Acknowledgments

Resumen

Abstract

List of Figures

3.1 User interface of the restaurant model. 24

3.2 A two input, two rule Mamdani FIS with crisp inputs. 28

3.3 Defuzzification Using the Center of Mass. 32

3.4 Defuzzification Using the Mean of Maximum. 33

3.5 A two Input, two rule Mamdani FIS with a fuzzy output. 34

5.1 The Post-Filtering architecture for Tijuana restaurants. 51

5.2 The Post-Filtering architecture for Tijuana restaurants. 52

5.3 The Post-Filtering architecture for Tijuana restaurants. 54

5.4 The chart shows the users preferences for questions from 1 to 6. . . . 57

5.5 The chart shows the users preferences for questions 7 and 8. 58

5.6 Time segmentation of contexts based on current user context. 60

5.7 Pre-filtering process for context-aware recommender system. 62

5.8 Gaussian Membership functions in the input are: a) RatingAverage, b) UserParticipation, and an output: c) Recommendation. 65

5.9 Fuzzy Inference System. 65

5.10 Recommender system architecture 67

5.11 RMSE results of matrix factorization test. 71

6.1 Representation of the percent of success for each task. 75

6.2 The radar chart that depicts the four axis evaluated in the questionnaire. . 77

List of Tables

3.1 The Boolean ”or”. 30

5.1 comparison of the different versions. 49

5.2 Contextual factors considered in the questionnaire. 57

5.3 Results of comparison by contexts in MovieLens dataset. 63

5.4 Example of contextual ratings in the user profile. 66

5.5 Datasets description. 67

5.6 Comparison of RMSE. 68

5.7 Level of similarity among items in datasets. 68

5.8 Contexts in InCarMusic dataset. 70

5.9 RMSE of datasets using matrix factorization. 71

6.1 Time on task data for 10 users and 13 tasks. 76

6.2 Confidence interval per task with a confidence level of 95%. 76

Contents

Acknowledgments **I**

Resumen **II**

Abstract **III**

1 Introduction **1**

2 State of the art **4**

3 Background **10**

3.1 User Modeling 10

3.1.1 Application Domains for user modeling 11

3.1.2 User models for providing task support 12

3.1.3 Decision support systems 13

3.1.4 Adaptive Hypermedia 14

3.1.5 Adaptive ubiquitous systems 14
3.1.6 User models for providing a personal experience. 16
3.1.7 Recommender Systems and User Adaptive Computer Games. 17
3.1.8 User Adaptive Computer Games. 18
3.1.9 User models for educational purposes. 19
3.1.10 Intelligent Tutoring Systems. 20
3.1.11 Adaptive Educational Games. 21
3.1.12 Methods for user modeling. 22
3.2 Interactive Evolutionary Computation. 24
3.3 Fuzzy Logic. 24
3.3.1 Fuzzy inference system. 25
3.3.2 Creating fuzzy rules. 26
3.3.3 Fuzzification. 27
3.3.4 Fuzzy combinations (T-norms). 28
3.3.5 Fuzzy "and" 29
3.3.6 Fuzzy "and" 30
3.3.7 Combining Outputs into an Output Distribution 31
3.3.8 Defuzzification of Output Distribution 31
3.3.9 Fuzzy Inputs. 33
3.4 Gamification. 34

3.4.1 Techniques. 35

4 Proposed method **36**

4.1 Graph-based user model 37

4.2 Interface 39

4.3 User activity 41

4.4 Database for interactive evolutionary computation 42

4.5 Fuzzy Inference 43

4.6 Decision Making 44

5 Case study **47**

5.1 EvoDrawings01 48

5.1.1 Interface. 50

5.2 EvoDrawing02 52

5.2.1 Fuzzy Inferance. 53

5.2.2 Interface 55

5.3 MovieLens dataset 59

5.4 Tripadvisor dataset 63

5.5 Datasets in matrix factorization 69

5.5.1 Results 70

6 System evaluation **72**

6.1 Metrics 72

6.2 Enviromental set up 74

6.3 Results 75

7 Conclusions and future work **78**

Publications **81**

A Pseudocode **83**

B USE Questionnaire **90**

Chapter 1

Introduction

It is a reality that the World Wide Web in recent years, is growing exponentially, which means the presence of millions of users on Web sites, Web applications, Web systems, etc. []. There is a wide variety of Web systems, where we have different users interacting with them. These users have different goals when using these Web systems. For example do a search in Google [] of particular topic, make a reservation for a room in a luxury resort, check your bank account or simply view your status on your Facebook account [].This variation of users represents a complex diversity as individuals []. This diversity lies in different skills, interests, preferences and ways of thinking, learning and knowledge [].For this reason users need different ways to interact with the information presented by the great variety of Web systems that exist.

When we intend to customize any element in Web system, we need to know the user's personal information. This information is a collection of needs, characteristics, feelings, tastes, etc. This information is required to be able to form the representation of knowledge about users. This is what is known as user modeling (UM).

A user modeling can be as simple as a profile systems where is basic knowledge of users. Also can be as complex as represent its characteristics, needs, interests, ways to feel. In order to understand specific users. The main goal of user modeling is to represent aspects of the real world of the user's in autonomous automatically way.

In this document we present a user modeling in the context of Web-based interactive evolutionary computation.

Interactive evolutionary computation (IEC) is a branch of evolutionary computation where users become a part of the evolutionary process by replacing the fitness function; evaluating individuals of a population based on their personal preferences[13]. These evaluations are subjective according to the user point of view based on their perceptions, interests and desires.

Normally such systems require users to evaluate large amounts of individuals iteratively, causing them to lose interest for participate by fatigue that is generated[13]. Nowadays some of these systems are migrating to Web technologies looking for volunteers users to collaborate in the evaluations for distribute the load and lower the fatigue. Having Web- based interactive evolutionary systems open the possibility for linked to social platforms in order to involve the largest number possible of users to assist in the evaluation of individuals produced by these systems applications.

Chapter 2

State of the art

Some 1986's Dawkins's research was the pioneer of a significant addition to the 1990s IEC algorithms research works[][Dawkins 1986].

There is two key research approach about his field:

Creative Approach: The Artificial Life (AL) was the base of creative approach. AL uses complex algorithms for biological life models emulation. To perform this task, it is needed to include some of the different techniques starting from right image treatment. Good graphic creation as well as a great music and quality sounds, [Sims 1991b], [Sims 1991c], [Sims 1994], [Dawkins 1986], [Disz 1997], [Unemi 2000]and [Unemi 2003].

Humanized technology approach: The concept of humanized technology approach comes from the approach that is focused on the IEC algorithms interface, this is the research of interaction between humans and computer systems. The main goal of this was to reduce the user's fatigue and to promote the inputs and outputs of algorithms to improve the efficiency of them. IEC has made his own way in practical fields such as engineering, education,etc., [Parmee 1993], [Ventrella 1994a], [Takagi 1996], [Poli 1997], [Parmee 1998] and [Takagi 1998].

Computer graphics (CG) The Biomorph of Dawkins was the first IEC research, from this research comes to many motivated works mostly about the Selfish Gene, come of these works are: [Ochoa 1998], [Mccormack 1993], and [Smith 2003].

In Dawkins work a conventional recursive algorithm was used as a baseline maintaining the main target of trees with an L-system Lindenmayer.

Some 1986's Dawkins's research was the pioneer of the significant addition to the 1990s IEC algorithms research works[Dawkins 1986].

There is two key research approach about his field:

Creative Approach: The Artificial Life (AL) was the base of creative approach. AL uses complex algorithms for biological life models emulation. In order to perform this task it is needed to include some of the different techniques starting from right image treatment , good graphic creation as well as a great music and quality sounds, [Sims 1991b], [Sims 1991c], [Sims 1994], [Dawkins 1986], [Disz 1997], [Unemi 2000]and [Unemi 2003].

Humanized technology approach: The concept of humanized technology approach comes from the approach that is focused on the IEC algorithms interface, this is the research of interaction between humans and computer systems. The main goal of this was to reduce the user's fatigue and to promote the inputs and outputs of algorithms to improve the efficiency of them. IEC has made his own way in practical fields such as engineering, education,etc., [Parmee 1993], [Ventrella 1994a], [Takagi 1996], [Poli 1997], [Parmee 1998] and [Takagi 1998].

Computer graphics (CG) The Biomorph of Dawkins was the first IEC research, from this research comes to many motivated works mostly about the Selfish Gene, come of these works are: [Ochoa 1998], [Mccormack 1993], and [Smith 2003].

In Dawkins work, a conventional recursive algorithm was used as a baseline maintaining the main target of trees with an L-system (Lindenmayer). This same L-system was the base for another experiment to create 2-D CG forms insects from a system called Blind Watchmaker who used L-system angles from L-system output intuitively selected; the creation was called biomorphs. These creations reach his target with the multiple selections of the users based on their preferences; all these selections acted like a natural adaptation filter.

We can find plenty of applications and works for fractal generation [Sims 1991a] and [Sims 1992], [Baluja 1993] and [Baluja 1994], [Lund 1995], or [Angeline 1996],[Raynal 1999] and [Lutton 2003], for rendering in tridimensional, [Todd 1991],[Broughton 1997], [Das 1994] and [Tam 2002], for generation of virtual creatures, [Sims 1994], [Rowland 2000], or aerodynamic surface design (wings), [NGuyen 1993], [NGuyen 1994] and [NGuyen 1997].

We can discover more than one additional way to use this research in the artistic field with several applications of IEC who are used for cartoon face construction and animations matters, like Mutator [Todd 1991], [Todd 1994] and[Todd 1999] or [Bentley 1999a].

The genetic programming (GP) applications offers a category called Interactive Genetic Programming (IGP) with many examples of successful application in tridimensional artwork for artistic animations or construction using mathematical equations as CAVE [Das 1994], [Papka 1996] and [Disz 1997], [Sims 1991], [Sims 1991a], [Sims 1992],[Sims 1993] and [Min 2004]. As this work consequence, Panspermia orPrimordial Dance was created.

Imagen

Imagen

The artistic field is only the first step of a great IEC implementation; it is important to mention another relevant projects called Galapagos, [Sims 1997], and SBART, [Unemi 2000]. The IEC application Galapagos Project is the exhibit in Tokio Multimedia Museum, (NTT Intercommunication Center) and this project originates engaging images to all visitors based on L-systems.

Imagen

There are created after one selection, to get a good solution through multiple repetitions. This action is performed with Genetic Programming (GP), after the calculation of each pixel value using trees of equations combining logarithm, maximum, and minimum, sine, root, cosine, exponential arithmetic operators. AnimationLab is found as an outstanding work who offer figures that can run or walk working with the user to receive more opportunities to be picked. A particular characteristic of all of the figures is that the figures extremities Mentioning open source works, we can find SBART as an IGP [Unemi 2000] tool to create graphics. SBART allow to users to evaluate 20 two-dimensional images, subsequently twenty new image has direction and angles.

Imagen

There are many examples for this field application as [McKenna 1990], [Ventrella 1994a], [Ventrella 1994b], or [Ventrella 1995], [Lim 1999] and [Lim 2000]. One of the Interactive Evolutionary Programming (IEP) artistic application was created by [Angeline 1996], as a fractal generation where the system allows the evolution of animations for the ones who were selected from the user, the application initially show only 10 animations to rate.

Music and sound

It is important to know how IEC was implemented in music generation, with several applications in this field. We will start mentioning the pioneer application

GENJAM, [Biles 1994], [Biles 1996] or [Biles 1999] and [Biles 2000]. Some other attractive works are Sonomorph, [Nelson 1993] and[Nelson 1995], or SBEAT, [Unemi 2003], [Horowitz 1994], [Onisawa 2000], [Tokui 2000] and [Fels 2002]. It is possible to hear a part of the music songs of these previously mentioned works broadcasted in the radio station WDYN. (100.1, New York, USA, WEBPage:http://www.wdyn.net/).

The IEC algorithms are the base for the functionality of the music generation systems, a visual representation of this is given in the below figure:

Imagen

Chapter 3

Background

This chapter present the fundamental concepts related this work. The formal definitions referring to fuzzy systems, contextual factors and recommender system techniques used in the proposed method.

3.1 User Modeling

User modeling can be represented as the technique of building a model of the user to personalize a system. The user model is commonly created as the user is working with the system. An example is an educational application that teaches students an individual skill: given the rules and knowledge in the user model, the difficulty level of the exercises in the form is altered as the user progresses. Formally definition of user modeling according to McTear (1993, p. 158): " user modeling is the process of gathering information about the users of a computer system and of using the information to provide services or information adapted to the specific requirements of individual users (or groups of users)". The purpose of the user model is to have a module containing the operations that are needed to personalize the system, and the user profile, which includes the personal data of the user (Mohamad et al., 2013). System personalization over user modeling is related to the research field of adaptive systems; this subject is beyond the scope of this research work. Focus on the human user, user modeling is a very cross-disciplinary research topic, comprehending the domains of artificial intelligence, computer science, and social science. Ideas have been coopted from an extensive range of subdomains, such as humancomputer interaction, elearning, information science, social computing, machine learning, data mining, cognitive science, and so on (Kay, et al., 2012; Kobsa, 2001). There is interest in user modeling from both a scientific and commercial perspective (Razmerita, 2009).

3.1.1 Application Domains for user modeling

Amount Research and implementation exist in this domain in which personalization and user modeling plays an important role. This section presents several works of these domains. To understand this topic, the different objects are divided into three general categories: * supporting a user during a task. * giving a user a specific personalized experience. * training and educating a user. The categories especially differ in the kind of user data that is used. For each domain, the general purpose of the domain and the more accurate purpose of the user model are discussed.

3.1.2 User models for providing task support

Task support systems are f systems that help a user during a task by either supporting the user perform the task or by completely taking over this task (Nurmi et al., 2007; Brun et al., 2010). For instance, an application that automatically categorizes the incoming emails of the user. The goal of the user model in these requests is to promote the efficiency of interactions with the user, to simplify these interactions and to make complex systems more usable (Razmerita, 2009; Fischer, 2001). To perform this personalization, data is collected through observations of the user. This information is related to the users goals and needs, but especially to the task that the user currently is accomplished, like the users task knowledge and background. Much research has been done in this domain, but because many separate research projects are focusing on an exact task or subject (Sannes, 2011), it is hard to make generalizations or to establish one delimited investigation topic. Commonly discussed research subjects are Decision Support Systems, Adaptive Hypermedia, and Adaptive Ubiquitous Systems, each having his or her own specific domain and way of personalization.

3.1.3 Decision support systems

Decision support systems are systems that support a user with making a decision in a complex, professional environment (Nurmi et al., 2007). For example, a system used at a pharmacy for automatically checking valid combinations of medicine. The method can be used to help the pharmacist in prescribing the right combinations and to give information for making a decision when a problem occurs. The purpose of the user model in decision support systems is to present the user with the right and appropriate information, giving different feedback or applying various decision steps according to the characteristics of the user.The data that is used is often associated with the users task and background knowledge. The adaptation takes place by adapting the amount and the content of the feedback provided by the system.

Decision support systems are traditionally ruled or logic-based systems, in which all the relevant information is represented in a knowledge base. This means that the content of the user model itself is also highly dependent on the way the rules and knowledge are represented.

3.1.4 Adaptive Hypermedia

Adaptive hypermedia system is a system that grant users to browse freely information network, structured by nodes and links, to retrieve items of information (Nurmi et al., 2007; Deepa et al., 2012). For instance an internet website application. The goal of the user model is to make the interface and structure of the system dynamic. This enables the application to adapt to the user and to make it easier for the user to search for and retrieve relevant information. The data used in the user model is related to the users abilities, knowledge, and goals in the application. The adaptation happens by adjusting the structure and the presentation style to the expected needs of the user. For example, by enhancing web search: promoting pages that might better correspond to the users characteristics, on the other hand by giving navigation support, through highlighting certain components of a page (Razmerita et al., 2012).

3.1.5 Adaptive ubiquitous systems

Ubiquitous systems are concerned with data handling applications integrated into everyday objects and activities (Nurmi et al., 2007). For example the smart energy meter, recording the energy usage in a household through small devices distributed in a house, supporting the user with managing this energy usage(Hargreaves et al., 2010). * The main purpose of the user model is to improve the system, facilitate the users preferences and thus make the overall use easier. * Because the personalization can take place in every situation and location, the data is focused on the user state and context. For example to enable the contextualization to a current environmental change. *The adaptation takes place by changing the behavior of, and the feedback given by the whole system. These objects can be inferred by looking at the properties of the objects in the user profile, or by looking at the objects in other user profiles that are similar to the user (Kobsa, 2001; Kay et al., 2012). Because of the predominantly commercial goal of these systems, the adaptations often take place in a very intrusive way, to make sure the user notices the change. Most recommender systems used on the Internet, which means that some typical technical difficulties are associated with these kinds of user models. First, the user profile is often only saved during the users visit, which means that fast and efficient adaption is necessary. Recommender systems often become more precise when the user spends more time on the system. Second, the systems structure is often split up in a client and on a server side, where the client side solely gathers user information and sends it to the server, where the actual computation takes place.

3.1.6 User models for providing a personal experience.

User models for providing the user with a personal experience have the goal to improve the user experience while using the system. This kind of user modeling is especially focused on more commercial fields, such as e-commerce, marketing, and computer games, and became popular with the rise of the Internet. The information that is used by the user in this main domain is mostly focused on the information that defines the user, such as the users preferences and interests. Since this data is regularly delicate, privacy is a big issue (Toch et al., 2012). While in other domains the privacy of the user data is also important, in this area it is even a greater topic of discussion because the incentive of the application developers is frequently contradictory to the incentive of the actual user, considering gaining and sharing the users personal information. For instance, user profiles are often shared among diverse components of the same application, or even with different applications (Brun et al., 2010; Karam et al., 2012), which presents additional weaknesses and possible undesirable information sharing. Ensuring personal data is not open to all people, in addition to defining strict privacy policies, is thus essential in these user models. Some investigation in this domain are.

3.1.7 Recommender Systems and User Adaptive Computer Games.

Recommender systems are concerned with presenting the user with relevant information and suggestions. They are commonly used on the Internet, for example on websites such as Facebook, to provide the user with personalized news, targeted advertisements and possibly new friends (Brun et al., 2010). The purpose of the user model is to give the system with information that is assumed to be important for the user. The information that is stored for this goal is associated with the preferences of the user to certain objects, like products, music or people. To benefit a classification of these objects, the interaction history of the user is stored, or the user is explicitly asked to rate certain objects. The content of the system is eventually adjusted by showing the recently inferred objects. In these senses, objects can be inferred by looking at the attributes of the objects in the user profile, or by looking at the objects in other user profiles that are related to the user (Kobsa, 2001; Kay et al., 2012). As a result of the predominantly commercial target of these systems, the adaptations often take place in a very invasive way, to make sure the user notices the change. Most recommender systems are based on the Internet, which means that some typical technical difficulties are associated with these kinds of user models. First, the user profile is often only stored while the users visit, which means that fast and efficient adaption is important. Recommender systems usually become more precise when the user spends more time with the system. Second, the systems architecture is usually client-server, where the client side gathers user information and sends it to the server, where the actual process takes place.

3.1.8 User Adaptive Computer Games.

User Adaptive computer games are games that focus on increasing the perceived value by providing a strongly individualized experience (Brisson, 2012). For example is a firstperson shooter that adapts the performance of the enemy according to the shooting accuracy of the player.

The fundamental idea of the user model is to identify or classify the user, so the appropriate adjustment is made in the computer game. The information that is used addresses the preferences and progress of the user, such as the users current difficulty level or even the employed strategy. This data is usually obtained through the interactions of the user with the game, and therefor first should be translated and formalized before it can be used to interpret conclusions on a higher level. The adaptation that takes place in the game concerns changing the content and role of the game, such as the game difficulty, the behavior of nonplayer characters or even the background music (Bakkes, et al., 2012) .

Because of the emphasis on the user, user adaptive computer games have relatively a lot of processing power available for personalization. In this sense the user adaptive computer games domain is a very interesting research domain.

3.1.9 User models for educational purposes.

Educational systems are systems developed with a teaching reason. They are commonly applied in elearning, where electronic media and Information technologies are used for education. However, in most educational systems, user modeling and adaptation plays a minor role. Content is presented, and only simple things such as the students progress in the course are registered. By adding personalization to these applications, the learning value can notably increase, ensuring that every learner achieves and reaches the highest standards possible (Heller et al., 2006). Also, the experience of the teacher or supervisor can be increased through personalization, for instance through inferring and employing the preferred teaching style. However, here we consider the student as the user to which the system will be personalized. Thus, it is preferable that the data stored in the user profile be interpretable by humans. When looking at the time of adjustment in educational systems, we can make a clear difference between adaptation while the student is doing an exercise, which we will refer to as online adaptation, and adaptation that takes place afterward, which we will refer to as offline adaptation. The most important investigation domains that do utilize considerable user modeling constructions are Intelligent Tutoring Systems and Adaptive Educational Games.

3.1.10 Intelligent Tutoring Systems.

Intelligent Tutoring Systems (ITS) are systems that provide students automated stepbystep instruction as the students complete training tasks and/or work on exercises. An ITS has the purpose to complement or even replace the human teacher. For example a system for teaching students how to program, with the ability to automatically detect common mistakes (ElsomCook, 1993).

The particular goal of the user model is to select educational activities and strategies and in addition delivered individual feedback that is most relevant to the users level of knowledge (Kobsa, 2001; McTear, 1993). The user information that is stored for this purpose is the students state, knowledge and level of achievement. This data is exclusively observed over the actions and results of the student, such as the answers the student gives. After observing this information, it is used to infer higher level properties, such as the students learning style and other preferences.

Traditionally, just like decision support systems, information technology systems are knowledge based systems, using formalized domain knowledge and rules to drive the user adjustment (adaptation). For instance stereotypes are widely used in information technology systems(Kay, 2000) and represent a set of default attributes that often cooccur in users or in a certain group of people. The different stereotypes that have been build differ in granularity of detail and complexity.

3.1.11 Adaptive Educational Games.

Adaptive Educational Games (AEGs) are complicated educational games that combine ideas from several investigations areas, to increase the students learning experience (Peeters, et al., 2012a). These are especially based on serious games: computer games with an educational approach, where things are taught to students by using a playful idea (Korteling, et al., 2011; Johnson, et al., 2005). For instance an AEG is a training application for fire fighters, letting the fire fighters train their skills and knowledge in a safe on a virtual environment.

The objective of the user model in an AEGs is to optimize the learning process and outcome. The user information is considered with the advance and knowledge of the student, but also with the students mental and cognitive characteristics. The gained data can be used to adjust the content, presentation, and system behavior to the students need, for example, by adjusting the content, tone, or amount of presented feedback. Adaptive computer games have a lot of processing power available for personalization, making a complex and interesting domain for user modeling.

3.1.12 Methods for user modeling.

In the user modeling topic, researchers have proposed more general design methods and frameworks to guide the developers in the process of user modeling. These general methods are useful in research projects, where the knowledge can be reused to adjust the user model to the systems characteristics. Also in commercial applications, these general methods have proven to be useful (Brun et al., 2010), because they make it easier and more feasible to implement personalization into a system. In early work, the process of user modeling was mostly based on the intuition and experience of the developer or researcher. In recent work, the techniques of user modeling were essentially based on the intuition and expertise of the developer or researcher. As the user modeling research field evolved, there has been put much effort in creating a general way for designing and constructing a user model, by basing decisions on more empirical grounds and by defining methods applicable to the whole field (Kobsa, 2001; Durrani, 1997).

Frameworks, methodologies, and architectures have been developed, defining the strict process, restrictions and choices on how to design and build a user model. In the early days of user modeling, the focus was put on developing one method applicable to the user modeling field as a whole. However, user modeling is a very cross-disciplinary research subject. Therefore, throughout the decades, the user modeling area of research has been influenced by the important research topics and trends of their time. For example, when information technologies became a major subject in the early nineties, user modeling methods were also mostly focused on the application of stereotypes, knowledge bases, and logic to define a user model. With the rise of the Internet, the objectives of the user modeling field change to Web-oriented applications and all the specific problems that arise with this. Thus this connection, also the general user modeling methods that were developed, were focused on the popular research domains of their time (Kay et al., 2012). The main approaches to user modeling did not change, but the specific fillingin of the user model, such as which technology to apply, did change. In this sense development of user modeling as a whole, most researchers eventually agreed that one method to solve all problems is not possible (McTear, 1993; Kobsa, 2001). Instead, a broad range of generic user modeling methods has been developed (Fischer, 2001); each of which supports only a few of the very different manifestations of personalization. In the rest of this section, the general user modeling architecture and the most interesting general and domain specific methods will be shortly discussed.

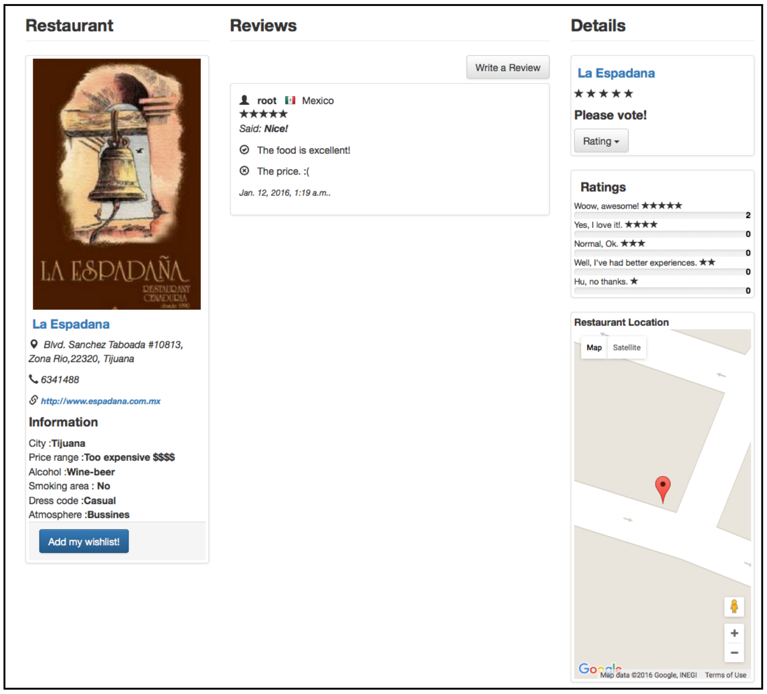


Figure 3.1: User interface of the restaurant model.

3.2 Interactive Evolutionary Computation.

3.3 Fuzzy Logic.

Fuzzy logic is an approach to computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based. The idea of fuzzy logic was first advanced by Dr. Lotfi Zadeh of the University of California at Berkeley in the 1960s. Dr. Zadeh was working on the problem of computer understanding of natural language. Natural language (like most other activities in life and indeed the universe) is not easily translated into the absolute terms of 0 and 1. (Whether everything is ultimately describable in binary terms is a philosophical question worth pursuing, but in practice much data we might want to feed a computer is in some state in between and so, frequently, are the results of computing.) It may help to see fuzzy logic as the way reasoning works, and binary or Boolean logic is simply a special case of it. Fuzzy logic includes 0 and 1 as extreme cases of truth (or "the state of matters" or "fact") but also includes the various states of truth in between so that, for example, the result of a comparison between two things could be not "tall" or "short" but ".38 of tallness."

Fuzzy logic seems closer to the way our brains work. We aggregate data and create some partial truths which we aggregate further into higher truths which in turn when certain thresholds are exceeded, cause certain further results such as motor reaction. A similar kind of process is used in neural networks, expert systems, and other artificial intelligence applications. Fuzzy logic is essential to the development of human-like capabilities for AI, sometimes referred to as artificial general intelligence: the representation of generalized human cognitive abilities in software so that, faced with an unfamiliar task, the AI system could find a solution.

3.3.1 Fuzzy inference system.

A fuzzy inference system (FIS) is a system that uses fuzzy set theory to map inputs (features in the case of fuzzy classification) to outputs (classes in the event of fuzzy classification). An example of a Mamdani inference system is shown in Figure x To compute the output of this FIS given the inputs; one must go through six steps:

1. **Determining a set of fuzzy rules.**

2. **fuzzifying the inputs using the input membership functions.**

3. **Combining the fuzzified inputs according to the fuzzy rules to establish a rule strength.**

4. **Combining the fuzzified inputs according to the fuzzy rules to establish a rule strength.**

5. **Combining the consequences to get an output distribution.**

6. **Defuzzifying the output distribution (this step is only if a crisp production (class) is needed).**

The following is a more detailed description of this process.

3.3.2 Creating fuzzy rules.

Fuzzy rules are a collection of linguistic statements that describe how the FIS should make a decision regarding classifying an input or controlling an output. Fuzzy rules are always written in the following form: if (input1 is membership function1) and/or (input2 is membership function2) and/or . then (output is output membership function). For example, one could make up a rule that says: if temperature is high and humidity is high then room is hot. There would have to be membership functions that define what we mean by high temperature (input1), high humidity (input2) and a hot room (output1). This process of taking an input such as temperature and processing it through a membership function to determine what we mean by "high" temperature is called fuzzification and is discussed in section 3.1.2. Also, we must define what we mean by "and" / "or" in the fuzzy rule. This is called fuzzy combination and is discussed in section 3.1.3.

3.3.3 Fuzzification.

The purpose of fuzzification is to map the inputs from a set of sensors (or features of those sensors such as amplitude or spectrum) to values from 0 to 1 using a set of input membership functions. In the example shown in figure 3.2, there are two inputs, x0, and y0 is shown in the lower left corner.These inputs are mapped into fuzzy numbers by drawing a line up from the inputs to the input membership functions above and marking the intersection point.

These input membership functions, as discussed previously, can represent fuzzy concepts such as "large" or "small", "old" or "young", "hot" or "cold", etc. For example, x0 could be the EMG energy coming from the front of the forearm and y0

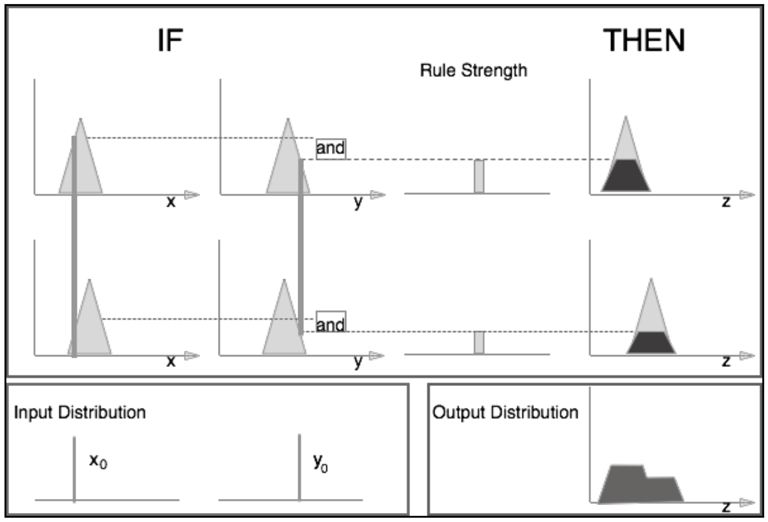


Figure 3.2: A two input, two rule Mamdani FIS with crisp inputs.

could be the EMG energy coming from the back of the forearm. The membership functions could then represent "large" amounts of tension coming from a muscle or "small" amounts of tension. When choosing the input membership functions, the definition of what we mean by "large" and "small" may be different for each input.

3.3.4 Fuzzy combinations (T-norms).

In making a fuzzy rule, we use the concept of "and", "or", and sometimes "not". The sections below describe the most common definitions of these "fuzzy combination" operators. Fuzzy combinations are also referred to as "T-norms".

3.3.5 Fuzzy "and"

The fuzzy "and" is written as:

$$\mu_A \cap \mu_B = T(\mu_A(x), \mu_B(x)) \tag{3.1}$$

where A is read as "the membership in class A" and B is read as "the membership in class B". There are many ways to compute "and". The two most common are:

1. Zadeh - $min(\mu_A(x), \mu_B(x))$. This technique, named after the inventor of fuzzy set theory simply computes the "and" by taking the minimum of the two (or more) membership values. This is the most common definition of the fuzzy "and".

2. Product - $\mu_A(x) + \mu_B(x)) - \mu_A(x)\mu_B(x))$. This technique uses the difference between the sum of the two (or more) membership values and the product of the membership values.

Both techniques have the following properties:

- $T(a, 0) = T(0, a) = a$

- $T(a, 1) = T(1, a) = 1$

Similar to the fuzzy "and", both definitions of the fuzzy "or" also can be used to compute the Boolean "or". Table 3.1 shows the Boolean "or" operation. Notice

Table 3.1: The Boolean ”or”.

Input 1	Input 2	Input 3
0	0	0
0	1	1
1	0	1
1	1	1

that both fuzzy ”or” definitions also work for these numbers. The fuzzy ”or” is an extension of the Boolean ”or” to numbers that are not just 0 or 1, but between 0 and 1.

3.3.6 Fuzzy ”and”

The consequence of a fuzzy rule is computed using two steps: 1. Computing the rule strength by combining the fuzzified inputs using the fuzzy combination process discussed in section 4.1.3. This is shown in Figure 4-1. Notice in this example, the fuzzy ”and” is used to combine the membership functions to compute the rule strength. 2. Clipping the output membership function at the rule strength. Once again, refer to Figure 4-1 to see how this is done for a two input, two rule Mamdani FIS.

3.3.7 Combining Outputs into an Output Distribution

The outputs of all of the fuzzy rules must now be combined to obtain one fuzzy output distribution. This is usually, but not always, done by using the fuzzy ”or”. Figure 4-1 shows an example of this. The output membership functions on the right-hand side of the figure are combined using the fuzzy ”or” to obtain the output distribution shown in the lower right corner of the figure.

3.3.8 Defuzzification of Output Distribution

In many instances, it is desired to come up with a single crisp output from a FIS. For example, if one was trying to classify a letter drawn by hand on a drawing tablet, ultimately the FIS would have to come up with a crisp number to tell the computer which letter was drawn. This crisp number is obtained in a process known as defuzzification. There are two common techniques for defuzzifying:

1. Center of mass - This technique takes the output distribution found in section 4.1.5 and finds its center of mass to come up with one crisp number. This is computed as follows:

$$z = \frac{\sum_{j=1}^{q} z_j \mu_c(z_j)}{\sum_{j=1}^{q} \mu_c(z_j)} \tag{3.2}$$

   where z is the center of mass and uc is the membership in class c at value zj. An example outcome of this computation is shown in Figure 4-2.

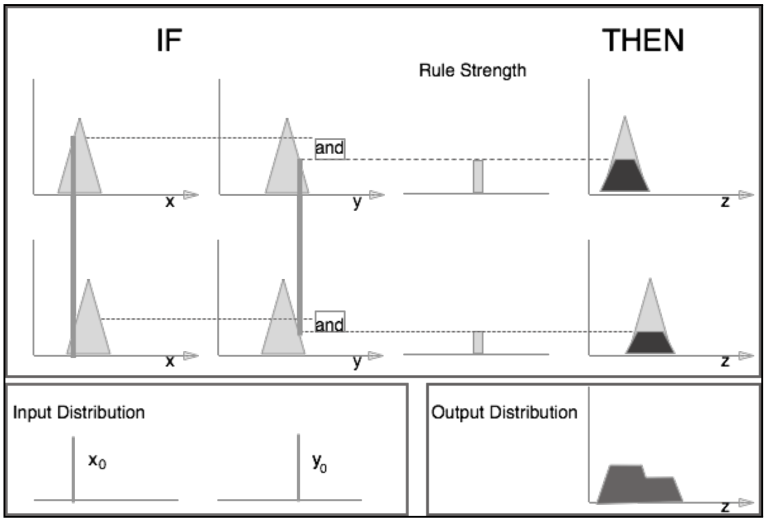


Figure 3.3: Defuzzification Using the Center of Mass.

Falta agregar imagen al path

2. Mean of maximum - This technique takes the output distribution found in section 4.1.5 and finds its mean of maxima to come up with one crisp number. This is computed as follows:

$$z = \frac{\sum_{j=1}^{l} z_j}{l} \tag{3.3}$$

where z is the mean of maximum, zj is the point at which the membership function is maximum, and l is the number of times the output distribution reaches the maximum level. An example outcome of this computation is shown in Figure x.

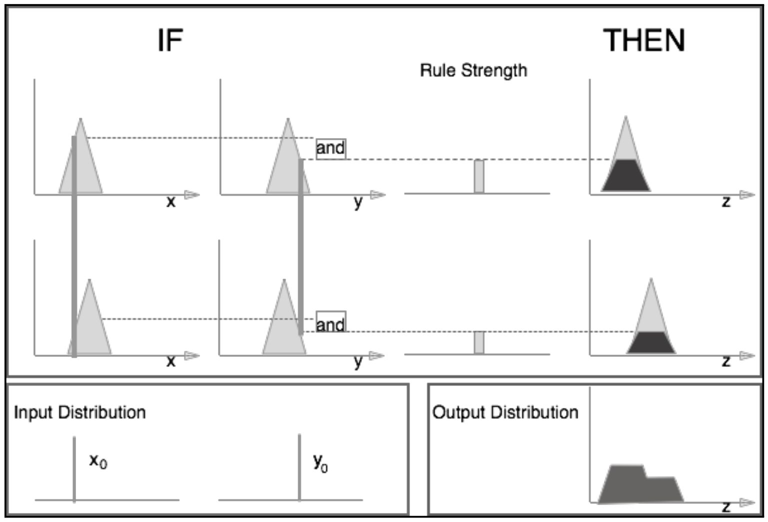


Figure 3.4: Defuzzification Using the Mean of Maximum.

Falta agregar imagen al path

3.3.9 Fuzzy Inputs.

In summary, Figure x-1 shows a two input Mamdani FIS with two rules. It fuzzifies the two inputs by finding the intersection of the crisp input value with the input membership function. It uses the minimum operator to compute the fuzzy "and" for combining the two fuzzified inputs to obtain a rule strength. It clips the output membership function at the rule strength. Finally, it uses the maximum operator to compute the fuzzy "or" for combining the outputs of the two rules. Figure x-4 shows a modification of the Mamdani FIS where the input y0 is fuzzy, not crisp. This can be used to model inaccuracies in the measurement. For example, we may

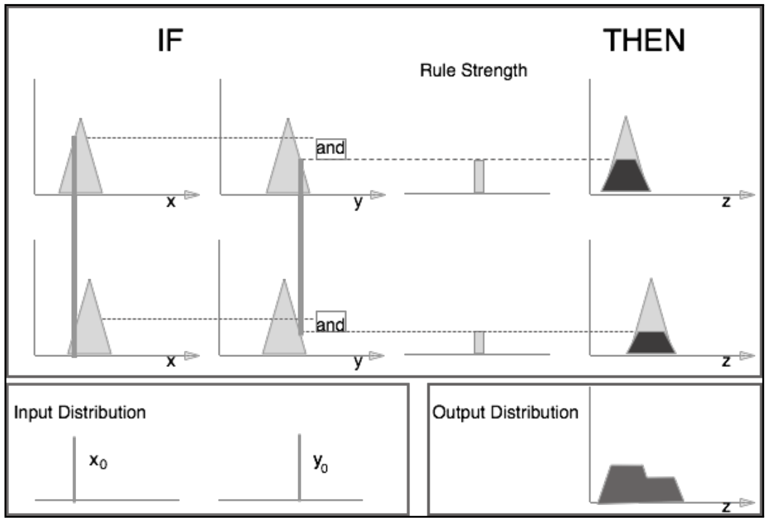


Figure 3.5: A two Input, two rule Mamdani FIS with a fuzzy output.

be measuring the output of a pressure sensor. Even with the exact same pressure applied, the sensor is measured to have slightly different voltages. The fuzzy input membership function models this uncertainty. The input fuzzy function is combined with the rule input membership function by using the fuzzy "and" as shown in Figure x-4.

falta agregar el path de la imgen

3.4 Gamification.

A definition given by referencia is Gamification is the process by which concepts are brought to the real world task associated with real people. Also gamification handle game design elements which are commonly known as non-game context in the presud to enhance user engagement, organizational productivity, flow, learning, evaluations, among others.

3.4.1 Techniques.

Techniques in this context seek to influence humans natural desires for competing, learning, mastery, achievement, status, self- expression, and socializing[][][][]. Early approaches use rewards for players who accomplish desired tasks or competition to engage players[]. For instance some sort of rewards include points,[25] achievement badges or levels,[26] the filling of a progress bar,[27] or providing the user with virtual currency.[26]. By Making the rewards for tasks achievements visible to other players or providing leader boards are ways of encouraging players to compete.[28]. Because the problematic consequences of competition, which can result in negative conduct, low cooperation and collaboration, or disadvantaging certain player demographics such as women,[29] best-practice gamification designs try to refrain from using this element.

Another techniques to gamification is to make existing tasks feel more like games.[30] Some techniques used in this approach include adding meaningful choice, onboarding with a tutorial, increasing challenge,[31] and adding narrative.[30]

Chapter 4

Proposed method

This chapter we explain the proposal method describing all the differents techniques and tools to apply user modeling in interactive evolutionary computation area using fuzzy logic, graphs theory and ”gamification” paradigm.

A fundamental aproach of this research is to develop a user modeling for interactive evolutionary computation in order to increase the user participation and to minimize the amount of evaluations for the evolutionary process in given web-based application.

The web-based IEC applications in this work are using EvoSpace framework mentioned in chapter X. It is important to mention that the model it is limited to the web-based interactive evolutionary computation context.

In following sections we explain the techniques mentioned above.

4.1 Graph-based user model

To develop a user model, it was necessary to know the factors that are involved in interactive evolutionary computation system . The key factors that drive modeling users are described here:

Individuals: The individuals are agents (entities) that create the population of the interactive algorithm.

Users: The users are the entities (agents) involved in the evaluation of individuals generated in the interactive genetic algorithm replacing the fitness function. User interface: It is the platform where users make their evaluations in an easy and fast way. User activity: They are the set of actions that the user performs in the application.

The following user modeling based on graph theory was conceived knowing these factors.

We can find the vertices that are defined by the users, individuals, and collections as we can observe in figure 1. The users has the following properties: The following user modeling based on graph theory was conceived knowing these factors. Identifier Creation Date Description

The individuals has properties as well: Identifier Creation Date Name Type

The collections has the following properties: Identifier Creation Date Name Type

The edges in Figure 1 represents the relationship between the vertices.

The edge KNOWS has the following properties: Identifier Creation Date

The edge HAS has the following properties: Identifier Creation Date

The edge LIKES has the following properties: Identifier Creation Date Rating

Finally the edge PARENT has the following properties: Identifier Creation Date

Imagen

When users interact with the application, a graph theory model is formed. This model is described below.

The model starts when users access the application for the first time through his "Facebook" account; after completion of this action a node, with the identifier that gives the social platform knowledge, is created. Internally, the call is made to the node object with the creation method where the identifier parameters, creation date, type, and name are executed. If users had previously logged, then the application searches through the identifier to prevent the node replication.

Later when users start the participation with the individual's evaluation, in the application an edge is created; this edge is called "LIKES" and represents the relationship between the individual node and the node of the user who is evaluating.

It is important to mention that an EvoSpace initial population of individuals replication, in the form of graphs, is made. This replication has the main goal to generate the individual's nodes and perform the current assessment.

If the user creates a collection in the application, a collection node type, with attributes identifier and name, is generated. Now if the user intends to store a digital painting according to his preference, in one of the collections previously created, a relationship called HAS is created; this relationship is between the collection node and the individual node which represents the paint chosen to be stored.

The relationship between the user node and nodes that represent all his friends is represented by an edge called "KNOWS." To create this edge, it is necessary to search for users who have installed the application "Facebook" and subsequently create this relationship.

To create the edge that represents the "KNOWS" relationship is necessary to search for users who have installed the application "Facebook" and subsequently create this connection between the user node and nodes that represent all the friends.

4.2 Interface

In particular, we shall explain how it was necessary to create an application called EvoDrawings to improve the way users interact collaborate with an interactive evolutionary computing system. One important class of this application is the user interface, which is the model we will explain in this chapter.

EvoDrawings is a web-based application of interactive evolutionary computing system. In these days when social media platforms has become the perfect tool for masses to exchange information/ideas, interests and creations, EvoDrawings include as a common requirement to the users to have an account of the social platform Facebook to have access to the application and consequently participate in it.

The main point to have users logged in the application is that they can evaluate individuals that are formed as a digital paint composed for a single chromosome of nine positions of real numbers. Each post of the chromosome represents some figure, color, performing behavior, etc. In figure X we can observe a chromosome composed of 15 elements from a digital paint. Once the individual is shown, the users can proceed to evaluate the individual subjectively. This means that according to the user's preferences the user can print his taste in all the evaluations. The evaluation method consists of giving the desired rating through an interactive visual component of five stars. The interactive visual component allows the user to select from one to five stars to evaluate the individual, having the one star as a slightly liking rate and the five stars like a total satisfaction rate. The application enables the user to visualize his social network and the personal evaluation of the users from his social net. In addition to this, the users can create collections with the main goal to allow the users to stock the more pleasing digital paints based on his preferences. Also, the user interface contains a section called About where the application explains to users in a general way what the application is all about.

We can find the vertices that are defined by the users, individuals, and collections as we can observe in figure 1. The users has the following properties:

4.3 User activity

The user activity is not so different from how the graph is formed, with the difference that the information generated is formed using the standard specified by "JSON activity stream 1.0," which is stored in the engine database NoSQL "Redis ."

An activity consists of 4 elements: an actor, a verb, an object and a target. The activity generates the history of a user and performs an action on an object. For example - "Christian likes the individual 3" or "Mario created a collection". In most cases, the components will be explicit, but may also be implicit.

The primary goal of this specification is to provide sufficient metadata about the activity, to help the consumer of these data to present the information to the user, in a straightforward and user-friendly format. This involves building simple sentences on the activity that is happening, as well as the visual representation of the activity.

So far, we can say that an "Activity Stream" is a collection of one or more activities of an individual (user). This specification does not define the relations between the activity within the collection, therefore remains to the interpretation of the user who implements it.

4.4 Database for interactive evolutionary computation

The database for interactive evolutionary computation consists of two databases, as we observe in figure X. In one hand we have the EvoSpace database that uses Redis database engine that is already explained in Chapter X section X.

This database contains the structure of individual user data, where every user participation is being stored as well as the fitness of the individual, the user identifier, the representation of the chromosome, etc.

It is necessary to have this information because it is used for fuzzy inference block, which is explained in another chapter. On the other hand, we have a relational database, where basic user information is stored, such the ID, the email, the session, etc. This database also stores everything needed to meet the requirements of login for "Facebook". One important additional information is that contains the structure for storing collections as we can see in Figure X.

4.5 Fuzzy Inference

In this chapter, we will focus on the explanation of fuzzy inference block. This block uses fuzzy inference to acquire a parameter having the weight function to adjust the participation in interactive evolutionary computing applications. This parameter is used in the decision block which will be explained later. Additionally, the parameter is also acquired from the information generated in different databases.

The fuzzy inference system is Mamdani type and is composed of three inputs and one output. Where entries are defined by the preference variable that is also composed of three functions of triangular membership; these features are called low, medium, high, and have ranged from 1 to 5. This range is given by the preference that the user assigns to the individual at the time of assessment.

We also consider the input variable called ”experience” and it is defined by three functions of triangular membership under the name of low, medium and high in a range of 1 to 100. The range of this variable are acquired from the activities that the user performs in the application, and empirically with each activity, a score is assigned. For example, if the user makes a login, a three-point score is assigned, as well if the user evaluates an individual, a two-point score is assigned, all the activities has a score punctuation and also the user has a score limit of one hundred points.

The third variable which is called ranking it is also defined by three triangular membership functions with the name of low, medium, high, in a range of 1 to 30. This range is defined by a ranking process as well as is also performed in video games; the range is adjusted according to the user participation.

This involvement is acquired from all the cardinality of the graph that the user has and passes through the logarithmic equation X that calculates the value of ranking. Finally, the exit "fuzzyrate" is in a range of 1 to 100 defined by three triangular membership functions with the name of bad, normal and good. In figure X we can see this fuzzy inference system.

Falta imagen aqui

We need the rules here please. ??????

4.6 Decision Making

The decision block it is defined by equation 1 representing the value of fitness that takes the individual to be evaluated by the user, as well as everything else that makes in the application.

equation please

Where represents the fuzzy fitness function for all users who have evaluated an individual of the population in particular. Likewise represents the range that a user assigned to the individual according to their preferences. The function represents our fuzzy inference and is defined by the equation 2.

equation please

Where is still the range that users assigned to a particular individual. The variable represents the user experience, which is a function that is defined in equation 3

equation please

Where represents user activity. The variable represents the taste generated of the verb "like" from the user activity stream. The variable represents the access that like the variable is obtained from the verb "join" from the user activity stream. The variables are represented by the verbs "save" and "open" from the user activity stream.

equation please

In equation 2, we can find the variable as the last entry and represents a range of the user; this variable is defined by a logarithmic scale where intervenes our user modeling based on graphs and is defined by Equation 4.

equation please

In equation 4, the variable represents the scale, levels represents the highest level that the users can have. The variable represents the endpoints, which are the maximum points that a particular user can have.

In order to acquire the user's level of range, a function floor is calculated, the main point of this is to increase the difficulty for the high ranked users to level up. In other words the expert users needs to have more participation if they want to raise their level range. This function is defined by equation 5

equation please

Where represents the level of range, represents the scale and are all the participations that the user has made. These participations represents the degree that the user has within his own graph and is defined by the vicinity of its vertex that is given by the adjacent vertices to , defined in Equation 6.

equation please

In this case the degree of the vertex is number of neighbors of:

equation please

we represent this participation as we can see in figure x.

Chapter 5

Case study

This chapter will explain the experiment used in the Study Case. Three versions of EvoDrawing were used to fulfill this research.

Starting off with the hypothesis which will allow increasing the user experience of having a drawing modeling using diffuse logic and use techniques.

It was necessary to design Three experiments to prove this thesis work hypothesis; each experiment used a different version of EvoDrawing. The results as well as each version of EvoDrawing compared to other applications of interactive evolutionary computing and also were compared between other Evo Drawing versions itself. Equality of data, parameters, and users compared to all experiments which examined as follows:

It is necessary to compare our proposal with other which we have explained early in the last chapter, due evolutionary interactive computing applications with the same amount of parameters are difficult to find we decided to build two additional ones in order to run pertinent tests so our hypothesis could be answered., This required to design experiments the following way.

The X table contains Evo Drawing versions and comparison. The first column will specify each version of Evo Drawing including its characteristics and each subsequent column describes the characteristics of each Evo Drawing version.

Now we will explain each version developed details and goes as follows; we will explain each version specific and technical reasons.

5.1 EvoDrawings01

This version initial configuration of our evolutionary algorithm goes as follows:

1. **Having an initial 80 individual population which we will represent in the following way:**

2. **evaluations equals an evolution parameter.**

3. **Its fitness provided by equation x.**

4. **Combining the fuzzified inputs according to the fuzzy rules to establish a rule strength.**

Table 5.1: comparison of the different versions.

Version	EvoDrawing01	EvoDrawing02	EvoDrawing03
User Interface	This version have an easy to handle and simple gui.Observe figure X	This version have an easy to handle and simple gui. Observe figure X	As the previous versions this one have the same interface but differs from the others on the competitiveness visual components based on video games and we can observe it in figure X
Fuzzy Inferance	It does not have Fuzzy inference system	A fuzzy inference system has been added to this version. It is composed by two inputs Experience and Preference; it also has a type Mamdani output called fuzzy rate.	A fuzzy inference system has been added to this version. It is composed by three inputs Ranking, Experience and Preference; it also has a type Mamdani output called fuzzy rate.
Graph-based user model	User modeling is not available	User modeling is not available	User modeling is available in this version based on graphs and is used to adapt inside diffuse inference aptitude function on ranking entrance.

5. **Combining the consequences to get an output distribution.**

6. **Defuzzifying the output distribution (this step is only if a crisp production (class) is needed).**

5.1.1 Interface.

An easy use individual evaluation web interface is develop to this version. We can notice in the navigation bar the access application functionality, worth mentioning that the app will only work with the social platform Facebook. Once accessed with a Facebook account the navigation bar will show an avatar as well as the exit application functionality through a logout. Also inside this interface exist the visual element Friends that will show all the participating friends in the form, to be able to view friends public collections is another functionality this element has. Continuing with the interface explanation, we found the Collections panel where the user's list of created and saved collections is shown as well as the functionality that give the ability to create new collections. An About this section will be shown inside; this will explain in a general way the functionality to the participating users. Now when it comes to users individual evaluation they will have a visual canvas element who will show behavior of a to be evaluated individual (animation) and in the lower part the individual will have the ability to star rank evaluate where one star means

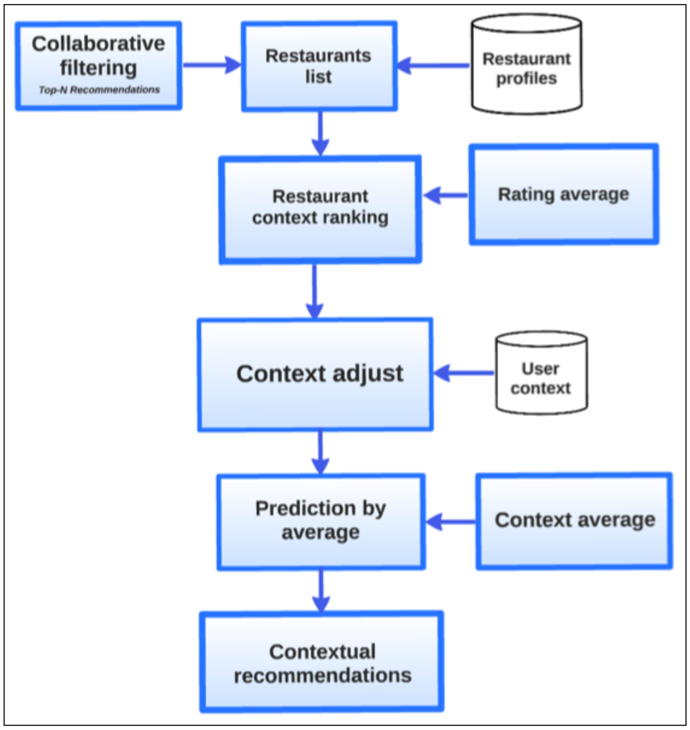


Figure 5.1: The Post-Filtering architecture for Tijuana restaurants.

he/she did not like it much and five stars meaning he/she like it too much. Now in the lower section of the star rank, a button will provide the ability to add that individual to a collection previously created. In the upper-rank star section, we will find a link which reads Click here to see my DNA History this will take us to detail about the individual to be evaluated that way we could review the detail of the individual in image x.

Detailed Important information about individual DNA history is shown here such as how many evaluations in likes he/she has received, how many visitors as well as ascendancy, genetic crossing operator, chromosome numeric representation

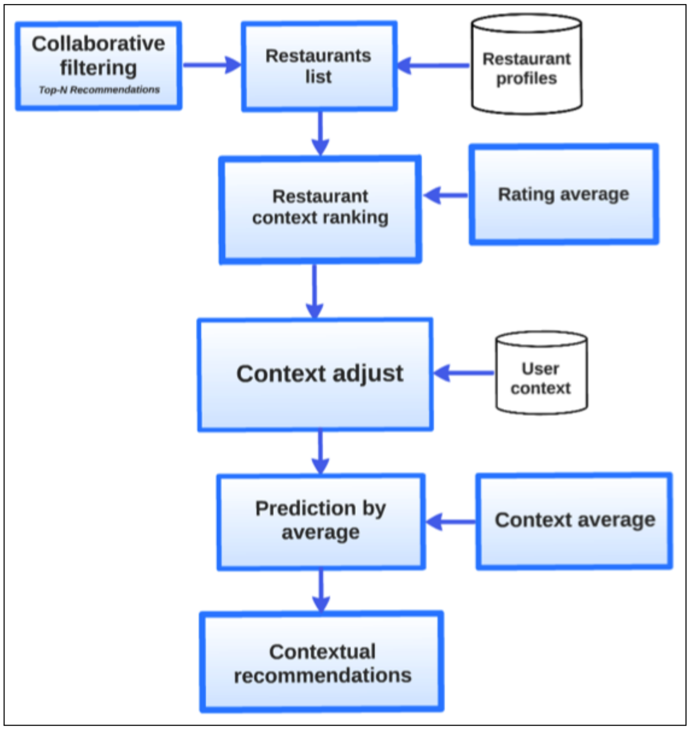


Figure 5.2: The Post-Filtering architecture for Tijuana restaurants.

and inside population identifier.

5.2 EvoDrawing02

Inside this version we will find an initial configuration of the interactive evolutionary algorithm in the following way:

As in the Las version it counts with 80 individuals as initial population and individuals are represented in the same way as in the previous version: Eight evaluations as one evolution parameter. One genetic Operator as aleatory selection between competition and ... One horizontal genetic crossing Operator. Its fitness function is provided by equation x.

1. **Having an initial 80 individual population which we will represent in the following way:**

2. **As in the Las version it counts with 80 individuals as initial population and individuals are represented in the same way as in the previous version.**

3. **Eight evaluations as one evolution parameter..**

4. **Combining the fuzzified inputs according to the fuzzy rules to establish a rule strength.**

5. **One genetic Operator as aleatory selection between competition and ...**

6. **Its fitness function is provided by equation x.**

5.2.1 Fuzzy Inferance.

Where represents all the individuals. Is the rank given to the individuals by the users. Represents one diffuse rank function composed by a diffuse inference system and will be represented by equation x.

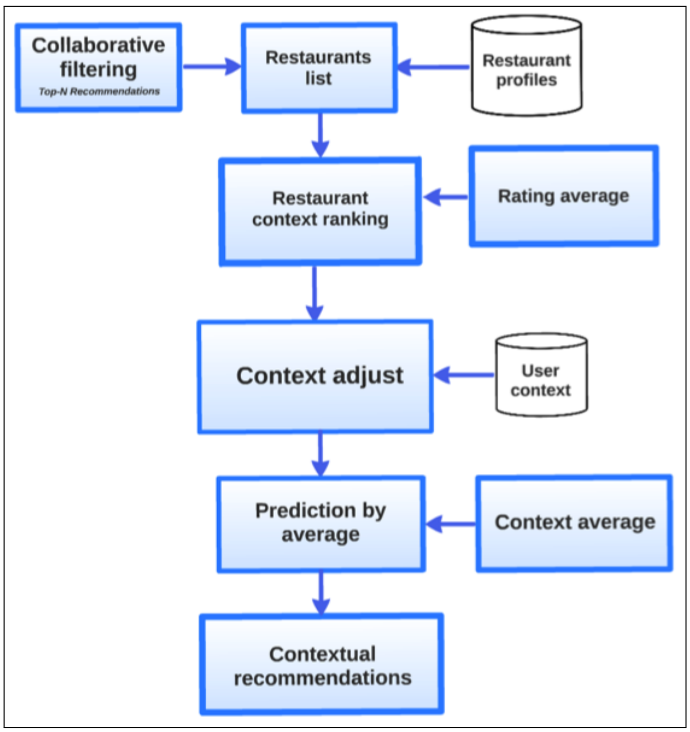


Figure 5.3: The Post-Filtering architecture for Tijuana restaurants.

Where represents the diffuse rank. Will continue been users given rank according to their preferences to each. Is the experience which is defined by the activity already explained in the previous chapter.

The diffuse inference system type is Mamdani, and it is composed by two entrances represented by and a fuzzy rate exit as we can observe represented in figure x.

It counts with 9 if/then rules which were designed in an empiric way and are the following:

If experience is low and preference is low then fuzzy_rate is bad If experience is mid and preference is low then fuzzy_rate is bad If experience is high and preference is low then fuzzy_rate is normal If experience is low and preference is mid then fuzzy_rate is bad If experience is mid and preference is mid then fuzzy_rate is normal If experience is high and preference is mid then fuzzy_rate is good If experience is low and preference is high then fuzzy_rate is normal If experience is mid and preference is high then fuzzy_rate is good If experience is high and preference is high then fuzzy_rate is good

5.2.2 Interface

The user interface in this version is the same as in the previous version. It only differs in how inference is made between versions.

In order to validate the proposed approach, data about restaurant preferences of users in different contexts was collected. The study subjects were students enrolled in a computer engineer major, a masters program and professors of the Tijuana Institute of Technology. A total of **50 users** answered a questionnaire; the questions were about their preferences for nearby restaurants and the technology used by them. The *questionnaire* consisted of **8 questions** and they rate restaurants from a list of 40 restaurants. Each restaurant chosen was rated 6 times one per context considered, a matrix rating with *1,422 ratings* were collected. The questions are shown in the table **??**.

The user's answers from question 1 to question 6 are represented in the figure 5.4. **Figure 5.4a** represents the percentage of surveyed students and teachers; **figure 5.4b** the percentage of the element that users consider the most important to visit a restaurant; **figure 5.4c** represents the preferences of devices when are using Internet for restaurant recommendations; **figure 5.4d** represents the percentage of operating system that often used, **figure 5.4e** shows the percentage of users that use the Internet to search restaurants in Tijuana; and **figure 5.4f**, shows the percentage of users that would like using a restaurant recommender system of Tijuana. For questions 7 and 8 only the top-ten restaurants are shown, without/with the contextual situation. In figure 5.5a, the favorite restaurant is **Daruma**(178 votes), whereas in figure 5.5b, **Daruma** does not appear in the top-ten. When considering

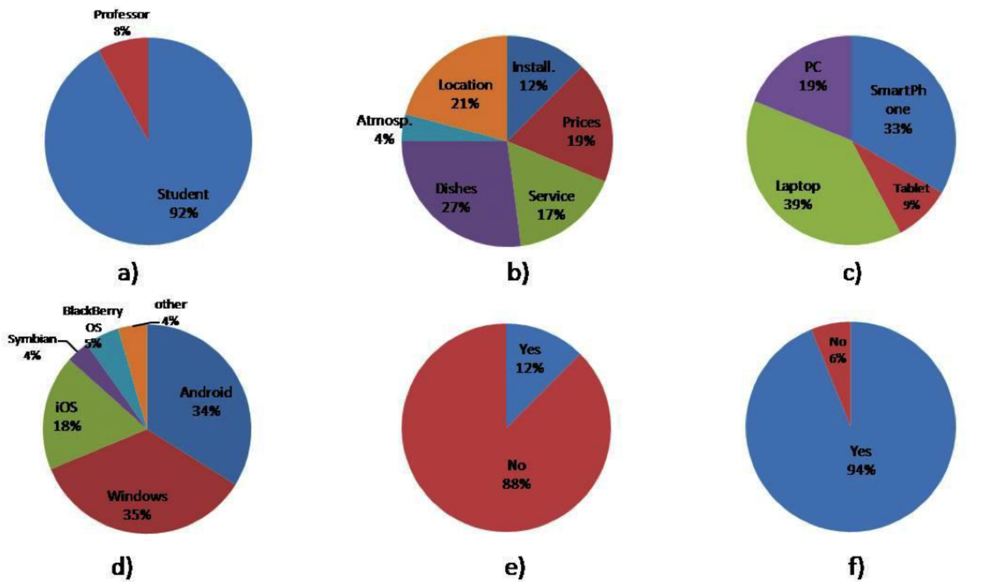


Figure 5.4: The chart shows the users preferences for questions from 1 to 6.

Table 5.2: Contextual factors considered in the questionnaire.

Contextual Factor	Context
Day	1.Midweek(Monday, Thuesday,Wednesday and Thursday) 2.Weekend(Friday,Saturday and Sunday)
Place	1.School 2. Home 3.Work

the context *midweek*, the favorite restaurant was **Carls Jr.**, which appears in both graphs; this restaurant was also the most voted in the different contexts. Contextual recommendations of post-filtering approach depends of context *midweek* or *weekend*, which is the day when the restaurants were rated. Subsequently, the result of the query is refined according to the user context; the 6 contexts mentioned correspond to combinations of contextual factors shown in table 5.2. The dataset was explicitly

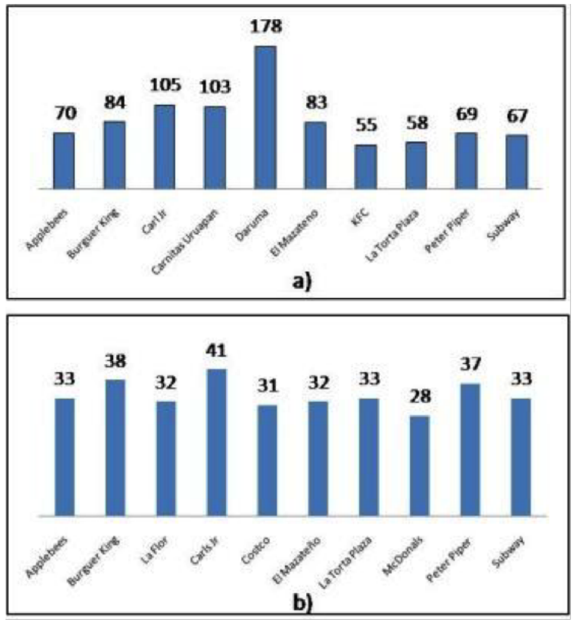


Figure 5.5: The chart shows the users preferences for questions 7 and 8.

collected from **50 users** whom answered questionnaire (see table **??**). A total of 172 predictions was made for different users and the error **MAE=0.5859** when the context **midweek** for current user was considered. The observation for this result is that using a small dataset the performance of the method proposed is limited. By other hand, having only one contextual factor does not improve the accuracy of the recommendations in this domain.

5.3 MovieLens dataset

GroupLens Research has collected and made available rating data sets from the MovieLens web site (http://movielens.org).

The data sets were collected over various periods of time, depending on the size of the set.

- **MovieLens 1M Dataset**: Stable benchmark dataset, 1 million ratings from 6000 users on 4000 movies. Released 2/2003.

  Downloaded from *http://grouplens.org/datasets/movielens/1m/*.

- **MovieLens 10M Dataset**: Stable benchmark dataset, 10 million ratings and 100,000 tag applications applied to 10,000 movies by 72,000 users. Released 1/2009.

  Downloaded from *http://grouplens.org/datasets/movielens/10m/*.

The recommender system proposed for MovieLens uses post-filtering and time segmentation. Time in recommender systems is used as a contextual factor in the research reviewed [**?**], [**?**], [**?**], and [**?**], results vary according the techniques that were done.

In [**?**] the pre-filtering approach was used, time was divided in time intervals and the size of time intervals is directly proportional to the distance from initiating the historical information to the current user context. In [**?**] a tracking model of

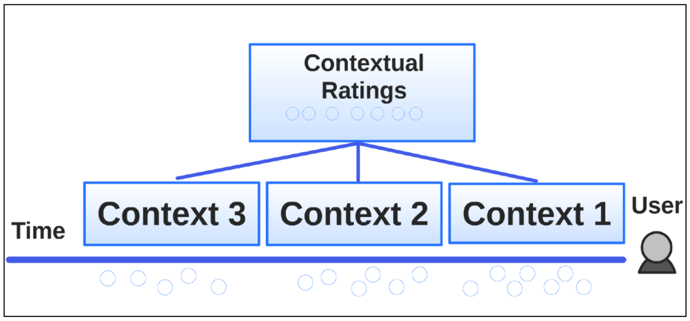


Figure 5.6: Time segmentation of contexts based on current user context.

user behavior over the life-time of data is proposed, in order to exploit the relevant components of all data instances , while discarding only what is modeled as being irrelevant.

In [**?**] it is shown that the time division is beneficial and its performance depends on the items selection method and influence of contextual variables in item ratings. In [**?**] the user profile is segmented into micro-profiles corresponding to a particular context, each context represents a time span in which recommendations for users are derived.

This experiment implements fuzzy logic on time segmentation, in order to improve user satisfaction by providing recommendations based to context and recents user preferences without discarding tastes in the past, as they include important information for the recommender system proposed. The first phase is division of three time segments based on the current context of the user is performed such as in is depicted in figure 5.6.

In recommender system, the first step is get the current user context (user-application interaction), from this information three contexts (figure 5.6) will be obtained that representing a time segment of three months each one, in total the algorithm considers all the ratings users did during nine months prior the current context. Subsequently, ratings are classified by contexts and reused as contextual rating matrix being, a ratings matrix for each context.

The size of matrix depends of users participation during the last nine months. One of the aims is to identify the user behavior through recent information, in order to, for instance, know whether the user changes ratings constantly; whether usually assign high, low or mixed ratings; whether user likes to see different items or whether have a favorite category.

Recommender systems use the collaborative filtering algorithm in order to find relevant items for the user [**?**]. User's profiles are used for determine the similarity between users calculated with Pearson correlation. The similarity between users can provide valuable information as long as user participation is enough (less than 10 ratings). The next step is to obtain recommendations list (Top-N), three contextual lists are the outputs of collaborative filtering algorithm and contain items with user's predictions for each context.

Popularity's prediction considers other variables: 1) users participation in respect of an item in the context and, 2) the rating's average that users have given for item

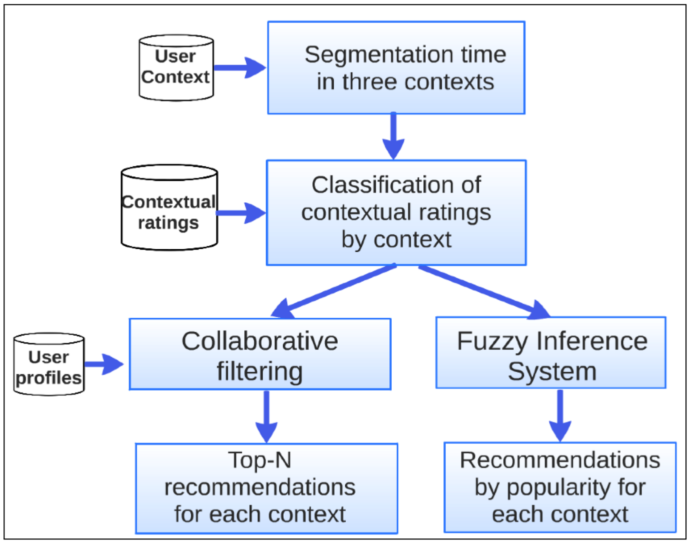


Figure 5.7: Pre-filtering process for context-aware recommender system.

in the same context.

A Fuzzy Inference System (FIS) uses these parameters to assign a weight within a scale from 1 to 5 (prediction value). These recommendations are used when the ratings matrix is sparse, a popularity prediction is done.

Finally, the system gets the recommendations list for each user in different contexts. The recommendation process of pre-filtering is depicted in figure 5.7. The dataset used to test the algorithm was MovieLens(100000 ratings) with 943 users and 1,682 movies. The ratings were collected in a period of 2 years. MovieLens is not a contextual dataset, however, the timestamp was used to determine the rating time, i.e., in this way it was noted the day to know whether the rating time was in weekday or weekend. In this terms, the context was used. Then, the time for each context was

Table 5.3: Results of comparison by contexts in MovieLens dataset.

Context	# Preditions	MAE
1	12235	0.28
2	21049	0.24
3	1075	0.38

divided in 3 months each one, this span covers 9 months before the user's current context.

The neighbors in each context are considered to recommend movies in that context only. An average of predictions are considered for add a movie to the top-N list of contextualized recommendations.The result in table 5.3 shows the error in three contexts. The error increase in context 3, in this context the ratings matrix is a little bit sparse; the error is justifiable because user has less participations.

5.4 Tripadvisor dataset

The dataset used to evaluate the algorithm was TripAdvisor in two versions downloaded [**?**], this datasets was used in [**?**], [**?**] to evaluate the performance of context-aware recommender systems.

The first dataset contains 4669 contextual ratings, 1202 users and 1890 hotels; the second dataset contains 14175 contextual ratings, 2731 users and 2269 hotels. Data were collected of reviews online in tripadvisor.com. There is only one context: type of trip (family, friends, bussines, romantic and relax).

The proposed method consists of three algorithms to recommend: Fuzzy Inference System, collaborative filtering and content-based. Each one uses rating matrix to get recommendations.

The context-aware recommender system uses the post-filtering paradigm[**?**] for adjust recommendations in context. The recommendation by popularity is through the Fuzzy Inference System depicted in figure 5.9, the Fuzzy Inference System contains the variables that are involved in the process to recommend in a human interaction, this process is the same that the recommender system does.

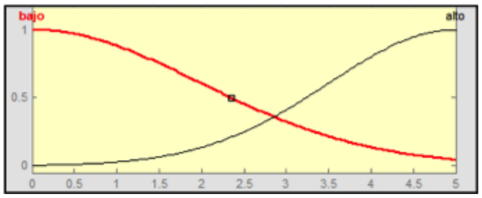
The output represents how matter each item into the users community, i.e. if it was a popular item for users.

The FIS has Gaussians membership functions and are depicted in figure 5.8. The Fuzzy Inference System uses fuzzy rules to infer the inputs and output (a numeric value) that represents the weight of the recommendation. The rules are following:

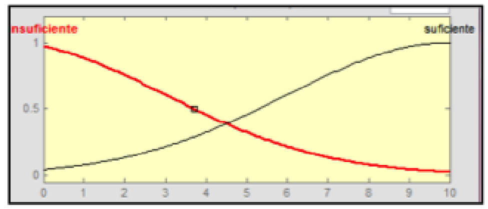
1. *If* ***RatingAverage*** *is low and* ***UserParticipation*** *is insufficient then* ***recommendation*** *is low.*

2. *If* ***RatingAverage*** *is low and* ***UserParticipation*** *is sufficient then* ***recommendation*** *is high.*

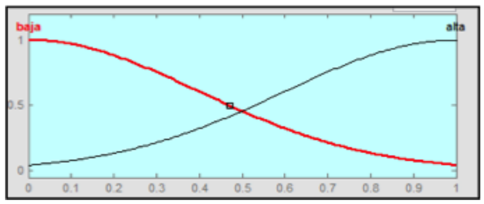
3. *If* ***RatingAverage*** *is high and* ***UserParticipation*** *is insufficient then* ***rec-***



(a)



(b)



(c)

Figure 5.8: Gaussian Membership functions in the input are: a) RatingAverage, b) UserParticipation, and an output: c) Recommendation.

ommendation *is low.*

4. *If* ***RatingAverage*** *is high and* ***UserParticipation*** *is sufficient then* ***recommendation*** *is high.*

Content-based uses cosine similarity to compare the binary vectors representing the profile of each item, thereby obtaining a numerical value that determines similarity,

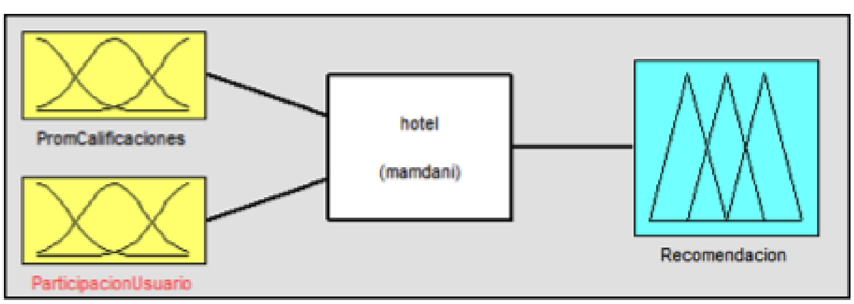


Figure 5.9: Fuzzy Inference System.

based on a threshold.

In other words, it makes a comparison of profiles of each item to determine the most similar to items the user has rated with highest score, context-aware recommender system proposed has a scale from 1 to 5. In the next step the outputs of every recommender algorithm is represented by a list of recommended items. Subsequently applies the context filter and context-aware recommender system gets the final contextual recommendations.

Table 5.4: Example of contextual ratings in the user profile.

User profile		
Item1	Rating1	Context1
Item2	Rating2	Context2
Item3	Rating3	Context3

Context-aware recommender system identifies contextual data of the user profile (see table 5.4), and compares recommended items to filter those items that are adjusted to the user context.

The context filtering is the last step before to get the recommended items. The schema of architecture for context-aware recommender system is depicted in figure 5.10. Two experiments were performed using TripAdvisor dataset, table 5.5 describes the data sets and the scarcity percentage of the specified data. Scarcity of 99% mean that there are problems to recommend items because the information is not enought to get good recommendations.

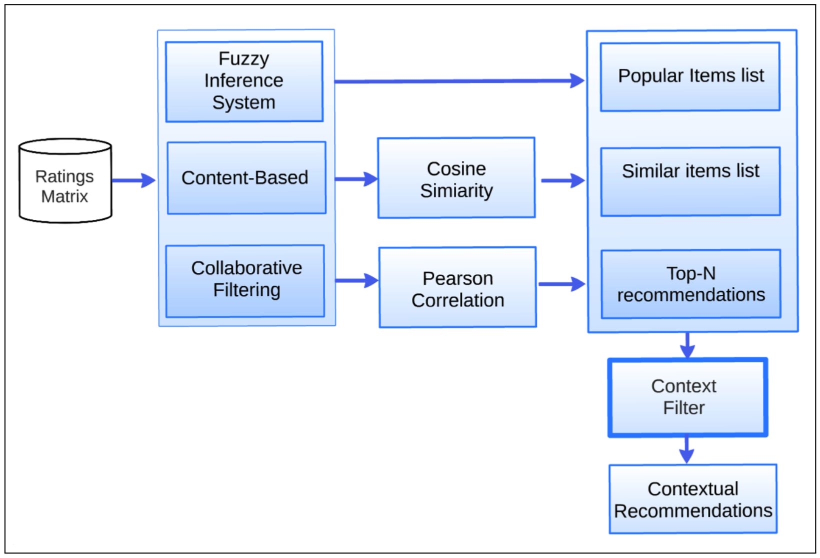


Figure 5.10: Recommender system architecture

Table 5.5: Datasets description.

Dataset	Users	Items	Ratings	Scarcity (percent)
TripAdvisor v1	1202	1890	4669	99.79
TripAdvisor v2	2731	2269	14175	99.77

By other side, in table 5.6 the comparison shows that the algorithm has a acceptable performance, i.e., the error falls into the range of results obtained with others algorithms. Then, contextual recommendations were evaluated with the Root Mean Square Error in order to compare the results with context relaxation algorithm[**?**] that is evaluated with the same dataset.

The fundament of content-based is the cosine similarity; this means that if similarity value among items is high, the recommendations will improve the degree of user satisfaction. This is observed when calculating the similarity average in each

Table 5.6: Comparison of RMSE.

Dataset	Algorithm	RMSE
TripAdvisor v2	FC + Post-filtering	0.504
	FC	0.994
	Pre-filtering + Relaxation	0.985

Table 5.7: Level of similarity among items in datasets.

Dataset	Similarity	Avg.votes per user.
TripAdvisor v1	0.448	5
TripAdvisor v2	0.508	8

dataset as shown in table 5.7.

FIS can provides a list of popular items for each dataset, recommendations through averages are obtained, and recommendations are conditioned to show it when the collaborative filtering and content- based are not delivering recommendations because of data scarcity. However, the majority of popular items of dataset were rated in contexts: romantic, family and bussines, that means that the dataset has biases.

In this experiment the context-aware recommender system proposed involves the paradigm of post-filtering for contextual recommendations. The structure of the datasets facilitated the evaluation of recommendations although the rating matrix has been scarce in both cases. Anyway, information of items and users was used to test the system and a good performance of the system was done.

With respect the performance, post-filtering allows select relevant items that are adjusted into the context, indeed, post-filtering and implementation of different recommendation techniques the system has suitable performance and the datasets help the processes performed.

5.5 Datasets in matrix factorization

Filmtrust dataset

FilmTrust is a small dataset crawled from the entire FilmTrust website in June, 2011. Filmtrust contains a ratings matrix of 35498 ratings, 1504 users and 2071 movies. The dataset has a density of 1.14% and was used in [**?**] using the trust level such as context. The web page is *http://www.librec.net/datasets.html*.

InCarMusic dataset

InCarMusic dataset[**?**] has 8 contextual factors and the possible values for contextual conditions are explained in table 5.8. Music tracks were ten different genres. There is not unified music genre taxonomy, for this reason the recommender system uses the genres defined in [**?**]: classical, country, disco, hip hop, jazz, rock, blues, reggae, pop and metal, 50 music tracks and 42 users in dataset.

Table 5.8: Contexts in InCarMusic dataset.

Context	Values
Driving style	elaxed, driving, sport driving.
Road type	city, highway, serpentine.
Landscape	coast line, country side, mountains/hills, urban.
Sleepiness	awake, sleepy.
Traffic conditions	free road, many cars, traffic jam.
Mood	active, happy, lazy, sad.
Weather	cloudy, snowing, sunny, rainy.
Natural phenomena	day time, morning, night, afternoon.

5.5.1 Results

For experiments with matrix factorization technique the Graphlab toolbox was used. Both mentioned datasets and Movielens (1 millon and 10 millions) were used to test the algoritmh. The test involves K factors that are increasing for 50 iterations. previously, was done a test to identify what number of iterations are enough to get a good result with no overload of process in the algorithm. Results are depicted in the chart 5.11 where the axis (x, y) represent the K value and the error value, respectively. The observations deal to small differences among the datasets, in a range of 0.80-0.90, and the high variability is in MovieLens dataset 10 millions. The big dataset implies more unstable behaviour, while in a small dataset (Filmtrust) the error is less variable. A comparison among MovieLens 1 million and 10 millions shows that there's not a significant difference.

By other side, other datasets were used to test matrix factorization under the

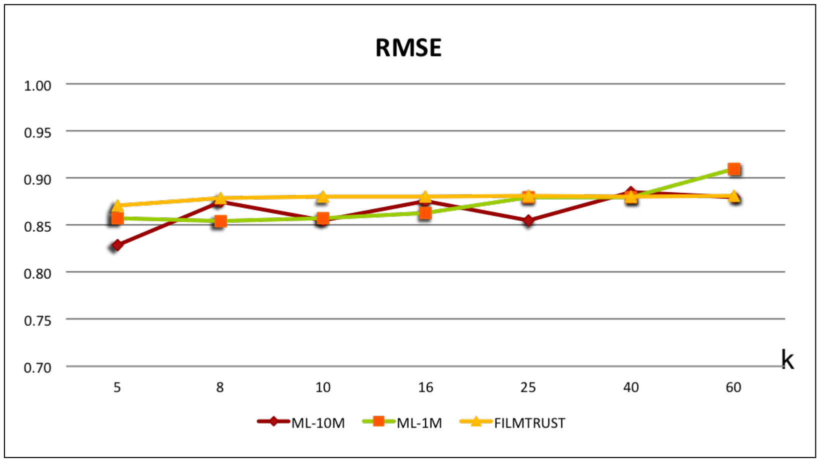


Figure 5.11: RMSE results of matrix factorization test.

Table 5.9: RMSE of datasets using matrix factorization.

Dataset	Ratings	Cosine Sim.	RMSE
Tijuana Rest.	896	0.67	0.60
Mexico Rest.	1161	0.25	0.54
InCarMusic	4012	0.45	0.93
TripAdvisor	4669	0.17	0.85
MovieLens	10000	0.46	0.51
Movielens	100000	0.94	0.42

same parameters to calculate the RMSE for each one. Table 5.9 presents the total of ratings of each dataset, the cosine similarity, it means how similar are the items into the dataset, and the RMSE error obtained in the test with matrix factorization technique. The datasets contain less ratings than the presented in the chart 5.9, according the table 5.9 is not possible to assum that matrix factorization has a better performance with small datasets, because TripAdvisor and InCarMusic datasets obtain an error in the same range that the large datasets of the previuos chart.

Chapter 6

System evaluation

6.1 Metrics

To evaluate context-aware recommender system was used the **task success** and **time-on-task** metrics.

The **task success metric** is perhaps the most widely used performance metric. It measures how effectively users are able to complete a given set of tasks. The **time-on-task metric** is a common performance metric that measures how much time is required to complete a task[**?**].

Task success is something that almost anyone can do. If the users cant complete their tasks, then something is wrong. When the users fail to complete a simple task can be an evidence that something needs to be fixed in the recommender system.

The usability test consist of a list of simple tasks for users that they shall perform in the system to complete the test. Before to start, a minimal description about the system for user was explained. The tasks list are the following:

1. *Rated a restaurant without context.*
2. *Add context to the user profile.*
3. *Filter restaurants by favorite context.*
4. *Find information of a specific restaurant.*
5. *Find all the reviews of a specific restaurant.*
6. *Find section of my favorite restaurants.*
7. *Add a review of a restaurant.*
8. *Find the most popular restaurants.*
9. *Add a restaurant to your wishlist.*
10. *Get recommendations based on expert opinion.*
11. *Get the recommendations content-based.*
12. *Get the collaborative recommendations.*
13. *Get recommendations of the nearby restaurants.*

6.2 Enviromental set up

Each user did the task list, one by one, with previous instructions. It gives a brief explanation about the general features of system before to start. The time average for each user was around 10 minutes to finished all activities without disruptions. After, the results was depicted in a chart to observe the user behaviour for each task, in the figure 6.1 the axis (x, y) represent the task number and percent of success, respectively. The chart shows that only 3 tasks werent accomplished successfuly, the task 5, 6 and 7.

The issue with task 5 was that users can not found easily the reviews section in the interface, the issue in task 7 is derived of task 5 because the user couldnt find the manner to add a review. The task 6 correspond to the favorite restaurants, but the issue is that it was confused to chose favorite restaurants in place of wishlist section. In general, these results mean a possible redisign in the interfacte to facilitate the performance of these tasks. The time it takes a participant to perform a task says a lot about the usability of the application. In almost every situation, the faster a participant can complete a task, the better the experience. In fact, it would be pretty unusual for a user to complain that a task took less time than expected [**?**]. Then, task-on-time was applied to measure time that an user did the task. A resume of the time tasks for each user it is in table 6.1, *null* values mean that the user didn't

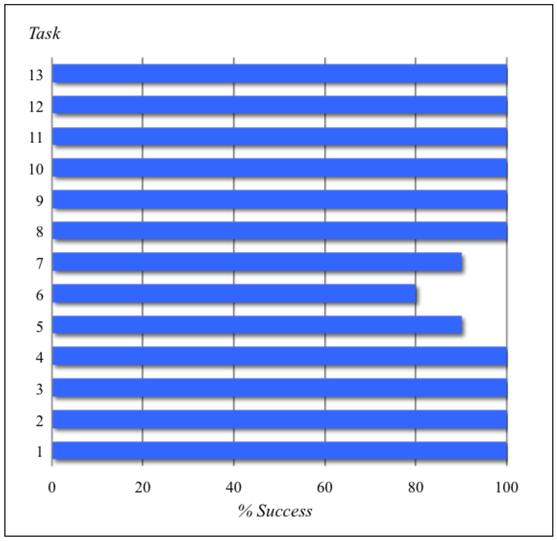


Figure 6.1: Representation of the percent of success for each task.

the task.

6.3 Results

To measure the efficiency of the metric it was chose an confidence interval. In this way, it is observed the time variability within the same task and also helps visualize the difference across tasks to determine whether there is a statistically significant difference between tasks. The obtained information is in table 6.2, the median was used to calculate the confidence interval. In the next step the USE *(Usefulnes, Satisfaction, and Ease of Use)* questionnaire [**?**] was applied in order to get the user's feedback and comments for to know about the difficults in the test. The USE questionnaire consists of 30 rating scales divided into 4 categories:

Table 6.1: Time on task data for 10 users and 13 tasks.

Task	Us1	Us2	Us3	Us4	Us5	Us6	Us7	Us8	Us9	Us10
1	12	28	24	30	19	33	23	16	5	7
2	3	4	17	5	17	134	9	16	12	11
3	123	69	159	53	69	113	44	41	70	98
4	20	4	86	40	13	4	17	3	20	3
5	50	10	63	50	7	11	10	5	20	Null
6	10	30	28	27	5	46	Null	7	Null	34
7	10	20	16	8	15	Null	9	24	16	28
8	18	24	10	10	5	3	27	4	5	6
9	5	6	31	4	45	9	12	5	3	8
10	15	17	15	11	10	19	13	10	20	20
11	30	15	20	16	20	22	15	13	18	20
12	12	14	19	14	40	10	17	17	15	15
13	25	15	15	14	10	10	11	10	10	25

Table 6.2: Confidence interval per task with a confidence level of 95%.

Task	Median	CI 95%	Upper bound	Lower bound
1	20	5.96	25.96	14.04
2	11.5	0.81	12.31	10.69
3	69.5	25.57	95.07	43.93
4	15	16.34	31.34	-1.34
5	15.5	14.84	30.34	0.66
6	27.5	11.57	39.07	15.93
7	16	5.19	21.19	10.81
8	8	5.80	13.80	2.20
9	7	9.43	16.43	-2.43
10	15	2.44	17.44	12.56
11	19	3.00	22.00	16.00
12	14.5	5.51	20.01	8.99
13	12.5	3.89	16.39	8.61

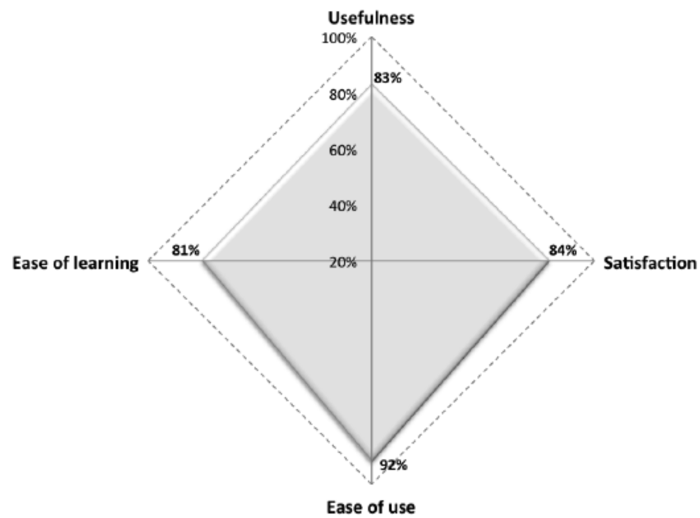


Figure 6.2: The radar chart that depicts the four axis evaluated in the questionnaire.

Usefulness, Satisfaction, Ease of Use, and Ease of Learning. Each is a positive statement to which the user rates level of agreement on a 7-point Likert scale. The USE questionnaire(see appendix B) allows to get values for Usefulness, Satisfaction, Ease of Use, and Ease of Learning, the visualizing the results is in the Fig.6.2 , where the four axis of the radar chart represent the values of percent which users rated positively this factors with respect to their interaction with the context-aware recommender system. The accurate values are *Usability 83%, Satisfaction 84%, Easy of use 92%, and Easy of Learning 81%.*

Chapter 7

Conclusions and future work

7.1 Conclusions

Using a user model in Web-based interactive evolutionary computation overall with the different approaches such as fuzzy logic and gamification it demonstrated in experiment EvoDrawing03 that the users increase their participation with respect to other versions (EvoDrawing01, EvoDrawin02).

In this sense the results have shown a phenomenon in the users which is competitiveness. This phenomenon occurs naturally because as human beings is our nature to be competitive regardless of the topic or activity that we assign [Reference]. This gave support to users return to evaluate more individuals within the experiment EvoDrawings03 and consequently the participation increase exponentially.

In this research work also found that the way individuals was presented to be evaluated and how to evaluate them helped the user to take their evaluations so easy and quick. This means that users evaluated on average 40 or more individuals in an iteration, reducing the risk of demotivation assessment and therefore lose interest in participation. However, it was concluded that the biggest problem of interactive evolutionary computing systems remains on user fatigue.

The fatigue can be generated by many factors, such as how to evaluate individuals, the subject of the application, how to present the individuals, the objective within the application, expertise by users, and more. The resulting method of this research helps motivate users on the issue of participation in interactive evolutionary computing applications.

7.2 Future work

Publications

1. *Restaurant Recommendations based on a Domain Model and Fuzzy Rules. Xochilt Ramírez-García, Mario García-Valdéz. International Seminar on Computational Intelligence. Tijuana Institute of Technology. Tijuana Mexico. (2012).*

2. *Post-filtering for a Context-Aware Recommender System. Xochilt Ramírez-García, Mario García-Valdéz. Recent Advances on Hybrid Approaches for Designing Intelligent Systems . Springer International Publishing Switzerland. (2013).*

3. *Recomendaciones contextuales basadas en el enfoque de post-filtrado. Xochilt Ramírez-García, Mario García-Valdéz. Modelado computacional de Habilidades Linguisticas y Visuales. Vol.74. Research in Computer Sciences, IPN. 2014.*

4. *Context-aware Recommender System Based in Pre-filtering Approach and Fuzzy*

Rules. Xochilt Ramírez-García, Mario García-Valdéz. Recent Advances on Hybrid Approaches for Designing Intelligent Systems . Springer International Publishing Switzerland. (2014).

5. *Context-Aware Recommender System Using Collaborative Filtering, Content-Based Algorithm and Fuzzy Rules. Xochilt Ramírez-García, Mario García-Valdéz, 2016.*

6. *A Hybrid Context-aware Recommender System for Restaurants. Xochilt Ramírez-García, Mario García-Valdéz, 2016.*

Appendix A

Pseudocode

Algorithm 1 Get Cosine similarity values

Require: The list of itemProfilesUser and itemProfilesAll in binary format.
Ensure: The list of cosine similairty value for each item of the itemProfilesUser with each element of itemProfilesAll.

$allProfiles \leftarrow [\ ]$
for $itemu$ to size of $itemProfilesUser$ **do**
  for $itema$ to size of $itemProfilesAll$ **do**
    if $itemu = itema$ **then**
      jump next item
    else
      $cosineSimilarityValue \leftarrow$ among $itemu$ and $itema$
      $itemProfiles \leftarrow itemu, itema, cosineSimilarityValue$
    end if
  end for
end for
return $allProfiles$

Algorithm 2 Collaborative filtering algorithm

Require: The userId.
Ensure: The Top-N list of recommendations for the current user.
  $ratingMatrix \leftarrow allRatings$
  Call $Recommendations \leftarrow getRecommendations()$ module
  return $Recommendations$

Algorithm 3 Content-Based Algorithm

Require: The user id.
Ensure: The Top-N list of recommendations.
  $RV \leftarrow$ All items that user rated with 5
  for $item$ to size of RV **do**
    if $item$ is not in RV **then**
      $UV \leftarrow itemid$
    end if
  end for
  $allItems \leftarrow [\,]$
  $getItemsProfilesUser \leftarrow$ Binary vectors of RV
  $allRatings \leftarrow$ Rating matrix
  for $item$ to size of $allRatings$ **do**
    if $itemid$ is not in $allItems$ **then**
      $allItems \leftarrow item$
    end if
  end for
  $getAllItemsProfiles \leftarrow$ Binary vectors of allItems
  $getCosineSim \leftarrow$ getItemsProfilesUser,getAllItemsProfiles
  for $item$ to size of $highCosineSim$ **do**
    if $itemsimilarity \geq 0.8$ **then**
      $highCosineSim \leftarrow item$
    end if
  end for
  Sort $highCosineSim$ list
  return $itemProfiles$

Algorithm 4 Get item profiles

Require: The UV vector, allItems vector and boolean value of userProfile.
Ensure: The list of temProfiles in binary vectors.

```
if userProfile true then
    getItemsProfilesUser ← UV
    for itemp to size of UV do
        get binary vector of itemp
        itemProfiles ← itemp
    end for
else
    allItemProfiles ← allItems
    for itemp to size of allItems do
        get binary vector of itemp
        itemProfiles ← itemp
    end for
end if
return itemProfiles
```

Algorithm 5 Calculate Cosine similarity

Require: The itemProfileUser and itemProfileAll, both vectors in binary format.
Ensure: The cosine similarity value.

```
sum ← 0
normaItemUser ← 0
normaItemAll ← 0
for position to size of itemProfileUser do
    sumProduct ← sumProduct + (itemProfileUser[position] * itemProfileAll[position])
end for
for item to size of itemProfileUser do
    normaItemUser ← normaItemUser + itemProfileUser[item]^2
end for
for item to size of itemProfileAll do
    normaItemAll ← normaItemAll + itemProfileAll[item]^2
end for
squareRootUser ← squareroot(normaItemUser)
squareRootAll ← squareroot(normaItemAll)
cosineSimilarity ← sumProduct/(squareRootUser * squareRootAll)
return cosineSimilarity
```

Algorithm 6 Create a binary vector of item profile

Require: The tem profile content in r.
Ensure: The temProfile of r in a binary vector.
  $price \leftarrow [4]$
  $payment \leftarrow [2]$
  $alcohol \leftarrow [2]$
  $smokingarea \leftarrow [2]$
  $dresscode \leftarrow [3]$
  $parking \leftarrow [3]$
  $installation \leftarrow [4]$
  $atmosphere \leftarrow [5]$
  $cuisine \leftarrow [30]$
  $price[positionPriceId - 1] \leftarrow 1$
  $payment[positionPriceId - 1] \leftarrow 1$
  $alcohol[positionPriceId - 1] \leftarrow 1$
  $smokingarea[positionPriceId - 1] \leftarrow 1$
  $dresscode[positionPriceId - 1] \leftarrow 1$
  $parking[positionPriceId - 1] \leftarrow 1$
  $installation[positionPriceId - 1] \leftarrow 1$
  $atmosphere[positionPriceId - 1] \leftarrow 1$
  $cuisine[positionPriceId - 1] \leftarrow 1$
  $itemProfile \leftarrow price+payment+alcohol+smookingarea+dresscode+parking+ installation + atmosphere + cuisine$
  return $itemProfile$

Algorithm 7 Get recommendations

Require: The currentUser and ratingMatrix.
Ensure: The Top-N list of recommendations for the current user.

```
Dictionaries totals ← {}, sumSimilarity ← {}
predictions ←[ ]
for otherUser to size of ratingMatrix do
   if otherUser = currentUser then
      jump next otherUser
   end if
   similarityValue ← get pearsonSimilarity
   if similarityValue ≤ 0 then
      jump next otherUser
   end if
   for item to size of profileOther do
      if item is not in profileUser then
         if profileUser[item] = 0 then
            Set in totals ← item
            totals[item] Add ratingMatrix[otherUser][item] ∗ similarityValue
            Set in sunSimilarity ← item
            sumSimilarity Add similarityValue
         end if
      end if
   end for
end for
for each (item, total) in totals do
   predictions ← [(total/sumSimilarity[item], item)]
end for
Ranking of predictions
return predictions
```

Algorithm 8 Get Pearson correlation

Require: The currentUser, otherUser and preferences.
Ensure: The pearsonCorrelation score.
  Dictionaries $itemsRatedMutually \leftarrow \{\}$
  for each $item$ in preferences of $currentUser$ **do**
    if $item$ is in preferences of $currentUser$ **then**
      jump next $itemsRatedMutually[item] \leftarrow 1$
    end if
  end for
  $numberElements \leftarrow$ size of $itemsRatedMutually$
  if $itemsRatedMutually = 0$ **then**
    return 0
  end if
  for $item$ to size of $itemsRatedManually$ to get all preferences **do**
    $sumCurrentUser \leftarrow preferences[currentUser][item]$
    $sumOtherUser \leftarrow preferences[otherUser][item]$
  end for
  for $item$ to size of $itemsRatedManually$ to get squares **do**
    $squareCurrentUser \leftarrow square(preferences[currentUser][item])^2$
    $squareOtherUser \leftarrow square(preferences[otherUser][item])^2$
  end for
  for $item$ to size of $itemsRatedManually$ to get sum of products **do**
    $sumProduct \leftarrow preferences[currentUser][item] * preferences[otherUser][item]$
  end for
  $pearsonNumerator \leftarrow sumProduct - ((sumCurrentUser * sumOtherUser)/numberElements)$
  $pearsonDenominator \leftarrow square(squareCurrentUser - ((sumCurrentUser)^2/numberElements) * squareOtherUser - ((sumOtherUser)^2/numberElements))$
  $pearsonCorrelation \leftarrow pearsonNumerator/pearsonDenominator$
  return $pearsonCorrelation$ among two users

Algorithm 9 Matrix factorization

Require: R is a matrix to be factorized, dimension N * M, P an initial matrix of dimension N * K, Q an initial matrix of dimension M * K, K is the number of latent features, steps for the maximum number of steps to perform the optimization, alpha is the learning rate and beta is the regularization parameter.

Ensure: The factorized matrix P and Q.

$alpha \leftarrow 0.0001, beta \leftarrow 0.001$
$QMatrix \leftarrow QMatrix * T$
for $step$ to $rangeSteps$ **do**
  for i to size of $RMatrix$ **do**
    for j to size of $RMatrix[i]$ **do**
      if $RMatrix[i][j] > 0$ **then**
        $e_{i,j} \leftarrow RMatrix[i][j] - dotProduct(PMatrix[itoend], QMatrix[inittoj])$
      end if
      for k to range of $KFactors$ **do**
        $PMatrix[i][k] \leftarrow PMatrix[i][k] + alpha * (2 * e_{i,j} * QMatrix[k][j] - beta * PMatrix[i][k])$
        $QMatrix[k][j] \leftarrow QMatrix[k][j] + alpha * (2 * e_{i,j} * PMatrix[i][k] - beta * QMatrix[k][j])$
      end for
    end for
  end for
  $eR \leftarrow dotProduct(PMatrix * QMatrix)$
  for i to range of $RMatrix$ **do**
    for j to size of $RMatrix[i]$ **do**
      if $RMatrix[i][j] > 0$ **then**
        $e \leftarrow e + (beta/2) * PMatrix[i][k]^2 + QMatrix[i][j]^2$
      end if
    end for
  end for
  if $e < 0$ **then**
    break
  end if
end for
return $PMatrix, QMatrix * T$

Appendix B

USE Questionnaire

Usefulness

- It helps me be more effective.
- It helps me be more productive.
- It is useful.
- It gives me more control over the activities in my life.
- It makes the things I want to accomplish easier to get done.
- It saves me time when I use it.
- It meets my needs.
- It does everything I would expect it to do.

Ease of Use

- It is easy to use.
- It is simple to use.
- It is user friendly.
- It requires the fewest steps possible to accomplish what I want to do with it.
- It is flexible.
- Using it is effortless.
- I can use it without written instructions.
- I don't notice any inconsistencies as I use it.
- Both occasional and regular users would like it.
- I can recover from mistakes quickly and easily.
- I can use it successfully every time.

Ease of Learning

- I learned to use it quickly.
- I easily remember how to use it. It is easy to learn to use it.
- I quickly became skillful with it.

Satisfaction

- I am satisfied with it.
- I would recommend it to a friend.
- It is fun to use.
- It works the way I want it to work.
- It is wonderful.
- I feel I need to have it.
- It is pleasant to use.

Source: From the work of Lund (2001). Note: Users rate agreement with these statements on a 7-point Likert scale, ranging from strongly disagree to strongly agree. Statements in italics were found to weight less heavily than the others.