Publicaciones en Ciencias y Tecnología

Publicaciones en Ciencias y Tecnología. Vol.12, N°2, Julio-Diciembre (2018) 69–81

Artículo de investigación

Fuzzy queries aid in medical diagnosis

Consultas difusas en asistencia al diagnóstico médico

Josué Ramírez[a], Leonid Tineo[a]

[a] *Universidad Simón Bolivar. Caracas, Venezuela*

Recibido: 15-03-2018 Aceptado: 12-10-2018

Abstract

This paper proposes the utilization of a fuzzy database engine for supporting medical diagnoses. Expert know how is stored in a relational database and then it is modeled diagnoses rules with fuzzy queries that pulls out the most accurate information related to the sickness and therefore supporting doctors with the medical diagnostic. A solution prototype has been developed with information related to respiratory disease characterization and it is built with fuzzy queries using SQLf. This case study can be used to define a roadmap for future developments in medical diagnosis supported on fuzzy databases. As always, the diagnosis can only be given by a specialist, these systems only provide help in their work task.

Key words: SQLf, fuzzy querying, automated support system for medical diagnosis.
UNESCO Code: 120304- Artificial intelligence.

Resumen

Este artículo propone el uso de un motor de base de datos difuso para ayudar en el diagnóstico médico. El conocimiento experto se almacena en una base de datos relacional y luego se modela mediante reglas de diagnóstico con consultas difusas que extraen la información más precisa relacionada con la enfermedad y, por lo tanto, apoyan a los médicos con su diagnóstico. Hemos construido un prototipo de sistema con una base de datos que almacena la caracterización de enfermedades respiratorias. Esta aplicación se ha creado utilizando un sistema de gestión de bases de datos que admite el lenguaje de consulta difusa SQLf. Este trabajo encamina desarrollos futuros en el diagnóstico médico soportado sobre bases de datos difusas. Como siempre, el diagnóstico solo puede ser dado por un especialista, estos sistemas solo brindan ayuda en su labor médica.

Palabras clave: SQLf, consultas difusas, sistema automatizado de apoyo al diagnóstico médico.
Código UNESCO: 120304- Inteligencia Artificial.

1. Introduction

At present time, there are several examples of automated medical diagnosis systems [1, 2, 3, 4, 5, 6]. Due to the vagueness nature of this field, in earlier works, fuzzy logic has been proposed to be applied for medical diagnosis [7, 8]. Expert systems have been used to gather the knowledge and experience of physicians in the form of rules that facilitate diagnosis for diseases such as Parkinson [1], diabetes [4], lung disease [6], preeclampsia [3] and respiratory [5]. There is an observable rise in interest for the topic of uncertainty in data and database management systems [9, 10, 11]. Nevertheless, as far as we know, none of the existing automated medical diagnosis systems takes advantage from these

Impresa: ISSN 1856-8890. Dep. Legal pp200702LA2730. Digital: ISSN 2477-9660. Dep. Legal ppi201402LA4590.
Email addresses: `ramirezjosue@usb.ve` (Josué Ramírez), `leonid@usb.ve` (Leonid Tineo)

database management systems. Several works deal with fuzziness in databases. Most of them concern with the design and implementation of fuzzy querying languages [9]. However, as the research area is relatively new, there are few applications being developed based on such languages [9]. In [12] stand by that fuzzy querying might be used in lots of applications for decision support and massive information systems.

The initial goal of the work reported here was to show a practical and real example of the application of fuzzy querying in provide support for medical diagnosis. We have chosen the case of respiratory diseases due to their high incidence and relatively simple medical theoretical foundations. Based in this study case, we propose a general model for medical diagnosis using fuzzy queries. To the best of our knowledge, a similar solution has not been previously proposed. A medical doctor has participated in the interdisciplinary team performing this work.

One of the widest medical specialties is that of respiratory diseases. Given the large volume of information they hold, it is difficult to identify, at first glance, which disease affects certain patients, mainly due to the multiple factors playing a significant role in the diagnosis. In addition, the presence of these factors implies the use of linguistic terms susceptible of conveying a fuzzy interpretation, since each physician or individual may give his/her own interpretation to a specific linguistic term. For instance, if a physician wants to deliver a diagnosis just by the patient's symptoms, the physician will try to detect them by inquiring from the patient. However, the physician does not only want to know the symptoms but also their intensity and duration, since these will tell about the possible condition's progress. Although there are some standards as to this term valuation, there are also many possible linguistic values. Intensity, for instance, may be low, moderate, or severe, while duration may be acute, semi acute or chronic.

We might say that delivering a medical diagnosis is an art [13] whereby several factors and variables take part and it is the result of experts' judgment. Nevertheless, diagnoses imply a very precise knowledge basis and a group of criteria or basic rules, which may be logically represented. So, it is possible to automate and emulate a diagnosis based on fuzzy logic. The possibility of obtaining a diagnostic approach would result in a more productive examination because the patient may provide useful information to deliver a medical diagnosis.

The objective of this work is to provide a support tool for medical diagnosis based on the use of fuzzy queries. For this, the simple diagnosis was first modeled, involving only the symptoms reported by the patients in consultation. Then the most specialized diagnostic was modeled, including the signs observed by the doctor. With these models, a prototype of a system was built in which the specialist can adjust preferences according to his knowledge and experience; with which you can obtain possible diagnoses automatically.

The rest of this paper is organized as follows: Section 2 shows a brief comparison of our proposal with some recent Related Works in the field of automated medical diagnosis. Section 3 points out some aspects of a Fuzzy Querying Language. Section 4 is devoted to use fuzzy queries for modeling simple diagnosis in a general way. Section 5 provides a fuzzy querying based solution for specialized diagnosis. Section 6 describes main system features of our automated medical diagnosis prototype, including user interfaces. Finally, in section 7, we address some Conclusions and Future Work.

2. Related works

Several recent works [1, 2, 3, 4, 5, 6] concern with development of automated medical diagnosis system. The article [1] through practical examples illustrates the indispensability of the cognition of logic and the ways to apply it in some areas of medical diagnosis. They model the diagnosis using rules of logic for a specific diagnostic context. According to [2], the field of medicine has greatly benefited from expert systems based on fuzzy rules since these systems provide practical and novel solutions in the presence of uncertainty, specifically in the detection of medical complications and diseases. The problem with these previous two works is that, despite trying to be general, the rules for each specific diagnostic context could be many, generating an overload when using the inference engine.

In [3], the experience of implementing an expert system based on fuzzy logic to detect the risk level of preeclampsia is reported, which allows early diagnosis, and makes possible the monitoring of the pregnant woman. They used MySQL for the database and JAVA to program the expert system. The article [4] proposes a Fuzzy Expert System for diagnosis of diabetes using the jFuzzyLogic library, which offers the implementation of fuzzy inference and the Java API for XML Web Services (JAX-WS). These last two works have the virtue of using fuzzy logic, which is more similar to human reasoning and can yield better results in the midst of uncertainty, however, the problem is that it does not present a general model, but only specific rules for a disease.

In [5], an internet-based diagnostic expert system for common respiratory diseases is presented. The system contains a large amount of data and a high efficiency function for its analysis and for the diagnosis of various diseases. They used the "pattern of forward reasoning" technique and the ASP.NET technologies with C #, Microsoft SQL Server 2005.

This system has the advantage of being Internet-based, which makes it highly available; also, the support in database is an advantage. However, it does not exploit the potentialities of the DBMS, but implements the rules through external application. In addition, these rules are not general, but they belong to a specific specialty.
In [6] presents the implementation of an expert system for the diagnosis of pulmonary diseases. The knowledge base and the inference rules were programmed in SWI-Prolog. This last system uses classical logic with specific rules for the considered diseases, which constitutes a disadvantage, because medical diagnosis involves the manipulation of imprecise terms of natural language, it is not possible in classical logic.
With respect to all these related works, there are several advantages of the proposal that we present in this article. The benefits of the database manager are used. The representation of knowledge through a relational database is simple. The diagnostic rules expressed as fuzzy queries are general, so that they can be applied to the context of any medical specialty. The diagnostic system implementation is much less complex compared to implementing an expert system.

3. Background: fuzzy querying

Traditional database systems lack flexibility. They consider data to be perfectly known and their query languages do not allow natural expression of user preferences. For solving this rigidity problem of database systems, several previous works have been performed. In this tendency, some of most remarkable efforts are the following: RankSQL [14] intended for retrieving the top-k ranked answers; Skyline [15] retrieves non-dominated rows based on a crisp multi criteria comparison; SQLf [11] conceptualized as a complete extension of SQL with fuzzy conditions; FSQL [16] based on fuzzy sets for imperfect data representation. Fuzzy sets based approach has been demonstrated to be the most general for dealing with vagueness and uncertainty in databases [11]. Several research works based on this approach has been made by different authors [9]. Fuzzy Databases designates the research area dealing with the application of fuzzy sets or fuzzy logic to databases.
A fuzzy set F [17] is a special kind of collection of elements from a classical set X (universe) where membership is defined by a gradual function μ_F on X that ranks on the Real interval $[0, 1]$. Here the degree 0 means the total exclusion while 1 means total inclusion. Elements with membership degree in the open interval $]0, 1[$ are said to be in the border of the fuzzy set, they are partially included. As the degree approaches 1, the statement "the element is included in the set" is more certain. Fuzzy sets give a mathematical meaning to vague linguistic terms of natural language. Such meanings are established according to the context and human preferences. For example, Figure 1 shows the graphical representation of membership functions for fuzzy sets defining fuzzy terms describing fever symptom on the patient's temperature as light, moderate or high.

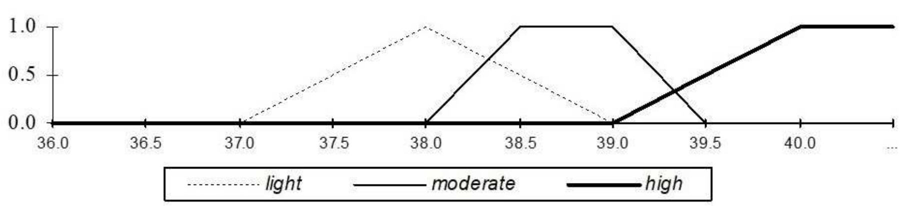

Figure 1: Functions defining fuzzy terms light, moderate and high, to describe fever on a patient according to physician appreciation or preference.

The membership function in Figure 1 for the fuzzy set high corresponds to $\mu_{high}(x)$ given by equation (3.1). Temperature values over $40°C$ are completely included ($\mu_{high}(x) = 1.00$) in the fuzzy set, while those under $39°C$ are completely excluded ($\mu_{high}(x) = 1.00$), and as the temperature approaches $40°C$ increases the degree. In the same way, definitions of fuzzy sets for light and moderate are straight forward form Figure 1.

$$\mu_{high}(x) = \begin{cases} 0.00, & \text{if} \quad x \leqslant 39, \\ x - 39, & \text{if} \quad 39 < x < 40, \\ 1.00, & \text{if} \quad x \geqslant 40. \end{cases} \tag{3.1}$$

In a fuzzy querying system, we may define the fuzzy terms light, moderate and high to describe fever, and use them for querying a table with temperatures of patient. For doing so, we may use SQLf [11], as well as another existing fuzzy query language. SQLf allows using linguistic terms such as predicates, modifiers, comparators and quantifiers; they are interpreted by means of fuzzy sets concepts. SQLf is rather a complex language [11], we do not explain it in detail, but we do illustrate some features that we will use for diagnosis. Let see a simple SQLf example addressing a query to data set in Table 1. We could want to identify persons with certain possible disease, for so doing, patients with a high fever symptom must be obtained from this table. If we use a classic querying system, rigid bound must be established. We must ask for "patient with temperature over $40°C$". The following traditional query in SQL would retrieve only the individual E whose temperature is greater or equal to 40: $SELECT * FROM$ patient $WHERE\ temperature \geqslant 40$.

Table 1: Patient table: example data set of individuals with their temperature (transposed table)

Individual	A	B	C	D	E	F	G
Temperature	37.5	39.5	36	39.2	40	36.5	39.8

SQLf enables user to specify a semantic for linguistic terms, as the term high for fever symptom. The following SQLf sentence defines high as a fuzzy term (predicate) interpreted by means a fuzzy set whose membership function is that of equation (3.1) that we have already seen: $CREATE\ FUZZY\ PREDICATE\ high\ ON\ NUMBER\ AS$ $(39, 40, INFINIT, INFINIT)$. Then, we may formulate the following query in SQLf: $SELECT * FROM$ patient $WHERE$ temperature $IS\ high$. The result of this query would be the fuzzy set of rows meeting, at least gradually, the fuzzy condition "temperature IS high". Each row would be provided with the satisfaction degree to the fuzzy condition that would be its membership degree to result, as Table 2 shows in third (unnamed) attribute. For example, for the individual B, $\mu_{high}(39.5) = 0.5$. As we can see, fuzzy queries give discriminated answers that are ranked by the satisfaction degree to the fuzzy condition. User would like to limit the answer set according to its size or a satisfaction level threshold. This kind of restriction is named calibration. For example: $SELECT * FROM$ patient $WHERE$ temperature is $high\ WITH\ CALIBRATION$ 0.5. This last query applies the fuzzy predicate high to the patient's temperature and filters the result to obtain only those that have a degree of satisfaction greater than or equal to 0.5, obtaining the result shown in Table 3.

Table 2: Fuzzy query result (transposed table).

Individual	B	D	E	G
Temperature	39.5	39.2	36.5	39.8
Membership	0.50	0.20	1.00	0.80

Table 3: Resulting fuzzy table (transposed) for the fuzzy query with calibration 0.5.

Individual	B	E	G
Temperature	39.5	36.5	39.8
Membership	0.50	1.00	0.80

Different fuzzy logic operators may be used for combining conditions. The conjunction (AND) is interpreted in SQLf with the min operator. That is, the satisfaction degree for a conjunctive condition is the minimum of the satisfaction degree for individual conditions in the conjunction. As the dual, disjunction (OR) in SQLf is interpreted by means of the maximum. To illustrate conjunction and disjunction operators in fuzzy queries, suppose we have a relation with incidence of symptoms in some diseases as in Table 4, incidence is measured in a scale from 0 to 100. Let us define the fuzzy predicate slender whose meaning is given by the fuzzy set with membership function in equation (3.2), it would be done with the SQLf statement: $CREATE\ FUZZY\ PREDICATE\ slender\ ON\ 0 \cdot\cdot 100\ AS\ (INFINIT, INFINIT, 20, 40)$. For retrieving diseases with a $slender$ incidence of all the three considered symptoms, we may address the SQLf query with conjunctive condition: $SELECT * FROM$ characterization Of Diseases $WHERE$ dyspnea Incidence IS slender AND cough Incidence $IS\ slender\ AND$ wheezing Incidence $IS\ slender$. Table 5 shows this query result. Remark that Pneumonia disease is in the result set, but Tuberculosis is not. It obeys the fact satisfaction degree for Tuberculosis results in 0, remember that 0 is the measure of complete exclusion in fuzzy sets. Table 5 rightmost column is the satisfaction degree, which is computed as the minimum of satisfaction degrees for individual conditions, given by equation (3.3) for Pneumonia disease case.

Table 4: The case of simplified relation "characterizationOfDiseases"

disease Name	dyspnea Incidence	cough Incidence	wheezing Incidence
Pneumonia	15	35	25
Tuberculosis	75	30	85

$$\mu_{slender}(x) = \begin{cases} 1.00, & \text{if} \quad x \leqslant 20, \\ \dfrac{40-x}{20}, & \text{if} \quad 20 < x < 40, \\ 0.00, & \text{if} \quad x \geqslant 40. \end{cases} \tag{3.2}$$

Table 5: Result of a conjunctive query.

disease Name	dyspnea Incidence	cough Incidence	wheezing Incidence	$\mu(x)$
Pneumonia	15	35	25	0.25

$$\min\{\mu_{slender}(15); \mu_{slender}(35); \mu_{slender}(25)\} = \min\{1.00; 0.25, 0.75\} = 0.25. \tag{3.3}$$

Now we want to find diseases with a slender incidence of some symptoms dyspnea or cough or wheezing. It would be a SQLf query with a disjunctive condition as follows: $SELECT * FROM$ characterizationOfDiseases $WHERE$ dyspneaIncidence *IS slender OR* coughIncidence *IS slender OR* wheezingIncidence *IS slender*. It will produce the result shown in Table 6. The satisfaction degree is computed as the maximum of satisfaction degrees for the individual conditions.

Table 6: Result of a disjunctive query.

disease Name	dyspnea Incidence	cough Incidence	wheezing Incidence	$\mu(x)$
Pneumonia	15	35	25	1.00
Tuberculosis	75	30	85	0.50

SQLf provides an extension for *SQL* set operators *UNION* and *INTERSECT*. These operators combine the result fuzzy set of operand queries. As consequent with *OR* fuzzy logic connector in SQLf, fuzzy set *UNION* gives each row the maximum satisfaction degrees in both queries results. Fuzzy set *INTERSECT*, being the dual of *UNION* gives the minimum.

In traditional logic, existential and universal quantifiers (*SOME* and *ALL*) are the extension of disjunction (OR) and conjunction (*AND*) over the domain universe of a variable. In fuzzy logic, these traditional quantifiers are rigid and therefore allow the definition and use of fuzzy quantifiers such as *fewOf, halfOf and mostOf*. SQLf allows using this kind of quantifiers in the *HAVING* clause of a partitioned query. This is an important feature that is not provided by other known fuzzy querying languages. As it would be necessary for expressing diagnosis queries, we must use SQLf rather than other fuzzy querying languages.

The term *mostOf* is a fuzzy quantifier of proportional nature. That is, it describes the proportion of a whole satisfying a given condition in quantifier scope. Therefore, it is defined by means of a fuzzy set of numbers in the real unit interval. The fuzzy quantifier *mostOf* is of an increasing behavior, the closer proportion is to the unit, the closer its membership degree is to 1. Per context and user preference, this quantifier could be defined in SQLf as follows: *CREATE FUZZY QUANTIFIER mostOf AS* $(0.5, 0.75, 1.0, 1.0)$. This statement defines the linguistic term *mostOf* by means of a fuzzy set in the universe $[0, 1]$, whose membership function is that of equation (3.4).

$$\mu_{mostOf}(x) = \begin{cases} 0.00, & \text{if} \quad x \leqslant 0.50, \\ \dfrac{x-0.5}{0.25}, & \text{if} \quad 0.50 < x < 0.75, \\ 1.00, & \text{if} \quad x \geqslant 0.75. \end{cases} \tag{3.4}$$

Suppose we want to obtain individuals having intensity moderate for *mostOf* presented symptoms with the relation, show in Table 7, where first column is the individual identification; second column is the name of symptom that such individual

Table 7: Example presentedSymptom relation

Individual	Symptom	Intensity
A	Chills	25
A	Dry cough	50
A	Dyspnea	90
A	General Malaise	44
A	Headache	65
A	Loss of appetite	30
A	Sneezing	40
A	Sweating	55
B	Sweating	50
B	Loss of appetite	25
B	Chills	10
B	Dyspnea	15
B	General Malaise	35
B	Headache	70

presents; and third column is a measure in range 0 to 100 of the intensity of observed symptom for the individual. Let us define fuzzy term moderate as follows: *CREATE FUZZY PREDICATE* moderate *ON* $0 \cdot\cdot 100$ *AS* $(20, 40, 60, 80)$, whose membership function is that of Equation (5). Let us formulate the SQLf query: *SELECT* individual *FROM* presentedSymptom *GROUP BY* individual *HAVING mostOf ARE* intensity *IS* moderate. It will produce the result in Table . The column at right side shows the satisfaction degree computed for the group according semantics of SQLf and given user defined terms.

$$\mu_{moderate}(x) = \begin{cases} \dfrac{x-20}{20}, & \text{if} \quad 20 \leqslant x < 40, \\ 1, & \text{if} \quad 40 \leqslant x \leqslant 60, \\ \dfrac{80-x}{20}, & \text{if} \quad 60 < x \leqslant 80, \\ 0, & \text{otherwise.} \end{cases} \tag{3.5}$$

Table 8: Result of a fuzzy quantified partitioned query

Individual	$\mu(x)$
A	0.50
B	0.25

Hereafter, we explain how we obtain this result. Despite there are improved evaluation strategies for fuzzy quantified queries [18], we do the explanation using the Naïve procedure operative semantic, because it will be clearer for the reader.

The evaluation procedure in Table 9 is as follows: First, we form the groups according the GROUP BY clause. In this case, we have two groups given by individual A and B. For each row, we compute the satisfaction degree of the condition under quantifier. This corresponds to the column $\mu_C(r_i)$ whose value is calculated as $\mu_{moderate}(intensity)$ for the row r_i. We sort groups' rows in descending order of these degrees. As we can see, values of column $\mu_C(r_i)$ appear in decreasing order for each group. After the sort, each row in a group is considered as the last of a subgroup that comprises rows from the first one. For example, let consider the group of the individual B. This group have 6 rows that were sorted in decreasing order of satisfaction degree $r_{\leqslant} Sweating, 50, 1 >$, $r_2 \leqslant$ *General Malaise*, 35, 0.75 >, $r_3 \leqslant$ *Headache*, 70, 0.50 >, $r_4 \leqslant$ *Loss of appetite*, 25, 0.25 >, $r_5 \leqslant$ *Chills*, 10, 0 >, $r_6 \leqslant$ *Dyspnea*,15, 0 >. Thus we have six subgroups $SG_1 = \{r_1\}, SG_2 = \{r_1, r_2\}, SG_3 = \{r_1, r_2, r_3\}, SG_4 = \{r_1, r_2, r_3, r_4\}, SG_5 = \{r_1, r_2, r_3, r_4, r_5\}, SG_6 = \{r_1, r_2, r_3, r_4, r_5, r_6\}$ The size of each subgroup represents a proportion regard to the whole group cardinality. This proportion is the fraction shown in column p_i. In the case of the individual B, corresponding p_i for each SG_i is $\frac{i}{6}$ because cardinality of the whole group is 6. The satisfaction degree of this proportion for the fuzzy quantifier is in column $\mu_Q(p_i)$ which, in this case, is computed as $\mu_{mostOf}(p_i)$. For each row, we take the minim between $\mu_C(r_i)$ and $\mu_Q(p_i)$. The maxim of these combinations

Table 9: Naïve evaluation of a fuzzy quantified partitioned query in SQLf.

individual	Symptom	intensity	$\mu_(r_i)$	p_i	$\mu_Q(p_i)$	$\min(\mu_c(r_i),\mu_Q(p_i))$	$\max(\min(\mu_c(r_i),\mu_Q(p_i)))$
A	Dry cough	50	1	1/8	0	0	0.50
	Sneezing	40	1	2/8	0	0	
	Sweating	55	1	3/8	0	0	
	General Malaise	44	1	4/8	0	0	
	Headache	65	0.75	5/8	0.50	0.50	
	Loss of appetite	30	0.50	6/8	1	0.50	
	Chills	25	0.25	7/8	1	0.25	
	Dyspnea	90	0	8/8	1	0	
B	Sweating	50	1	1/6	0	0	0.25
	General Malaise	35	0.75	2/6	0	0	
	Headache	70	0.50	3/6	0	0	
	Loss of appetite	25	0.25	4/6	0.66	0.25	
	Chills	10	0	5/6	1	0	
	Dyspnea	15	0	6/6	1	0	

would be the satisfaction degree for the group $\max_i(\min(\mu_C(r_i),\mu_Q(p_i)))$. This aggregation with minim and maxim is known as a Sugeno's fuzzy integral [19].

4. Proposal for simple diagnosis

Diagnosis is "the art or act of identifying a disease from its signs and symptoms" [13]. Technically, there are different types of diagnoses, namely differential, early, clinical, and pathological diagnosis. The work that we present here is based on clinical diagnoses, established through case history (anamnesis), physical examination and supplementary examinations. Each specialist may come to a diagnosis through different forms, depending on own experience and knowledge. In short, dealing with medical diagnosis software means using information system technologies and a knowledge base supplied by a medical specialist, to emulate decision making. Symptoms and diseases represent leading entities in diagnosis. A prior simple diagnostic may be taken just with some symptom that characterizes the disease. During the anamnesis prior to a clinical study, other key factors take part to discard *and/or* reach a specialized or approximate diagnosis. Some of these factors are: personal, hereditary, and family history; medical examination results (signs), and other supplementary examinations. Simple diagnosis is possible through the existing relationship between diseases and characterization of symptoms. The selection of disease depends on the incidence of these symptoms under a specific condition. We propose using fuzzy querying for providing automated diagnosis. For this purpose, existing relationship between diseases and their characterizing symptoms must be modeled in a database. Each symptom has a level of incidence (intensity and duration) on the diseases. Intensity level is measured in percentage scale, according the specialist. In the same way, duration level is measured in terms of days presenting the symptom. Thus, the relationship between diseases and their characterizing symptoms is modeled with the relational schema: symptom_disease(diseaseID, symptomID, intensity, duration). A medical specialist must contribute with the necessary knowledge to feed the database. In our practical experience, we have populated the database with a definite universe of common symptoms linked to several respiratory diseases. We have worked with the advisory of a physician who provided diseases information and validated diagnosis.

For each symptom, fuzzy terms with respect to intensity and duration must be established. Terms light, moderate and high are those that physicians currently use for describing symptoms intensity. Moreover, duration is often described by terms acute, subacute and chronic. "Subacute" is the rather recent onset or somewhat rapid change. In contrast, acute indicates very sudden onset or rapid change, and chronic indicate indefinite duration or virtually no change [20]. These terms are expressed in SQLf as fuzzy predicates. Membership functions are established by the expert according own preferences and knowledge. Our prototype diagnosis system has a predefined setting and gives user final interface for the redefinition of these terms. Intensity is measured in a percentile rank from 0 up to 100. Thus, definition of terms for intensity might be as follows: CREATE FUZZY PREDICATE light ON NUMBER AS (INFINIT, INFINIT,20,40); CREATE FUZZY PREDICATE moderate ON NUMBER AS (20,40,60,80); CREATE FUZZY PREDICATE high ON NUMBER AS (60,80, INFINIT, INFINIT). Duration is measured in number of days. In this sense, definition of terms for duration might be expressed in SQLf as follows: CREATE FUZZY PREDICATE acute ON NUMBER AS (INFINIT, INFINIT,

15,45); CREATE FUZZY PREDICATE subacute ON NUMBER AS (15,45,90,120); CREATE FUZZY PREDICATE chronic ON NUMBER AS (90,120, INFINIT, INFINIT).
In a consultation, patient might present several symptoms. Each one would show intensity and duration described by corresponding fuzzy term. A final user interface (UI) allows choosing the presented symptoms. For each symptom, the corresponding fuzzy terms for intensity and duration are selected by means of the UI. Diagnosis system would build and execute the fuzzy query in (4.1). This query models natural language requirement: Find the diseases characterized by some of the symptoms presented by patient, taking into account their intensity and duration, and ranking of those diseases according these characteristics. This is what specialist makes as simple diagnosis.

$$SELECT\ DISTINCT\ diseaseID\ FROM\ symptom\_disease\ WHERE\ symptomsFuzzyCond \tag{4.1}$$

The condition $symptomsFuzzyCond$ is built as follows: Assume user selects M symptoms to be in system variable $\$symID[1]\ldots\$symID[M]$. In the same way, the corresponding fuzzy terms to intensity and duration are on system variables: $\$symIntensity[1]\ldots\$symIntensity[M]$ and $\$symDuration[1]\ldots\$symDurationl[M]$. Then $symptomsFuzzyCond$ would be the disjunctive form SQLf fuzzy logic expression (4.2).

$$\begin{array}{lll}
(& (& symptomID = \$symID[1] \\
 & & AND\ (intensity\ IS\ \$symIntensity[1]\ AND\ duration\ IS\ \$symDuration[1]) \\
) & & \\
 & OR & \ldots OR \\
 & (& symptomID = \$symID[M] \\
 & & AND\ (intensity\ IS\ \$symIntensity[M]\ AND\ duration\ IS\ \$symDuration[M]) \\
 &) & \\
) & &
\end{array} \tag{4.2}$$

We might want to obtain more precise diagnosis based on patient symptoms. A disease characterization by all its symptoms is a pattern. We would like to know in what measure patient state approaches this pattern. According to the medical advisor who supported this work, it would be difficult a patient having all symptoms of a disease in corresponding levels. Therefore, we think in “most of” the symptoms instead of all. For this diagnosis, the system builds and executes a fuzzy query like (4.3) that uses the fuzzy condition $symptomsFuzzyCond$, i.e. expression (4.2). The main structural and semantic difference between fuzzy queries (6) and (8) is that the latter uses a partitioning structure with a fuzzy quantifier, while the former does not involve fuzzy quantifiers.

$$\begin{array}{l}
SELECT\ diseaseID\ \ FROM\ \ symptom_disease\ GROUP\ BY\ diseaseID \\
HAVING\ mostOf\ \ ARE\ \ symptomsFuzzyCond
\end{array} \tag{4.3}$$

Fuzzy query (4.3) corresponds to natural language requirement: *Find the diseases such that most of characterizing symptoms are presented by patient, taken in account their intensity and duration, and ranking those diseases according how near patient is to disease pattern.*

5. Proposal for specialized diagnosis

As we have seen, there is a relationship between diseases and their characterizing symptoms and we take advantage of it to deliver simple diagnosis approach. For obtaining a more specialized diagnosis, other existing relationships may be taken in account. Diseases are also characterized by their signs. Symptoms are subjective observed by the patient, while signs are objective measurable by the physician [13]. The intensity of a sign is in the range of its corresponding measure: for example, the temperature is in Celsius degrees. The relational schema *sign_disease*(*diseaseID*, *signID*, *intensity*), models the relationship between diseases and their characterizing signs In the knowledge base, sign intensity level may be in the measure scale or in percentage. In accordance with specialized knowledge and experience, a level is stored for each sign. According to the medical advisor who supported this work, despite that signs are measurable, specialists prefer to use linguistic terms. Three values (linguistic labels) with respect to intensity may be established. These values are light, moderate and high. For example, for sign fever temperature, the label moderate may be defined by the SQLf statement: CREATE FUZZY PREDICATE moderate ON NUMBER AS (38,38.5,39,39.5).
When the signs are used, physician may obtain a more accurate diagnosis. Diagnosis based on symptoms, expressed as fuzzy queries (4.1) and (4.3), may be specialized combined one of those queries with one of these others (5.1) or (5.2).

$$SELECT\ diseaseID\ \ FROM\ \ sign\_disease\ \ WHERE\ \ signsFuzzyCond \tag{5.1}$$

Fuzzy query (5.1) means that: disease is characterized by some of the signs presented by patient, taken in account their intensity. The fuzzy condition signsFuzzyCond used in query (5.1) is the SQLf fuzzy logic expression (5.3).

$$\begin{array}{l} SELECT\ diseaseID\ \ FROM\ \ sign\_disease\ \ GROUP\ BY\ \ diseaseID \\ HAVING\ mostOf\ \ ARE\ \ signsFuzzyCond \end{array} \tag{5.2}$$

The meaning of fuzzy query (5.2) is: *diseases such that patient presents most of characterizing signs, taken in account their intensity*. Expression (5.3) defines the fuzzy condition *signsFuzzyCond* used in (5.2). System builds the expression (5.3) in the same way that previous condition *symptomsFuzzyCond* in expression (4.2).

$$\begin{array}{l} ((\quad signID\ =\ \$signID[1]\ \ AND\ \ intensity\ =\ \$signIntensity[1]\ \ )\quad OR \ldots OR \\ (\quad signID\ =\ \$signID[N]\ \ AND\ \ intensity\ =\ \$signIntensity[N]\ \ )) \end{array} \tag{5.3}$$

Combination of queries (4.1) or (4.3) and (5.1) or (5.2) for diagnosis may be done by means of UNION or INTERSECT operators, according desired diagnosis. For being be more flexible, we use (4.1) UNION (5.1). On the other hand, when a more accurate diagnosis is desired, we use (4.3) INTERSECT (5.2). These combinations are demand by the physician by means of system user interface.

For a specialized diagnosis, physician may consider other evidence such as patient antecedents. Therefore, we include this relation in the knowledge base, with the relational schema: *antecedent_disease*(*diseaseID*,*antecedentID*) Physician may obtain a more accurate diagnosis using symptoms and signs with antecedents. As in previous case, depending how weak or strong is desired a diagnosis, physician executes query ((4.1) UNION (5.1)) INTERSECT (5.4) or, alternatively, this other combined query (4.3) INTERSECT (5.2) INTERSECT (5.5).

$$SELECT\ diseaseID\ \ FROM\ \ antecedent\_disease\ \ WHERE\ \ antecedentsCond \tag{5.4}$$

Query (5.4) is not fuzzy, nevertheless, for diagnosis it is combined with another query that do is fuzzy. The combination is with the INTERSECT operator because having an antecedent is not sufficient to deliver a diagnosis. The condition *antecedentsCond* used in queries (5.4) and (5.5) is the boolean expression (5.6). System builds this expression in the same way that (4.2) and (5.3).

$$\begin{array}{l} SELECT\ diseaseID\ \ FROM\ \ antecedent\_disease\ \ GROUP\ BY\ \ diseaseID \\ HAVING\ mostOf\ \ ARE\ \ antecedentCond \end{array} \tag{5.5}$$

Query (5.5) is intended for: *filter diseases such that patient presents most of characterizing antecedents, taken in account their intensity*.

$$(\ antecedentID = \$antecedentID[1]\ \ OR\ \ldots\ OR\ antecedentID = \$antecedentID[L]\ ) \tag{5.6}$$

6. Prototype system features

As we have mentioned previously, we have built a prototype automate diagnosis system where rules of diagnostics and decision making are implemented by means of fuzzy queries as we have explained in previous sections. Due to the fact that for diagnosis we need partitioned queries with fuzzy quantified conditions, SQLf querying language must be used, because existing others do not support this kind of conditions. We considered three available implementations of SQLf, they are [21] PostgreSQLf, SQLf-pl and SQLfi. However, PostgreSQLf, up to present development status, does not provide fuzzy quantified partitioned queries. On the other hand, SQLf-pl provides queries fuzzy quantified partitioned, but does not provide application programming interface. Thus we use SQLfi that provides an API for fuzzy querying on top of an RDMBS and implements fuzzy quantified partitioned queries. The system prototype has been built using SQLfi on top of ORACLE RDBMS, their interfaces are developed with JSP and Java Script. User interface messages are in Spanish because this is the official language of Venezuela.

With the interface of Figure 2, patients may choose own symptoms and indicate intensity and duration for performing a simple diagnosis. Left side of the screen shows the list of symptoms. User may select one a time. Right side allows indicate the values (*light*, *moderate* and *high*) for intensity (in Spanish *leve*, *moderado* and *severo*) and the values (*acute*, *subacute* and *chronic*) for duration (in Spanish *agudo*, *subagudo* and *crónico*). Button “Aceptar” adds the symptom and its description to the fuzzy query condition.

Figure 2: Interface for choosing presented symptoms with corresponding intensity and duration.

In this case, system gives information about possible diseases and diagnostic tests that physician could indicate in these cases, it is shown in Figure 3. First column shows the satisfaction degree obtained with fuzzy query evaluation. These degrees establish the order in which results are displayed. A point check allows user to select the disease to see more details about. Second column has the returned diseases names. Third column contains the suggestions of possible diagnostic tests that physician could demand. Also, information about clinic laboratories is available.

Figure 3: Screen with result of a diagnosis.

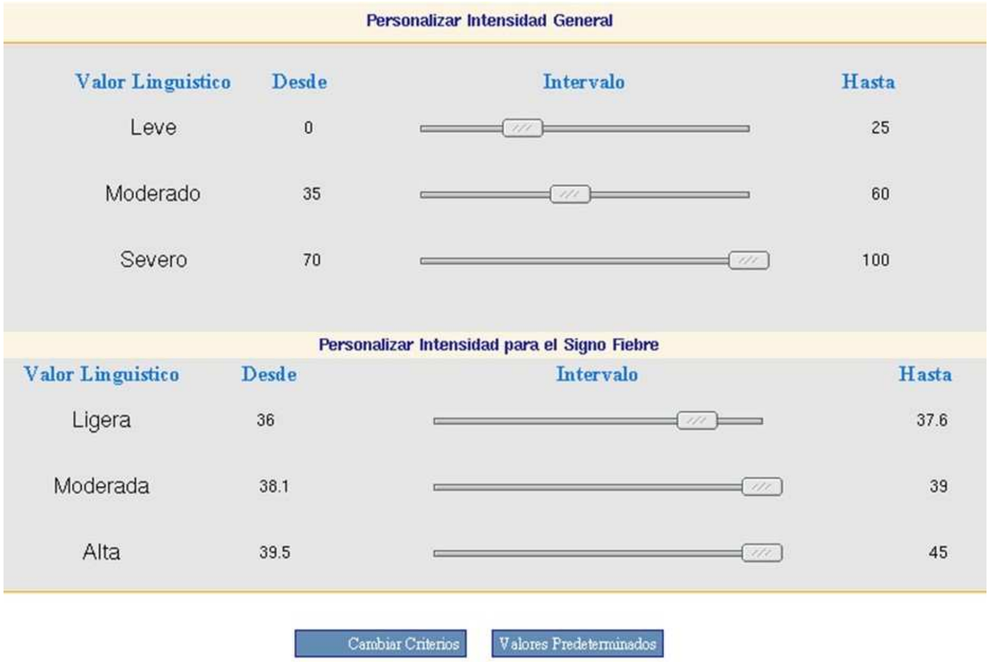

Figure 4: Fuzzy label adjust by means of graphic user interface.

However, other relevant aspects that allow meeting this requirement are also considered and translated into transactional requirements such as incorporation and updating of symptoms, diseases, antecedents, signs, diagnostic tests, and results, among others, these being the most relevant. System also allows insertion and update of items in relationships: symptom disease, sign disease and antecedent disease. This is made with user interface like that of Figure 5. Left side of the screen shows the list of symptoms. Right side allows indication of values for intensity in percentage scale and duration in number of days.

Figure 5: Interface that allows to physician to update the symptom-disease relationship.

These values express specialist appreciation according expertise that would be stored in the knowledge base for future diagnosis. These operations that modify knowledge base are available just for specialists.

7. Conclusions and future work

We have modeled medical diagnosis by means of fuzzy queries to an expert knowledge base. The use of fuzzy queries is because linguistic terms are involved in medical consultation and diagnosis. We provide models for simple diagnostic and specialized diagnosis. Simple considers just with symptoms that characterizes the disease. Specialized involves also signs and antecedents. Resulting models use a querying structure based on fuzzy quantification that only the fuzzy query language SQLf provides. We have taken the case of respiratory diseases, but proposed models are as general that they could be used for other medical fields. Here resides the main contribution of this research work. We just need a specialist to feed the knowledge base and implement these query models. Furthermore, we have developed an automated medical diagnosis system. This system gives support both physicians and patients in the determination of diseases. It was conceived with a graphic user interface that makes easier the use of fuzzy logic inside for non-experienced users in a very intuitive way. This application is supported by a relational database, which is accessed through fuzzy queries in SQLf. Main difference between presented solution and existing fuzzy decision making and expert systems in medicine is the use of a fuzzy querying engine. Here knowledge is modeled and stored in a relational database, diagnosis rules are modeled and processed by means of fuzzy queries. A new kind type of queries is used, that is fuzzy quantified queries. It constitutes an innovation regarding other works. Nevertheless, it would be useful in the future to combine different techniques previously used in automated medical diagnosis with fuzzy querying use, in order to obtain more powerful tools. The evaluation of system performance compared to other diagnosis systems will be mater of future work. It is necessary that future work evaluates the effectiveness of the system in terms of true positive (TP), true negative (TN), false positive (FP) and false negative (FN).

Acknowledgements

We give thanks to our students, who worked in the implementation of the automated medical diagnosis system prototype. “Praise be to the name of God for ever and ever; wisdom and power are his. He changes times and seasons; he deposes kings and raises up others. He gives wisdom to the wise and knowledge to the discerning” (Daniel 2:21-20, New International Version)

References

[1] A.B. Bhattacharya; A.Bhattacharya. *Implementation of fuzzy technology in complicated medical diagnostics and further decision*, pages 935–968. IGI Global, 02 2017.

[2] A.V. Senthil Kumar. *Fuzzy expert systems for disease diagnosis*. Advances in Medical Technologies and Clinical Practice:. IGI Global, 2014. OnLine.

[3] E. Núñez Flórez; R.Vergara Ortiz; J. Bocanegra García. Sistema experto basado en lógica difusa tipo 1 para determinar el grado de riesgo de preeclampsia. *INGE CUC*, 10(1):43–50, 2014. OnLine.

[4] V. Cruz-Gutiérrez; A.S. López. Un sistema experto difuso en la web para diagnóstico de diabetes. *Research in Computing Science*, 107:145–155, 2015.

[5] W.H. Xu; Y. L. Chen; Z. Yan. Development of expert diagnostic system for common respiratory diseases. *Zhejiang da xue xue bao. Yi xue ban= Journal of Zhejiang University. Medical sciences*, 43(2):252–256, 2014.

[6] A. Lamesgin; M. M Sirajudeen. Implementation of an expert system for Lung disease diagnosis. Disponible: https://www.researchgate.net.

[7] Y. Hata; O. Ishikawa; S. Kobashi; K. Kondo; T. Nakano. Automated medical diagnosis system (amds) with normal degree based on fuzzy logic. In *Proc. 2nd IASTED Int. Conf. on Biomedical Engineering*, pages 590–593, 2004.

[8] P. R. Innocent; R. I. John. Computer aided fuzzy medical diagnosis. *Information Sciences. Medical Expert Systems. Elseiver*, 162(2):81–104, 2004.

[9] J. Galindo. Introduction and trends to fuzzy logic and fuzzy databases. In *Handbook of Research on Fuzzy Information Processing in Databases*, pages 1–33. IGI Global, 2008.

[10] Z. M. Ma; L. Yan. A literature overview of fuzzy conceptual data modeling. *J. Inf. Sci. Eng.*, 26(2):427–441, 2010.

[11] O. Pivert; P. Bosc. *Fuzzy preference queries to relational databases*. Imperial College Press, London, 2012.

[12] E. Cox. Relational database queries using fuzzy logic. *AI Expert*, 10(1):22–29, 1995.

[13] Merriam-Webster. Medical Dictionary. OnLine.

[14] C. Li; K. C. Chang; I. F. Ilyas; S. Song. RankSQL: query algebra and optimization for relational top-k queries. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 131–142. ACM, 2005.

[15] S. Borzsony; D. Kossmann; K. Stocker. The skyline operator. In *17th IEEE International Conference on Data Engineering*, pages 421–430. IEEE, 2001.

[16] L. Galindo. New characteristics in FSQL, a fuzzy SQL for fuzzy databases. *WSEAS Transactions on Information Science and Applications*, 2(2):161–169, 2005.

[17] L. A. Zadeh. Fuzzy sets. *Information and control*, 8:338–353, 1965. OnLine.

[18] Y. López; L. Tineo. About the performance of sqlf evaluation mechanisms. *CLEI Electronic Journal*, 9(2):8–1, 2006.

[19] D. Dubois; H. Prade; A. Rico; B. Teheux. Generalized sugeno integrals. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 363–374. Springer, 2016.

[20] Inc MedicineNet. MedicineNet, Inc. OnLine.

[21] A. I. Aguilera; L. Borjas; R. Rodriguez; L. Tineo. Experiences on fuzzy DBMS: Implementation and use. In *XXXIX Latin American Computing Conference*, 2013.

Sobre los autores

Josué Ramírez
Ingeniero en Computación, Magister Scientiarum en Computación. Profesor contratado en la Universidad Simón Bolívar, Venezuela. Adscrito al Departamento de Computación y Tecnología de la Información.
Correo: ramirezjosue@usb.ve - ORCID

Leonid Tineo
Ingeniero en Computación. Magister Scientiarum en Computación. PhD. en Computación. Profesor e Investigador en la Universidad Simón Bolívar, Caracas, Venezuela. Adscrito al Departamento de Computación y Tecnología de la Información. Correo: leonid@usb.ve - ORCID