Public Health Alert System for Health Networks: Application to Cardiology

W Manzi de Arantes Júnior[1,2], C Verdier[1]

[1]Lyon Research Center for Images and Intelligent Information Systems (LIRIS), Lyon, France
[2]Calystène Informatique Santé, Meylan, France

Abstract

The authors propose a system that allows healthcare professionals like physicians and nurses to define medical alerts from patient and environmental data by using fuzzy linguistic variables. Such variables are associated to three importance levels (very important, important or less important) indicating their relative importance in the context and can be developed separately from alerts. Each time a predefined alert is activated by the system, it has two quality indicators which are used for filtering: an 0 to 1 applicability level stating how much the patient is concerned and a trust level indicating its reliability and calculated according to the amount of information that is available at the moment. Finally, lack of information, very common in medical records, is treated transparently thanks to the new concept of modifier, which allows to express the influence variables have on each other by means of a weighted oriented graph.

1. Introduction

Medical alerts are a powerful tool for preventing the occurrence of potential health problems. Unfortunately, nowadays alert detection systems suffer from several drawbacks: they take into account only a few number of parameters provided by physical devices; they use simple range-based algorithms that are not flexible and do not provide quality indicators; they cannot be easily moved from a medical domain to another; they do not treat lack of information properly.

The system we propose aims to detect/fire user-defined medical alerts based on patient records and environmental data. It will be used as part of a geriatric care network integrating healthcare establishments located in the Paris region (France). Geriatrics is a domain in which alerts are very useful because aged people are subjected to chronic diseases conditioned by dozens of parameters which are difficult to survey all the time.

Our algorithms rely on three main concepts: *variables*, *alerts* and *modifiers*. Its main features are: medical alerts are defined through fuzzy linguistic variables, which are intuitive and flexible; relationships between variables are expressed by a graph constructed by an expert user; alerts are associated with a 0..1 trust level (TL) indicating their reliability and with an applicability level (AL) indicating their conformity to a given patient; lack of information is transparently treated and reflected in the trust levels; all alerts are double-filtered before being delivered to users.

2. Proposed system

Our medical alerting system contains three modules for processing information from five databases – medical records, environmental data, detection rules, user profiles and archive – and for detecting and displaying alerts. The first one translates all relational data into XML and is able to detect semantic errors. The second one uses fuzzy sets theory to analyze information and create new medical alerts each time their definition rules match the data from a given patient. Finally, the third one receives the alerts created by the second and displays them by using a multi-agent layer. Only the alert detection module is discussed in this paper.

2.1. Variables

Fuzzy linguistic variables [1] carry information about patients (*weight, age*, *height*, *gender*, etc.) and about their environment (*external temperature*, *air humidity*, etc.). They can be created independently from the alerts using them and are normally domain-independent (a 190 cm tall women is considered *very tall* regardless of the medical domain in which the variable *height* is used: geriatrics, cardiology, etc.). We have also extended the definition of linguistic variable to include some important features that are not found in its original form (the main one being the notion of trust levels).

2.2. Modifiers

All data from patients and the environment must be fuzzified before being used for defining alerts. Traditional fuzzification algorithms treat variables as isolated entities, which is a serious drawback: we cannot state that a 40 kg person is *thin* unless his age is known, as this same value can be considered *fat* or *normal* if he is 8 or 17 years old.

The *age* (as well as the *height* and the *gender*) has thus a very strong influence on the *weight* as it can completely change the linguistic label of later. In this context, we call *modifier* a variable that is combined to another in order to express the influence it has on the second. We say then that the *weight* is modified by the *age*.

The modifiers of a variable have not the same relative importance. The *weight*, for example, is more influenced by *age* than by *gender*, as its linguistic label changes if the patient is 5 or 30 years old but is likely to remain the same if it is a man or a woman. Therefore, we decided to associate an importance level – *very important, important or less important (VI, I and LI)* – to each modifier of our system in order to explicitly state how strong is its impact on the variable it modifies. The modifiers of the variable *weight* can then be expressed as follows:

height-modifiers = age: VI; height: VI; gender: LI.

The influences our variables have on each other are modeled by means of a weighted oriented graph in which a variable at the start of an edge is viewed as a *modifier* of the one at the end of the edge:

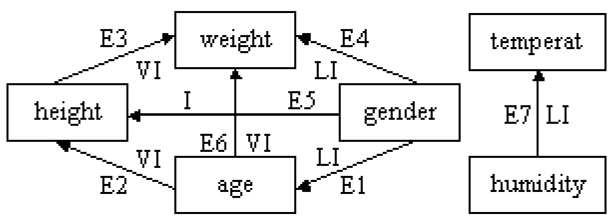

Figure 1: dependencies graph for our example

An edge linking a modifier to a variable is materialized through a collection of *specialized fuzzy sets* representing the specialization of the sets of the variable in the ones of the modifier. For modeling the edge E2, for example, we specialize the sets of the variable (*small*, *normal* and *tall*) in the ones of the modifier (*teenager*, *adult* and *senior*) as follows (specialization in *senior* is not showed) :

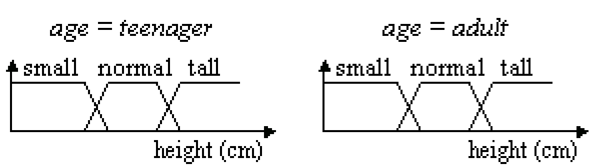

Figure 2: materialization of the edge *age-height*

All edges of the graph are then modeled in this way. On the other hand, isolated and fuzzifiable leaf variables are modeled by using *canonical fuzzy sets* (in opposition to specialized ones). In such sets, membership degrees do not depend on any category and are calculated directly from the crisp values of the variable. For example, the variable *humidity* is fuzzifiable and has no modifier, then it is not possible to specialize its fuzzy sets (*dry*, *normal* and *humid*) in any other. Then, its canonical sets are:

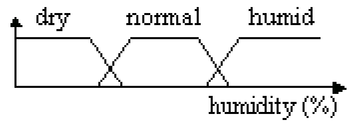

Figure 3: canonical fuzzy sets for the variable *humidity*

Finally, non numeric variables like *gender* are not fuzzifiable and are always leaf nodes in the graph.

2.3. Fuzzification of a variable

Fuzzifying a patient or environmental variable consists in transforming its numerical value (a.k.a. *crisp value*) into one or more tuples of the form <linguistic-label (LL), membership degree (MD), trust level (TL)>. This process requires choosing the right path in the graph. All paths leading to the fuzzifying variable are candidate, but the resulting LL, MD and TL may be very different. Then, we choose the one that yields the best TL as our goal is to provide linguistic variables as reliable as possible.

The fuzzification algorithm visits all candidate paths. Then, each time it traverses an edge, a new LL and MD are created from the specialized fuzzy sets representing the variable and the modifier:

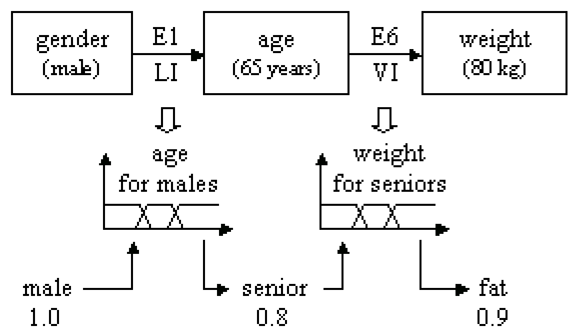

Figure 4: fuzzification of *weight* through the path E1-E6

In this example, the algorithm traverses the path E1-E6 to fuzzify the variable *weight*. In the first step (E1), it uses the LL of the modifier *gender* (*male*) to choose the right fuzzy sets of the variable *age* (the ones specialized to males). Its crisp value (65 years old) is then assigned to this sets and the resulting LL/MD are *senior*/*0.8*. Next (E6), *age* becomes the modifier and *weight* becomes the variable. Again, the LL outputted by the first (*senior*) is used to choose the appropriate sets of the later (the ones specialized to seniors), and the crisp value of the variable (80 kg) is assigned to them resulting in *small*/*0.9*. Finally, the final MD is calculated by multiplying all MD created throughout the path: 1.0 x 0.8 x 0.9 = 0.72. Then, a 80 kg patient belongs to the set of *fat people* with a MD of 0.72.

Bifurcations are always possible at each node because a same crisp value can belong to two fuzzy sets. If such behavior occurs, two sub-paths are created from the node belonging to two fuzzy sets and they are traversed until the fuzzifying variable with the respective LL/MD.

Now, to calculate the TL of the fuzzified variable, the algorithm takes into account the importance levels of the traversed edges and the missing modifiers. First, our three levels are associated to numerical values by an equation that indicates their relative weights:

$$W_{VI} = 2W_I = 3W_{LI}$$

where W_{VI}, W_I and W_{LI} stand for relative weights of VI, I and LI modifiers in the system. This equation states that the impact of VI modifiers is two and three times greater than the impact of I and LI ones on the TL of a variable. This is a crucial parameter setting since different ratios allow to change the sensibility of the system to lacking information. Setting the value of W_{VI} to 6 (to have integer weights), we have:

$$W_{VI} = 6,\ W_I = 3,\ W_{LI} = 2$$

Replacing the symbols LI and VI with the respective numeric values in our example, the TL of the this path is 8 and the TL of the other candidate paths are E1-E2-E3 = 14, E5-E3 = 8, E4 = 2. Therefore, if there is no missing variable in the graph, the most reliable path to fuzzify the *weight* is E1-E2-E3, which is then chosen by algorithm. Finally, before passing the alert to the filtering module, we convert its TL to a 0 to 1 scale by dividing it by the best TL the graph can produce – 14 – and we have TL = 14/14 = 1.0. Then, the fuzzification of the variable *weight* results in the tuple <*LL=fat, MD=0.72, TL=1.0*>.

Note that bifurcations do not have any impact on TL. Then, two paths created by bifurcation may have different LL and MD, but their TL will be the same.

2.4. Dealing with lack of information

Values of missing variables are approximated through a historical database containing all graph paths used in fuzzifications in which *all* variables were known. Then, each time an unknown variable is found in the graph, a similarity analysis is performed between the path we are traversing and each path of the historical database: the later will be retained if it corresponds exactly to the one we are traversing and if the remaining variables of the two paths have similar values. For example, if the age is unknown in the path E1-E2-E3, its LL and MD will be approximated by using all historical paths corresponding to this one (having the same nodes) and whose variables (*gender*, *height* and *weight*) are similar to the ones of our patient. Two non linguistic variables are similar if they have the same value (*male* and *male*) and two linguistic ones do if the absolute difference between their values is less than 10% (*weight = 50 kg* and *weight = 53 kg*).

Once the necessary paths of the historical database are chosen, we examine *only* their nodes corresponding to the missing variable and the approximated LL et the DM of the later are calculated as follows:

LL_{apx} *= LL that has the largest number of occurrences*

DMapx = average of the MD in such occurrences

If two more LL have the same number of occurrences, a bifurcation occurs. Finally, *the importance level of a missing modifier whose value were approximated is not included when calculating the TL of its variable*. This way, a missing modifier, even approximated, reduces the reliability of the variable, which is a logical step.

2.5. Alerts

In our system, medical alerts are constructed by using variables and the logical operator *and* as follows:

alert: (var-1 op value-1) and … and (var-n op value-n)

where *op* stands for arithmetic operator (=, ≥ or ≤) and *value-i* is a LL. A physician can, for example, create an alert which will be fired each time *old* or *very old* and *fat* people are exposed to *hot* external temperatures:

alert: (age ≥ old) and (weight = fat) and (temp. = hot)

However, we estimate that the variables of an alert are not equally important. In the above example, *age* and *temperature* are essential, while *weight* is less significant (for old people, an external temperature of 40 degrees is harmful regardless of their weight). Then, we associate an importance level to each one of its variables:

alert: (age ≥ old; VI) and (weight = fat; LI) and (temp. = hot; VI)

Then, for deciding whether an alert must be activated or not, the system searches for its variables in patient and environmental data. Each time a variable is found with *the same linguistic label* in a medical record, the chances that the alert will be fired increase. For example, patient P1 will be more concerned with our alert than P2 because he has three variables with the same LL that the alert (*old*, *fat* and *hot*) while P2 has only two (*old* and *hot*):

Environment: temperature = hot
P1: gender = male, age = old, height = tall, weight = fat
P2: gender = male, age = old, height = tall, weight = thin

The final applicability level (AL) of a medical alert to a given patient is the lowest MD of its variables:

$$AL(alert) = \min(MD(var_1), \dots, MD(var_n))$$

Analogously, the final trust level of a medical alert to a given patient is the lowest TL of its variables:

$TL(alert) = min(TL(var_1), \ldots , TL(var_n))$

Then, in our example, if the fuzzified variables of the patient P1 have the following MD and TL:

AGE: LL = old, MD = 0.8, TL = 0.9
GENDER: LL = male, MD = 1.0, TL = 1.0
HEIGHT: LL = tall, MD = 0.9, TL = 1.0

The final applicability and trust levels of the alert for the patient P1 are given as follows:

AL(P1) = min(1.0, 0.8, 0.9) = 0.8

TL(P1) = min(1.0, 0.9, 1.0) = 0.9

2.6. Filtering generated alerts

In order to avoid flooding physicians with unnecessary information, all medical alerts generated by our system are filtered before they are displayed. We filter them first by TL then by AL, which eliminates all unreliable ones (e.g., TL < 0.9) and, among the most reliable, selects only those whose probability to occur is high (e.g., AL > 0.9).

3. Related work

Medical alerting systems has been used in intensive care units [2, 3] and in controlling abnormal laboratory results [4, 5, 6] and adverse drug reactions [7, 8, 9]. On the other hand, alerting systems based on medical records are less numerous. In [10], the authors propose an alerting system whose inputs are given by a medical information software, but they come essentially from ICU. In [11], a medical record system provides information to an alert generation algorithm, but the rules for defining alerts are complete C modules (there is no user-friendly language). Finally, very few medical alerting systems deal with lack of information. In [12], the authors propose to replace all missing values by *normal* ones, but it is not clear how we can obtain such values. In [3], a trend-based algorithm is proposed in which user-defined physiological trends (like average heart rate) are evaluated to *true*, *false* or *unknown* due to missing values. Then, an score is assigned to each trend composing an alert and their sum is compared with a threshold value to determine whether the alert should be activated. This approach is similar to ours, but detecting trends requires medical data to be continually analyzed and cannot be successfully realized in small datasets like classical medical records.

4. Conclusion and future works

We have created a system for detecting medical alerts from patient records and environmental parameters. It has three main components – variables, modifiers and alerts – which bring modularity and transparently deals with lack of information. Future works include taking into account time constraints on variables so that users can express information like “if temp. is hot *during two days*, then”.

References

[1] Zimmermann HJ. Fuzzy set theory and its applications. Second Edition. Kluwer Academic Publishers; 1991.

[2] Chen HT, Ma WC, Liou DM. Design and Implementation of a Real-Time Clinical Alerting System for Intensive Care Unit. In: American Medical Informatics Association 2002 Annual Symposium Proceedings; Nov. 2002. p. 131-135.

[3] Schoenberg R, Sands DZ, Safran C. Making ICU alarms meaningful: a comparison of traditional vs. trend-based algorithms. In: Proc AMIA Symp; 1999. p. 379-83.

[4] Shabot M, LoBue M, Leyerle B, Dubin S. Inferencing strategies for automated alerts on critically abnormal laboratory and blood gas data. In: Proc 13th SCAMC; 1989. p. 54-57.

[5] Bradshaw KE, Gardner RM, Pryor TA. Development of a computerized laboratory alerting system. Comput Biomed Res. 1989; 22:575-587.

[6] Kuperman GJ, Teich JM, Bates DW, et al. Detecting alerts, notifying the physician and offering action items: a comprehensive alerting system. Proc AMIA Annual Fall Symp 1996; 20:704-708.

[7] Dormann H, Criegee-Rieck M, Neubert A, et al. Implementation of a computer-assisted monitoring system for the detection of adverse drug reactions in gastroenterology. Alimentary Pharmacology & Therapeutics; 19(3):303-309.

[8] Classen DC, Pestotnik SL, Evans RS, Burke JP. Computerized surveillance of adverse drug events in hospital patients. JAMA. 1991; 266:2847-2851.

[9] Evans RS, Pestotnik SL, Classen DC, et al. Preventing adverse drug events in hospitalized patients. Ann Pharmacother 1994; 28:523-527.

[10] Shabot M, LoBue M, Chen J. Wireless clinical alerts for critical medication, laboratory and physiologic data. In: Proc. of the 33rd Hawaii Int. Conf. on System Sciences; Jan 4–7, 2000.

[11] Oppenheim MI, Mintz RJ, Boyer AG et al. Design of a clinical alert system to facilitate development, testing, maintenance, and user-specific notification. Proc AMIA Symp. 2000; 630-634.

[12] Ennett CM, Frize M, Walker CR. Influence of missing values on artificial neural network performance. Medinfo 2001; 10(Pt 1):449-53.

Address for correspondence

Wilmondes Manzi de Arantes Júnior
LIRIS – Bât. Blaise Pascal – Campus de la Doua
69622 – Villeurbanne Cedex - France
wilmondes.manzi@insa-lyon.fr