present the supporting tools implemented to aid in the development of the student model.

Chapter 4: MindLand Database

To develop and validate our student model, it was necessary to collect information from students using the implemented system. In this chapter, we present the employed methodology and the sensors used to collect the information that comprised our database. We developed an adaptive video-lecture as part of the methodology. We also present the sensor we built to capture eye metrics. As part of our methodology, we use a subjective classification of *attention* from short video-segments of students in our database. The last part of the chapter presents a few metrics extracted from our database and relates the information contained in it with other databases in the literature.

Chapter 5: Feature Extraction

The fifth chapter presents the algorithms developed to extract the features from the data recorded in the MindLand database. We use signal processing filters to evaluate and extract relevant information from the EEG and EDA sensors. We present the filters we developed to capture and process eye video signals and algorithms to extract numerical features like blink rate, pupil size

and gaze information (pupil movement). We also describe the computer vision algorithms applied for pupil detection. We conclude the chapter by presenting a list of features that we extracted and how they can be used in future research.

Chapter 6: Student Attention Classifier

The classifier module is the main focus of Chapter 6. The chapter presents the classification obtained based on unsupervised learning techniques from the extracted features of Chapter 5. We describe the learning procedure of this module, how it is implemented and relate the results with student *attention*. The automated classification output by the model is compared with that obtained from the subjective classification performed by faculty members.

Chapter 7: Engagement Model Base on Log Analysis

In this chapter, we analyze two years of logs collected from the CEDERJ video lectures. We present a few statistics obtained from the metrics extracted from the logs. In addition, we propose an engagement model that can be used to classify a video-lecture.

Chapter 8: Conclusion

We conclude by summarizing the contributions of this thesis and presenting possible future research that may spawn from our work.

Chapter 2

Background

To improve the student learning process, researchers and developers of educational systems have to understand the behaviour of students in the different learning stages. Craig et al. (2004) shows that the learning process can be expressed as a sequence of cognitive mental states. For example, the student can be engaged in a task or confused by an explanation. To detect these states, we need to obtain some form of feedback from the student while she interacts with the system. This interaction can be captured from the user inputs to the system or from the user's behaviour in response to the system information, measured by sensors. The feedback is processed to generate features that can be analysed by a classifier, or predictor, to infer the user's mental state. This is basically a computational model of the student's behaviour. The classifier indicates the most probable mental state the student is expressing and can be used by a learning system to adapt the content to each user specific needs. So, if the student loses engagement during the lecture, the system can adapt to attract the student back to the lecture by presenting an exercise, for example.

Figure 2.1 summarizes these ideas. Recall that our objective in this thesis is to develop an adaptive learning system based on video lectures that can adapt the presentation based on the student engagement towards the lecture. For that, we employ a passive monitoring system based on sensors and a new learner model that can infer the student engagement.

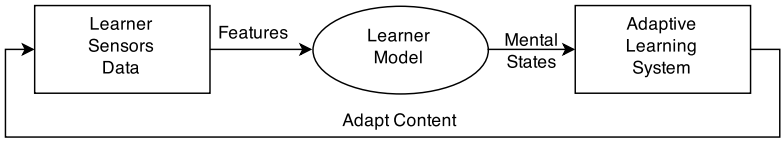


Figure 2.1: Overview of Detection Process

In this chapter, we present the necessary background to understand the research topics we address and the related work of this thesis. We begin by clarifying the concepts in the learning process and how the engagement of a student is related to that process. We show how sensor's data can be used to capture *engagement*. By sensors, we mean any method to catch feedback from students. We conclude this chapter overviewing other Adaptive Learning Systems proposed in the literature that have the ability to adapt their content, in particular by employing sensors.

2.1 A Brief Overview of the Learning Process

Mental state is an indication of the *subject*'s current state of mind (el Kaliouby, 2005). It can be affective or cognitive. Affective mental states are those expressed by emotions, such as sadness, happiness, anger, fear, etc. Cognitive mental states are related to mental processes, as thinking, concentrating and engagement. While some studies correlate cognitive and affective mental states, only recent studies have show the cognitive-affective states that have some impact on the learning process (D'Mello and Graesser, 2012). Emotions like frustration, boredom, surprise and anxiety are more likely to occur during cognitive tasks of the learning process, like trying to solve a problem or to comprehend a new information. In these process, engagement is a special state which indicates that the student is motivated to perform the task. In experiments conducted by D'Mello et al. (2007), the engagement was the most observed state from students while interacting with an intelligent tutoring system and it is present more than 45% of time.

An important research issue is to identify both the current mental state of a student and how these states are correlated. The cognitive disequilibrium theory, a theory from the psychology that have its origins in the cognitive dissonance theory (Festinger, 1962) and the Piaget's equilibrium theory (Inhelder et al., 1976), have been used by D'Mello and Graesser (2012) to model the interaction of mental states in the learning process, as shown in Figure 2.2.

This theory proposes that a student in equilibrium, when engaged in the learning process, can became confused by an explanation or exercise and enter in a disequilibrium process. The equilibrium can be restored if the student manage to solve the problem. Although this model has only been partially validated (D'Mello and Graesser, 2012), this theory gives some insights on how these mental states are related.

The state of disequilibrium, or confusion mental state, is desired to occur during the learning process. It makes the student think about the learning material and helps in the comprehension of the information. The learning can also occurs in the equilibrium, where the student presents deep engagement in the learning task

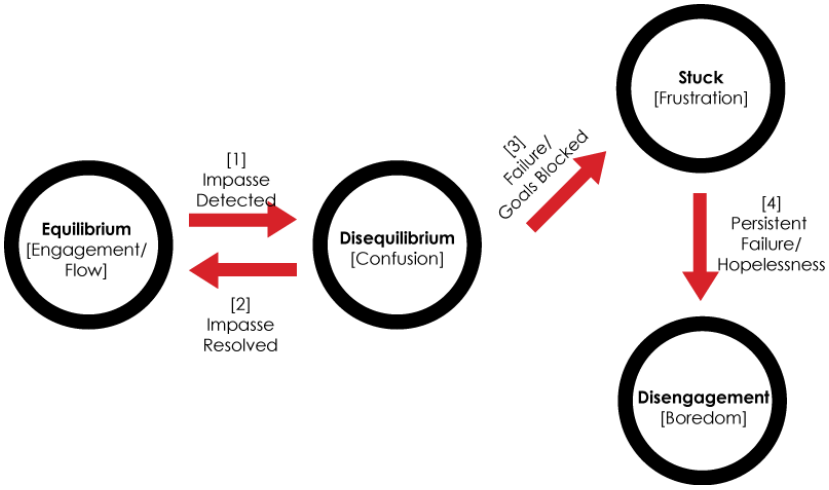


Figure 2.2: Equilibrium-Disequilibrium as proposed by D'Mello and Graesser (2012)

(Nakamura and Csikszentmihalyi, 2002).

The valence-arousal model (Baker et al., 2010; Lang, 1995) is a study that tries to categorise the affective mental states. Valence are associated with positive, pleasure mental states, like happiness and joy. Arousal represents mental states that have a higher activity in the body, usually caused by reaction to an external stimuli. They are associated with an increase of blood pressure and can be easily seen on facial expression. Happiness is an example of high arousal. Baker et al. (2010) analyzed how the learning mental states are correlated in the valence-arousal model. Figure 2.3 shows this correlation. In that figure, four quadrants are presented. The valence dimension (pleasure to displeasure) presents negative mental states on the left and positive on the right. The arousal dimension (activation to deactivation) presents high activity mental states on the top and low on the bottom. For example, frustration(FR) represents a negative mental state with high activity. Other states in the picture are: Surprise (SU), Delight (DE), Neutral(NU), Boredom (BO), Confusion (CO) and Engagement (EC).

Engagement is considered positive in this study, but they don't have a consensus on its classification in the arousal level (thus, classifying as neutral). The study only demonstrate the idea that engagement tends to fluctuate during a learning session, usually increasing on novel inputs and decreasing over time. The experiments conducted by Baker et al. (2010) point out that engagement has the tendency to be high during the interaction with Intelligent Tutoring System. Other findings from this study suggest that, when the student loses the engagement state (which is called "disequilibrium" in the D'Mello and Graesser (2012) theory), it is hard to return to engagement. Usually, the student gives up and returns to the topic later. This perception of engagement is also studied by other authors, particularly Nakamura

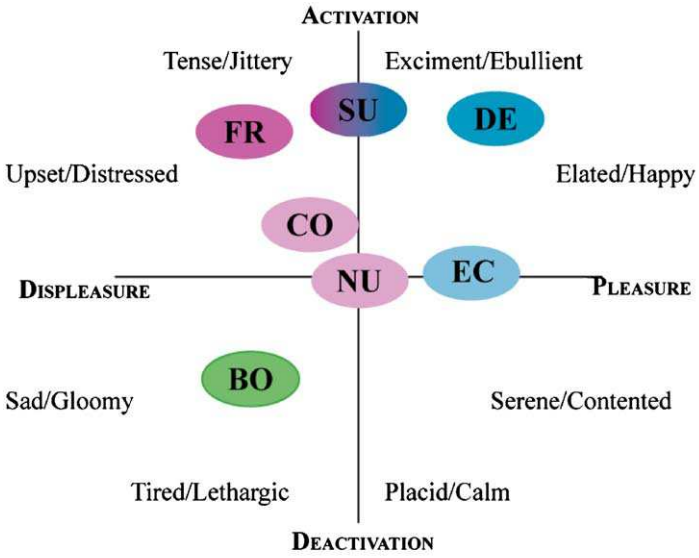


Figure 2.3: Valence-Arousal (Baker et al., 2010)

and Csikszentmihalyi (2002), with similar findings.

2.1.1 Cognitive Load Theory

Cognitive load is the cognitive effort that a person spends during a task. In the learning process, it is related to the amount of information a student receives and the effort needed to absorb this information. It is based on the perception that humans have a limited working memory and an unlimited long-term memory (Antonenko et al., 2010). A student can't retain much information due the limitation of his working memory, but information that has already been learned and understood can reduce the load of the working memory by simplifying the received information. Therefore, the cognitive load can be a measurement of difficulty on learning a specific topic in accordance with the student previous knowledge.

Cognitive load is also related to the way a subject is presented to the student (Sweller, 2010). For example, an instructional procedure (a way to teach a topic) may work for novices but may not work for more advanced students. This problem is known as the Expertise Reversal Effect. Advanced students have a consolidated long-term memory, requiring less effort to understand a topic than inexperienced students.

The Cognitive Load Theory has an important role in designing E-Learning Systems. Van Merrienboer and Sweller (2005) shows that an Adaptive Learning System needs to measure the load on students in order to adapt its contents to meet the needs of these students and maintain their engagement in the lecture. But if the load imposed on a students remains high over a long period of time, the student

begins to burn-out and leaves the engagement state. This corroborates with the findings of Baker et al. (2010), in which the student can feel bored. One of the experimental studies in Van Merrienboer and Sweller (2005) indicates that engagement can be controlled provided that the load can be maintained on an optimal level. Sometimes, switching activities (from a reading material to an exercise, for example) can decrease the load and maintain the engagement. This is considered as a balance between Extraneous Cognitive Load (generated by exercises) and Intrinsic Cognitive Load (generated by processing information).

2.2 Measuring Engagement

As mentioned above, engagement is one of the fundamental states in the learning process. If the student is engaged, her learning process is maintained. In classroom, engaged students may be easy to perceive, as they usually ask questions and are more active than others. But during relatively passive activities, like reading a book or watching a video, it is difficult to measure engagement. Even in classroom, it is difficult to notice students that are paying attention and focused on explanations if they are not active. Some authors call the ability of students to block out distractions and maintain a sustained attention in class as cognitive engagement (D'Mello and Graesser, 2012; Risko et al., 2013; Rotgans and Schmidt, 2011). Trained teachers can perceive the level of cognitive engagement of students and adapt their class correspondingly.

In Adaptive Learning Systems, detecting automatically the time instants during the lecture in which the students are on the verge of leaving the state of engagement may indicate that a lecture should be adapted to achieve better understanding of the topic (D'Mello and Graesser, 2012). Picard (1997) points to four main ways to detect mental states in today's systems: vocal, observable behaviour (user actions), facial expressions and physiological signals (heart rate and electrodermal activity, for example). In this section, we present some of the sensors that can be employed to automate the mental states detection process. Using the theories of valence-Arousal and Cognitive Load, we can understand and extract features from the sensors that may indicate if the student is engaged or not on a lecture. This may provide a valuable information to allow teachers to design modifications on the video lectures flow. In our system, this information will be used to indicate the best point to intervene, trying to maintain an optimal engagement.

2.2.1 Electroencephalograph

The electroencephalograph (EEG) signal is the measurement of electrical activity generated by the brain (Basar, 1999). Each neurone generates a small electric voltage and the sum of voltages generated by the neurones in a specific region of the brain can be measured by an electrode placed on the surface of the scalp. A typical setup uses 19 electrodes and a reference signal, usually measured at the earlobe. Figure 2.4 shows this montage, known as 10-20 system.

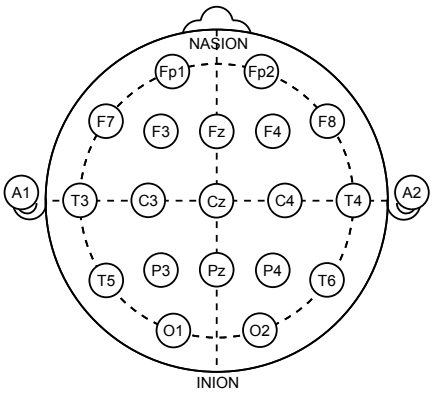


Figure 2.4: Electrode placement from 10-20 system

Cognitive activities and mental states causes variations in neural activity (Basar, 1999). These variations are identified by rhythmic fluctuations in the EEG signal, which occur in particular frequency bands. The study of these frequency bands is an active field of neuroscience. The exactly location and function of each band is still unknown. Researches usually vary in the range of bands by a few hertz, and it's known that sub-bands exist. Although the main functionality of each band is a consensus, new functions and enhancements are not rare. The major frequency bands, as presented by Basar (1999) and Klimesch (1999), are:

Delta: Very low frequency waves presented in the range of 0.1 to 4 Hz. They are normally associated to deep sleep states and mental restructuring.

Theta: Presented in the range of 4 to 8 HZ. They are associated to meditation and increase of creativity.

Alpha: Range from 8 to 12 Hz. Some authors indicate that this band can increase until 15 Hz. Normally associated to memory and attention. This is the most studied band in the literature.

Beta: Range from 12 to 30Hz. Beta lower frequencies are usually associated to memory, as of alpha band. The higher frequencies are usually associated to the motor behaviors, like active movements.

Gamma: Range from 30Hz to 100Hz. Although many studies exist on this band(Basar, 1999), to our knowledge there is no consensus on its meaning yet.

Frequencies above 70Hz are considered related to muscle, motion or pure noise, although Basar et al. (2001) point out that frequencies up to 100Hz deserve more study. Another important research topic is the correlation between these bands. Theta, alpha and beta waves were studied combined to measure engagement (Chaouachi and Frasson, 2010). Correlation between alpha and theta have been used to measure cognitive load (Antonenko et al., 2010; Basar et al., 2001) and as an indication of attention (Klimesch, 1999).

These frequency bands are usually segmented from the raw EEG data by using some kind of signal transform, like Fast Fourier Transformation. Other signal processing algorithms, like Wavelets (Murugappan et al., 2010) and High Order Crossing analysis (Petrantonakis and Hadjileontiadis, 2010), were also used over the raw EEG signal for specific analysis.

The use of EEG devices in emotion recognition is a recent research area. Chaouachi and Frasson (2010) show that EEG has good results in measuring user engagement and other mental states normally related to the learning process, like frustration and boredom. EEG is particularly good to measure mental states with lower arousal, that is, less activity. EEG has also proven to be good in measuring cognitive load associated with learning materials (Antonenko et al., 2010).

2.2.2 Electrodermal Activity

It is known that the dermis and hypodermis, when supplied with blood flow, possess good electrical conductivity (Boucsein, 2012). This conductivity can fluctuate depending on the blood flow intensity on the specific area of the skin. So, the skin can act as a resistor with variable resistance depending of the blood flow. This phenomena is called skin conductance, or electrodermal response. The Electrodermal Activity (EDA) is the measurement of this conductance. When the blood flow increases in the skin, the sweat glands activity also increases, which increases the conductivity of skin. Since the blood flow, and consequently the sweat, is controlled by the nervous system, skin conductance is used as an indicator of psychological arousal. When a person becomes more stressed and nervous, the blood flow tend to increase, and so the sweating. On the other hand, as a person becomes calm and relaxed, less sweat is produced.

To measure the conductivity, a sensor applies a small voltage to the skin. This voltage is usually a few micro volts. This voltage is read from another part of the skin, near to the initial one, and amplified. The sites used to apply and read this

voltage are the medial finger phalanx and the palm. These are the best locations to measure, as they have many nervous terminations and, consequently, a high blood flow. One of the problems in these sites is the artefacts induced by movements of the hand or fingers. These artefacts have to be filtered out after the measurement.

Figure 2.5 shows a typical Electrodermal Response (EDR). After an external stimulus, the nervous system takes some time to process and react to that information (EDR latency). Depending on the stimulus, the reaction may generate an increase in the sweat glands and a corresponding increase in conductivity, which, in turn, amplifies the measured current (EDR amplitude). This amplification is not instantaneous, as the skin takes some time to adapt (EDR rise time). After a peak in amplitude, the body starts to calm down and the conductivity level fall to a resting value. Usually, measures of interest are the rise time, maximum amplitude and the time it takes to fall 1/2 and 2/3 of the peak. The maximum amplitude is important because it indicates the level of stress generated by the stimulus.

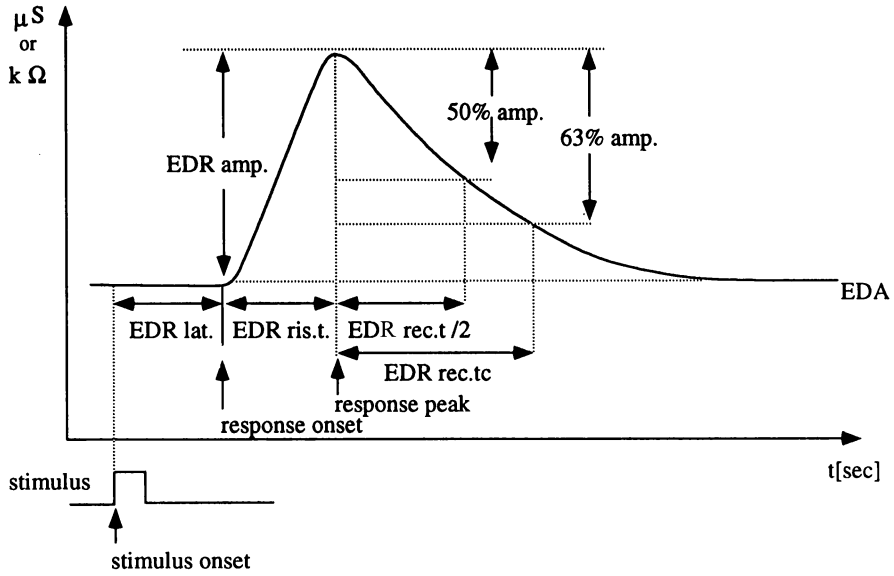


Figure 2.5: Eletrodermal Response (Boucsein, 2012)

In most experiments, the generations of stimulus is known. It can be a visual stimulus, like showing a picture to the subject to measure her reaction, for example. Some experiments also measure the EDR with unknown stimulus, also known as Non-Specific EDR (NS-EDR). In those cases, the idea of the experiment is to analyse the reaction of the subject during a period of time. Counting the number of peaks in the EDR is a typical measure of interest in those experiments. One problem related to this measures is the overlapping of EDRs. If a stimulus occur during the response of a previous stimulus, it can generate interference in the measure obtained.

The analysis of an EDA signal can be made in two ways: phasic or tonic. In a phasic analysis, each EDR is treated individually. Experiments that use phasic

analysis usually are in search of specific events. For example, a stimulus, like a picture or a sound, is presented to the subject and her reactions are analysed in that specific moment. In the tonic analysis, the experiment observes the variation of the Electrodermal Level (EDL) over long periods of time. The EDL is the resting value after an EDR event, and it can rise and fall during long experiments, particularly under stressful experiments. In a video lecture experiment, we can perform a phasic analysis during slide changes events, for example, and tonic analysis as the behavior of the student during the entire experiment.

Applications of EDA include stress monitoring and affective mental state classifications (Picard et al., 2001). Experiments show that cognitive states can be measured with EDA (Poh et al., 2010). EDA was also used on learning process to detect mental states (Cooper, 2011), but the results were inconclusive.

2.2.3 Facial Expressions

Since the works of Ekman (1993), studies have tried to automate facial emotion recognition with the help of computers (el Kaliouby, 2005). To infer the current mental state of a person, a computer uses a video camera to record the person's activity and analyzes its facial movements. Most emotions are expressed on facial signs (Ekman, 1993), so, a computer have a better chance to predict the current emotion by analyzing the video from the subject's face.

Frameworks to predict the emotion of a user based on the facial video feed are described in el Kaliouby (2005) and D'mello et al. (2005). In these frameworks, a computer first track a series of facial points. These are points in the face that follow a standard position, as show in Figure 2.6. Then, a computer can detect expressions in the face, called Action Units (AU), given by the movement of these points. This follows an standard, the Facial Action Coding System (FACS) (Picard, 1997), that shows the basic movements of facial muscles and their correlation with facial expressions. Although the expressions follow a standard, the process of correlation between FACS and emotions is still an open issue and authors have been trying different methods, searching for better results (D'mello et al., 2005; el Kaliouby, 2005).

The correlation between FACS Action Units and mental states from the learning process is an active area of study (Dragon et al., 2008; Rozin and Cohen, 2003). The action units that involve the eyes points and eye brows have shown important correlation with learning mental states (Rozin and Cohen, 2003). Eye movements also show correlations with the cognitive disequilibrium theory. On questions that involve pictures or graphs, a confused student tend to amplify the eye movements in the region where the picture or graph is located during the disequilibrium state(Graesser

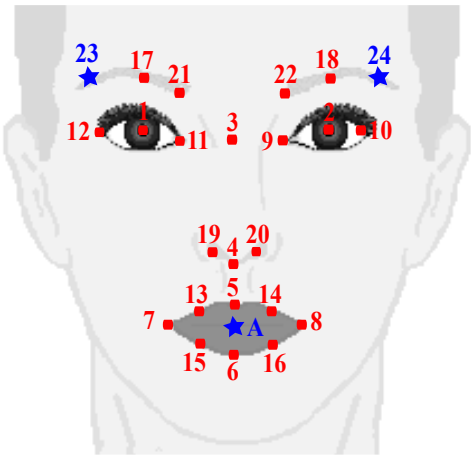


Figure 2.6: Facial Points (el Kaliouby, 2005)

et al., 2005b).

In studies that use the valence-arousal model(Dragon et al., 2008), mental states associated with activity, like Frustration and Confusion, have a better chance of being detected by facial expressions than mental states like concentration and engagement, as they show less activity. Those that show high activity can be detected mostly by head (Dragon et al., 2008) and eye movements (Rozin and Cohen, 2003). Particularly on education systems, Conati et al. (2013) adopt a standard metric in eye-tracking to measure attention: the total fixation time. This metric is related to the overall time a subject's gaze rest in from of the monitor. Similar experiments were executed by D'Mello et al. (2012).

Facial expressions also show correlation with difficulty and learning speed. Whitehill et al. (2008) and Lang (1995) made some experiments that show that the blink rate has high correlation with cognitive load, for example. These studies indicate that the eye region is more appropriate to predict states with less activity, like engagement. Pupil diameter, blink rate and blink interval are good measurements for cognitive states (Beatty, 1982; Kahneman and Beatty, 1966; Van Gerven et al., 2004).

2.2.4 Student Actions

Any action the user can make on the interface of the system is considered a user action. These actions are very related to the purpose of the system. Simulators can have a complex interface and the amount of actions generated by the user can be very high. Simpler interfaces, like those in question-answering systems, usually have only a dialog or multiple choices for the student to answer the question.

Question-answering systems rely more on the answer than on the action. The

answer can give much more information about the student's mental state, particularly on systems that interact with a natural language interface (Graesser et al., 2005a). Cooper (2011) used a summary of the user actions to predict the mental state. Indirect feedback given by the user actions, like the number of hints the user requests or the time she spends on solving a problem, have shown some correlation with mental states like interest or frustration.

Most Adaptive Hypermedia Systems base the user models solely on the user actions or actions patterns(Kobsa, 2001). This approach is similar to that used on recommendation systems, were just the interaction of the user with the system is sufficient to create a predictive model of the user.

In Hypermedia Systems, like learning systems based on video lectures, the user actions are basically the actions the user can perform when she is watching a video. de Vielmond et al. (2007) created a user behaviour model from the actions of play, pause, jump forward, jump backward and next slide. The learning system used in this work is similar to the one used in de Vielmond et al. (2007). The cognitive state prediction from these actions have yet to be explored.

2.3 Adaptive Learning Systems

Adaptive Learning Systems (ALS) are based on the idea that each student has a different learning style(Graf et al. (2009)). Some prefer a visual approach, like videos lectures, others prefer reading text books, and some prefer to do exercises to retain knowledge. The challenge is to detect the learning style and adapt the content to each student individually.

ALS are divided in two classes (Paramythis and Loidl-Reisinger, 2003): Adaptive Educational Hypermedia Systems (AEHS) and Intelligent Tutoring Systems (ITS). AEHS are directed to content learning. The student is presented with a series of hyperlinks to different content about the learning domain. The content is based on text, video and other kinds of medias. Those systems are indicated to students that like to study based on books and are close to the traditional learning. The most common inputs used to adapt the context of AEHS lectures are the interactions of the student, like the choice of the hyperlinks. ITS are based on problem solving. The student is presented with a series of exercises or tasks and the adaptations can be made based on the results of those exercises. Usually, AEHS also have a problem solving approach, but uses it only as a teaching strategy, while the ITS uses it as its main educational system. Traditional recommendation systems are usually Adaptive Hypermedia Systems (Romero et al., 2009), where the user navigates in the content of a site and this content adapts based on this navigation. For example, Amazon can adapt its interface and recommend new products based on the last

products the user have seen. Some ITS can also use recommendations to help the students solve the tasks (Schiaffino et al., 2008).

Figure 2.7 shows a traditional architecture for ALS (Magoulas and Chen, 2006; Paramythis and Loidl-Reisinger, 2003; Sottilare and Holden, 2013). These systems usually rely on three modules: The Learner Model, a Domain-Specific Knowledge and Instructional Strategies.

The Learner Model is responsible to track and learn information about the student. This model usually uses the information about the interactions of the user with the system, like page access and question-answer. More advanced models can use sensors to track the student mental state. The Domain-Specific Knowledge module contains all the lecture material. This includes pages, books, videos, etc. This is the content of the lecture. The Instructional Strategies are defined by the teacher beforehand. It is usually a set of rules of how to present the content based on information given by the Learner Model.

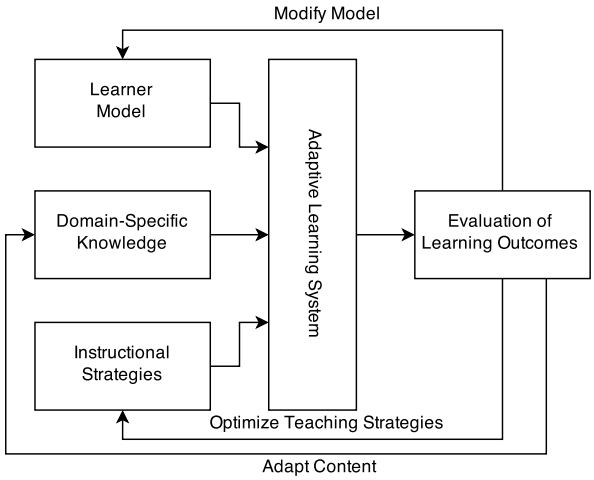


Figure 2.7: Adaptive Learning System Architecture (adapted from Sottilare and Holden (2013))

2.3.1 Adaptive Educational Hypermedia Systems

Brusilovsky (2012) divide the Adaptive Educational Hypermedia Systems (AEHS) into three generations. The first generation is composed of pre-Web hypermedia systems. The second are Web-based adaptive hypermedia systems. The third generation started in 2004. They are Web-based hypermedia systems that have adaptive tools that can track every interaction the student performs. These AEHSs use the collected information to adapt the knowledge domain during the student interaction. Two examples of third generation AEHS are the Adaptive Hypermedia Architecture and the Adaptive Educational System based on Cognitive Styles.

The Adaptive Hypermedia Architecture (AHA!) (De Bra et al., 2003) was created as a support architecture for online course with user guidance through conditional explanations and link hiding. Figure 2.8 show the adaptive technique used by AHA!. It consists of adaptive presentation, which shows or hide information from the user based on the user model, and an adaptive navigation, which colors the links also based on the user model. The user model is a series of attributes calculated from the user previous interactions. For example: if a student access a page, the system assumes that the student have acquired the knowledge presented in that page and increase a specific attribute. When the user access another page, it consults the previous attributes of the user to conditionally show or hide information, based on the user previous knowledge. The information of what to show based on the attributes are predefined on a series of rules specified by the lecturer.

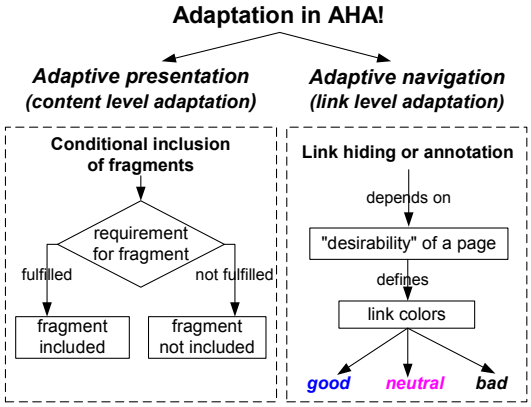


Figure 2.8: Adaptive System of AHA! (De Bra et al., 2003)

The Adaptive Educational System based on Cognitive Styles (AES-CS) changes the representation of the domain-specific knowledge to a linked list where the student can navigate to a specific knowledge faster (this is known as domain maps). This allowed the creation of larger knowledge bases. A similar approach is used in systems like QuizGuide and InterBook (Brusilovsky, 2012).

Usually, AEHS use simple user models based on rules. For example, if the student navigates through specific pages, her information is updated on a database and if she reaches a threshold, she is ready for more advanced topics or the next knowledge. So, the main difference on those systems is the domain-specific knowledge module and how it represents the information of the lecture. Advanced student models have been proposed (Brusilovsky and Millán, 2007), basically changing the deterministic, rule-defined student model to a more probabilist model based on bayesian networks, where the thresholds are probabilistic.

2.3.2 Intelligent Tutoring Systems

Most Intelligent Tutoring Systems are task-based systems that adapt its tasks based on the past interactions of the students. For example, in a question-answering ITS, the questions can increase in difficulty if the student is answering right most of the past questions. More advanced ITS started to rely on sensors in their adaptive module in order to adapt the lecture without the need of student interactions. Two examples of such systems are the Wayang Outpost and AutoTutor.

Wayang Outpost (Cooper, 2011; Dragon et al., 2008) is an example of ITS that presents problems to the students and asks for solutions from a list of multiple choices. The student can use hints in the form of text messages, audio and animations. To detect mental states on this system, a sensor that analyzes facial expressions from a camera input was used. This sensor is an extension of The MindReader software (el Kaliouby, 2005), which classifies the facial expressions to Mental states using a Hidden Markov Model. Mouse pressure, chair pressure and EDA sensors were also used. The system is also able to generate a stream of user actions, in the form of answers given by the user, but only a summary of these actions were used, like the time a user spends on a question. Figure 2.9 shows an overview of this adaptive system. To facilitate the portability of the adaptive system to others ITSs, the authors segmented the system in two modules, the ITS and the User Model System (UMS), which synchronises the sensors and apply the classifier. A simple classifier based on a linear model for each mental state is used. Mental states are based on those given by The MindReader software, which are confidence, frustration, excitement and interest.

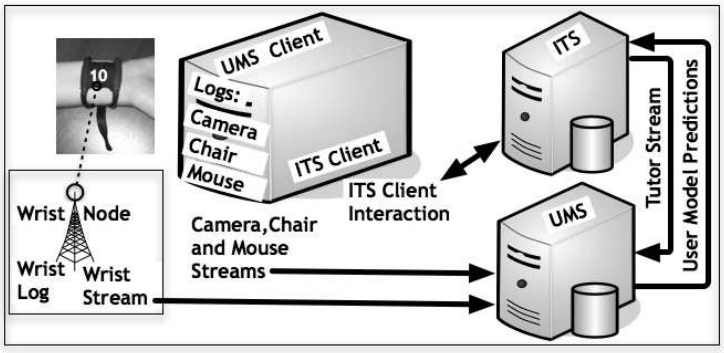


Figure 2.9: Wayang Adaptive System (Cooper, 2011)

Autotutor (Craig et al., 2004; D'Mello et al., 2007) is another ITS that can use sensors to detect the student mental state. In this system, the student has to answer a question about a topic in a short paragraph. The system helps the student by communicating with him through an interactive dialog box about the topic the student is learning. There are no user actions on the interface, as the user interacts

with the system using natural language. The camera sensor uses IBM blue eyes software, which focus on the facial expressions of the eye and head movements. Body posture is measured with a pressure chair (D'Mello et al., 2007). Figure 2.10 shows an overview of this system. The system does not use a single classifier, but a decision tree classifier for each sensor. Each sensor reports the mental state individually. Detectable mental states are boredom, confusion, engagement and neutral. The neutral mental states is identified as an "idle" state. This is the only system that work with the engagement mental state, a similar state that we are using in our work. They define engagement as a mental state in which the student is focused on the answering task. Autotutor have a mean accuracy for detecting engagement of 64% with the pressure chair sensor, while the camera sensor did not report the engagement state. The authors justify the lack of engagement classification from the camera sensor because this state don't generate overly expressive facial expressions.

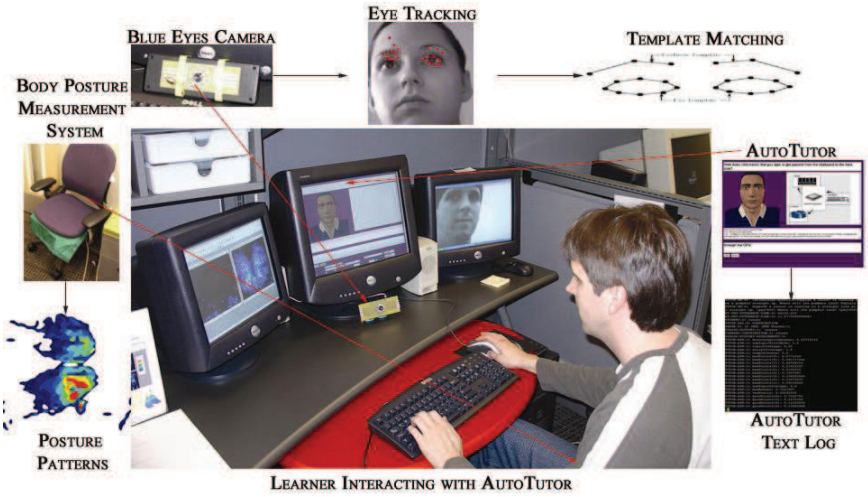


Figure 2.10: Autotutor Mental State Detection System (D'Mello et al., 2007)

Initiatives on standardization of sensor-based ITS also exist. The Generalized Intelligent Framework for Tutoring (GIFT) (Sottilare and Holden, 2013) is a research project from the US Army Research Laboratory. It's a modular architecture developed to address the constraints in present ITS. GIFT's main objectives are to provide authoring tools and experimental testbeds for ITS. The GIFT architecture consists of a learner model, that monitor the student using sensors and feedback; a domain model, that tries to adapt the lecture based on the feedback and evaluation of the student; and the expert model, that uses the feedback from the professor to optimize the learning strategy. A description of this architecture is presented in Figure 2.11. A basic implementation of the GIFT architecture exists as a game first person shooter tutoring system.

We also analysed other ITS approaches that are not sensor-based in order to

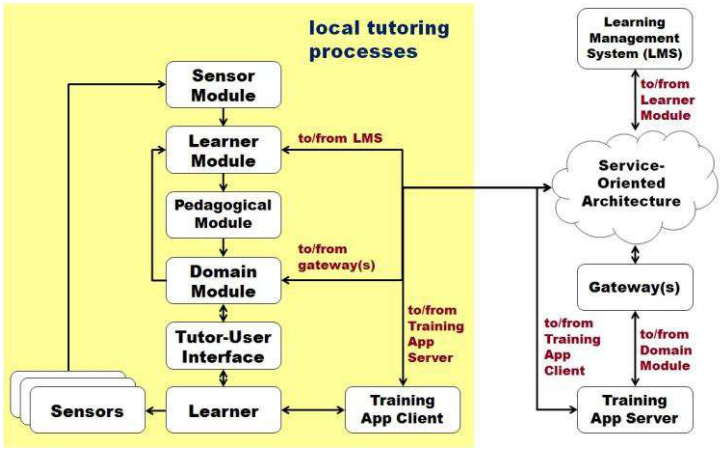


Figure 2.11: GIFT Architecture (Sottilare and Holden, 2013)

understand their adaptive module. One example is the SmartTutor (Cheung et al., 2003) proposal, from University of Hong Kong. This ITS has a student model based on a series of rules, which makes decisions on a method similar to decision trees. This model is used to give advices and recommendations to the user to help answering the questions from the tutoring system. A similar approach is used in the brazilian ITS Pat2Math (Jaques et al., 2013). eTeacher (Schiaffino et al., 2008) uses a probabilist approach based on bayesian networks to build a student model. This model can generate recommendations to the student, like study a different topic first or do more exercises. The model uses the student actions and profile (topic already seen) as an input.

2.4 Summary and Discussions

In this chapter, we defined how the engagement can impact the student during learning. We presented techniques to detect the engagement level of students and some systems that can adapt their content based on the behavior of the student.

One of the problems of AEHS is that it uses the student interaction for adaptation and, in these systems, the student usually has low interactivity. Most systems rely on preferences that the user set in the beginning and on tracking the user between multiple sessions. Besides, the adaptation model is based on predefined rules. On the other hand, modern ITS can use sensors to recognize the mental state the student is in and adapt the task she is performing. Some ITS can detect the user engagement based on her interactions with the system. The challenge is in developing a system that can adapt its content based on the engagement of the user with low interactivity.

As pointed out by Brusilovsky and Millán (2007), the main challenge of these systems is the user model. The majority of systems in use today rely on the profiles

of the user to adapt the system. These profiles store simple information about the user, like previous knowledge and learning goals. Brusilovsky and Millán (2007) conclude that future adaptation learning algorithms would be able to process each user's interactive behaviour information and simultaneously update the presentation to reflect this behaviour. This leads to a new class of adaptive systems which takes in consideration the cognitive information of the user and this is the main motivation in advanced systems today, like Autotutor. Also, in Brusilovsky (2012) it is mentioned that future systems should not be limited to the AEHS or ITS technologies alone, but should use a combination of these technologies to support the multiple needs of students.

Our proposed adaptive learning system behaves like an AEHS, adapting the lecture content to the student needs. Different from the existing AEHS, we overcome the problem of low interactivity in those systems by using sensors to monitor the student. To our knowledge, this is the first adaptive and recommendation learning system based on video lectures to employ sensors in the modelling of students, an approach only found in tutoring systems. There are other works that employ sensors but they have focus on affective states. As D'Mello et al. (2007) presented, engagement is important in the learning process and is difficult to measure. Autotutor is the only analyzed system to work with engagement, but the concept of engagement used in that research is an active attitude of the student towards the system. In our work, as video lectures are mostly passive to the student, our concept of engagement is different. We are using the attention of the students to the video lecture to characterize their engagement.

Chapter 3

Architecture

In this chapter, we present the architecture that we developed to support our adaptive learning system. The system is based on the RIO Multimedia System. This is a media distribution and presentation system that allows synchronization between different media types, like video and images. We implemented an adaptive module to the original RIO System. This module can recommend different media files to the student or automatically adapt the flow of the presentation based on active interactions of the student towards the lecture. We also developed an architecture that can interact with this adaptive module to give information about the student based on a trained student model.

In the next section, we present the RIO Multimedia System and our implemented adaptive module that extends the RIO System. Next, we show the architecture that captures the student information and gives feedback to this adaptive module, so the system can adapt the lecture without the need of student interactions, based only on sensors that monitor the student reactions. We also present the player system, which is a software implemented to support the development of our student model.

3.1 Background: RIO Multimedia System

The RIO Multimedia System is a client/server architecture. The server, named RIO Multimedia Storage Server, is a universal storage system and is responsible for the media distribution. It can store different types of media files, such as video, audio, text, images. Originally developed to store video files and 3D models at the University of California (UCLA) (Muntz et al., 1998), the server was designed to achieve high utilization and low latency, by using random allocation of disk blocks and replication. Later, the server was extended in a series of researches from our laboratory Land/COPPE-UFRJ (Botelho, 2008; Netto, 2004) to allow scalability and availability in the network, using replication of information in multiple servers. Currently, the National Research and Educational Network (RNP) offers

a service, Videoaula@RNP, deployed through the RIO Multimedia system. The Videoaula@RNP provides a service that enables the preparation, storage and delivery of video lectures. Our laboratory developed an interface to delivery the media files using HTTP protocol. This allowed a web browser to be used as a client.

The client side of the system is used for presentation of the media content. A client was developed for the CEDERJ consortium that requests the medias to the server and synchronize them to show to the student (de Souza e Silva et al., 2006). This client is also used in the Videoaula@RNP project. The main interface of the client is composed of a recorded video of the teacher, a list of topics of the lecture and slides, which can contain animations. Figure 3.1 shows an screenshot of the system client.

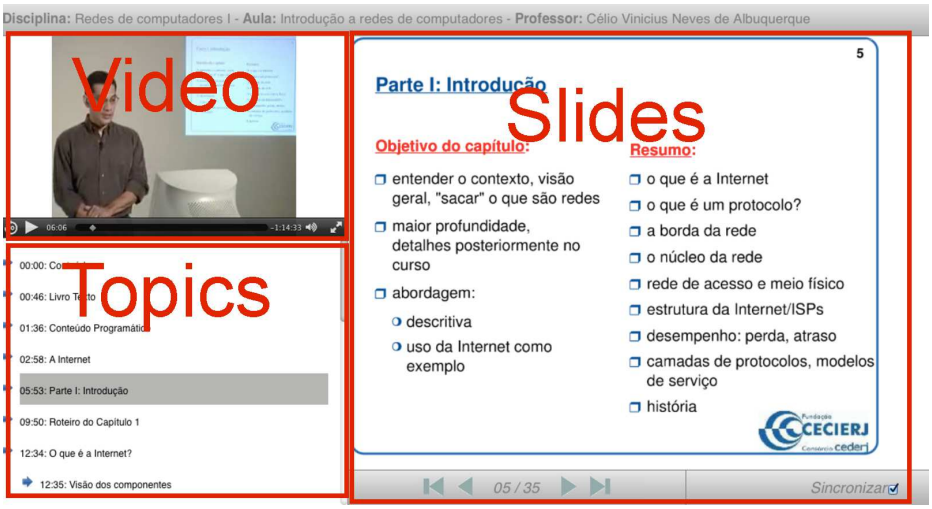


Figure 3.1: RIO multimedia client

The student can interact with the video lecture by moving the slide immediately below the video window to go forward and backward. In addition, he can select any specific topic in the list labeled "Topics" in the figure 3.1 and jump to this topic of the video lecture. While watching the video lecture, the student is able to navigate to review some concepts that may help him to understand the subject. To exemplify some of the features available, the lectures may ask the student to solve an exercise or initiate a simulator. The exercise or a pointer to the simulator would be indicated in the slide area. The student would pause to solve the exercise or execute the simulation and then resume the video lecture. The slides of the video lecture can be programmed to support interactivity with the students. Students may click on a slide to watch the solution of a problem or to request additional material (e.g. a PDF file, additional video files, external links, etc).

The current client is developed in HTML5 an is based on two main files: The video and the sync file. The video file sets the temporal information of the presen-

tation. The lecture duration is defined by the video file, as well as the timeline of the presentation. The client presents the lecture in a linear timeline, so the student can jump forward and backward in the video and all the associated material would follow. The sync file gives the timeline of the slides region. It is a series of XML elements with relative time associated to the video file. This time indicates when a specific learning object will be presented in the slide frame of the client. The learning object can be any HTML page containing image, video and even advanced resources like javascript interaction. A "time" attribute defines the time that the object have to be presented.

Listing 3.1: Example of RIO sync file

```
1  <slide time="1" relative_path="Topic01.html"/>
2  <slide time="90" relative_path="exercise.html"/>
3  <slide time="92" relative_path="Topic02.html"/>
4  <slide time="180" relative_path="Topic03.html"/>
5  <slide time="300" relative_path="end.html"/>
```

An example of sync file is presented in listing 3.1. The only element implemented in the sync file is the "slide", which indicates the time to present the media object defined in the attribute "relative_path". In the example, the sync file instructs the client to present the slides "Topic01", "Topic02", "Topic03" and "end" at the video time 1, 92, 180 and 300, respectively. The time is presented in seconds and the slides are HTML files. A special slide is presented at time 90: an exercise slide. This slide is an HTML file that allow interactions to solve a problem. The student have to manually pause the video to solve the exercise, or the client will present "Topic02" after two seconds.

3.2 The Developed Adaptive Module

To create an adaptive interface for the RIO client, we must allow communication between the different domain knowledge informations, like feedbacks from exercises. This is fundamental to support task based learning styles, like problem-solving in ITS. This interface must also allow conditional elements, like an AEHS. This allow the media objects to be conditionally presented to the student, based on a predefined attribute. Also, the current client allows the student to interact with the slides, but the client has no access to this interactivity. If the student is presented with an exercise in the slide frame, she must stop the video to solve the exercise, for example. And the client has no access to the student answers.

We modify the client to support an adaptive interface, following the architecture presented in section 2.3. Our Domain Knowledge Information is defined as the set of medias presented in the lecture, like videos and HTML pages. But the Instructional

Strategy, which is based on the sync file, must be modified to allow adaptation of the presented lecture.

We follow the design patterns defined by Brusilovsky (2003) and Wu et al. (2001) to develop a new sync file that can support conditional content. For that, we define elements that control the state of the variables in the system. The value of these variables can be defined by internal or external events to the client. For example, an element can be explicitly defined in the sync file to inform that the student watched a particular instant of the lecture (internal event) or an external object can modify the value of a variable, as the result of an exercise. Using these defined variables, we create a new conditional attribute that control the flow of the presentation. These conditional attribute can analyze the value of variables and evaluate a boolean expression, similar to what is done in SMIL W3C recommendation for adaptive time-based applications (Jansen and Bulterman, 2009).

To allow the adaptation of the lecture, we include three new XML elements in the sync file:

setvalue

Used to define the value of variables. Must have a "ref" attribute that indicates the name of the variable and a "value" attribute that indicates its new value. The value must be an integer.

pause

Can pause the video associated with the lecture. This element can use a new attribute "timeout", which defines that the video resume playing after the specified number of seconds. This attribute is optional.

goto

Alters the flow of the lecture. The video and all the associated material are set to the new time destination. The destination must be defined in the "jump" attribute. The value of this attribute is set in seconds.

A new attribute, "if", was created to allow the conditional control of the presentation. This attribute can be used in any XML element of the sync file, including the "slide" element. The value of this attribute is a boolean expression that evaluate to true or false. The expression must follow the syntax: `<variable> <operator> <value>`. The variable must be predefined with "setvalue" element. The operator can be "eq" (equal to), "lt" (less than) or "gt" (greater than). If an element has this attribute, it is evaluated prior to the execution of the element. If the expression returns false, the element is not executed.

To support the communication between the learning objects and the client, we define an Application Program Interface in javascript that gives the support for the

learning objects to alter the value of variables. The learning object must be an HTML file. The learning objects have access to the following functions:

RIOPause()

Pauses the video playback. The student can still interact with the object and can restart the video with the "play" button, located immediately below the video window.

RIOPlay()

Restarts the video playback. If the video is not in "pause" state, this function do nothing.

RIOJump(destination in seconds)

Alters the flow of the lecture. Works like the "goto" attribute in the sync file.

RIOSetVariable(Variable, Value)

Define the value of a client variable.

The system can also interact with the learning objects using the "action" attribute in the sync file. This attribute can call a javascript function defined in the object when a specific time arrives.

The Listing 3.2 shows an example of the new sync file. This sync file illustrates the following scenario: two topics are presented to the student and, depending on the student's answer to an exercise, a third topic is presented or not. If the student correctly solves the exercise, the lecture jumps the presentation of the third topic. In this video lecture, the video time is 360 seconds and it has 3 slides, one for each topic, and an exercise. A variable "result" has its value set to zero in the beginning of the lecture, to indicate that the system does not know if the student needs to watch the third topic. At 90 seconds, the student has to solve an exercise. The video lecture is paused and the student has 30 seconds to solve the exercise. The exercise send a command to resume the video earlier with the "RIOPlay" function and uses the "RIOSetVariable" function to alter the information of the "result" variable to one, if the student solves correctly the exercise. This is an indication that the student understood the topic and did not need additional information that is present in the third topic of the class. Even if the user did not solve correctly the exercise, the second topic is presented. At 180 seconds, the system evaluates the information in the "if" attribute of the "goto" element. If the expression is true, the system advance the video time to 300 seconds. If it is false, the video continue normally. So, if the student solved the exercise correctly, the third topic will not be presented. Figure 3.2 shows the timeline of this video lecture.

Listing 3.2: Example of RIO sync file

```
1  <slide    time="1"   relative_path="Topic01.html"/>
2  <setvalue time="2"   ref="result" value="0"/>
3  <slide    time="90"  relative_path="exercise.html"/>
4  <pause    time="91"  timeout="30"/>
5  <slide    time="92"  relative_path="Topic02.html"/>
6  <goto     time="180" jump="300" if="result eq 1"/>
7  <slide    time="181" relative_path="Topic03.html"/>
8  <slide    time="300" relative_path="end.html"/>
```

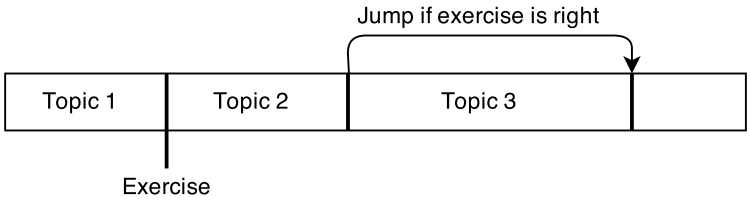


Figure 3.2: Timeline of Example Lecture

We also need to develop a mechanism that allows an external system, like a simulation software that was started by the client during the presentation or a monitoring application, to interact with the client in the same way a slide can interact with the javascript API. For that, we implemented a communication protocol based on the WebSocket interface (Hickson, 2012). This interface allows external systems to send messages to the client. We specified a protocol for the messages. Currently, the only message that can be sent to the client is one that changes the value of variables. So, an external system can indicate to the client the progress of a student, when, for example, she is doing an exercise. We also developed the mechanism for the client to send messages to the external systems using the WebSocket interface. The client can send messages generated by user events, like a topic change, or automatic events, like variable change or slide change described in the sync file. Using this information, an external system can track the state of the client.

The new adaptive module of the client allows the construction of adaptive lectures, but two premises have to be followed: all the adaptation has to be described by the lecturer in the sync file, in the form of variable changes, and all the adaptation is a result of a student interaction with the client, like solving an exercise. It is our objective to allow the adaptation to occur automatically, without the need of interaction of the student, based on his current engagement with the lecture. For that, we develop a system architecture that can use sensors to infer the state of the student and can automatically modify the client variables, allowing the lecture to change its flow without interaction. This architecture is described in the next section.

3.3 The Implemented Architecture

As presented in the last section, the implemented adaptive module in the client allows adaptability of the presentation based on a set of rules defined by the lecturer in the sync file. We are working with video lectures, a learning material that has low interactivity with the student. Also, the current implementation of the system does not track the student. This makes the adaptability of the lecture even harder, as we do not have information from past interactions of the student with the system. In the present configuration, a lecture that is only based on video material can have the same behavior for all the students, if the student does not interact with the video. Our objective with the developed system is to monitor the student while she is watching the video lecture and adapt the lecture for the needs of that student, even if she is passive towards the lecture.

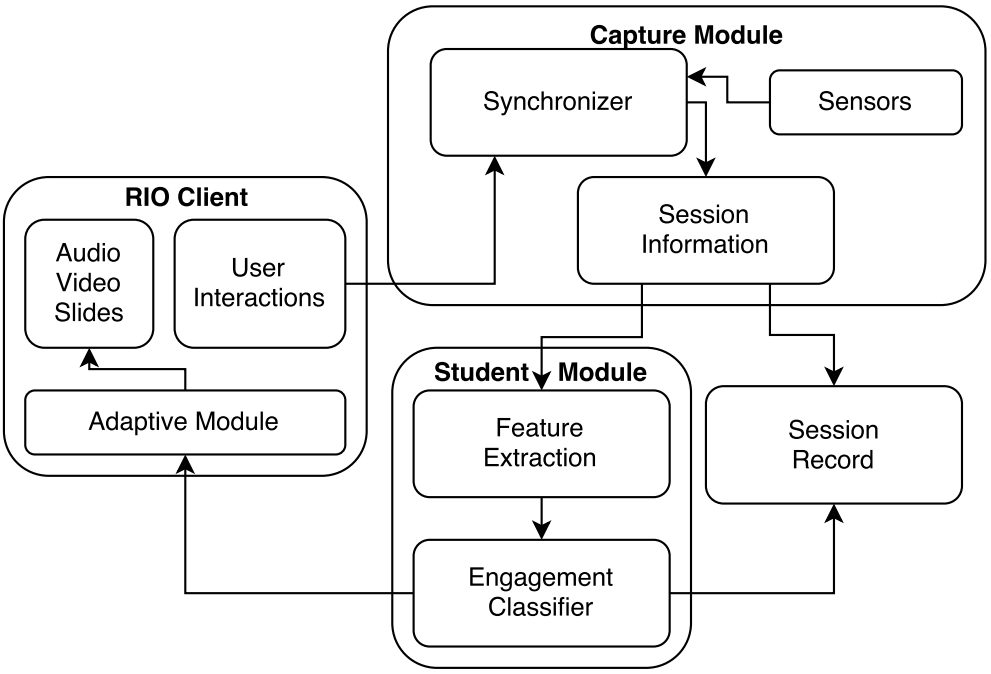


Figure 3.3: Overview of the System Architecture

Our system architecture is composed of two different modules that are interconnected: The Capture Module and the Student Module. The Capture module can monitor the student passively using sensors and the Student module can give feedback about the student engagement based on those readings. The Capture module is composed by the Sensors, Synchronizer and Session Information sub-modules. The Student module is composed by the Feature Extraction and Classifier. Each of these modules are presented in more details in the following sections. Figure 3.3 presents these modules and their interconnections.

3.3.1 The Capture Module

This module is responsible to collect and synchronize information from the student. The capture module consists of three sub-modules: Sensors, a synchroniser and Session Information. Figure 3.4 shows the interconnection of the sub-modules. The Sensors are the interface that captures the data. It can connect the system to hardware sensors, for example. The interface can be passive or active. A passive sensor interface can monitor information from students using hardware sensors, for example, while an active sensor interface needs an explicit feedback from the student, such as the answer from an exercise. We built two passive interfaces: one for hardware sensors that have outputs based on time series information and one for camera based sensors, which have an output based on frame image. The time series interface are used for sensors that reports data periodically, usually an integer or float value. Examples of sensors like these are EEG and EDA. Normally, each hardware has a different way to deliver this information and is necessary to build wrappers for these hardwares to communicate with the time series collector. The camera interface implements a standard video API to capture frames from different Operating Systems. In Linux, the system supports Video4Linux2 and OpenDV. In OS X, we use the native AV Foundation API and in Windows we use the DirectX API. The active sensor implemented is an interface to the WebSocket messages from the client. This interface can receive messages when the student makes an action in the client, like select a different topic.

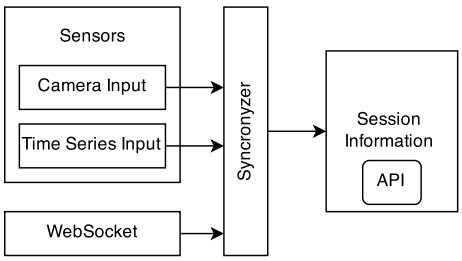


Figure 3.4: Capture Module

The synchronizer is responsible for the time stamp in each of the input modules. For the time series input module, we process each information in the rate of the sensor. Our system allows processing in a microsecond interval (10^{-6} seconds). In the case of the camera input module, each frame receive a timestamp, so we have a synchronization with other sensors based on the video frame rate.

All the collected data from the inputs is delivered to the Session Information module. This module organize the information so it can be passed to other modules in a standardized format. Time series information are converted to simple double precision stream of data, for example. Camera inputs are compressed using H.264

video encoding, with the time stamp stored inside. To allow maximum quality in the video frames, we use the lossless profile of H.264 as the default configuration, but this can be configured by the user. The API simple allows the streaming of this information via a local socket. To this moment, two modules uses this API. One is the Session Record sub-module. It uses the API to get the information of all the sensors and dump it to disk in a format that the Player System can reproduce. The other is the Feature Extraction sub-module from the Student module, which is described in the next section.

Although the capture module works in real-time, delivering information to the Student module for adaptation of the lecture while the student is watching it, the implementation of a sub-module that can record data, the Session Record sub-module, was necessary to create a database that can be used to build the student model, necessary for the Student Module. The methodology to create this database is described in chapter 4. The sensors described in section 4.2 are connected to the system using The Sensor interface implemented.

3.3.2 The Student Module

The Student module is composed of two sub-modules: The Feature Extraction and the Classifier. The primary objective of the Feature Extraction sub-module is to filter the inputs from the capture module to a more understandable time series that can be interpreted by the Classifier. For example, camera-based sensors have an output of frames in a specific rate. To allow the Classifier to process these information, the frames need to be processed to extract information, like the diameter of the pupil in an eye camera. We implemented a feature extraction mechanism for each input sensor. These implementations are presented in chapter 5.

The Classifier is the core that determines, in real time, the student's engagement level (based on its attention levels) while watching a video lecture. We defined two categories of engagement (attentive and not attentive) and the classifier can indicate the probabilities for the student to belong to one of the categories. These probabilities are made available to external systems using web socket. The Adaptive module in the RIO Multimedia client uses these probabilities to modify the content of its variables, for example. The Classifier uses a trained Student Model based on an unsupervised learning procedure known as Model-Based Clustering. A detailed description of the classifier is presented in chapter 6. Figure 3.5 shows the sub-modules interconnection.

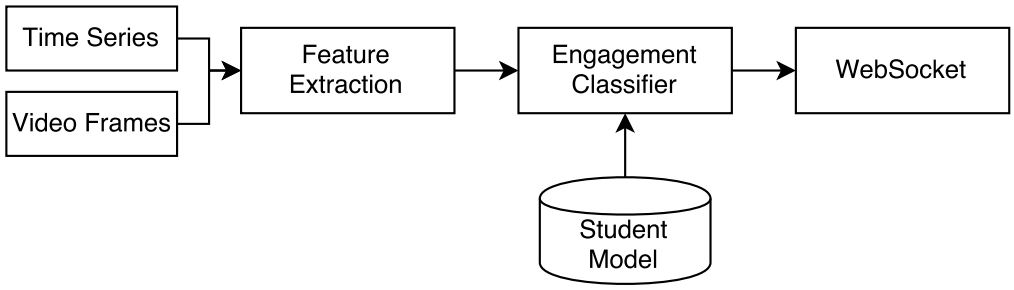


Figure 3.5: Student Module

3.4 The Player Software

The player is an auxiliary system developed to support the model construction and the test phase. It uses information recorded by the Session Record sub-module of the Capture module to playback all the information collected from a student during a video lecture. It can also work with the time series from the Features Extraction sub-module. Figure 3.6 shows a screenshot of the Player. This screenshot shows the player presenting a time series selected from one of the sensors, the output of the camera that captures the student's face, the output of the sensor that captures the eye surroundings and the video lecture being watched. The player can show overlapped time series in the same window or open additional time series windows, as well as additional video windows, as long as the Session Record module has recorded them during the same session.

Figure 3.6: The Player Software

Other information presented in the tool is classification information. A session can have one or more classification information associated with it. This information

is a float information in the range of [0 : 1] and is displayed in the time line of the player and in the time series window as a color intensity varying from red (0) to green (1). To this moment, this information is generated by the Student Module as the output of the classifier and by another external tool we developed to create subjective classification. This tool is presented in section 4.5. The Player can also be used as a tool to aid the lecturer to evaluate the student's reaction to parts of his/her video lecture.

3.5 Summary and Discussions

In this chapter, we presented two systems that were developed during this thesis: An adaptive system for the Rio Multimedia Client and an architecture that can monitor the student and report his engagement level. These two systems can interact with each other to adapt a video lecture based on the engagement of the student towards the video lecture.

The implemented adaptive system can be used by the lecturer to generate video lectures whose flow can be adjusted based on interactions from the student, like the answer to an exercise. The system uses basic programming language directives, like conditional statements and jumps, to alter the video flow. This allows the lecturer to develop complex adaptive video lecturers. We have a working prototype of this system and this is one of the contributions of this thesis.

We also have a working prototype of the system proposed architecture to passively monitor the student. To our knowledge, this is the first system that can measure the engagement information of a student in real time. We describe the modules of this implementation in the following chapters.

Our architecture is based on sensors that can monitor the student. To better understand the output of the sensors, we developed a supporting tool, the Player Software. This tool is generic enough to work with any time series in synchronization with the video lecture, and can be further extended to work with different sensors. This system is also available as a contribution of this thesis.

Chapter 4

MindLand Database

Our database was developed with the goal of creating a dataset for training and evaluation of the proposed user model for cognitive engagement. The database records the sensors output in response to students while watching an adaptive lecture created for our system, as presented in the last chapter.

We decided to create a new dataset in this work as the majority of the existing datasets uses actors or posed subjects. The few that use live recordings are based on an specific activity, like induced expressions (showing happy images to generate a happy emotion, for example).

We start this chapter by presenting the video lecture used in the data collection and the sensors used to monitor the students. Then we present our experiment methodology and some results from the collected database to this date. We end this chapter presenting a comparison of our dataset with others from the literature.

4.1 The Adaptive Lecture

We used an existing lecture of the data structures class from the Computer Systems Technology undergraduate course of CEDERJ. This lecture is presented by professor Jayme Szwarcfiter and originally have 44 minutes. We extracted 10 minutes from this lecture to use in this experiment. The extracted part represents the explanation of the Hanoi Tower game. We choose this specific part as it also includes exercises and some hints for the students to solve the game. As such, this specific part of the lecture can benefit from the new adaptive system of the RIO Multimedia Client, described in section 3.2. We use this lecture to train and evaluate our student model. We modified this lecture to make it possible to adapt according with the student response to the exercise.

The lecture has four distinct parts. The first is the explanation of the hanoi tower problem and have 390 seconds (6 and a half minutes). The second is an exercise. The student is presented with the hanoi tower game with 3 discs and is

asked to solve it. The time the student takes to solve the problem can vary. The third part contains a few hints the lecturer presents to help the student to solve the problem. This part has 130 seconds and is only shown to students that do not give the best solution to the exercise. We define the best solution as the one that uses the minimum number of movements to solve the hanoi tower game. In our exercise, the best solution has 7 movements. The forth part of the lecture is shown to all students and contains some curiosities about the hanoi tower problem. This part has 45 seconds. Figure 4.1 shows the flow of the lecture.

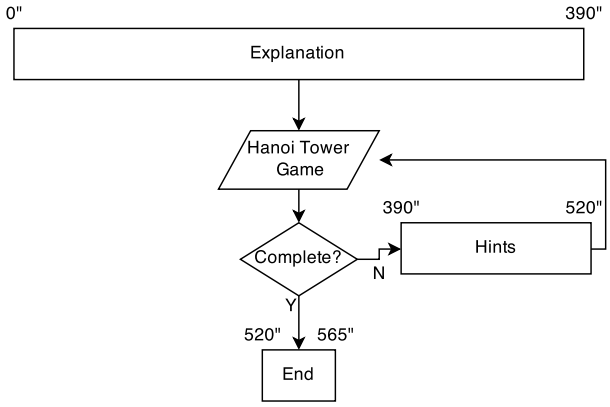


Figure 4.1: Lecture Flow

The flow of the lecture is controlled by the result of the exercise. The exercise is a hanoi tower game with 3 discs. In this game, 3 rods are shown and the first contains 3 discs arranged in descending order of decreasing disc size, the smallest being at the top. The objective of the game is to move the disc stack to the second rod using the third rod. The student can move only one disc at a time and no disk can be placed over a smaller one. Figure 4.2 shows a snapshot of the lecture.

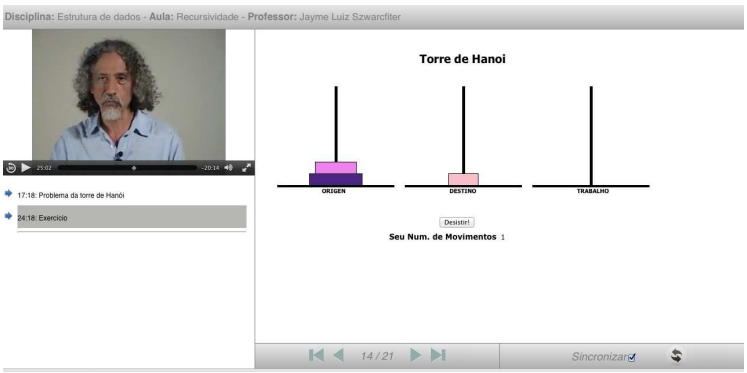


Figure 4.2: Lecture of the Hanoi Tower Game

The tower of hanoi can be solved using a specific algorithm. A minimum of $2^n - 1$

steps is required to move n disks from the first rod to the second rod. For people who do not know the algorithm, it is not easy to solve the puzzle. Crowley et al. (2010) show that, even for small values of n, the game can pose some challenge in terms of the time to complete the game and the number of steps. This information is used to adapt the lecture. If the student completes the puzzle but uses more than the minimum amount of steps, the lecture presents extra hints to the student and let her do the exercise again. When the student press the "Give up" button in the exercise page (build in HTML5), the following information is returned to the client: the time taken since the exercise began (in seconds), if the exercise was solved or not and the number of steps. This experiment was simple in terms of the possible ways a lecture can adapt. However we recall that more complex adaptive flow can be constructed.

4.2 Sensors used in this experiment

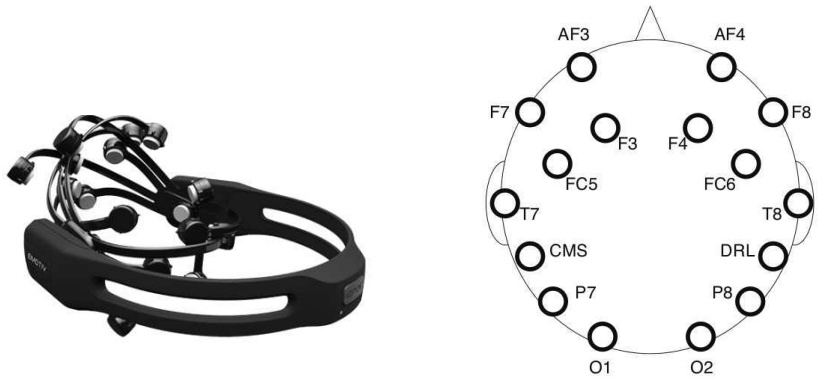
In this section, we describe the three sensors used in the experiment to monitor the student: Electroencephalogram (EEG), Electrodermal Activity (EDA) and an infrared eye camera. These sensors were connected to the architecture using the Sensor interface as presented in section 3.3.1. The database contains the raw information of each sensor, recorded using the Session Record sub-module of the Capture Module.

4.2.1 EEG

To capture the brain wave signals from the students, we use the Emotiv EPOC Scientific EEG Headset. Figure 4.3(a) shows an image of this device. This EEG device is a low-cost Brain-Computer Interface (BCI) that can capture raw EEG data and was already used in other studies (Duvinage et al., 2012; Liu et al., 2012). The EPOC Headset uses 14 electrodes placed according to the 10-20 location system. Figure 4.3(b) shows the electrode locations. It also has two reference electrodes (CMS/DRL) and a 3-axis gyroscope to measure the movements of the student head.

The electrodes needs direct contact with the scalp. Before each experiment, a saline solution is dropped on the sponge in each electrode. We use a contact lenses protection solution, easily found in drugstores.

The communication between the computer and the headset is made by a proprietary wireless device connected to the USB. This wireless device operates at 2.4 Ghz and allows a sampling rate of 128 Hz from each electrode. The Headset actually captures the EEG signal at 2048 Hz, but it is down-sampled to 128Hz to allow the communication with the computer. A Software Development Kit provided by



(a) Emotiv EPOC EEG Headset

(b) Emotiv EPOC Electrode Placement

Figure 4.3: EEG Sensor

Emotiv allows the capture of all the 14 signals and the gyroscope. To interface this SDK with our capture system, we developed a wrapper that delivers this information as a time series in our pre-defined format. This device can deliver 17 different time-series. One for each electrode containing the raw EEG signal at 128Hz and one for each axis of the gyroscope.

4.2.2 EDA

To measure the electrodermal activity (EDA) of students, we use the Affectiva Q Sensor. This sensor is a portable armband developed by MIT as the Handwave Skin Conductance Sensor, and latter commercialized as Affectiva Q Sensor (Poh et al., 2010). The sensor uses two dry silver chloride electrodes (Ag/AgCl) placed over the wrist.

Poh et al. (2010) and Boucsein (2012) show that, in cognitive tasks, the best electrode placement is at the fingers, therefore, we adapted the sensor to be able to collect the measurement at the fingertips. Figure 4.4 shows the used EDA sensor. We use the medial phalange of left hand index and middle finger in all experiments.

Real-time measurements were acquired using the bluetooth protocol available. The sensor can deliver the raw EDA signal in a standard serial bluetooth interface at 32 Hz. We developed a serial interface in our capture system and used this interface to read the EDA data during the experiment. This serial interface can be used with other sensors that use any standard serial protocols in addiction to bluetooth.

4.2.3 Camera to Capture the Pupil

We use an unmodified Logitech 920C camera to record the student face during the experiment. But, as described in section 2.2.3, the facial features that envolve the

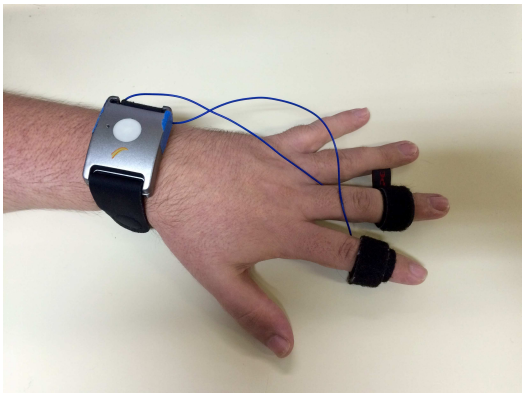


Figure 4.4: EDA Sensor

physiological responses from eyes have high correlation with engagement. So, we decided to use a specific device to measure the eyes features precisely.

We developed two different devices. One that records the student face, and another that focus at the pupil. This last device requires that the student wears a headset. We decided to develop our own device to measure the pupil due to cost constrains. Comercial devices that can perform this kind of measurements have a high price range, in the order of US$ 100K. We developed our device for under US$ 1K to meet our original requirement that the ITS would be a low cost system.

We develop the pupil sensor to track the pupil using infrared light. Infrared had been used in numerous works to track gaze movements. In an infrared tracking system, the eye is illuminated by a source of infrared light. Usually, this source is composed of infrared LEDs. This technique, also known as dark-pupil technique(Świrski et al., 2012), uses a known reflective property from the eye. Both the sclera and the iris strongly reflect infrared light while only the sclera strongly reflects visible light. The pupil can't reflect both. A camera that can capture infrared light, but filter visible light, is then used to capture the pupil. Both prototypes are based on this technique.

Our first prototype is based on a modified Logitech 920C webcam. This camera can capture imagem at fullHD quality (1920x1080 pixels). All normal cameras have a filter in the lens that remove infrared and only allows visible light to pass. We changed this filter to block the visible light and allows the infrared light to pass. We also built a source of 4 infrared LEDs of 840 nanometre (nm) of wavelength and 100mw/sr[1] of intensity. The camera and the light source were designed to be placed at a distance of 20 centimetres from the eyes, a normal range for a webcam. Vidyasankar (2013) used our first prototype to generate a series of experiments using our system. The result of those experiments are described in his master thesis.

[1]milliWatts per steradian, the LED radiant intensity information

The first prototype we develop could only capture the eye with low resolution. From this device we could get information of eye blink and gaze, but the pupil size measurement is compromised. Figure 4.5 shows a comparison of our first prototype and a normal camera. On the left of the figure 4.5 we show an image captured from our modified camera illuminated by infrared light. In the right side of figure 4.5 a "normal"camera is used. The eye is clearly visible in both cameras, but the pupil is only visible in the infrared. Even though we have a visible pupil, the resolution is too low to get a proper measure of the pupil size. The distance of the camera is too high, reducing the available resolution for the eye.

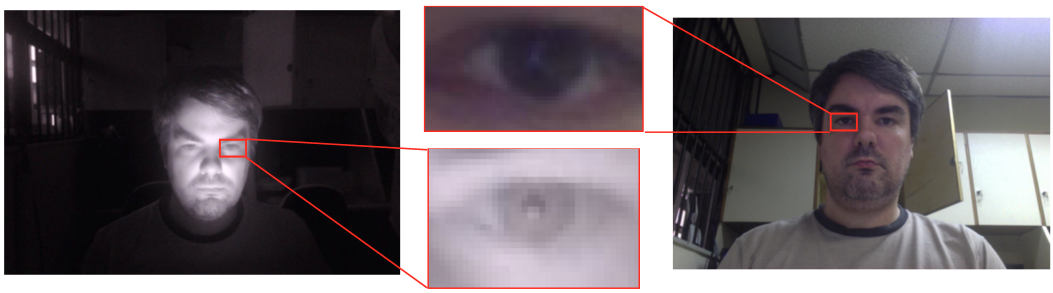


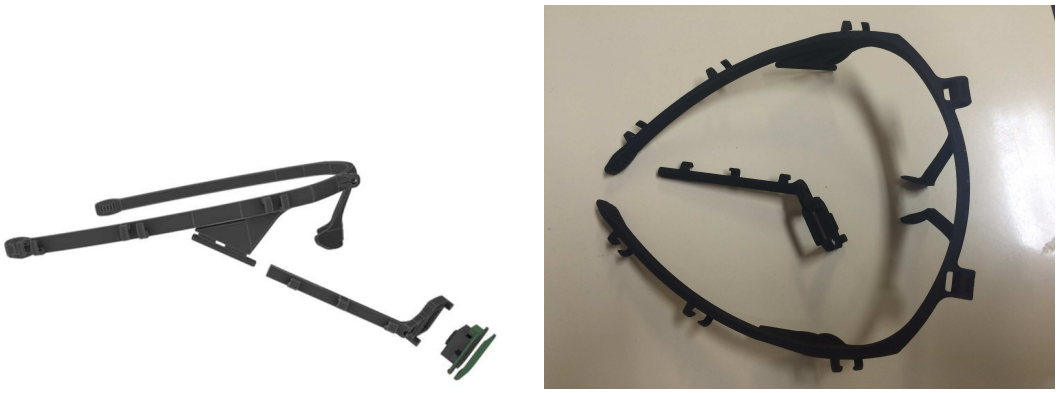
Figure 4.5: Comparison of the first prototype with a normal camera

In our second prototype, we increase the quality of the video captured from the eye by approaching our modified infrared camera to the eye. To place the camera at the proper distance from the eye, we developed a headset based on the specifications from Kassner and Patera (2012). This headset has a detached arm that can hold the infrared camera at 3 centimetres from the eye, in an angle that is away from the user's field of view. Figure 4.6(a) shows the 3D model of this headset and the camera arm. It was fabricated using a plastic polymer (ABS) on a Fused Deposition 3D-Printer. Figure 4.6(b) shows the actual fabricated headset.

In addition to the camera placement requirement, the second prototype had two other requirements: good image quality and a small size from factory to fit in the headset arm. We choose the Microsoft HD-6000 camera. This is a 720p camera (1280x720 pixels). The camera can be attached to the arm using its own lens holder.

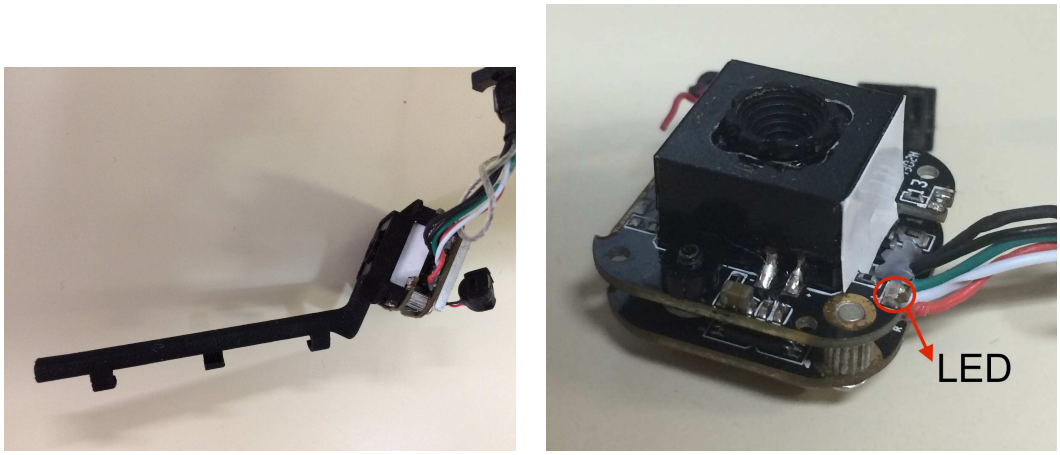
Figure 4.7(a) shows the camera attached to the headset arm. As in the first prototype, we changed the camera filter to allow the infrared light to pass and to block the visible light. The infrared light source is provided by a single infrared LED. We replaced the camera "on" LED indicator with a OSRAM SFH 4050 Infrared LED (850nm, 100mw/sr). Figure 4.7(b) shows the camera lens and infrared led.

Compared with the first prototype, this device has the disadvantage of being more intrusive. Figure 4.8(a) shows a user with the headset. However, the advantage in terms of high image quality by far outperforms the disadvantage as shown in



(a) 3D Model of Second Prototype (b) Second Prototype Headset

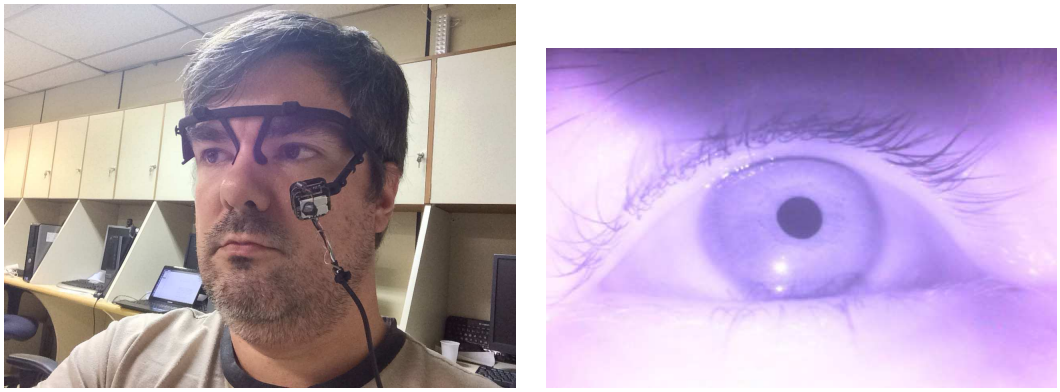
Figure 4.6: Headset and Arm of second prototype (Based on Kassner and Patera (2012))



(a) Camera Attached to Arm (b) Infrared Camera

Figure 4.7: Modified Infrared Camera

Figure 4.8(b). The eye image obtained from the camera shows a clearly visible pupil as the darkest circle in the image. This image should be compared with that at the bottom center of figure 4.5.



(a) User with Headset (b) Eye Captured with the Camera

Figure 4.8: Example of Eye Camera in Use

For safety reasons, we analyzed the device with respect to the Mulvey et al. (2008) report. This report is used to analyze the safety of infrared devices to the user, according to light emissions and distance to the eye. The report indicates that the only concern from our device specifications is the time the eye is exposed to the infrared light. The report presents two methods to analyze the duration of exposure: lower than 16 minutes ($t < 1000s$) and higher than 16 minutes. Our device has no known harm to the eye for both cases.

For our database, only the second prototype was used to record the raw video of student's left eye. We developed a module that can extract the features from the raw video as multiple time series. This module is presented in the next chapter. Our database comprehend only the raw video of the eye, to allow the data to be used in the future to extract new features or to enhance the developed feature extraction module.

4.3 Methodology

4.3.1 Participants

19 healthy participants (2 females), aged between 20 and 28 (mean age 22,5) participated in the experiment. All of them were students from the first semester of computer science course in the Federal University of Rio de Janeiro. Prior to the experiment, each participant was instructed on the manner by which sensors collect the data. The students were asked to explicitly consented to participate in the experiment. They were also instructed to inform of any medication they have

taken, including coffee, as it can indicate abnormal activities in the sensors (Roth et al., 2012). They were all given the opportunity to leave at any point during the experiment.

4.3.2 Experiment Setup

Each student executed the experiment individually and were accommodated in a chair in front of a Core I7 computer and a 23 inch LED display. The computer was running Ubuntu Linux 14.04 and our capture system integrated with the RIO system that was running in the Firefox browser. An isolated room was used for the experiment. Lights and air conditioner were regulated to maintain a constant flow and not affect the sensors or the attention of the student.

After an initial presentation of the experiment and consensus from the participant, the sensors were placed and the signal of each sensor checked individually. We develop a check interface coupled to the capture system to allow this procedure to be performed fast. This interface shows a green light on each sensor if it was placed correctly and the associated data is captured in a reliable form. After this initial check, the student had a few minutes to familiarize with the RIO client. An instructor showed the basic procedures of the system and explained how to interact with the lecture. When the student was comfortable, the instructor started the recording and leaved the room.

The experiment starts with a 10 seconds idle procedure. During this interval, the student is instructed to relax and a fixation cross is presented in the monitor. This part is used as a baseline. After this procedure, the lecture starts. Figure 4.9 shows a student during the experiment. The total amount of time a student spends during the experiment is the sum of the time spent in each part of the lecture. Given that the time spent in the exercise is a random variable and the student can wrongly answer the exercise, having to do it again, the total time can also be expressed as a random variable. In section 4.4 we present statistics concerning the amount of time the experiment took. When the student completed the lecture, the instructor returned to the room to stop the capture system and remove the sensors.

4.4 Results

From the set of students that participated in the experiment, only one decides to quit the lecture without concluding the exercise. Figure 4.10 shows the ordered time spent by each student in the exercise part of the lecture. On the average, students took a little more than a minute (67 seconds) to conclude the exercise with the minimum amount of moves. Nine students (47%) failed to complete the exercise

Figure 4.9: A Student During the Experiment

on the first attempt. On average, these students took more that two minutes (133 seconds) to do the exercise with more moves or to press the "give up" button. After this step, the student is presented with additional explanation about how to solve the problem and is offered a second attempt to solve the problem. In the second attempt, 77% of students solved the hanoi tower problem in seven moves, the minimum amount. Only two students had difficulties, one dropped the experiment and the other needed extra explanation to solve the game. Those students that solved in the second attempt took 69 seconds on average to solve the problem. This is similar to the results obtained for the students that answered in the first attempt and shows that the adaptive mechanism can help the students in the learning process, which is one of the objectives of an Intelligent tutoring system. Table 4.1 shows a summary of the results obtained from the exercise part of the lecture.

In this experiment, we collected over three hours of data. From that, over two hours are from the first part of the lecture and over 40 minutes are from students doing exercises, summing all attempts. Table 4.2 shows the total time collected during the experiment, in minutes, of each part of the lecture.

4.5 Subjective Classification

As presented in section 1.1, our goal is to develop a model to classify the cognitive engagement of students while watching a video lecture. In order to help the develop-

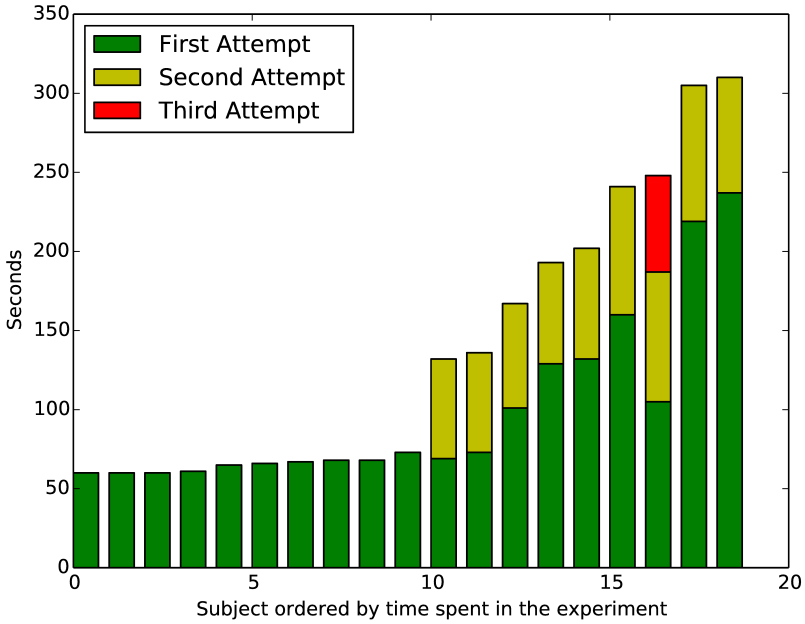


Figure 4.10: Time spent by each subject in the exercise

ment of our model and to analyze its accuracy, we need a way to subjectively classify the information that was recorded from the students. Other databases (el Kaliouby, 2005; Koelstra et al., 2012; Lucey et al., 2010; Soleymani et al., 2012) usually employ a participant self-report, experienced judges or previously classified information that can be correlated with the collected data. The problem with self-report is that the participant can be biased in the classification. Suppose we ask a student to indicate if she is engaged in a specific part of the lecture. Most of the answers will be yes, as the student really feel she is attentive, even if she was not.

Previously classified information needs a reliable source that can classify one or more of the signals to correlate with the other unclassified signals. Although many studies have presented a way to classify individual sensors, like EEG, a consensus between the authors is far from reach. In this research, we decided to use experienced teachers in presential lectures as the judges in the classification process.

The frontal facial camera was chosen as the classification attribute. We segmented the video of the facial camera in 30 seconds intervals. To reduce the number of videos to classify, we only select the first part of the lecture in this experiment, and only 11 students from the database (6 that correctly answer the exercise and 5 that not). As the first part of the lecture have six and a half minutes, each student produced 13 video segments of 30 seconds each. We remove the first 30 seconds from the first part, as it is the setup of the experiment and may have noise. We used 12 video segments from each student in this classification process, giving a total of 132

Subject	Leave the Experiment	Time (in seconds)		
		First Attempt	Second Attempt	Third Attempt
1	N	160	81	-
2	Y	237	73	-
3	N	129	64	-
4	N	65	-	-
5	N	66	-	-
6	N	105	82	61
7	N	60	-	-
8	N	69	63	-
9	N	73	63	-
10	N	60	-	-
11	N	61	-	-
12	N	73	-	-
13	N	101	66	-
14	N	60	-	-
15	N	68	-	-
16	N	67	-	-
17	N	68	-	-
18	N	219	86	-
19	N	132	70	-

Table 4.1: Exercises Results

video segments. We had to reduce the number of students to classify to reduce the burden generated to the teachers. Each classification session in this configuration already took over an hour to classify all the 132 videos and many teachers can't spend that much time. To facilitate the process, we developed a web system that allow the teachers to classify the videos at anytime. This system is presented in the next section.

Lecture Part	Total (minutes)
1	123,5
2	43
3	19,5
4	13,5
Total	199,5

Table 4.2: Total Duration of each part of the lecture Collected in the Experiments

4.5.1 Manual Classification System

A web system was developed aiming at performing a subjective classification of each video segment as mentioned above, and experienced lecturers were chosen for this task. Each lecturer is identified by an email she has to input before the system initiates a classification. If the user closes the browser before completing the classification process of all the videos in the system, she can return to the system at any time to classify the remaining videos. The system was developed using PHP5 and HTML5. It is compatible with modern browsers and is available in the internet to allow a lecturer to classify the videos anytime, anywhere. Figure 4.11 shows the classification screen presented to the user.

Figure 4.11: Classification System

When each video segment is presented to the user, three selections are available: Attentive, Neutral and Not Attentive. The user may select one of them by pressing the corresponding button located below the video. The video plays in loop until the user presses one of the choices. Note that the user can pause or skip the video at any time. The user is instructed to be precise in his answer. We instructed the user to choose the "Neutral" button in case she is not sure of the answer.

After the user selects among the three available choices, the system automatically presents the next video segment. In addition, the classified video is removed from the list of unclassified videos for this user. If the user chooses to skip the video, it is maintained as unclassified and can be presented to this user in the future.

We give priority to present the videos with smallest number of classifications. From the list of videos with smallest number of classifications, we give priority to those videos with ambiguous classification. For example, if a video is classified by one user as attentive and by another as not attentive, this video has priority to reduce the ties. If there are more than on tie, or no ties, the videos are chosen with uniform probability. When a lecturer classifies all the videos, the system shows a message congratulating her and ends the session.

4.5.2 Rater Reliability

The Rater reliability is a measure of the degree of agreement among the raters. The ideia of this metrics is simple: if the raters agree with each other on a classification, the information is considered reliable. But if they are split in their opinion, then we can't trust the classification and it must be discarded. In this context, we need to evaluate the raters and the information. For instance, we may have raters that are not well engaged in the process, or not well trained. These raters usually choose the classification randomly. In addition, the information itself can cause the raters to disagree with each other. If an information is too difficult to classify, usually we have an unreliable classification.

Many metrics are proposed in the literature to measure the reliability of the classified information (Banerjee et al., 1999; Fleiss et al., 2004). The simplest method is to calculate the percent of agreement. The Cohen's kappa coefficient and its variants are widely used as a measure of reliability between raters (Fleiss et al., 2004). Kappa extends the percent of agreement by considering the probability of chance agreement, that is, the probability of two rates to agree with each other even if they choose the classification at random. In our study, we will use a generalization of the kappa statistics presented in Landis and Koch (1977), called intraclass correlation coefficient (ICC). This generalization allows multiple raters and different raters per video segment, which is the case of our experiment.

Let n be the number of video segments to classify, and m_i be the number of classifications that the ith video segment has not considering the Neutral classification. Let x_i be the number of attentive classifications related to the ith video segment. Clearly, $m_i - x_i$ is the number of not attentive classifications. We remove the neutral classifications, as it is an indication of "doubt". We can now define two parameters: the overall proportion of attentive classifications, given by $\bar{p} = \sum^{n} \frac{x_i}{m_i}$, and the mean number of ratings per video segment, given by $\bar{m}$. The overall agreement of a video segment (Inter-Rater — SVS) and between each video segments (Intra-Rater — BVS) are defined as follows.

$$SVS = \frac{1}{n(\bar{m}-1)} \sum_{i=1}^{n} \frac{x_i(m_i - x_i)}{m_i} \tag{4.1}$$

$$BVS = \frac{1}{n} \sum_{i=1}^{n} \frac{(x_i - m_i\bar{p})^2}{m_i} \tag{4.2}$$

For example, if we have a high value of SVS, it means that the raters have a high agreement in the classification of a specific video segment, as the rates are near the mean rate of that segment. And if we have a high value in BVS, it means that we have a high agreement for a rater in each video segment, as her rates in each segment are near the mean rate of that segment. The final measure for ICC can be estimated as a proportion between Inter and Intra-Rater, when n is large, as shown in equation 4.3. Note that we can increase the agreement by removing the segments that have a low Inter-Rater agreement (those were the rates are far from the mean value) or by removing the raters that have a low Intra-Rater agreement.

$$\hat{r} = \frac{BVS - SVS}{BVS + (\bar{m}-1)SVS} \tag{4.3}$$

$\hat{r}$ can range between $-1/(\bar{m}-1)$ and 1. The higher the $\hat{r}$ value, the better the agreement, with 1 being the perfect agreement. Fleiss et al. (2004) and Landis and Koch (1977) have analyzed the values that $\hat{r}$ can receive and have characterized different ranges of values with respect to the degree of agreement. Values below 0.4 are considered poor agreement, while values above 0.75 represent excellent agreement, above chance. Values between 0.4 and 0.75 are fair to good agreement, beyond chance.

4.5.3 Results

15 lecturers have participated in the classification experiment. 8 are faculty members of the Federal University of Rio de Janeiro. The remaining 7 have, at least, two years of experience in lecturing for undergraduate students. On average, each lecturer classified 62.1 video segments (standard deviation of 45.5). Four lecturers classified all the 132 video segments. On average, each video segment has 7.1 classifications (standard deviation of 1.5). Figure 4.12 shows the distribution of classifications.

To calculate the rater reliability, we removed the video segments that have more than 60% of "Neutral" classifications. This leaves our database with 79 video segments, with an $\hat{r}$ value of 0.411. We then removed the three raters with the lowest Intra-Rater agreement (those that gave rates that are far from the mean rate of each segment). The 12 remaining raters have a $\hat{r}$ value of 0.614, which is considered a good agreement. From the 79 video segments, 48 are classified as attentive by the

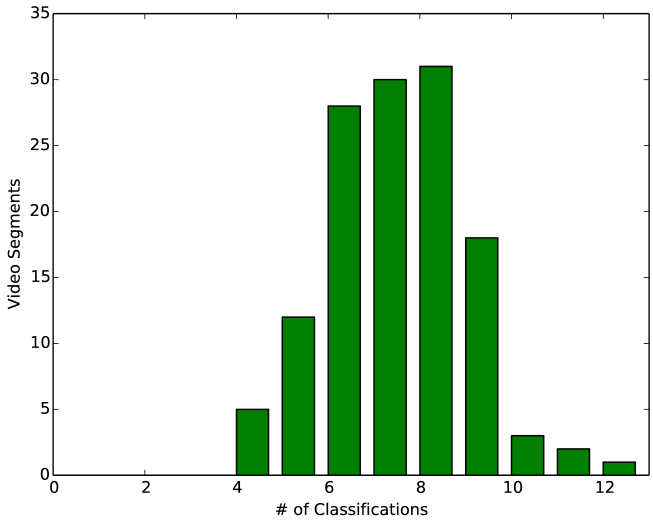


Figure 4.12: Distribution of Subjective Classifications

teachers and 31 are not attentive.

Figure 4.13 shows the classification of the segments during the first part of the lecture. Each 30 seconds of the first part have 11 video segments, one for each student. For example, the first 30 seconds have five segments classified as attentive by the lecturers. We included the video segments that have more that 60% of "Neutral" classifications in the figure, for reference. It is interesting to note that the segments in the end of the first part of the lecture are more difficult to be classified by the lecturers, since they have more neutral classifications than those in the beginning of the experiment. Also, we can see a consistent droop in the number of attentive classifications towards the end. An explanation of this behaviour can be that, as the students became tired, they lose the attentive state.

4.6 Other Databases

Many databases exist in the literature comprehending facial expressions and sensorial recordings. These databases were created mainly for research in affective recognition. One of the major drawbacks of these databases is the use of acted or posed expressed emotions, due to the fact that deliberate behaviour differ from spontaneously one (Zeng et al., 2009). In these databases, actors are hired to deliberately express the emotions while been recorded. In these cases, only the frontal video is available, as the physiological response does not exist. Examples of these databases are the CVPR (el Kaliouby, 2005), which comprehend of 16 volunteers acting in different mental states, and CK+ (Lucey et al., 2010), which is a collection

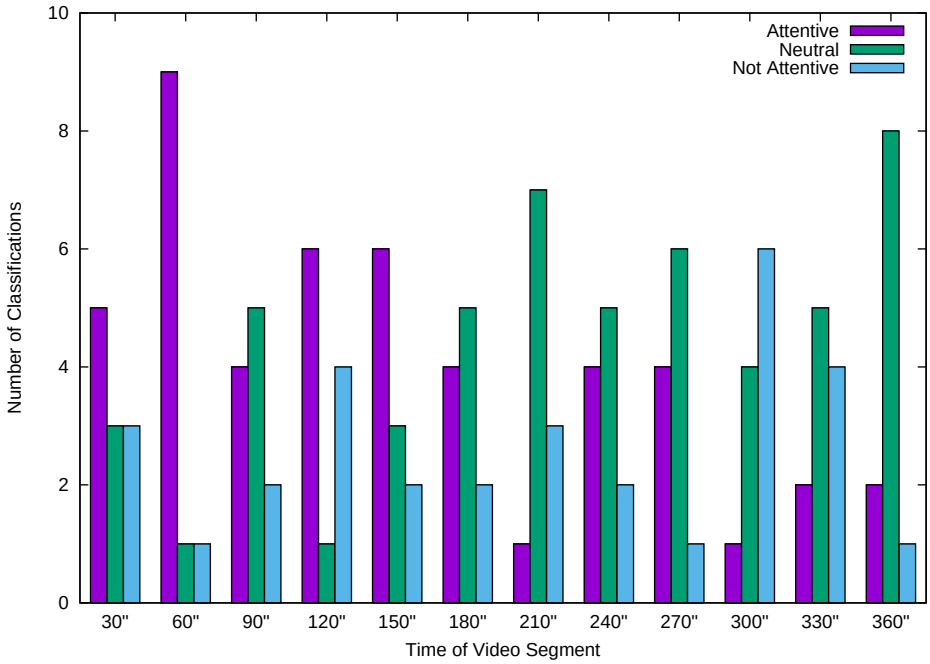


Figure 4.13: Classification over Time of the Experiment

of images of posed faces.

In the last few years, spontaneous emotional databases began to appear in the literature. These databases are recorded with real users, but usually with induced emotions. In these cases, users are presented with a stimuli to express the emotion, like a short horror video to trigger a fear response. The DEAP (Koelstra et al., 2012) is a collection of 22 participants watching videos of one minute. Frontal video was recorded, as well as physiological data. The MAHNOB Database (Soleymani et al., 2012) also used videos as stimuli. This database has 27 participants and recorded face video and physiological data. This is the only database we analyzed that included eye information, like pupil size and blink rate, although it did not include the video eye itself. A large review of other induced emotion databases can be found in Zeng et al. (2009).

Our database has the unique characteristic of being a natural expressed emotions database. Different from induced databases, the natural ones are recorded without a stimuli, in an environment close to real environment. One example of natural databases is the HUMAINE (Douglas-Cowie et al., 2007). It is a collection of videos from different TV programs used to classify emotions. Different from our database, HUMAINE database does not contain physiological data in natural environment and much of the dataset is copyright restricted.

One important characteristic of the studied databases is the labelling procedure. This is how the user information is classified, based on a set of labels. The databases of the literature usually use informations like valence and arousal, or affective states,

like happiness and sadness, as labels. None of the databases have attentive state as a label. The labelling procedure is done using self-report in almost all of the databases that do not use acting, in which case the label is the acted expression. The only one to report the Rater Agreement is MAHNOB Database, with a value of 0.32.

4.7 Summary and Discussions

In this chapter, we presented how we built a database that can be used in the studies of cognitive and affective states from students interacting with video lectures. We didn't find a freely available database that incorporate this student's specific task.

Using our system, we built a database composed by over 3 hours of session time, with 19 students and over 18 time series from 4 different sensors. The size of this database is comparable to existing ones. However, our database is unique since we employ live recording of real students (not actors or posed users) with natural reactions and used more sensors than those used to build existing open databases. In addition, we extracted features different from those included in the existing databases, as presented in the next chapter. This is another contribution of our work.

In summary, the contributions presented in this chapter are: the interface to 3 different sensors (EEG, EDA and infrared eye camera), the development of an eye camera prototype to capture pupil related features and an extensive database of students interacting with video lectures, with synchronized physiological data from students and attention labels. We also presented an initial analysis of the classification done by experienced professors.

Chapter 5

Feature Extraction

The main objective of this chapter is to extract the relevant information for our student model and classifier from the database built after collecting data from different sensors, as detailed in chapter 4. In most cases, the attributes from an isolated entry in the database do not provide enough information to the problem we want to solve. We can draw a simple analogy using the image of an object in a picture file. A single pixel does not contain enough information to infer what the object is. We need many pixels and the way they are displaced in space to recognize the object. Likewise, working with the raw data from time series is usually difficult and normally the data contains too much information. A "high-level" information must be extracted from these series for algorithms to recognize a pattern. These non-linear combinations of the original inputs are called feature extraction or feature construction (Murphy, 2012).

Each type of input can have different feature extraction functions. For example, it is common to apply signal processing functions, like Fourier transforms, to time series in order to extract frequency information. On the other hand, pattern recognition algorithms are usually applied to video information. We worked with three different sensors, EEG, EDA and an eye camera, and in this chapter we present the functions used to extract the relevant features from these sensors.

5.1 EEG Features

In the literature, many studies have proposed EEG features to evaluate attention levels and cognitive engagement (Antonenko et al., 2010; Basar et al., 2001; Klimesch, 1999). One of the most used features is presented in Klimesch (1999). Most studies are based on experiments that use *event-related potentials*, in which users are presented with a stimuli, like a warning signal prior to an event, in order to measure their reactions. Klimesch (1999) experimented with semantic memory performance, a well known reference of processing information, and sustained attention. In addi-

tion, he presented a feature that uses the band power in the range of 6Hz to 10Hz, which made the information simpler to compute. On our study, we use the feature presented in Klimesch (1999).

Our raw EEG data is composed of 14 channels. Each channel is the signal of a different electrode site placed in the head, as presented in figure 4.3(b). The sample rate is 128 samples per second. We start the analysis of raw EEG by removing the artifacts with a standard threshold method (Delorme et al., 2007). The threshold we use is based on 3.5 times the standard deviation of the signal. After that, we applied a Short Time Fourier Transform over one second with a Hamming window of two seconds (256 samples) to extract the frequencies between 6 Hz and 10 Hz. The final value is the sum of amplitude powers in each frequency in this range. This gives a time series with one sample per second. We applied this procedure on each channel individually. We also removed the channels that presented too much noise. Klimesch (1999) shows that the power is topographically widespread over the entire scalp and uses the mean value of the final power of each electrode. The Emotiv EPOC EEG headset that we use give the information of quality measures in each electrode, so we use only the electrodes that presented the highest quality information.

Klimesch (1999) shows that a decrease in band power in the range of 6Hz and 10Hz is an indication of attention or semantic memory performance and, consequently, an increase in the power may indicate a not attentive state. To measure this behavior, he uses the z-score, which is given by $z_t = \frac{x_t - \mu_t}{\sigma_t}$, where x_t is the band power measured at time t, μ_t is the mean value and σ_t is the standard deviation in the interval $(0 : t]$.

Figure 5.1 shows the EEG feature calculated over the EEG signal of a student. This figure was generated by the player system. The green line is the feature. We also presented the moving averages of the feature for better visualization. The green and red marks are the periods classified as attentive and not attentive by the teachers.

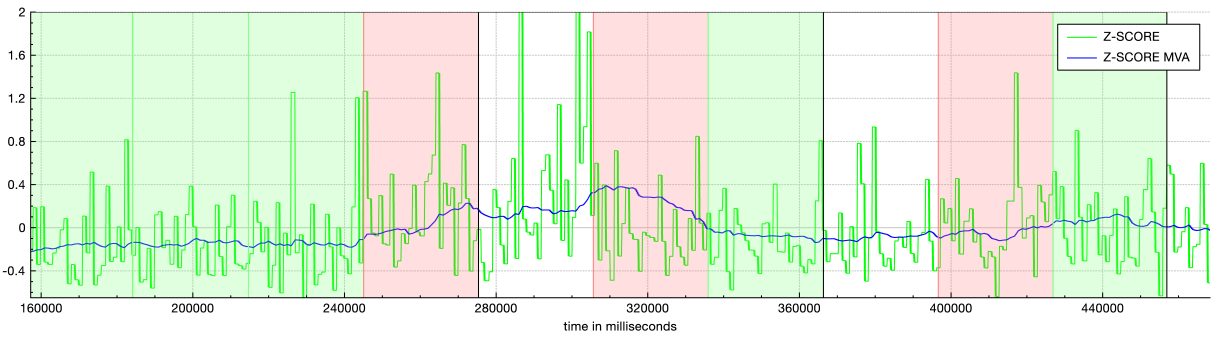


Figure 5.1: EEG Feature Extracted from Student 2

5.2 Eye Features

The use of eye based features in learning systems is relatively new. Most studies focus on eye-tracking features (Conati et al., 2013; D'Mello et al., 2012), but it is known that other eye features, like the pupil size and blink rate, are relevant in the process of engagement (Beatty, 1982; Kahneman and Beatty, 1966). Therefore, we also decided to extract and analyze these two features of the eye: pupil size and blink rate in our studies to try to detect attention. We also extract the eye gaze information, based on the pupil tracking, for future analysis.

In order to collect eye features we use the infrared video captured with our pupil camera, as described in section 4.2.3. The video was recorded in H.264 format at 10 frames per second. To analyze the video and generate the time series for the eye features, we extract all the frames from video in individual JPEG files, with respective timestamps. For each frame, we compute the pupil size and blink information. To process the video and extract the features, we developed a C++ system using the Open Computer Video Library (OpenCV) version 2.4 (Bradsky and Kaehler, 2009).

5.2.1 Pupil Size

To measure the pupil size, we developed an algorithm that can track the pupil in an image and measure its area, in pixels. Our algorithm is composed of three steps:

- Find the darkest blobs in the image,
- Find the blob that is most similar to an ellipse,
- Calculate the center and area of that ellipse. The pupil size is the area of that ellipse.

The first task of the algorithm is to find the pupil in the frame image. We assume that the pupil is the darkest blob in the image that is most similar to an ellipse. A blob is a region of the image that have most of its pixels following a specific property, like brightness. Figure 5.3(a) shows a sample of frame image. We first convert the image to grayscale based on the brightness of each pixel. So, all pixels have a value between 0 and 255, with 0 being the darkest. To reduce the noise in the image, we apply a Gaussian filter of size 9 and standard deviation of 2. This is simple a convolution of the image with a Gaussian function. The resulting image is smoother, with more pixels following the same bright value of its surroundings. After that, we apply a simple threshold function to create a binary image. All pixels that are above a specific value are set to zero, while all the others are set to one. The problem is how

to determine the best threshold value. We use the histogram of the pixel values of the grayscale image to determine this threshold(Kim et al., 2014). Figure 5.2 shows the histogram of the frame in Figure 5.3(a), after applying the Gaussian filter. Our implemented approach seeks for the lowest value in the histogram and increase this value in 30 pixels. This value is used as threshold. We experimented with other values for this increment and found that a range of 30 pixels is enough to segment the pupil pixels. Figure 5.2 shows this segmented area in purple. Although based on a fixed value, this approach is simple. Kim et al. (2014) made some experiments with this method and they show that it can give good results. Nevertheless, in the future, we can automate this procedure by using other approaches, like clustering algorithms (Świrski et al., 2012), to found the best increment for each frame. Figure 5.3(b) shows the image after applying the threshold function. The pupil and some of the eyelashes are clearly visible.

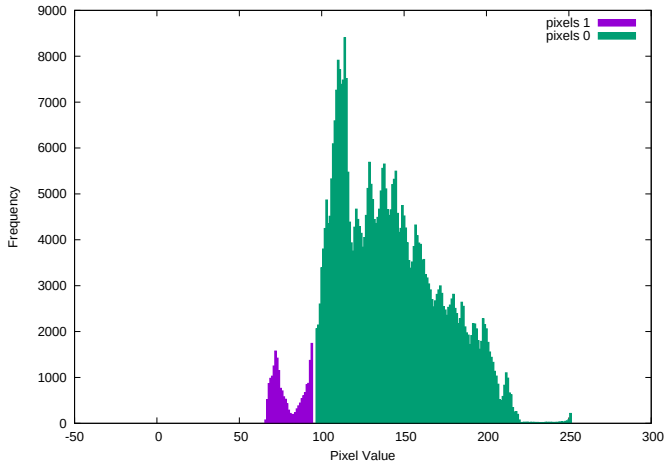
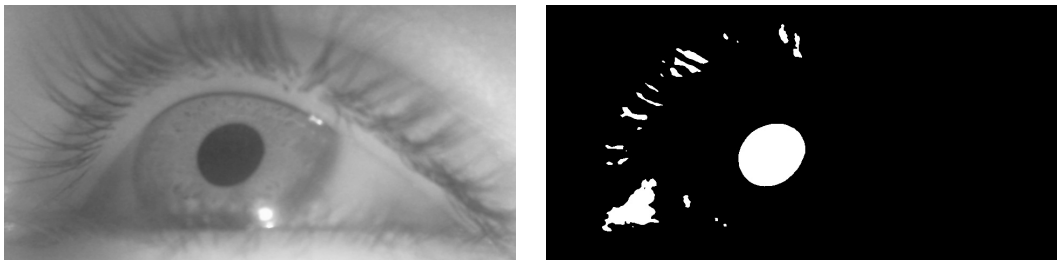


Figure 5.2: Pixel Histogram of the Filtered Frame



(a) Sample frame of eye camera

(b) Thresholded frame

Figure 5.3: Sample frame before and after threshold function

After we segmented the image in blobs that are composed of the darkest pixels, we need to discriminate the blob that is most probably the pupil. We assume that the pupil is the blob that is most similar to an ellipse or, in the best case, to a circle.

This part of the algorithm is divided in two sections: find the points that constitutes the border of the blobs and fit these points in a ellipse fitting algorithm. To find the points that belong to the border of each blob, we use the algorithm developed by Suzuki and be (1985). Because of its simplicity, this is one of the most used algorithms to find contours. The idea is to do a raster scan on the image and mark all the pixels with value one that have a pixel with value zero in its surroundings. These are border pixels. A composition of connected border pixels mark the contours of a blob.

Due to occlusion problems in the pupil, like when eyelashes are over the pupil or glints caused by reflections are visible, we use a convex hull algorithm. By definition, a convex hull is the smallest convex region enclosing a specific group of points. Using the original border pixels, we use the proposed algorithm of Sklansky (1982) to remove the occlusions. We then applied an ellipse fitting algorithm to the resulting group of points. The ellipse fitting is based on the proposal of Fitzgibbon and Fisher (1995), which minimizes the mean square error of the ellipse parameters at the given set of points. The result is the center, width and height and angle of the estimated ellipse that enclose all the pixels of the blob.

To detect if a specific ellipse is the pupil, we defined some constrains:

- The ellipse shape must resemble a circle. We empirically detect as a pupil the ellipses with ratio between the width and height higher than 85%. Kim et al. (2014) uses a value of 75%. We made some experimentations with a higher value and found that 85% can give a better detection rate with a small number of missed pupils.

- Since we are working with continuous frame sequences, the center of the pupil in a frame must not be distant from the center of the last frame pupil. So, we choose the ellipse with the lower euclidean distance between the two centers.

- The area of the pupil does not increase or decrease too fast. We empirically choose the ellipse that is not higher or lower than the pupil in the last frame by a factor of 0.3.

Using these thresholds, we return the information of a pupil ellipse if at least one of the blobs pass in all constrains. If more than one blob passes, the last one is returned. If none passes, an indication that the image have no pupil is returned. From this data, we return two time series: one that have the center information of the pupil and other that have the size of the pupil, which is the area of the ellipse. Figure 5.5 shows the implemented algorithm. Figure 5.4 shows the original frame with the detected pupil (the blue point is the center of pupil in the previous frame).

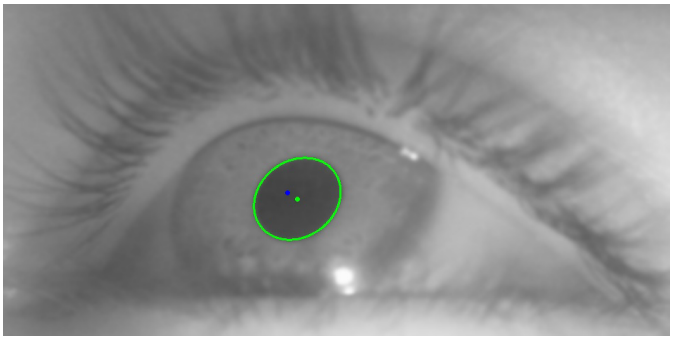


Figure 5.4: Frame with detected pupil

To evaluate if our implemented algorithm provides a good estimate of the pupil size, we use our player software. The player can show the image of the eye and the extracted time series both synchronized in time. Figure 5.6 shows 6 seconds of the extracted time series of pupil size. The pupil size is presented in number of pixels from the ellipse area. We overlapped the pupil image over the time series for better reference. We can observe that the pupil can almost double its size during the test. Comparing the pupil size with the pupil image at the same time, shows that the extracted time series is accurate. Beatty (1982) shows that the pupil diameter can increase during difficult tasks, like solving a problem.

5.2.2 Blink Information

The blink information is computed based on the eye state in the frame. We need to detect when the eyelids are open or close. For that, we use a property of the infrared image of the eye. When the eye is closed, the eyelid occupy a major area in the eye image. Using infrared light, the eyelid is much brighter than the pupil (Kim et al., 2014). When the eyelid is open, the pupil and much of the eyelashes are visible and the image becomes darker. We use again the pixel value histogram of the filtered frame, however, for this task, we build two histograms: one made from previously selected frames of open eyes and the other made from closed eyes. The selected frames were extracted from the student database. Figure 5.7 shows the histogram of open and closed eyes. Open eyes have much more pixels with bright intensity below 150, while closed eyes have more pixels with bright intensity above 200. For reference, one of the images used for open eyes is shown in figure 5.8(a), and one for closed eye is shown in figure 5.8(b).

The blink state algorithm starts by using the information from the pupil size algorithm. If the pupil is found, the eye state can be defined as open, otherwise, the state is undefined. The undefined state occurs because the pupil can still be visible in the frame, but some of the constrains are not met. For example, when

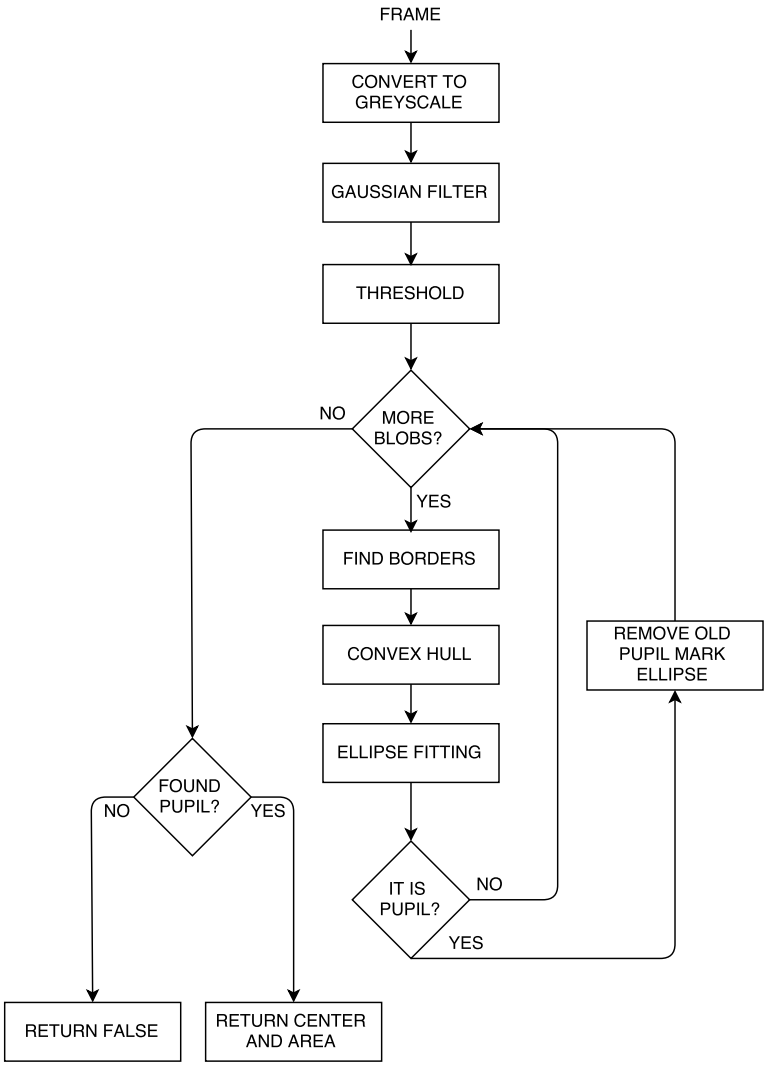


Figure 5.5: Implemented Algorithm to Detect Eye Pupil

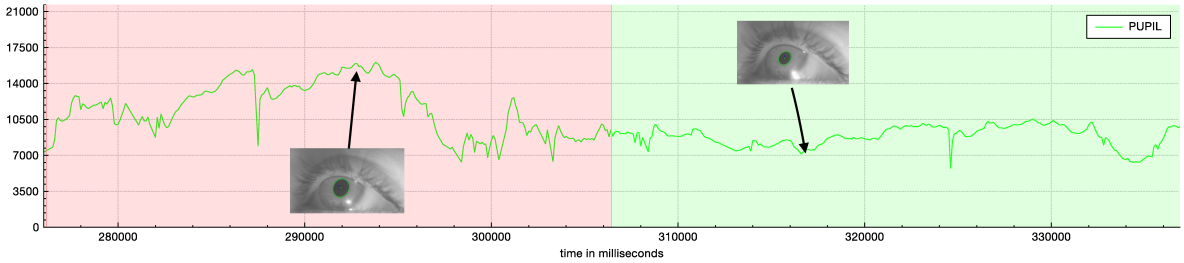


Figure 5.6: Pupil size extracted from student 11

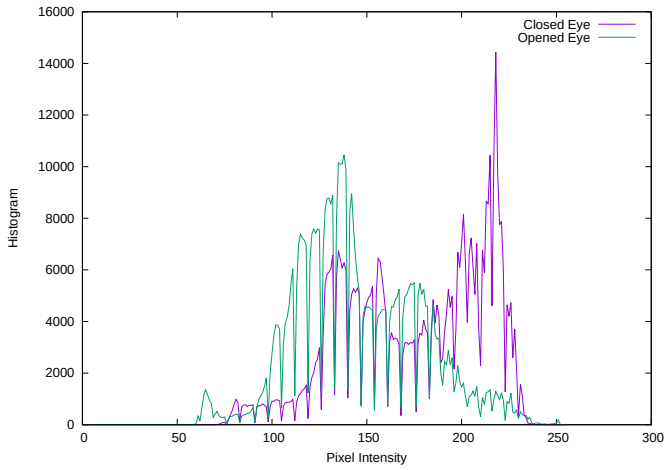
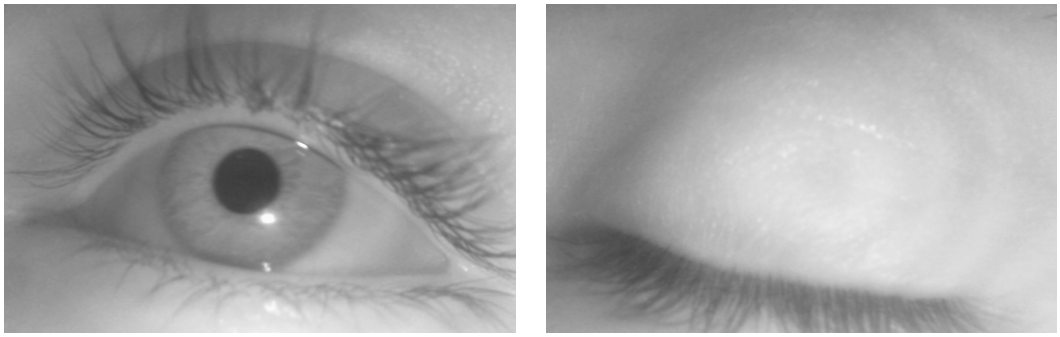


Figure 5.7: Histogram of Open and Closed Eyes



(a) Open Eye

(b) Closed Eye

Figure 5.8: Sample Frame of Eyes

the student is looking to one side, the pupil have a shape that is elliptical and will not meet the circle proportions. If the state is undefined, we have to search the histograms to set the blink state. So, for each frame that is set as undefined, we compare the histogram of that frame with the calculated histogram of open and closed eyes using the Kolmogorov-Smirnov test. The blink state is set as open or closed eye, depending on the results of the Kolmogorov-Smirnov test.

The output of this algorithm is a binary time series. A value of one is assigned if the eye is closed, and zero otherwise. Figure 5.9 shows the blink state extracted from a student. The figure shows two seconds, divided in intervals of 100 milliseconds. This is the time between each image captured from our eye camera (recall that we use a camera with frame rate of 10 frames per second). In the figure, three blinks occur, one with duration of 100ms, one with 200ms and the last with 400ms. This was measured by the number of consecutive frames with value one. Because of the frame rate from the eye camera, blinks less than 100ms are difficult to measure.

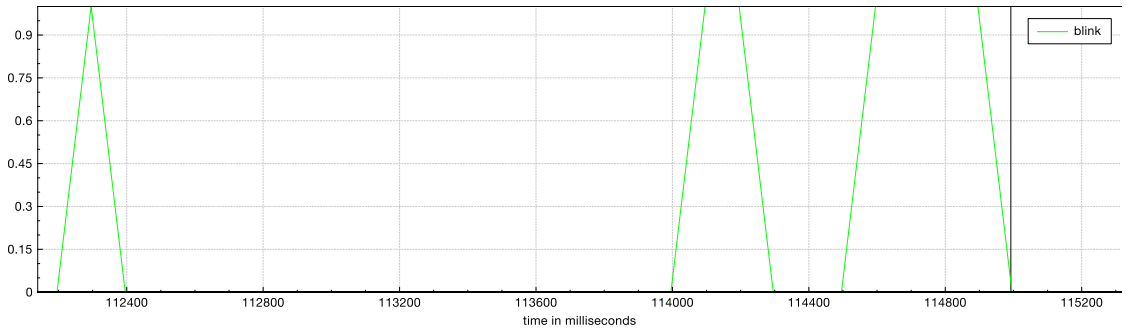


Figure 5.9: Blink State extracted from student 11

5.3 Other Features

In addition to the three features presented in the last sections, we also processed two more features: one based on the EDA sensor and the other based on the head movement, extracted from the EEG sensor.

For the electrodermal activity, we follow the research presented in Cooper (2011), as he is using the same sensor that we use in our work. Cooper (2011) performed a tonic analysis over the original raw signal (recall section 2.2.2). The tonic analysis uses the electrodermal level to measure the level of stress of the student. A high EDL is an indication of high level of stress. As the stress is also an indication of arousal, we will use the EDL as a feature. The raw signal that the EDA sensor outputs is the EDL information, but it needs to be normalized, as each student have different expected levels. For that, we used the z-score as presented in the EEG

Table 5.1: List of Available Time Series

Sensor	Time Series
EEG	Power Signal from 6 to 10 Hz
	Head Movements
Eye Camera	Pupil Size
	Pupil Gaze
	Blink State
EDA	EDL Signal

feature, but with the mean and standard deviation over a period of 30 seconds.

Figure 5.10 shows the EDL of a student during the entire period of the experiment (six and a half minutes). The student presented a negative value until the half of the experiment. After that, his EDL starts to raise and remains high to the end, with a peak in the last minute. This indicates that the student is calm in the beginning of the lecture, and starts to stress towards the end (Boucsein, 2012). Prokasy (1973) shows that attention is the inverse of EDL (more stress leads to less attention).

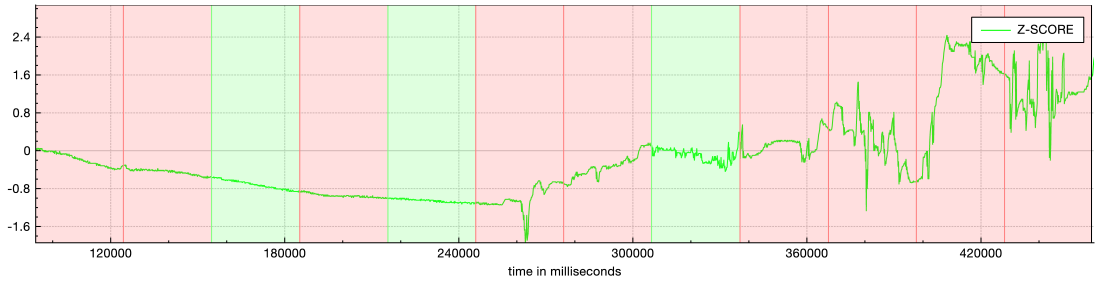


Figure 5.10: EDL extracted from student 11

5.4 Summary and Discussion

In this chapter, we described how we extracted a workable time series from the database information obtained from the sensors presented in the last chapter. These time series are used to extract relevant information for the student model that we present in the next chapter. Table 5.1 shows a summary of the extracted information from the sensors.

To extract a viable time series to work with, we implemented a signal processing procedure for each sensor. Although many works exist in the field of feature extraction, there are just a few implementations that can actually be used, as these extractions are heavily dependent on the sensor information. Particularly, for the eye camera sensor, many approaches exist to find and track the pupil. However, those approaches are usually proprietary. In this chapter we present an implemen-

tation that can extract pupil size information and data related to the blinking of an eye from an infrared eye camera. Our objective with this implementation is to allow the classifier to use these features. The lack of open source implementations make it difficult to compare our results with others, but is our intention to further explore this area.

We can resume the contributions in this chapter as:

- a new algorithm, and its implementation, that can extract: (a) the pupil size, (b) blinking duration and (c) gaze information from video captured with an infrared camera attached to the front head of the user and;

- an implementation of an EEG filter for attentive states, using results which are presented by Klimesch (1999).

Both implementations work in real time to process data acquired from sensors. These implementations are used in our system's architecture as described in section 3.3.2.

Chapter 6

Student Attention Classifier

One of the most important building blocks of our architecture, as presented in section 3.3, is the classifier. The classifier is responsible for infering the student's engagement state and the output is used to decide if the lecture being watched should or should not be adapted. In this chapter, our objective is to determine if it is possible to develop a classification model that would detect, in real time, the engagement level of the student using the system.

While most AEHSs use simple student models which are based on a set of predefined rules, systems like ITS, considered more advanced than AEHSs, try to predict the student mental state during the execution of tasks and games. The student model used in these systems are usually based on a supervised learning algorithm, where judges or the user inputs information about the student's mental state. This can pose a major problem to the model, as supervised machine-learning algorithms need accurate ground truth information to train the model, and mental states are difficult to judge (D'Mello et al., 2007). Different from existing approaches, we use an unsupervised learning algorithm in the search for patterns that can be interpreted as an *attention state*. Recall that we use *attention information* to measure the cognitive engagement. In our work, we use the time series extracted from the database collected, as presented in chapters 4 and 5, to generate the observations that are clustered by a Model-Based procedure (Bouveyron and Brunet-Saumard, 2014; Fraley and Raftery, 1998).

We start this chapter by presenting the procedure used to generate the observations. We also introduce the Model-Based Clustering algorithm employed. We then present our results followed by a discussion on these.

6.1 Features and Observations

Our observations are based on the segmentation of a particular topic taught as part of a video lecture. This same topic was used for the experiment that includes

faculty members that were asked to rate each segment as described in section 4.5. Recall that we divided the lecture in segments of 30 seconds each. We worked with 11 students and each generated 12 video segments, producing a total of 132 video segments.

To generate the observations for the classifier, we used only a single value for each 30 seconds segment of the whole time series extracted from each sensor. This procedure is called *summarizing a time series* Cooper (2011); Murphy (2012). and was necessary to generate data compatible with the ratings done by faculty members. Briefly, this value can be thought of as a *summary* of the information included in each 30 seconds interval. Summarizing a time series is an active research area. In our work, we favor simplicity and tested a few different simple metrics commonly used to *summarize* each segment such as the maximum value of the series in the interval, the mean value and the standard deviation Cooper (2011); Murphy (2012). The maximum value was that which produced the best results when used in the classifier. This was an expected behavior, as a higher value in each of the features is normally an indication of a lack of attentive state. After the application of this procedure a total of 132 observations was obtained, one for each video segment. It should be noted that an observation is an array of five values, one for each feature.

The number of values for each sensor varies because of errors in the sensor readings. In the case of EDA and EEG sensors, for example, a minor displacement of an electrodes may generate noise that can't be removed. We analyzed each video segment individually and, if an error in the sensor readings was detected, the corresponding value was removed. The sensor that produced the least number of error readings was the EDA, considering all the 132 video segments. For this sensor, 124 values from the possible 132 were obtained with no error reading. However, the blink indicator, for example, produced only 103 useful readings out of 132. The camera sensor is sensitive to noise because of the feature extraction method used, as described in section 5.2. For blink detection, for example, if the algorithm does not output any information about the pupil, the implemented system may erroneously report that the student is blinking. Instead, the output was the result of an erroneous pupil recognition. To improve the blink detection, and to reduce the noise in this sensor feature, we perform a manual inspection to remove the wrong information.

Table 6.1 presents the five features extracted from the sensors that were used for clustering. The table also shows the number of observations from each sensor that have no errors.

Sensor	# of Observations	Feature
EEG	105	Power Signal from 6 to 10 Hz
	114	Head Moviments
Eye Camera	107	Pupil Size
	103	Blink Duration
EDA	124	EDL Signal

Table 6.1: List of Available Features.

6.2 Model-Based Clustering

Clustering techniques have been studied for years. Clustering is a data analysis technique used to group data according to their degree of similarity. The earlier approaches for clustering usually rely on geometric procedures, like the k-means algorithm. Model-based clustering (Bouveyron and Brunet-Saumard, 2014; Fraley and Raftery, 1998) is a probabilistic approach to clustering and, from this approach, the notion of clusters can be formalized through the probability distribution of their observations. This approach has the potential to be more flexible as compared to other techniques, since the partitions can be interpreted from a statistical point of view, that is, data from each cluster is likely to be samples from the same distribution.

Model-based clustering is based on the finite mixture of probability distributions. Among the possible probability distributions for the mixture components, the gaussian distribution is certainly the most widely used. The Gaussian Mixture Model is the simplest form of *latent variable models* which constitute the basis for unsupervised learning algorithms and is used in many applications, such as signal processing and pattern recognition (e.g. Murphy (2012)).

In model-based clustering, it is assumed that the data is generated by a mixture of probability distributions in which each distribution represents a different group or cluster. Therefore, given a series of observations $y = (y_1, y_2, \ldots, y_n)$, a mixture model with M components is given by equation 6.1, where f_m is the conditional density of the mth cluster, with parameter θ_m, and w_m is the prior probability that an observation belongs to the mth cluster (with the constraint $\sum_{m=1}^{M} w_m = 1$).

$$g(y) = \sum_{m=1}^{M} w_m f_m(y|\theta_m) \tag{6.1}$$

Our proposed student model is a mixture of gaussian distributions. $f_m(y|\theta_m)$ is the probability density function of a gaussian distribution, as presented in equation 6.2. $\theta_m = (\mu_m, \sigma_m)$, where μ_m represents the mean of the mth cluster (its center) and σ_m defines the standard deviation.

$$f_m(y|\theta_m) = \frac{1}{(2\pi)^{1/2}\sigma} exp\left[-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right] \doteq \mathcal{N}(y;\mu,\sigma^2) \tag{6.2}$$

We can use this formulation to model the student engagement employing a single feature only. In this case, y are the observations of a feature extracted from a sensor. For our purposes, we are in the search of two clusters ($M = 2$) that can describe the behavior of students as *engaged* or *not engaged*. Our problem is how to find the best parameters θ_m, w_m for each cluster m and how to define the best label for each cluster.

For the parameter estimation problem, we use the Expectation-Maximization (EM) algorithm for clustering via Gaussian Mixture Models, as presented by Bouveyron and Brunet-Saumard (2014). The algorithm is presented below. We initialize $w_m^{(0)} = 1/M$ for $m = (1, \ldots, M)$. To improve convergence, we initialize the parameter $\mu_m^{(0)}$ with the centroids returned by a previous application of k-means clustering over data set y. Parameter $\sigma_m^{(0)}$ is initialized by one, as we assume the observations are uncorrelated[1]. The convergence criteria is defined as the log-likelihood below 0.0001.

When we work with multiple features for clustering, we use the multivariate generalization of the gaussian mixture model. In this generalization, each observation $y_i \in \mathbb{R}^D$, where D is the number of dimensions. In our case, D is the number of features used simultaneously for clustering. The multivariate gaussian distribution uses a covariance matrix $\Sigma \in \mathbb{R}^{DxD}$ instead of a single variance information, as defined in equation 6.8. The estimation of this matrix is also more complex than the simple σ parameter. To estimate Σ, we need to change equation 6.7 in the EM algorithm. The method to estimate Σ can vary, depending on the model (Fraley and Raftery, 1998). Most of these methods are used to reduce the number of parameters to estimate. As we are not working with a high number of dimensions, we can work with the full covariance matrix, as presented in equation 6.9.

$$f_m(y|\theta_m) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{6.8}$$

$$\hat{\Sigma}_m^{(q)} = \frac{\sum_{i=1}^{n} tim^{(q)}(y_i - \hat{\mu}_m^{(q)})(y_i - \hat{\mu}_m^{(q)})^t}{n_m^{(q)}} \tag{6.9}$$

Considering that we are working with the full covariance matrix, the total number of parameters to estimate is defined by $(M-1) + MD + MD(D-1)/2$. In the case of our student model, where we have $M = 2$, the number of parameters is given by $D^2 + D + 1$.

[1]Actually, the observations are pre-whitened prior to applying the EM(Murphy, 2012). This ensure that the empirical variance is one

Initialize $\theta_m^{(0)}$, $w_m^{(0)}$, $q = 1$
repeat
  Expectation Step:
  for $i = (1, \ldots, n), m = (1, \ldots, M)$ **do**

$$t_{im}^{(q)} = \frac{w_m^{(q-1)} f_m(y_i|\theta_m^{(q-1)})}{\sum_{l=1}^{M} w_l^{(q-1)} f_l(y_i|\theta_l^{(q-1)})} \tag{6.3}$$

  end
  Maximization Step:
  for $m = (1, \ldots, M)$ **do**

$$n_m^{(q)} = \sum_{i=1}^{n} t_{im}^{(q)} \tag{6.4}$$

$$\hat{w}_m^{(q)} = \frac{n_m^{(q)}}{n} \tag{6.5}$$

$$\hat{\mu}_m^{(q)} = \frac{\sum_{i=1}^{n} tim^{(q)} y_i}{n_m^{(q)}} \tag{6.6}$$

$$\hat{\sigma}_m^{(q)} = \sqrt{\frac{\sum_{i=1}^{n} tim^{(q)} y_i^2}{n_m^{(q)}} - \hat{\mu}_m^{(q)}} \tag{6.7}$$

  end
  increase q
until *convergence criteria are satisfied*

Algorithm 1: Expectation-Maximization algorithm for clustering via GMM

6.3 Results from Model-based Clustering

We start by a divide and conquer procedure, that is, we analyze each individual feature. The reason for that is to study each sensor in isolation to better understand the insights that can be obtained from each sensor towards our goals. Then we study the effects of using the information from subsets of sensors, aiming at selecting the best features that can represent the attentive state of the student. As described in Section 6.1, we selected 132 observations, one for each video segment. Recall that an observation is an array of five values, one for each feature. Then, we use the values of the array to create a cluster for each feature.

Figure 6.1 shows the application of a gaussian mixture model for each feature. The blue bars represent the histogram of values obtained from each of the features. As explained before. we use two clusters ($M = 2$). The solid line in each figure represents the fitted distribution in the mixture and each dotted line shows one of the distributions of the mixture and are associated to one of the clusters. It is clear from the figures that, for a subset of the features considered, the distributions that compose the mixture of gaussians clearly distinguish the results into two clusters. For other features, the distinction is not so clear.

Boucsein (2012) experimented with EDA sensors. The author observed that lower EDL values are related with *subjects* under observation feeling calm and this is an indication that the *subject* is probably in the *attentive state*.

From this interpretation, Figure 6.1(a) shows that most of the EDA readings relate to a *calm feelings* while the *stressful periods* (relatively high EDA values) are less common.

Klimesch (1999) performed similar experiments with EEG sensors. The results show that lower EEG sensor values may represent an *attentive state*. Applying these findings, we observe that most values we collected from the EEG sensor indicates an *attentive state* (Figure 6.1(b)).

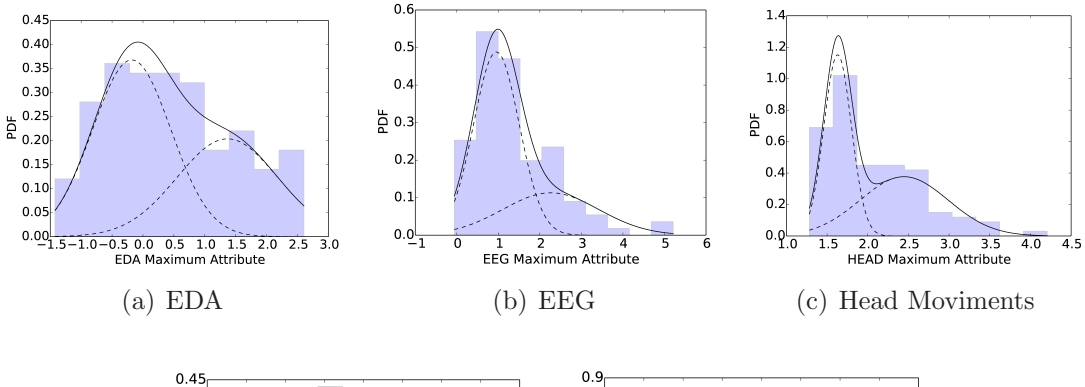
Considering the eyes sensor, the experiments performed in Lang (1995) showed that the blink duration and pupil diameter are important physical responses when the subjects are in the *attentive state*. The experiments in that work show that the *attentive state* relates with a short blink and a contracted pupil. We can relate these findings with the results in Figures 6.1(d) and 6.1(e). For instance, the rightmost Normal distribution of Figure 6.1(e) has higher expected blink duration and points from this distribution can be associated with *attentive states*. Once the distributions are obtained, the issue is then to determine the probability that a point belongs to one of the two distributions.

When head movements are considered, we use the work of el Kaliouby (2005) which shows that concentration is characterized by a steady head position. We can then relate *attentive states* with the absence of head movement.

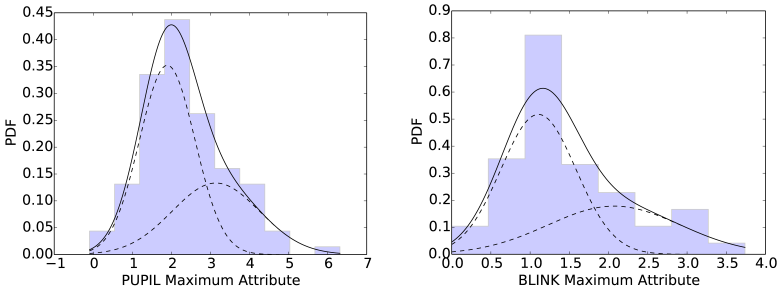
In the above paragraphs we associate the values of each cluster with results from the literature related to attention. We can then use the clusters to classify each video segment generated by participating students of our experiment as follows. For a video segment with n features, we employ a simple "majority vote" to classify the segment. In addition, although we collected five features (from the five sensors), not all of them produce a reliable readings due to sensor readings errors. Therefore, we choose to require that $n \geq 3$, that is, we only classify segments that have at least three features. From the 132 segments, 119 were classified ($n \geq 3$). From this total, 82% were classified as attentive and 18% as not attentive. We choose to define as a *weak classification* when the percentage of features with matching classification is less than 70%. In our experiment, 42% of the classifications were weak and 58% strong.

6.3.1 Accuracy Analysis

In the previous section, we used model-based clustering to determine if the sensor's data collected from our experiments would naturally divide into two distinct sets



(a) EDA (b) EEG (c) Head Moviments

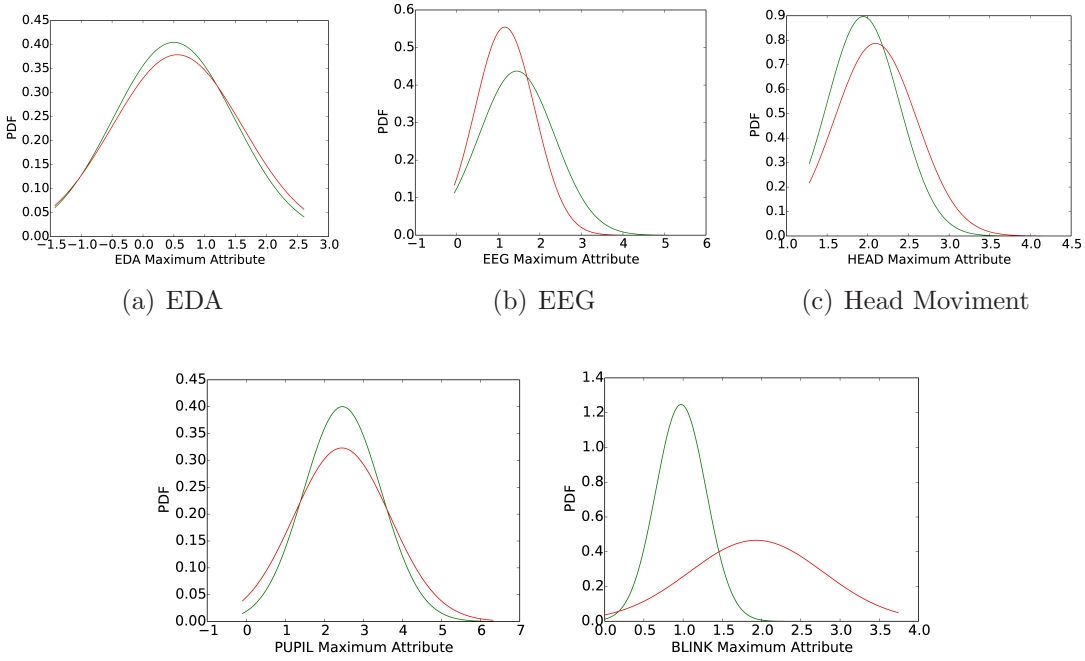


(d) Pupil Size (e) Blink Duration

Figure 6.1: Unsupervised Classification of Features

and, if so, if the sets would have a meaningful interpretation according to existing results in the literature. The results, which are plotted in Figure 6.1, clearly show distinctive clusters. As shown in previous section, the references in the literature provide a baseline to interpret the results. The purpose of this section is to analyze the accuracy of this interpretation, that is, the accuracy of the unsupervised classification obtained. For that, we employ the outcome of the experiment described in section 4.5, based on a subjective classification from experts. Recall that the experts classified 79 video segments, 48 as attentive and 31 as not attentive. The video segments with neutral classification were removed.

Each video segment classified by experts is associated with a set of values, one for each feature. For each feature, we compute the total number of *attentive* (equivalently *not attentive*) answers from all experts that were associated with a value of that feature, considering all video segments that were classified. Figure 6.2 shows the resulting histogram (distribution) of the subjective classification. The green line represents the video segments classified as attentive, while the red line represents the video segments classified as not attentive. In Figure 6.2(a) and 6.2(d), the green and red curves overlap almost entirely. This indicates that, in this experiment, the EDA and pupil size information cannot adequately distinguish between attentive and not attentive, as the distributions for the sensors (shown in Figures 6.2(a) and 6.2(d)) are similar. This is not the case for the blink duration feature. From Figure 6.2(e), it is clear that attentive and not attentive distributions are distinct. This is



(a) EDA (b) EEG (c) Head Moviment

(d) Pupil Size (e) Blink Duration

Figure 6.2: Subjective Classification done by Experts

an evidence that the blink duration is a feature that the experts took into account during their classification.

We proceed by checking if the unsupervised classification corresponds to the subjective classification from experts. In order to achieve this goal, we first label the clusters generated by the **unsupervised** classification using the results from the literature, as discussed in section 6.3. For example, from Figure 6.1(e), the two Normal distributions can be used to distinguish a video segment according to blink duration. We argued in section 6.3 that the rightmost Normal corresponds to *attentive* video segments, because of the higher expected blink duration as compared to the leftmost Normal. From the Gaussian mixture model, each video segment has a probability of being associated with a given cluster. A video segment S is said to belong to cluster A (or N) if the probability that S is in A (N) is greater than the probability that S is in N (A). (There is, if $P[S \in A] > P[S \in N]$ then segment S is associated to cluster A.)

After labeling the clusters obtained from the unsupervised classification (model-based clustering), we construct a *confusion matrix* (sometimes called *contingency table* or *matching matrix*) for each feature. The rows of each matrix represent the instances of the actual class which is considered the classification from experts and the columns are obtained from the model-based clustering. The resulting matrices are shown in Table 6.2 for all the five features.

Table 6.2: Confusion Matrix of individual sensors

(a)

EDA	**A**	**N**
A	31	17
N	18	13

(b)

EEG	**A**	**N**
A	34	14
N	27	4

(c)

Head Moviment	**A**	**N**
A	29 (TP)	19 (FN)
N	11 (FP)	20 (TN)

(d)

Pupil Size	**A**	**N**
A	29	19
N	23	8

(e)

Blink Duration	**A**	**N**
A	46	2
N	14	17

For example, from Table 6.2(c), from the 48 video segments classified as *attentive* from the experts, 29 of them were also classified as *attentive* from the model-based clustering (combined with the labeling from the literature results) but 19 were classified as *not attentive*. Likewise, the model-based clustering classified 40 video segments as *attentive* and 39 as *not attentive*. Also shown in Table 6.2(c), the *true positives* (TP), *true negatives* (TN), *false positives* (FN) and *false negatives* (FP).

From the the confusion matrices, several metrics can be obtained. For instance, the *true positive rate* defined as $TPR = TP/(TP + FN)$ is $TPR = 29/48 = 0.60$ and the accuracy ($ACC = (TP+TN)/(TP+FN+TN+FN)$) is $ACC = 49/79 = 0.62$ for the *head movement* feature.

Table 6.3(a) shows the *sensitivity* (*TPR* for *attentive* and *TNR* for *non attentive* for all features). Video segments classified as attentive are defined by "A", while not attentive are defined by "N". From the table, the blink rate, head movement and pupil size features are candidates to provide good predictive capability. Surprisingly, the EEG had poor predictive capability.

In order to obtain the results above we assumed that the results from the literature were adequate for labeling the clusters obtained from the unsupervised clustering. One question that immediately comes to mind is how similar are these results as compared to those when we label the clusters from the unsupervised clustering using the feature values provided by the experiments performed with experts. In order words, for a given feature value would the literature agree with what our experts indicate in the experiments we performed? The answer is: from the five features,

only the interpretation of the values of two of them (EEG and pupil size) are not in agreement with the interpretation from our experiment. For example, the results of Klimesch (1999) indicate that lower EEG sensor values may represent an *attentive state*. However, our experiments conclude exactly the opposity. In what follows, we label the clusters from the Gaussian mixture model in accordance with the results from the experiment with experts as follows.

A simple majority vote from the **subjective** classification is used for labeling a cluster that was obtained by the **unsupervised** classification. Therefore, if one of the clusters (from the **unsupervised** classification) has more than 50% of its video segments classified as attentive by the experts, we label this cluster as *attentive* and the other as *not attentive*. Using this simple labeling procedure, we can obtain another set of accuracy values for the unsupervised classification as compared with the subjective one. Table 6.3(b) shows the results for this new classification. Note that only the EEG and pupil size had their accuracy metrics changed, since only these two features were not in agreement with the literature.

Table 6.3 summarizes the percentage of agreement between the unsupervised classification and the classification from experts using both the literature that label the unsupervised classifier (Table 6.3(a)) and the experts (Table 6.3(b)). We can observe that the blink duration feature has a high sensitivity (hit rate). This feature can classify correctly almost all the attentive video segments classified by the experts, and almost 55% of the not attentive segments. (Note that approximately 60% of samples are attentive (positives in the confusion table) and 40% are non attentives (negatives), which indicates a reasonable balance. Other features present higher sensitivity as compared to that for the blink rate for *not attentive* but, if we consider both classifications, attentive and not attentive, the blink rate has an accuracy of over 80% for all video segments, the highest percentage of agreement among all features.

6.3.2 Feature Correlations

If we use two features to classify the video segments, the best classification is achieved when both the blink duration and pupil size features are used. Figure 6.3 shows the correlation of the two features. The clusters generated by the mixture of Gaussians are represented by the solid blue ellipses. The green dots represent the video segments classified by experts as attentive, while the red dots represent those classified as not attentive. The grey small dots represent the video segments not classified by experts. We label each cluster of the unsupervised classification based on the information of the subjective classification, as follows: the cluster is labeled as attentive if more than 50% of video segments are classified as attentive. The dots in a cluster

Table 6.3: Sensitivity and Accuracy of individual sensors

(a) Literature

Feature	Sens.	Acc.
EDA	A: 64.58 N: 41.93	55.69
EEG	A: 70.83 N: 12.90	48.10
Head Moviment	A: 60.41 N: 64.52	62.03
Pupil Size	A: 60.41 N: 25.81	46.84
Blink Duration	A: 95.80 N: 54.80	79.75

(b) Experts

Feature	Sens.	Acc.
EDA	A: 64.58 N: 41.93	55.69
EEG	A: 29.16 N: 87.09	51.89
Head Moviment	A: 60.41 N: 64.52	62.03
Pupil Size	A: 39.58 N: 74.19	53.16
Blink Duration	A: 95.80 N: 54.80	79.75

represent the video segments that receive the same classification both from experts and from the unsupervised classification. On the other hand, the crosses indicate that the classification by experts is different from that given by the unsupervised classification.

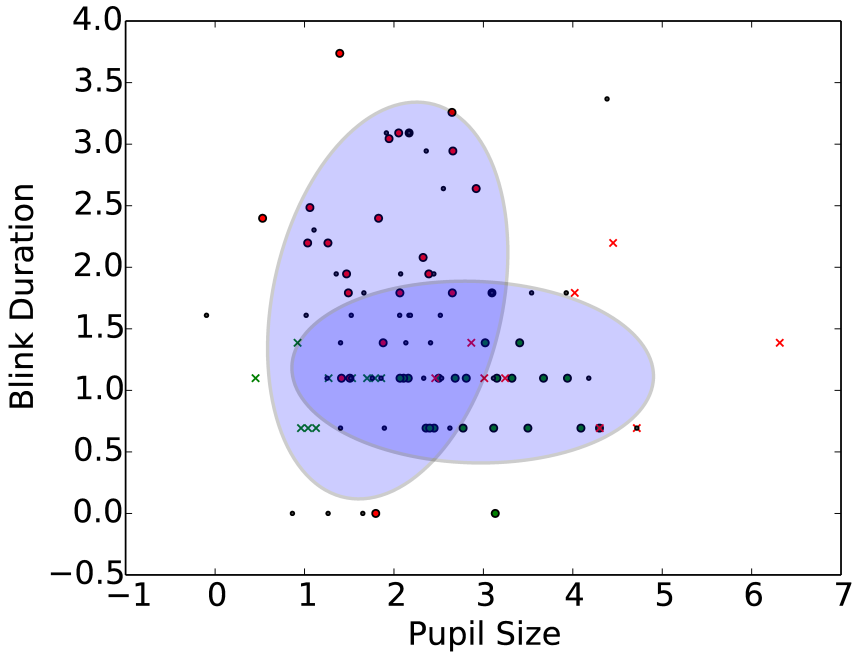


Figure 6.3: Relation between Blink Duration and Pupil Size

Our best result obtained by the use of two sensors gives 68.75% of agreement with the classification from experts for attentive video segments, and 70.96% of agreement for not attentive. If we compare this information with the best classification using a single feature, we have a better balance between the attentive and not attentive

classification. This is probably due to the fact that the maximum value of pupil size may accurately represent the not attentive video segments, while the maximum value of blink duration may represent reasonably well the attentive segments. Overall, this clustering classification achieved over 70% of agreement with the experts.

We also observe from Figure 6.3 that several video segments that are not correctly classified are those which are near the border of the attentive cluster (these are the green crosses). This means that, although the probability of these video segments belong to a cluster is higher than the probability of belonging to the other cluster, this probability is approximately 50%. This is an indication that, with a higher number of points or a better tweak of the feature extraction method, these video segments can change cluster and, as such, increase the overall agreement. This is also observable in Figure 6.4, which presents the probability of each video segment belonging to a defined cluster. If we use the probability of 50% for the video segment to be classified as attentive, almost 70% of the video segments will be assigned to the attentive cluster. However, if we lower this threshold to 40%, we increase the number of video segments to almost 80%.

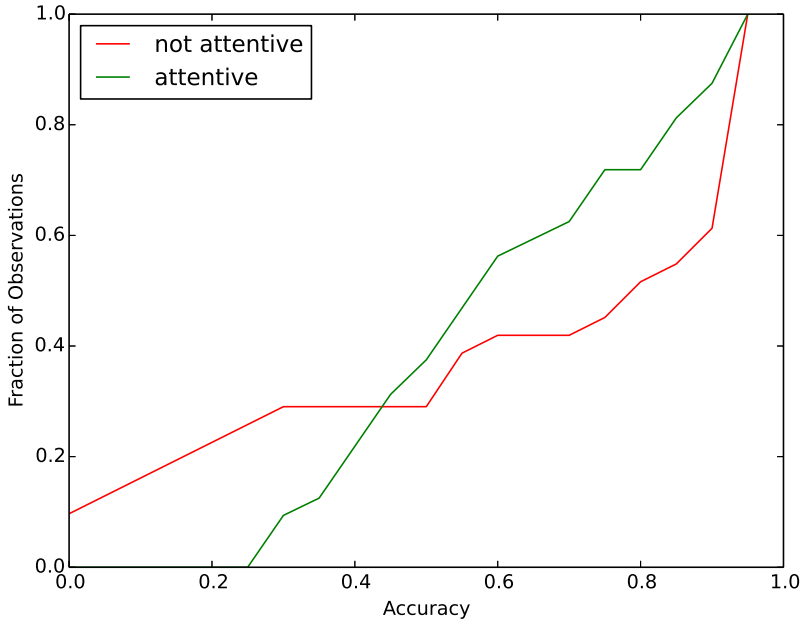


Figure 6.4: Probability of a Video Segment to belong to a Cluster (using Pupil and Blink features)

We may add additional features and increase the dimensionality of the two defined clusters. Our best result for three features arose when the EDA sensor was added to the other two best features, blink and pupil. Figure 6.5 shows the clusters generated by using the best three features (Blink, Pupil and EDA). Using this con-

figuration, we obtained 70.96% of agreement with the classification given by experts for the not attentive video segments, and 62,5% for the attentive video segments. Overall, the agreement of this classification is less than 66%. This result shows that, for this experiment, the classification using the blink duration and pupil size gives the best classification.

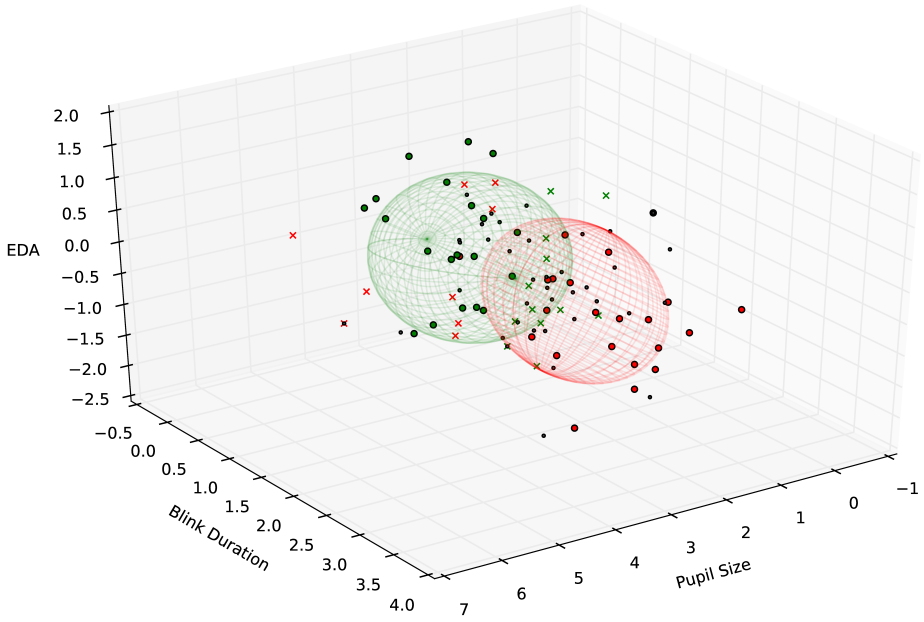


Figure 6.5: Relation between Blink Duration, Pupil Size and EDA

6.3.3 Classification over Time of the Experiment

In this section we analyse the result of the classification over time that is, using the video segment streams. We employed the classifier that gives the best classification, which is the two dimension classifier with blink and pupil features. We arrange the video segments according to their generation time during the experiment. Recall that we are using the video segments from the first part of the lecture with length equal to six and a half minutes. This result can becompared with Figure 4.13, which presents the classification from experts. Figure 6.6 shows the observations divided in three classes: attentive, neutral and not attentive. In this figure, we considered only the observations such that the probability of belonging to a specific cluster is higher than 60%. The observations with probability smaller than 60% are considered "Neutral".

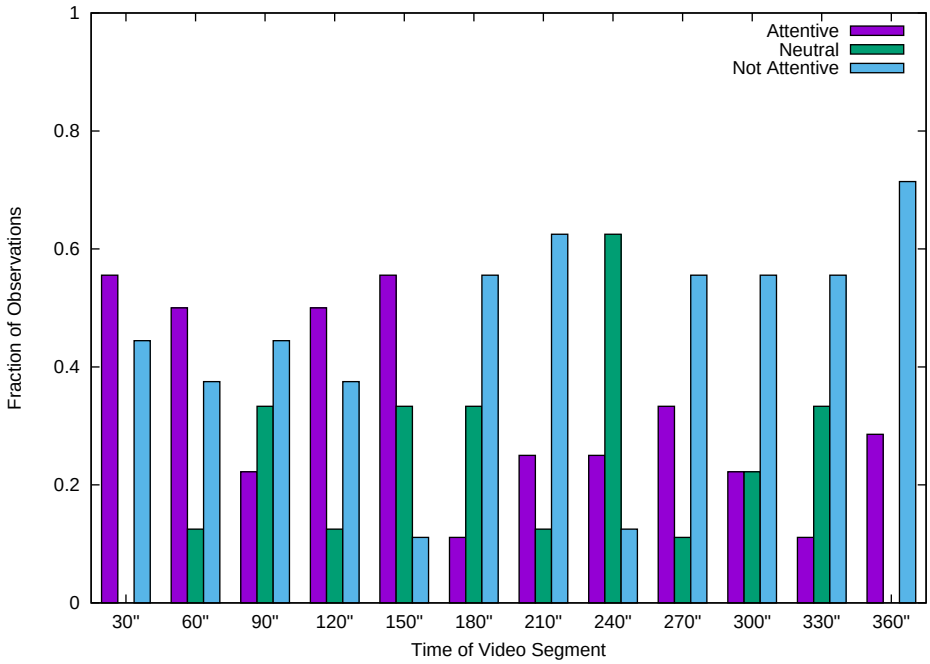


Figure 6.6: Clustered classification over Time of the Experiment

It is interesting to observe that the video segments towards the end of the first part of the lecture are classified as "not attentive". This contrasts with the results presented in Figure 4.13, in which the experts classified these video segments as "Neutral". We also observe a consistent droop in the number of attentive classifications towards the end.

6.4 Summary and Discussions

We proposed a classifier employing model-based clustering techniques that can be used to adapt video lectures according to the engagement of a student. To our knowledge, this model is the first that can be employed to automatically adapt a video lecture without relying on any student interaction. The change in the flow of the lecture is achieved only from the output of sensors monitoring a student. By using a Gaussian mixture distribution for cluster analysis, we devised an unsupervised classification method for engagement that achieved, in our experiments, over 80% of agreement compared to subjective classification given by experts. It is remarkable that our initial classification based on unsupervised learning matched so closely with that given by experts. In addition, the best classifier in literature for the engagement state of students can achieve 64% of accuracy with a pressure chair sensor (D'Mello et al., 2007). We obtained similar results with our cheap and easy to build eye sensor.

Based on the analysis of each sensor individually, we found that the eye features

are the ones that give the best classification. This is an indication that these features are relevant to teachers to detect the student attention, an indicative of engagement, during a lecture. This is also important for scalability of the entire system, as the eye sensors rely only on a webcam, while other features depend on expensive sensors.

Chapter 7

Engagement Model Base on Log Analyzis

In the last chapters, we described a methodology to measure the student engagement while watching a video lecture based on their biological feedback. This information can be used to adapt the lecture in real time in order to maintain the student's motivation in the lecture. This methodology was implemented in the VideoAula@RNP client, presented in section 3.1. This new client is not currently in use in the VideoAula@RNP service. The client in use in the VideoAula@RNP logs information about the student' interactions during a video lecture.

In this chapter, we analyze the information collected over two years in order to study the student behaviour in the lectures. We explore metrics based on the time the student spend in a lecture. We also propose a method that can be used to evaluate the impact of the adaptation process. For that, we developed a mathematical model that uses time-based "Engagement Metrics" of video (Balachandran et al., 2013; Dobrian et al., 2011). These metrics are based on the length of the video lecture the student watched and on the popularity of each part of the video lecture. Our model can be used to analyze the engagement of the students in each part of the video lecture. This can also be used by the lecturer to improve the video lecture.

7.1 The VideoAula@RNP Log Database

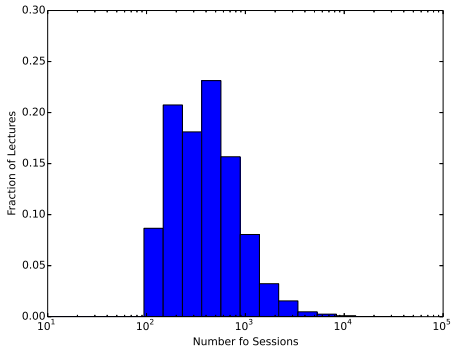
The studies reported in this chapter are based on data collected over two years from the CEDERJ Computer Systems Technology undergraduate course. Recall that the CEDERJ consortium uses the RIO Multimedia system to store and retrieve multimedia objects, in our case video lectures with slides, as presented in section 3.1. The system has been developed in our Laboratories and is in use since 2005 by students of the Computer Systems Technology undergraduate course. The users of

the RIO system have access to video and slides of the lectures. In addiction, they may interact with the video lecture, either by using forward and backward commands or via the slide area. The data used in this research was collected over the period of june 2012 to june 2014. This data is segmented over sessions each representing an user distinct access to the system. Each session contains the student's interactions from the beginning of the session until the browser window is closed. As an example, if the student moves forward or backward, the system logs the final position of each jump. The system also logs informations about the video in a 30 seconds interval. This is useful to track the length of time the student is watching each specific part of the video lecture. We don't track individual students in the database, so each access to a video lecture is recorded as a new session. All this information was collected and stored in a SQL database (mysql).

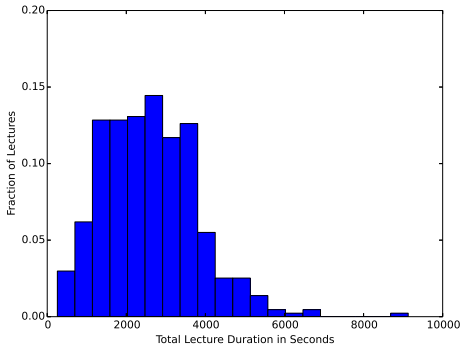
The log system is an HTML server that receives information from the RIO client. Each log entry in a session can be generated by an active action from the student, or a passive instruction from the client. Active entry information is generated when the student make an interaction with the client interface, like change a topic or move the video forward. A passive entry information is generated every 30 seconds by the client, regardless of the student interactions. When the student first access a video lecture, the log system initialize a session with a special entry that contains a unique identification for the session, the server time and information from the browser and client, like the ip address and operation system. Every subsequent entry in the log contains the id of the session, the server time and an identification if the entry is active or passive. Active entries also contain information about the interaction the student has done, like the destination of a jump. The passive entries contain the video lecture state, like the position of the video lecture the student is currently watching, in seconds.

The dataset used in this study contains 722511 sessions of 436 video lectures. We removed from the dataset the sessions that have only the initial entry, as this indicates that the student has watched less than 30 seconds of the lecture (There was any passive entry for this session in the log).

Figure 7.1(a) shows the histogram of the number of sessions. The mean number of sessions of a video lecture is around 1000. From figure 7.1(a), note that only a small fraction of video lectures have a large number of access (or sessions). This is expected in a video databases, as a small number of videos are much more popular than the others(Cha et al., 2007). Figure 7.1(b) shows the histogram of the total lecture duration. The mean lecture duration is 45 minutes with a few videos having over two hours. Some statistics of the dataset are presented in table 7.1.



(a) Popularity Histogram



(b) Histogram of the Total Lecture Duration

Figure 7.1: Some histograms of our dataset

Popularity (Sessions)	Lecture Time (seconds)
Min: 97	Min: 256
Max: 49894	Max: 9129
Mean: 1657	Mean: 2647
Median: 769	Median: 2557

Table 7.1: Summary of dataset

7.2 Metrics

Popularity metrics are comonly used to characterize video databases(Cha et al., 2007; Chatzopoulou et al., 2010; Richier et al., 2014), and the number of access is the most popular of these metrics. But recently, other types of metrics have gained attention, like time based metrics(Balachandran et al., 2013; Bendersky et al., 2014; Dobrian et al., 2011). Time based metrics are related to the time students spend during the video lecture session.

Our work focus on four metrics: three are time based metrics and one is based on the viewer's interactivity. They are defined as follows:

Watch Time: Is defined as the time a student spends watching only new video content of a given lecture. This metric counts only the non overlapping intervals that were watched during the session. If the student jumps backwards to watch a video segment a second time, it is counted only once.

Play Time: The total time a student spends watching a lecture. Overlapped intervals are all added. This metric is greater than or equal to the Watch Time. For example, suppose that a student watches a specific part of a video lecture for three times during a session. In this case, if the length of that interval is 10 seconds, the total Play Time will be 30 seconds and the total Watch Time will be 10 seconds.

Session Time: The total time of a session. This metric includes the Play Time and the time the video is not been played, when the pause button is pressed, for example. We can calculate the session total time by subtracting the time of the first entry log in the session with the time of the last one.

Number of Jumps: In order to study how students interact with the video lecture we consider the number of jumps that occur during a session. A jump results from the forward and backward commands issued, for instance when the student moves to another topic in the lecture, or review the current topic being studied.

7.2.1 Metrics Analysis

We first consider the time related metrics defined above. Figure 7.2 shows the Session Time, Play Time and Watch Time histograms, in intervals of one minute. We observe that the Session Time have the highest mean value among the three metrics. This is expected, as the Session Time includes the Play Time plus the time the video stays in pause. Some sessions have over three days (figure 7.2 is truncated at 100 minutes) . A close look of these sessions shows that the video lecture was paused most of time (more than two days). This behavior can be explained as the students leave the browser open after watching the video lecture, usually on weekends. The session only ends when the students loose connection to the server (if a passive log is not sent by the RIO client in the 30 seconds interval). Figure 7.2 also shows that the number of sessions with Play Time and Watch Time above 60 minutes is relatively low.

Figure 7.3 shows the histogram of number of jumps per session. Many sessions ($> 10^3$) have a considerable number of jumps (> 10) and over two thirds of the jumps are forward.

This result shows that the students do not watch a video lecture continuously, from the beginning to the end. They prefer to watch small parts of the lecture and jump from topic to topic. This observation should be taken into account when lectures prepare their class. Preferably, the video lectures should be organized in self contained short topics.

7.3 The Engagement Model

We define the engagement of a student while watching a video lecture as the fraction of the video lecture the student watched in a session, based on the watch time metric. Intuitionally, if a video lecture catches the student's attention, the student watch a large portion of the video lecture. On the other hand, less interesting video lectures

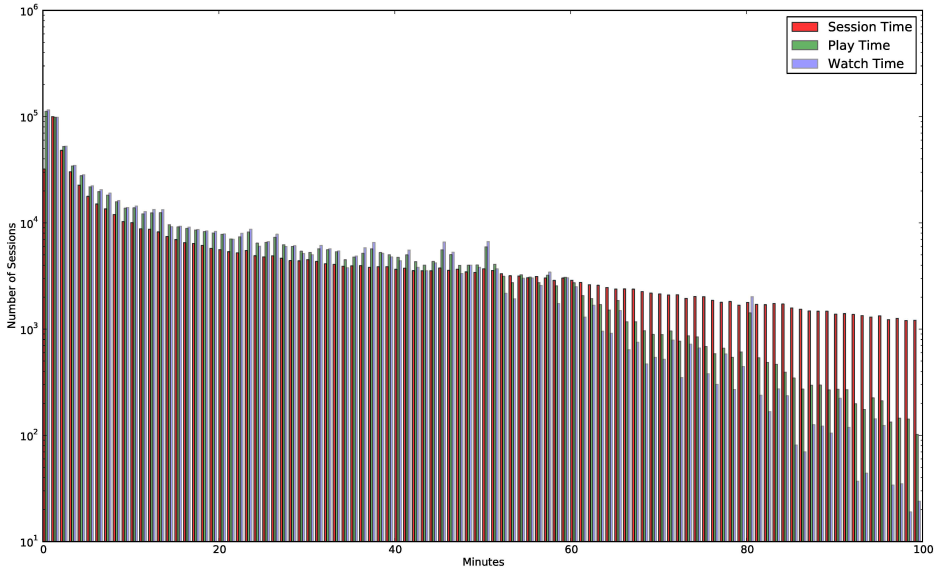


Figure 7.2: Histogram of Session Time, Play Time and Watch Time

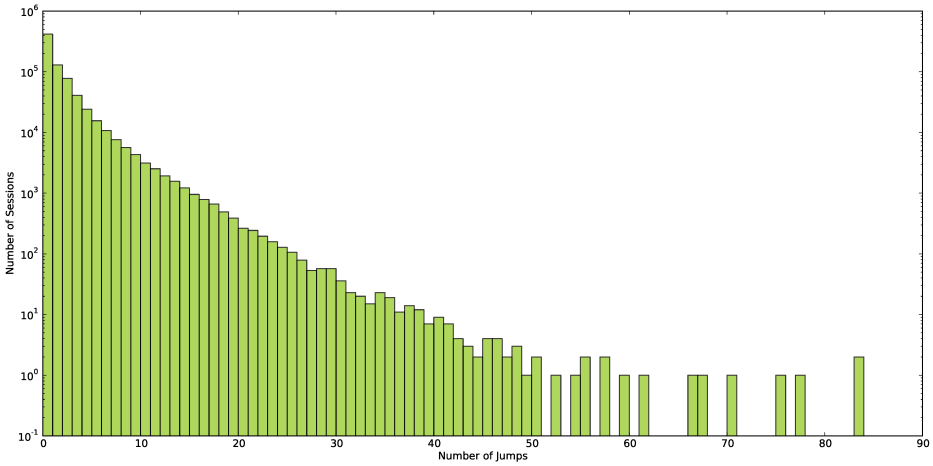


Figure 7.3: Histogram of Jumps

may cause the student to watch only a small fraction of it. Balachandran et al. (2013) used an engagement predictive model based on metrics such as video play time aiming at obtaining a QoE measure. Bendersky et al. (2014) also used a similar metric in recommendation algorithms. All these works use the watch time as a metric value extracted from video information. To our knowledge, no work has presented a model that can study the user's engagement using the watch time metric.

We model the student's engagement as a system with two attractors: one that is driving the student to leave the lecture and one that is holding the student to continue watching the video. If the force to leave the lecture is much higher than that of holding the student, the lecture is assumed to be less interesting and the user can eventually leaves after watching only a small fraction of the lecture. If these forces are balanced, we expect that the student probability to leave is uniformly distributed with time, as students have equal chance to leave at any point. Models like this are particularly common in reliability and biological studies (Gupta and Nadarajah, 2004). These studies usually shows bathtub shaped ("U-Shape") or unimodal shaped ("J-Shaped") distributions. Most of these studies based their models on the Beta distribution for simplicity. To study the behaviour of the student's engagement, we model the watch time metric using the beta distribution.

Let w be the watch time given as a fraction of the lecture time, so $0 \leqslant w \leqslant 1$, the probability density function of the beta random variable is given by the equation 7.1. $B(\alpha, \beta)$ is known as the beta function, and is used as a normalisation constant in the beta distribution. The beta function is a particular application of the incomplete beta function, were $x = 1$. The incomplete beta function is defined in equation 7.2. The cumulative distribution function of a beta random variable is given by $F(x; \alpha, \beta) = \frac{B(x;\alpha,\beta)}{B(\alpha,\beta)}$.

$$Beta(w; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} w^{\alpha-1}(1-w)^{\beta-1} \tag{7.1}$$

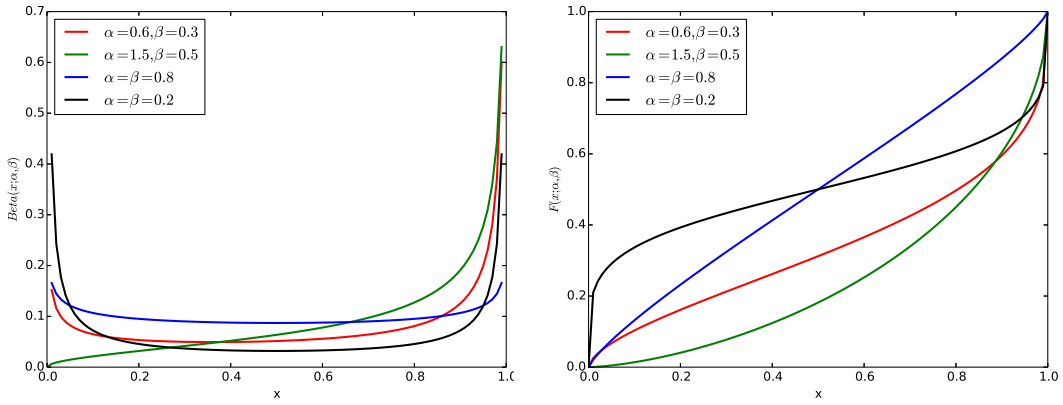
$$B(x; \alpha, \beta) = \int_0^x u^{\alpha-1}(1-u)^{\beta-1} du \tag{7.2}$$

In what follows, we analyze the engagement information that can be obtained from the parameters of the beta distribution, given by α and β. The beta probability density function has a "U-Shape" when $\alpha < 1$ and $\beta < 1$. The probability of a high value of w increases with α, while the probability of a low value of w increases with β. In our model, w is the watch time, therefore, we may relate parameter α with the force that maintains the student engaged in the lecture, while β may be related to the fraction of students that leave the session. If $\alpha > \beta$, the Beta distribution is skewed towards 1. This indicates that most of the students watched more than

Parameters	Analizys
$\alpha > \beta$	Most of the students watched more than 50% of the lecture
$\alpha < \beta$	Most of the students watched less than 50% of the lecture
$\alpha = \beta \to 1$	Students may drop at any point with equal chance (Uniform distribution)
$\alpha = \beta \to 0$	A large fraction of students drop earlier in the video lecture, other fraction stays to the end

Table 7.2: Summary of Beta parameters relationship

50% of the lecture. If $\alpha > 1$ and $\beta < 1$, the distribution has a "J-Shape" form with a left tail. It is strictly increasing, which indicates that a large fraction of students watched the entire lecture. If $\alpha < 1$ and $\beta > 1$, the "J-Shape" form have a right tail, indicating that a large fraction of students had dropped at the beginning of the lecture. One special case occurs when the density function is symmetric ($\alpha = \beta$). In these cases, it is interesting to observe the shape of the density function as the parameters approach zero. When $\alpha = \beta \to 0$, there is a relatively large fraction in the extremes (0 and 1). Therefore, there is a high probability that students either leave the video lecture in the very beginning, or they watch the entire video lecture. If $\alpha = \beta \to 1$, the probability is approximately uniformly distributed, and students have an equal chance of leaving at any point. Figure 7.4 shows examples of the beta distributions with different parameter values. Table 7.2 summarizes the above comments.



(a) Probability Density Function (b) Cumulative Distribution Function

Figure 7.4: Example of Beta Distributions

7.3.1 Estimating the Parameters

To estimate the parameters of beta distribution, we choose to use the method of moments for its simplicity. The beta distribution does not have a closed form for the likelihood, and as such it is not trivial to estimate the maximum likelihood. On the other hand, the lecture sample sizes have mean 1000, which favors unbiased estimations. We define N as the number of video lectures, and M_k the number of sessions of lecture k, $k = \{1..N\}$. Let $X^k = \{x_1^k, x_2^k, ..., x_{M_k}^k\}$ be the fraction of lecture k that each student x_i watched during a session, $0 < i < M_k$. We can estimate the parameters $(\hat{\alpha^k}, \hat{\beta^k})$ using the first two moments, the sample mean $(\overline{X^k} = \frac{1}{M_k}\sum_{i=1}^{M_k} x_i^k)$ and the sample variance $(\overline{X^{k^2}} = \frac{1}{M_k - 1}\sum_{i=1}^{M_k}(x_i^k - \overline{X^k})^2)$ (Gupta and Nadarajah, 2004). The estimators are presented in equation 7.3.

$$\begin{aligned} \hat{\alpha^k} &= \overline{X^k}\left(\frac{\overline{X^k}(1-\overline{X^k})}{\overline{X^{k^2}}} - 1\right) \\ \hat{\beta^k} &= (1-\overline{X^k})\left(\frac{\overline{X^k}(1-\overline{X^k})}{\overline{X^{k^2}}} - 1\right) \end{aligned} \tag{7.3}$$

In order to obtain the goodness of fit, we use the mean error rate (MER) metric presented in Richier et al. (2014). This metric shows good quality when compared with similar metrics, like chi-test, and have an easier interpretation. The MER is the mean error rate generated by the model in relation to the observations. In general, the error generated by the model is less than or equal to the MER metric. For example, if we have a $MER \leqslant 0.1$, the error is lower than 10% on the average. It is also similar to the Kolmogorov-Smirnov test (KS), which is a good measure to test if distributions are alike. In KS, we are interested in the upper bound, while in MER we are interested in the average. We choose to work with the cumulative distribution, as the density may have very small values and increase the error in the computation[1]. We also need to discretize the linear space. Let L be the number of values in this space (bins), we have $S_j = j/L$, with $0 \leqslant j \leqslant L$. We chose L as 100. Our tests with $L > 100$ only increase the processing time with little gain in precision. The MER metric is given by equation 7.4.

$$MER^k = \frac{1}{L}\sum_{i=0}^{L}\frac{|F(S_i; \hat{\alpha^k}, \hat{\beta^k}) - P[X^k \leqslant S_i]|}{P[X^k \leqslant S_i] + 1} \tag{7.4}$$

Figure 7.5 shows the histogram of MER over all the lectures in the database. We see that the error is smaller than 5% (The X axis in the graph is from 0 to 0.1) and most of the lectures have MER less than 3%. Richier et al. (2014) consider a mean error of 5% a reliable fitting, indicating that the beta distribution is a good

[1]This problem is presented in Richier et al. (2014)

representation for the watch time metric.

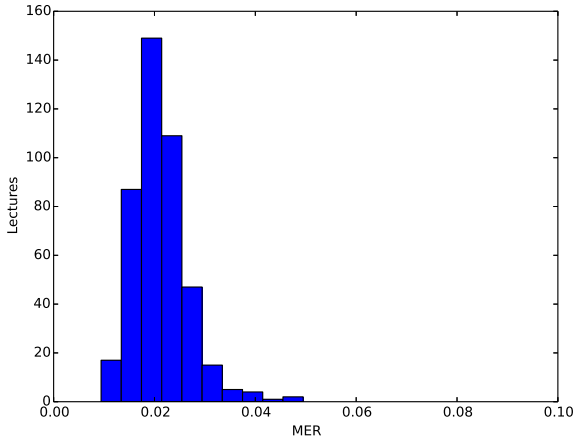


Figure 7.5: Histogram of MER metric

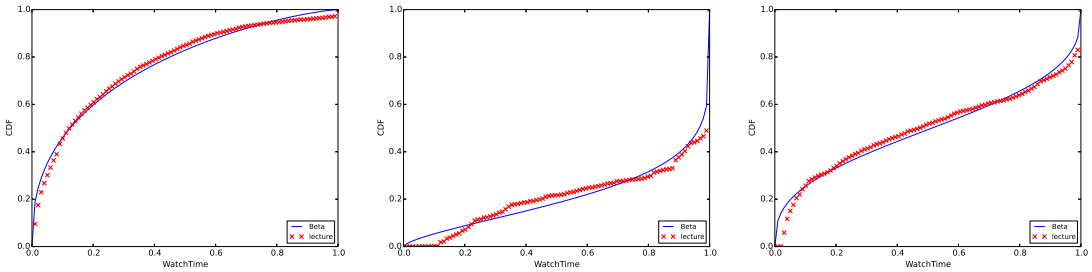
7.3.2 Results

We first choose three out of the 436 existing lectures in the CEDERJ distance learning computer system technology course to illustrate the distribution fitting with distinct α and β values, as discussed above. Figure 7.6 shows the results. The red line indicates the empirical cumulative distribution function and the blue line represents the beta distribution with the best fitting parameters, according to the parameter estimation procedure described in section 7.3.1. The first lecture presented (figure 7.6(a)) indicates that the beta distribution of the watching time for this lecture has a β value higher than the estimated value of α. In this particular case, $\beta > 1$, which indicates that the lecture have an inverse "J-Shaped" appearance. From all the students who accessed the lecture, 80% of them watched less than 40% of the lecture. Only 20% watched 40% of the lecture. Therefore, according to the engagement model we adopt (section 7.3), the student engagement is low.

In contrast, Figure 7.6(b) shows that more than 80% of students watched more than 40% of the second lecture. This is an example of a lecture with high engagement and $\alpha > \beta$. It is also interesting to note that, in this case, more than 50% of the students watch the entire lecture.

The watch time of the third lecture (figure 7.6(c)) has a symmetric distribution, with $\alpha = \beta$. Table 7.3 shows the estimated parameters of each of the three lectures.

The analysis presented above motivates a method to classify the lectures. We first obtain the parameters of the beta distribution for each of the lectures. Using our engagement model, we classify all the 436 lectures of the CEDERJ distance learning course according with the parameters α and β. Figure 7.7 plots the estimated parameters α and β for each of the 436 lectures. Each cross represents a lecture.



(a) $\alpha < \beta$ (b) $\alpha > \beta$ (c) $\alpha = \beta$

Figure 7.6: Example of Beta Fitting

Lecture	α	β
$\alpha < \beta$	0.260	1.355
$\alpha > \beta$	0.653	0.171
$\alpha = \beta$	0.350	0.350

Table 7.3: Estimated parameters for graphs in Fig. 7.6

From the figure we observe that most lectures have $\alpha < 0.5$ and $\beta < 1.0$. This indicates that the majority of lectures have a higher β and, consequently, a lower engagement from the students. A few lectures have $\beta > 1.0$, indicating lectures that the students drops at the very beginning and just a small fraction of students watched the entire lecture.

Lectures with symmetric Beta distributions are those in which $\alpha = \beta$, after parametrization. We consider that a lecture has an approximate symmetric distribution if the absolute difference of their parameter values is small, that is $|\alpha - \beta| < 0.1$. A dashed line is presented in Figure 7.7 to indicate the region that we consider symmetric. Table 7.4 classifies the lectures based on the parameters values. Almost one third of the lectures are symmetric.

Lecture		Total
$\alpha = \beta$		29%
$\alpha > \beta$		4%
$\alpha < \beta$	$\beta > 1$	2%
	$\beta < 1$	65%

Table 7.4: Lectures Classification in accordance with the Engagement Model

Using the proposed model, we may estimate the probability that a student leaves a lecture based on the current watch time information of a session. We may use such probability to adapt the lecture in real time and try to increase the student engagement if this probability is high, that is, if the student is likely to leave. Let x be the amount of time a student has been watching a lecture measured as a fraction of the total length of the lecture. If we have obtained a model of this lecture from

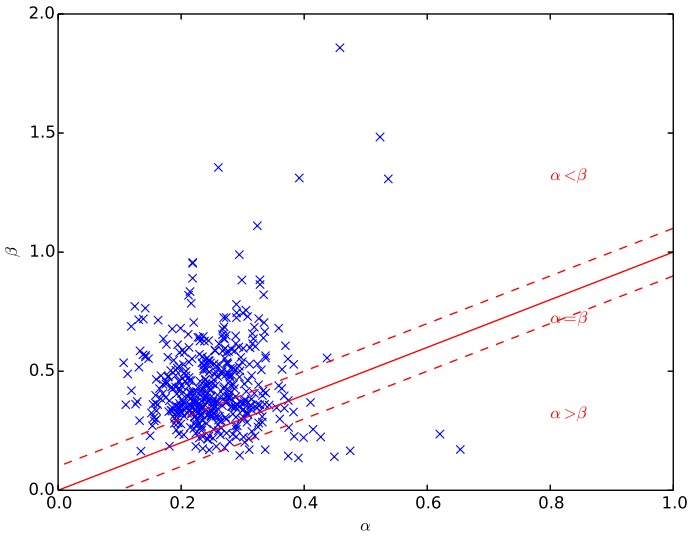
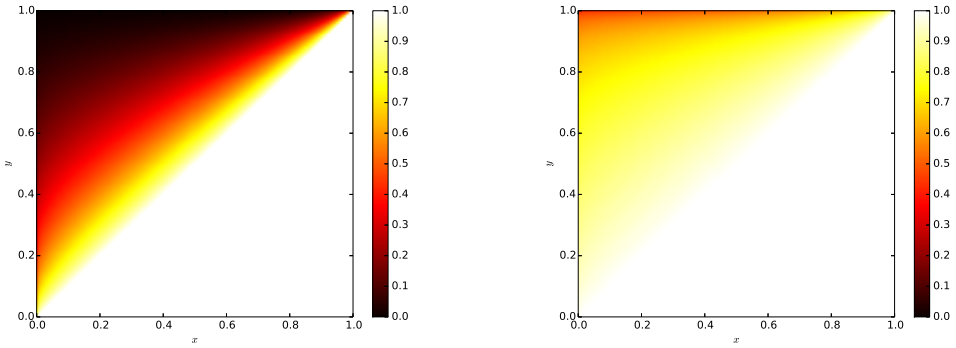


Figure 7.7: Lecture Classification based on α and β Parameters

past accesses, the probability of the student keeps watching for at least time y can be easily estimated from, $P[w > y|w > x]$ for $y > x$.

Figure 7.8 plots $P[w > y|w > x]$ for the lectures of Figure 7.6(a) and 7.6(b), respectively. Figure 7.8(a) corresponds to the lecture with parameter values for the model equal to $\alpha < \beta$ and $\beta > 1$. The figure shows that the probability of a student to watch the entire lecture is negligible, regardless of the fraction watched. But if 40% of the lecture has already been watched, there is a 50% chance that 60% of the lecture will be seen.

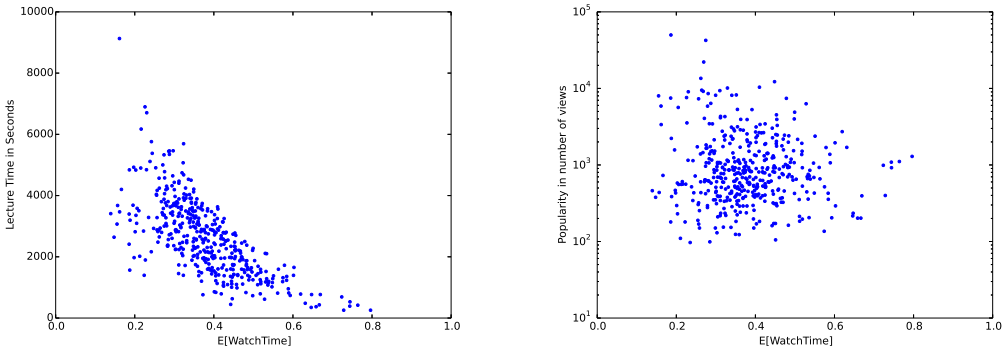
The lecture model of Figure 7.8(b) has parameter values $\alpha > \beta$. In this case, the probability that more than 80% of the lecture is watched is high, regardless of the amount of time the student has been watching the lecture.



(a) $\alpha < \beta$ (b) $\alpha > \beta$

Figure 7.8: $P[w > y|w > x]$ for lectures of Figure 7.6(a) and 7.6(b)

We also study how the engagement relates with total lecture duration and lecture popularity. Figure 7.9(a) plots the total lecture duration in seconds versus the expected watch time for all the lectures of the CEDERJ course. We observe that students watch up to 30% of long lectures (i.e. lectures that last for more than one hour and a half), while short lectures (those that last less than 20 minutes) keeps the student for more than 60% of its length, on average. This indicates that students have limited tolerance to watch long video lectures. Therefore, we conclude that one should avoid long lectures or long explanations.



(a) Lecture Time (b) Popularity

Figure 7.9: Correlation of Engagement with other metrics

Figure 7.9(b) shows the engagement and the popularity of lectures. The figure shows that highly popular lectures, with more than 10^4 sessions, are watched less than 30% of its length on average per session. An important conclusion from this findings is that the popularity metric based on the number of accesses is not a good indication of the engagement of students in the lecture. Popularity may be a good indicator that a topic is popular or the student expectation towards the lecture is high. Popularity does not indicate that most of the lecture will be seen.

We also analyze the popularity of each second in each video lecture. This can give a more precise information than the popularity metric, as we can detect which part of a lecture is more popular. For example, if a student have watched from the instant 10 seconds to the instant 60 seconds of a video lecture, only this range is counted as one access for this session.

Figure 7.10 shows the popularity of each second of a specific video lecture (class EAD5007 lecture 2). This lecture have around 2600 seconds (little more than 43 minutes). We computed the result conditioned on the value of the engagement metric. We want to analyze if a higher engagement can affect the popular parts of a lecture. We use the sessions from students that watched at least 30 seconds of the lecture ($w > 0$), students that watched more than 20% of the lecture ($w > 0.2$) and students that watched more than 50% of the lecture ($w > 0.5$).

We see in the figure that some parts of the lecture have some spikes in popularity. These parts coincide with the start of a topic, indicating that the student usually jump to a specific part of the lecture. Another interesting behavior is that, after the start of a topic, the number of students drops drastically, indicating that, after a few seconds, the students usually jumps to another topic or stops watching the lecture. The less popular part of this lecture is around second 1600 (around 60% of the lecture). We see that this part reach the lower popularity of the lecture. Only 25% of the students that have started the lecture ($W > 0$) have watched this part. The highest popularity is at the beginning of the lecture, but students with a high engagement ($W > 0.5$) also have high interest in the part around 1000 seconds of the lecture (around 35%). It is interesting to note that this part is the beginning of an exercise.

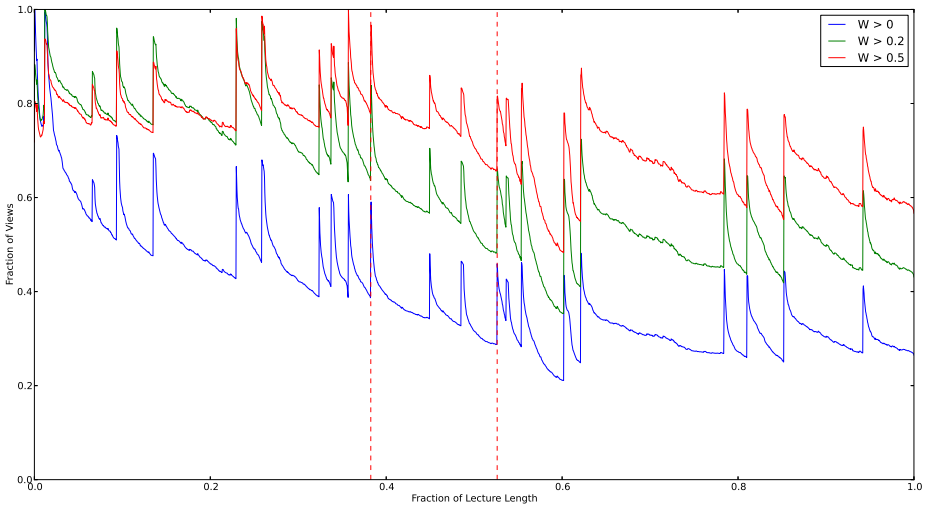


Figure 7.10: Popularity of Each Second of Video Lecture EAD05007

The lecture analyzed in figure 7.10 is the one that presents the game of hanoi tower. The part marked in red are the six and a half minutes that we used to generate the MindLand database, as presented in section 4.1. We see that this part have a high demand, particularly from students with high engagement. But the students lose interest during the exercise. The popularity decreases as the explanation of the exercise reaches its end. It is interesting to compare this result with those presented in figures 4.13 and 6.6. Figure 4.13 is the classification done by the experts about the engagement of the students, and Figure 6.6 is the automatic classification done by our classifier. We see a similar behavior in the three figures, with the students losing interest in this part of the lecture as it approaches its end. This indicates that this information of popularity can also be used as an engagement information

about the students and can be used by the lecturer to improve a specific part of his lecture.

7.4 Summary and Discussion

The engagement metrics is an important metric to study videos. As of 2012, YouTube has started to record a similar metric and adapted its ranking algorithm to take in consideration the time spent by the user in a video (Bendersky et al., 2014). As pointed out by Dobrian et al. (2011), the engagement in a video is directly associated with revenue, like the number of ads.

In this chapter, we have shown that the engagement metric is also important in the analysis of video lectures to measure the engagement of students. Our proposed model of the engagement metric shows that we can use this information to compare different video lectures. We expect to use this information to analyze the improvement of the developed adaptive system over the current system. Also, by analyzing the popularity of each second in a lecture, we can identify the parts of the lecture that have lower interest from the students. This is a valuable information for the lecturer to improve the lecture and for our adaptive system, to intervene in order to improve the engagement.

Chapter 8

Conclusion

In this thesis, we presented an intelligent recommendation system based on video lectures for distance learning. This system is the first to allow the adaptation of a video lecture flow based on passive monitoring of the student using sensors. By using attention to detect the engagement state of the student, we are able to modify the presented material in our adaptive learning system to allow a better experience of the student towards a lecture.

Another contribution is the development of a database of recorded reactions from students while watching a video lecture and interacting with an exercise (The hanoi tower game). Using our system, we built a database composed by over 3 and a half hours of session time, with 19 students and more than 18 time series obtained from 4 different sensors. Our database is unique since we employ live recording of real students (not actors or posed users) with natural reactions and use more sensors than those used to build existing open databases. Particularly, we employed an infrared eye camera. In addition, we extracted features different from those included in the existing databases, like blink rate and pupil size.

Our proposed student model, based on clustering, is a novel approach to the area. The majority of learning systems have their student model based on some kind of supervised learning algorithm or simple rule based models. Also, our is the first to predict engagement of student based on attentive states. By using gaussian mixture model, we built an automatic classification process for engagement that can achieve over 80% of agreement compared to manual classification given by teachers. Based on the analysis of each sensor individually, we found that the eye features are the ones that give the best classification. This is an indication that these features are relevant to teachers to detect the student attention, an indicative of engagement, during a lecture. This is also important for scalability of the hole system, as the eye sensors rely only on a webcam, while other features depend on expensive sensors.

Using a database of over two years of log information from students, we developed a model to analyze the engagement of students in the video lectures. We expect to

use this information to analyze the improvement of the developed adaptive system over the current system. Also, by analyzing the popularity of each second in a lecture, we are able to identify the parts of the lecture that have lower interest for the students. This is a valuable information for the lecturer to improve the lecture and for our adaptive system, to intervene in order to improve the engagement.

8.1 Future Developments

The creation of this intelligent recommendation system based on video lectures for distance learning open new doors for future research. The next step is to improve the system to put it into production, allowing the CEDERJ course and other initiatives to use this system. As more students start to use the system, and more adaptive lectures are developed, we expect an increase in our current knowledge in the actual student behavior towards the system and her engagement in the lectures. This will allow the upgrade of the current system and the development of new methods to improve the student experience in video lectures.

The MindLand database can be expanded from the current 19 students and new sensors can easily be incorporated in the database.

We can expect improvements in the student model, with higher accuracy for the classification of engagement, and new models for other kinds of cognitive states.

Bibliography

Pavlo Antonenko, Fred Paas, Roland Grabner, and Tamara van Gog. Using electroencephalography to measure cognitive load. *Educational Psychology Review*, 22(4):425–438, 2010.

Ryan SJd Baker, Sidney K D'Mello, Ma Mercedes T Rodrigo, and Arthur C Graesser. Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, 68(4):223–241, 2010.

Athula Balachandran, Vyas Sekar, Aditya Akella, Srinivasan Seshan, Ion Stoica, and Hui Zhang. Developing a predictive model of quality of experience for internet video. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 339–350, New York, NY, USA, 2013. ACM.

Mousumi Banerjee, Michelle Capozzoli, Laura McSweeney, and Debajyoti Sinha. Beyond kappa: A review of interrater agreement measures. *Canadian journal of statistics*, 27(1):3–23, 1999.

Erol Basar. *Brain function and oscillations*, volume II of *Springer Series in Synergetics*. Springer, 1st edition, 1999.

Erol Basar, Canan Başar-Eroglu, Sirel Karakaş, and Martin Schürmann. Gamma, alpha, delta, and theta oscillations govern cognitive processes. *International Journal of Psychophysiology*, 39(2):241–248, 2001.

Jackson Beatty. Task-evoked pupillary responses, processing load, and the structure of processing resources. *Psychological bulletin*, 91(2):276, 1982.

Michael Bendersky, Lluis Garcia-Pueyo, Jeremiah Harmsen, Vanja Josifovski, and Dima Lepikhin. Up next: Retrieval methods for large scale related video suggestion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*,

KDD '14, pages 1769–1778, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2956-9. doi: 10.1145/2623330.2623344. URL http://doi.acm.org/10.1145/2623330.2623344.

Ebenezer Rangel Botelho. Patching interativo eficiente: Implementação e analise de desempenho. Master's thesis, Federal Univeristy of Rio de Janeiro, 2008.

Wolfram Boucsein. *Electrodermal activity*. Springer, 2 edition, 2012.

Charles Bouveyron and Camille Brunet-Saumard. Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis*, 71:52 – 78, 2014.

Gary Bradsky and Adrian Kaehler. *Learning OpenCV*. O'Reilly, 2009.

Peter Brusilovsky. Developing adaptive educational hypermedia systems: From design models to authoring tools. In *Authoring tools for advanced technology Learning Environments*, pages 377–409. Springer, 2003.

Peter Brusilovsky. Adaptive hypermedia for education and training. In *Adaptive Technologies for Training and Education*, chapter 3, page 46. Cambridge University Press, 2012.

Peter Brusilovsky and Eva Millán. User models for adaptive hypermedia and adaptive educational systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, chapter 1, pages 3–53. Springer-Verlag, Berlin, Heidelberg, 2007.

CAPES. Universidade aberta do brasil. http://uab.capes.gov.br, august 2015.

CEDERJ. Centro de educação superior a distância do estado do rio de janeiro. http://cederj.edu.br, august 2015.

Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 1–14, New York, NY, USA, 2007. ACM.

Maher Chaouachi and Claude Frasson. Exploring the relationship between learner eeg mental engagement and affect. In *Intelligent Tutoring Systems*, volume 6095 of *Lecture Notes in Computer Science*, pages 291–293. Springer Berlin Heidelberg, 2010.

G. Chatzopoulou, Cheng Sheng, and Michalis Faloutsos. A first step towards understanding popularity in youtube. In *INFOCOM IEEE Conference on Computer Communications Workshops , 2010*, pages 1–6, March 2010.

B Cheung, L Hui, J Zhang, and Siu-Ming Yiu. Smarttutor: An intelligent tutoring system in web-based adult education. *Journal of Systems and Software*, 68(1):11–25, 2003.

Cristina Conati, Natasha Jaques, and Mary Muir. Understanding attention to adaptive hints in educational games: An eye-tracking study. *International Journal of Artificial Intelligence in Education*, 23(1-4):136–161, 2013.

David G Cooper. *Computational affect detection for education and health*. PhD thesis, University of Massachusetts Amherst, 2011.

Coursera. `https://www.coursera.org/`, august 2015.

Scotty Craig, Arthur Graesser, Jeremiah Sullins, and Barry Gholson. Affect and learning: an exploratory look into the role of affect in learning with autotutor. *Journal of Educational Media*, 29(3):241–250, 2004.

K. Crowley, A. Sliney, Ian Pitt, and D. Murphy. Evaluating a brain-computer interface to categorise human emotional response. In *Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on*, pages 276–278, July 2010.

Paul De Bra, Ad Aerts, Bart Berden, Barend de Lange, Brendan Rousseau, Tomi Santic, David Smits, and Natalia Stash. Aha! the adaptive hypermedia architecture. In *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*, HYPERTEXT '03, pages 81–84, New York, NY, USA, 2003. ACM.

Edmundo de Souza e Silva, Rosa Maria Meri Leão, Anna D. Santos, Bernardo C. Machado Netto, and Jorge Allyson Azevedo. Multimedia supporting tools for the cederj distance learning initiative applied to the computer systems course. In *22nd ICDE World Conference on Distance Education*, pages 1 – 11, 2006.

Carolina C. L. B. de Vielmond, Rosa M. M. Leão, and Edmundo de Souza e Silva. Um modelo hmm hierárquico para usuários interativos acessando um servidor multimídia. In *Simpósio Brasileiro de Redes de Computadores*, volume 1, pages 469–482, 2007.

Arnaud Delorme, Terrence Sejnowski, and Scott Makeig. Enhanced detection of artifacts in eeg data using higher-order statistics and independent component analysis. *Neuroimage*, 34(4):1443–1449, 2007.

Sidney D'Mello and Art Graesser. Dynamics of affective states during complex learning. *Learning and Instruction*, 22(2):145–157, 2012.

Sidney D'Mello, Rosalind W. Picard, and Arthur Graesser. Toward an affect-sensitive autotutor. *IEEE Intelligent Systems*, 22(4):53–61, 2007. ISSN 1541-1672. doi: http://doi.ieeecomputersociety.org/10.1109/MIS.2007.79.

Sidney D'Mello, Andrew Olney, Claire Williams, and Patrick Hays. Gaze tutor: A gaze-reactive intelligent tutoring system. *International Journal of human-computer studies*, 70(5):377–398, 2012.

Sidney K. D'mello, Scotty D. Craig, Barry Gholson, Stan Franklin, Rosalind Picard, and Arthur C Graesser. Integrating affect sensors in an intelligent tutoring system. In *Affective Interactions: The Computer in the Affective Loop Workshop at 2005 Intl. Conf. on Intelligent User Interfaces*, pages 7–13, 2005.

Florin Dobrian, Vyas Sekar, Asad Awan, Ion Stoica, Dilip Joseph, Aditya Ganjam, Jibin Zhan, and Hui Zhang. Understanding the impact of video quality on user engagement. In *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, pages 362–373, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0797-0. doi: 10.1145/2018436.2018478. URL `http://doi.acm.org/10.1145/2018436.2018478`.

Ellen Douglas-Cowie, Roddy Cowie, Ian Sneddon, Cate Cox, Orla Lowry, Margaret McRorie, Jean-Claude Martin, Laurence Devillers, Sarkis Abrilian, Anton Batliner, Noam Amir, and Kostas Karpouzis. The humaine database: Addressing the collection and annotation of naturalistic and induced emotional data. In Ana C.R. Paiva, Rui Prada, and Rosalind W. Picard, editors, *Affective Computing and Intelligent Interaction*, volume 4738 of *Lecture Notes in Computer Science*, pages 488–500. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-74888-5.

Toby Dragon, Ivon Arroyo, Beverly P Woolf, Winslow Burleson, Rana El Kaliouby, and Hoda Eydgahi. Viewing student affect and learning through classroom observation and physical sensors. In *Intelligent tutoring systems*, pages 29–39. Springer, 2008.

Matthieu Duvinage, Thierry Castermans, Thierry Dutoit, M Petieau, T Hoellinger, C De Saedeleer, K Seetharaman, and G Cheron. A p300-based quantitative comparison between the emotiv epoc headset and a medical eeg device. *Biomedical Engineering*, 765:2012–764, 2012.

edX. `https://www.edx.org/`, august 2015.

Paul Ekman. Facial expression and emotion. *American psychologist*, 48(4):384, 1993.

Rana el Kaliouby. *Mind-Reading Machines: Automated Inference of Complex Mental States*. PhD thesis, University of Cambridge, 2005.

Leon Festinger. *A theory of cognitive dissonance*, volume 2. Stanford university press, 1962.

Andrew W. Fitzgibbon and Robert B. Fisher. A buyer's guide to conic fitting. In *Proceedings of the 6th British Conference on Machine Vision (Vol. 2)*, BMVC '95, pages 513–522, Surrey, UK, UK, 1995. BMVA Press. ISBN 0-9521898-2-8.

Joseph L. Fleiss, Bruce Levin, and Myunghee Cho Paik. *The Measurement of Interrater Agreement*, chapter 18, pages 598–626. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., 3 edition, 2004. ISBN 9780471445425. doi: 10.1002/0471445428.ch18.

C. Fraley and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8): 578–588, 1998.

AC. Graesser, P. Chipman, B.C. Haynes, and A Olney. Autotutor: an intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4):612–618, Nov 2005a. ISSN 0018-9359. doi: 10.1109/TE.2005.856149.

Arthur C. Graesser and Natalie K. Person. Question asking during tutoring. *American Educational Research Journal*, 31(1):104–137, 1994.

Arthur C Graesser, Shulan Lu, Brent A Olde, Elisa Cooper-Pye, and Shannon Whitten. Question asking and eye tracking during cognitive disequilibrium: Comprehending illustrated texts on devices when the devices break down. *Memory & Cognition*, 33(7):1235–1247, 2005b.

Sabine Graf, Tzu-Chien Liu, Nian-Shing Chen, and Stephen JH Yang. Learning styles and cognitive traits–their relationship and its benefits in web-based educational systems. *Computers in Human Behavior*, 25(6):1280–1289, 2009.

Arjun K Gupta and Saralees Nadarajah. *Handbook of beta distribution and its applications*. CRC Press, 2004.

Ian Hickson. The websocket API. Candidate recommendation, W3C, September 2012. `http://www.w3.org/TR/2012/CR-websockets-20120920/`.

Stefan Hrastinski. Asynchronous and synchronous e-learning. *Educause quarterly*, 31(4):51–55, 2008.

B. Inhelder, H.H. Chipman, and C. Zwingmann, editors. *Piaget and His School*. Springer Study Edition. Springer-Verlag Berlin Heidelberg, 1 edition, 1976.

Jack Jansen and Dick CA Bulterman. Smil state: an architecture and implementation for adaptive time-based web applications. *Multimedia Tools and Applications*, 43(3):203–224, 2009.

Patricia A Jaques, Henrique Seffrin, Geiseane Rubi, Felipe de Morais, Cássio Ghilardi, Ig Ibert Bittencourt, and Seiji Isotani. Rule-based expert systems to support step-by-step guidance in algebraic problem solving: The case of the tutor pat2math. *Expert Systems with Applications*, 40(14):5456–5465, 2013.

Daniel Kahneman and Jackson Beatty. Pupil diameter and load on memory. *Science*, 154(3756):1583–1585, 1966. doi: 10.1126/science.154.3756.1583. URL `http://www.sciencemag.org/content/154/3756/1583.abstract`.

Moritz Philipp Kassner and William Rhoades Patera. Pupil : constructing the space of visual attention. Master's thesis, Massachusetts Institute of Technology, Dept. of Architecture., 2012.

Khan Academy. `https://www.khanacademy.org/`, august 2015.

Hyun-Cheol Kim, Jihun Cha, and Won Don Lee. Eye detection for gaze tracker with near infrared illuminator. In *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*, pages 458–464. IEEE, Dec 2014. doi: 10.1109/CSE.2014.111.

Wolfgang Klimesch. Eeg alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain Research Reviews*, 29(2–3):169 – 195, 1999.

Alfred Kobsa. Generic user modeling systems. *User modeling and user-adapted interaction*, 11(1-2):49–63, 2001.

Sander Koelstra, Christian Muhl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras. Deap: A database for emotion analysis; using physiological signals. *Affective Computing, IEEE Transactions on*, 3(1):18–31, 2012.

J Richard Landis and Gary G Koch. A one-way components of variance model for categorical data. *Biometrics*, 33(4):671–679, 1977.

Peter J Lang. The emotion probe: Studies of motivation and attention. *American psychologist*, 50(5):372, 1995.

Yue Liu, Xiao Jiang, Teng Cao, Feng Wan, Peng Un Mak, Pui-In Mak, and Mang I Vai. Implementation of ssvep based bci with emotiv epoc. In *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2012 IEEE International Conference on*, pages 34–37, July 2012. doi: 10.1109/VECIMS.2012.6273184.

P. Lucey, J.F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101, June 2010.

George D Magoulas and Sherry Y Chen. *Advances in web-based education: personalized learning environments*. Information Science Publishing (IGI Global), 2006.

Fiona Mulvey, Arantxa Villanueva, David Sliney, Robert Lange, Sarah Cotmore, and Mick Donegan. Exploration of safety issues in eyetracking. Technical Report IST-2003-511598, COGAIN EU Network of Excellence, 2008.

Richard Muntz, Jose Renato Santos, and Steven Berson. A parallel disk storage system for real-time multimedia applications. *International Journal of Intelligent Systems*, 13(12):1137–1174, 1998.

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

M. Murugappan, N. Ramachandran, and Y. Sazali. Classification of human emotion from eeg using discrete wavelet transform. *Journal of Biomedical Science and Engineering*, 3:390–396, 2010.

Jeanne Nakamura and Mihaly Csikszentmihalyi. The concept of flow. In *Handbook of positive psychology*, chapter 7, pages 89–105. Oxford University Press, 2002.

Bernardo Calil Machado Netto. Patching interativo: Um novo método de compartilhamento de recursos para transmissão de video com alta interatividade. Master's thesis, Federal Univeristy of Rio de Janeiro, 2004.

NSSE. National survey of student engagement. `http://nsse.indiana.edu/`, august 2015.

Alexandros Paramythis and Susanne Loidl-Reisinger. Adaptive learning environments and e-learning standards. In *Second European Conference on e-Learning*, pages 369–379, 2003.

P.C. Petrantonakis and L.J. Hadjileontiadis. Emotion recognition from eeg using higher order crossings. *Information Technology in Biomedicine, IEEE Transactions on*, 14(2):186–197, March 2010.

Rosalind W. Picard. *Affective Computing*. MIT Press, Cambridge, MA, USA, 1997. ISBN 0-262-16170-2.

Rosalind W. Picard, Elias Vyzas, and Jennifer Healey. Toward machine emotional intelligence: Analysis of affective physiological state. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(10):1175–1191, 2001.

Ming-Zher Poh, Nicholas C Swenson, and Rosalind W Picard. A wearable sensor for unobtrusive, long-term assessment of electrodermal activity. *Biomedical Engineering, IEEE Transactions on*, 57(5):1243–1252, 2010.

William Prokasy. *Electrodermal activity in psychological research*. Elsevier, 1973.

C. Richier, E. Altman, R. Elazouzi, T. Jimenez, G. Linares, and Y. Portilla. Bio-inspired models for characterizing youtube viewcout. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pages 297–305. IEEE Computer Society, Aug 2014. doi: 10.1109/ASONAM.2014.6921600.

Evan F Risko, Dawn Buchanan, Srdan Medimorec, and Alan Kingstone. Everyday attention: mind wandering and computer use during lectures. *Computers & Education*, 68:275–283, 2013.

Cristóbal Romero, Sebastián Ventura, Amelia Zafra, and Paul De Bra. Applying web usage mining for personalizing hyperlinks in web-based adaptive educational systems. *Computers & Education*, 53(3):828–840, 2009.

Jerome I Rotgans and Henk G Schmidt. Cognitive engagement in the problem-based learning classroom. *Advances in health sciences education*, 16(4): 465–479, 2011.

Walton T Roth, Michael E Dawson, and Diane L Filion. Publication recommendations for electrodermal measurements. *Psychophysiology*, 49:1017–1034, 2012.

Paul Rozin and Adam B Cohen. High frequency of facial expressions corresponding to confusion, concentration, and worry in an analysis of naturally occurring facial expressions of americans. *Emotion*, 3(1):68, 2003.

Silvia Schiaffino, Patricio Garcia, and Analia Amandi. eteacher: Providing personalized assistance to e-learning students. *Computers & Education*, 51(4): 1744–1754, 2008.

Jack Sklansky. Finding the convex hull of a simple polygon. *Pattern Recognition Letters*, 1(2):79 – 83, 1982. doi: http://dx.doi.org/10.1016/0167-8655(82)90016-2.

M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic. A multimodal database for affect recognition and implicit tagging. *Affective Computing, IEEE Transactions on*, 3(1):42–55, Jan 2012. ISSN 1949-3045.

Robert Sottilare and Heather Holden. Motivations for a generalized intelligent framework for tutoring (gift) for authoring, instruction and analysis. In *AIED 2013 Workshops Proceedings*, volume 7, pages 1–9, 2013.

Satoshi Suzuki and KeiichiA be. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32 – 46, 1985. doi: http://dx.doi.org/10.1016/0734-189X(85)90016-7.

John Sweller. Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational psychology review*, 22(2):123–138, 2010.

Lech Świrski, Andreas Bulling, and Neil A. Dodgson. Robust real-time pupil tracking in highly off-axis images. In *Proceedings of the Symposium on Eye Tracking Research and*

Applications, pages 173–176. ACM, March 2012. URL http://www.cl.cam.ac.uk/research/rainbow/projects/pupiltracking/.

Pascal W. M. Van Gerven, Fred Paas, Jeroen J. G. Van Merriënboer, and Henk G. Schmidt. Memory load and the cognitive pupillary response in aging. *Psychophysiology*, 41(2):167–174, 2004. ISSN 1469-8986. doi: 10.1111/j.1469-8986.2003.00148.x. URL http://dx.doi.org/10.1111/j.1469-8986.2003.00148.x.

Jeroen JG Van Merrienboer and John Sweller. Cognitive load theory and complex learning: Recent developments and future directions. *Educational psychology review*, 17(2):147–177, 2005.

Thothadri Rajesh Vidyasankar. Intelligent tutoring applied to videolectures: A supporting tool for experimentation and data collection. Master's thesis, Federal Univeristy of Rio de Janeiro, 2013.

Jacob Whitehill, Marian Bartlett, and Javier Movellan. Automatic facial expression recognition for intelligent tutoring systems. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–6. IEEE, 2008.

Hongjing Wu, Erik De Kort, and Paul De Bra. Design issues for general-purpose adaptive hypermedia systems. In *Proceedings of the 12th ACM Conference on Hypertext and Hypermedia*, pages 141–150. ACM, 2001.

Z. Zeng, M. Pantic, G.I. Roisman, and T.S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(1):39–58, Jan 2009. ISSN 0162-8828.