Managing Diabetes: Pattern Discovery and Counselling supported by user data in a mobile platform

Diogo Machado[†], Tiago Paiva, Inês Dutra[*], Vítor Santos Costa[*], Pedro Brandão[†]
[†]Instituto de Telecomunicações, Faculty of Sciences, University of Porto
[*]CRACS, INESC-TEC, Faculty of Sciences, University of Porto
dmachado@dcc.fc.up.pt, up200903816@fc.up.pt, {ines, vsc, pbrandao}@dcc.fc.up.pt

***Abstract*—Diabetes management is a complex and a sensible problem as each diabetic is a unique case with particular needs. The optimal solution would be a constant monitoring of the diabetic's values and automatically acting accordingly. We propose an approach that guides the user and analyses the data gathered to give individual advice. By using data mining algorithms and methods, we uncover hidden behaviour patterns that may lead to crisis situations. These patterns can then be transformed into logical rules, able to trigger in a particular context, and advise the user. We believe that this solution, is not only beneficial for the diabetic, but also for the doctor accompanying the situation. The advice and rules are useful input that the medical expert can use while prescribing a particular treatment. During the data gathering phase, when the number of records is not enough to attain useful conclusions, a base set of logical rules, defined from medical protocols, directives and/or advice, is responsible for advise and guiding the user. The proposed system will accompany the user at start with generic advice, and with constant learning, advise the user more specifically. We discuss this approach describing the architecture of the system, its base rules and data mining component. The system is to be incorporated in a currently developed diabetes management application for Android.**

I. Introduction

Diabetes is a chronic disease that in 2015 affected 415 million people around the world [1]. Despite all the efforts to stale the growth of this disease, the number of people with diabetes is continuously increasing and it is estimated to reach 642 million in 2040 [1].

In order to monitor and help a user, it is important to obtain information in order to understand and adapt to the user's unique characteristics.

With the dissemination of mobile smart devices, mHealth applications became an useful tool that clinicians recommend [2]. MyDiabetes is a mobile application that started as a way for users to register their daily data. The user's diabetes diary can travel with her can be transferred to a health professional.

For users, having their records available at all times can be very useful. To the common user, a list of values is a way of knowing if the current management is being effective. However, *inside* these values, usage patterns and miscalculations can be hidden from the patient. Our approach is to add,to the MyDiabetes application, a system capable of advising the user.

Our system is composed by two components: an Advice Rule Based System (ARBS), and a data-mining system. The ARBS embeds medical protocols and medical approved advice, translated to logical rules. These rules and protocols are being developed and implemented with the cooperation of the endocrinology service of the S. João's Hospital. The data-mining algorithms serve as a way to uncover hidden patterns in the registered data.

The main objective of this work is to uncover usage patterns and to advise the user in these situations, through the use of the ARBS and data-mining, guiding the user in general diabetic issues.

II. State of the art

Continuous monitoring of diabetic patients is a complex problem, as it is impossible for medical experts to be permanently available and monitoring their patients.

A. Expert systems

"An expert system is a computer program that provides expert advice as if a real person had been consulted where this advice can be decisions, recommendations or solutions" [3]. In the diabetes' field there are implementations of this type of system for diagnosis [4] and for treatment advice [5].

The proposal by [5] is implemented using VP-Ex-pert Shell, a tool for expert system's development. The knowledge acquisition for this project was obtained through direct interviewing of medical specialists and nurses from the diabetes' field and from the study of other related scientific resources. This system was evaluated by the internees and diabetes specialists of Hasheminezhad Teaching Hospital. The users were asked particular questions about their condition and diabetes' history like e.g. "Is your blood sugar rate equal or greater than 120?". With the answers from these questions the system is able to conclude a treatment advice. This approach was tested on 30 diabetics of various types. The results were compared with the diagnosis and advice given by the specialists. This project concluded that an expert system is able to facilitate the treatment of diabetic patients as the system's consistency was approved by the internees.

Another approach to diabetes' managements using an expert system was the Diabetes Advisor project [6]. Developed using the JESS expert system shell, it was designed as an interactive, menu based, system. At the end of the consultation, the Diabetes

978-1-5386-1629-1/17/$31.00 ©2017 IEEE

Advisor displays a summary of the main recommendations. If the situation justifies, the system may even recommend the user to seek medical council. This project was found to be useful even as a prototype that still requires improvements in the user interface and accessibility.

B. Data-mining algorithms

Data-mining research has broadened different medical fields such as cancer detection [7]. In the diabetes' field, data-mining is also a useful tool for monitoring and diagnosis [8].

Lee, Gatton and Lee [9] aim to provide diabetes' management for diabetic patients, taking into account the glycaemic level. This proposal considers two different approaches to the diabetes monitoring problem: a rule-based solution and a data-mining solution. The data-mining solution is obtained through K Nearest Neighbor (KNN), a classification algorithm used in data-mining and machine learning, applied to a sample set. The main goal of this project is to obtain an optimum treatment recommendation. This project is implemented as web services and as a Personal Digital Assistant (PDA). The results presented show that the system was able to calculate calorie intake based on the patient's condition and was able to manage meals, exercise therapy and glycaemic values.

After comparing the two methods, the prescription algorithm using the KNN classifier was selected as the optimum treatment method as it only needed the sample data, whereas the prescription algorithm based on rules needed prior knowledge about diabetes to select a treatment [9].

III. System architecture

In its core MyDiabetes is a mobile application composed of the following elements: 1) a User Interface (UI), to interact with the user collecting and showing information/data; 2) a DataBase (DB) where the application is able to store and use the user's information or its own; 3) the *Inference* component, used by the Advice Rule Based System (ARBS) whose task is to provide advice that follows medical practice and standards.

These components are managed and used by the *Core* of the application. We introduced the *Inference* component to enable logic programming in this environment. This component requires the integration of a Prolog compiler to perform logic operations. The ARBS, through the *Inference* component, uses a YAP [10] port to the Android system. YAP is a high-performance Prolog compiler developed at LIACC/University of Porto and at COPPE Sistemas/UFRJ. This port is still being developed at Faculty of Sciences of the University of Porto, and as such it is still not possible to test the ARBS in an Android environment.

The *Core* is the only component able to call the ARBS. This event is triggered by the user's initiative as a new register option is selected. By calling the ARBS the *Core* verifies if the user needs to be advised. The ARBS and the *Core* both can access the DB component. All data obtained by the *Core* is stored in the DB. In contrast, the ARBS needs to read the data from the DB, but currently does not update the DB.

IV. The advice rule based system

The ARBS is a Prolog-based language set of rules, that analyses data, based on medical knowledge and guidelines. The implementation of the ARBS uses three different types of rules, as we will discuss on the following sub-sections.

A. System rules

The *system rules* are the ARBS core. For a medical collaborator the perfect work environment would be one where development and work could be done with clauses that can be read and understood as simple text.

Predicates like `atom_concat/3`, used to concatenate two atoms (single data item), while useful in building rule IDs, or rules that filter certain facts, must be avoided during rule development, with medical collaboration, in order to avoid unnecessary, confusion.

Thus system rules include the rules that support the different sub-components of the ARBS, joining them as a working unit. Currently, the system rules also incorporate user's preferences such as how much advice should be given and a specification for unwanted advice. As users gain experience using the application, some advice may become dispensable. With this in mind we allow users to block certain advices. These blocks are not permanent as the advice may become relevant again in the future.

The advice's text is specified in different files according to the user's language. The *Core* informs the ARBS of what language is currently being used. The ARBS, through the system rules, then selects the corresponding advice file so that when advice terms are returned to the core they have the chosen language. Currently the system contains English and Portuguese advice files.

B. Advice query rules

One of the most important elements in the ARBS is the ability to access the stored data, i.e., the database. As the main goal of the application is to know the user's current health state so to advise accordingly. The *Advice Query Rules* are Prolog facts for accessing the database store. They are the link needed to ascertain facts of the user's clinical condition such as the value of the last recorded glycaemia. These rules, used to access the contents of the DB, are a direct use of the YAP's library capabilities of importing database tables as terms in the logical program's environment [11].

C. Medical rules

The ARBS, when called by the *Core* verifies if the recorded data fits a range of known clinical conditions. Medical Rules are clauses that represent different clinical conditions. Every one of these rules places the user in a particular situation e.g.: “the patient exercised recently (less than 2 h) and is going to take insulin” will trigger an advice to warn the user for the need of taking precautions.

Clinical conditions are represented by facts and symptoms e.g.: “Hypoglycaemia is defined by abnormally low blood glucose (blood sugar) levels, normally with values below

70 mg/dl”[1]. People with hypoglycaemia may show symptoms such as: shakiness, nervousness, anxiety, sweating, chills and clamminess. Likewise, Medical Rules are composed by conditions and symptoms. In this case every condition is a fact about the user’s health state, and is represented by an Advice Query Rule (AQR). If all the conditions of the Medical Rule are met, it denotes that the user is in a certain clinical condition.

The system is not able to verify symptoms automatically. The solution encountered for this problem was the creation of a particular interaction with the user. When needed, the system will ask the user if a certain symptom is present. If the response is affirmative, a new fact is added to the ARBS indicating that this symptom is present. With this new information the *medical rules* can determine the clinical condition more accurately.

The connection between the analysis and the advice’s text is made through an `ID`. Every medical rule has an unique `ID`, which, when the rule is triggered, is used to search the corresponding advice.

Advising the user requires data input knowledge, but also the current context. An advice is only meaningful if given at the right time. There are two possible occurrences when an advice is given by the application: 1) before receiving a record – when the user has the intention of inserting a new record; 2) after a record is stored – after new information was inserted into the database.

The two states mentioned embed different types of information. When the user is going to insert a new register, the system knows which type of register the user wants to insert, and knows what the user did recently. This creates the opportunity to call for the user’s attention, to ponder about the action that is going to be taken. For example, considering a case where the user wants to insert a new meal register. If the system detects that the user recently exercised, then the system must warn the user to the fact that this exercise must be taken into consideration while calculating the meal’s bolus.

After a new register is inserted, the system can evaluate the user’s current state. If the user, for example, inserts a flu condition, the system’s reaction should be to warn the user to the increased risk of glycaemia fluctuations, which translate on a need to check glycaemic values more frequently.

This brings up an important parameter of the *medical rules*, the *urgency* value. The urgency value may range from 1 to 10, 1 being the least urgent and 10 the most urgent advice to give. This attribute is specially important when trying to filter or sort the triggered advice. Thus, the ARBS uses this urgency to deliver to the Core a list of sorted advice.

In order to better understand medical rules, an example is represented in listing 1. In this example the rule verifies if the user has a low glycaemia value, when the user has the intention of registering a new exercise. Since exercise lowers blood glucose, it is important to take precautions before and after exercising, even more if the user already has low glycaemia values.

This medical rule has the following arguments: `start` defines the moment of intervention as the time the user enters the ’register new activity’; `exercise` defines that the system will only trigger this rule in an exercise (new) register situation; `6` is the indication of urgency[2] of this rule; `hasLowGlucoseBeforeExercise` was the chosen `ID`, in the advice’s text file there is an advice with the same `ID` that will be returned if the rule is triggered. As for conditions, the system knows the user is going to insert a new exercise register (defined by the arguments), so it only needs to confirm if the user registered recently a low glycaemia value. This is evaluated by the AQR `hasRecentValueLow` with the argument `glucose`.

```
inRisk(start ,exercise ,6,hasLowGlucoseBeforeExercise)
    :- hasRecentValueLow(glucose).
```

Listing 1. Medical Rule that verifies if the user had a recent low glycaemia value, accounting if the user is going to do or did exercise.

D. Crisis situations and possible causes

To deduce possible causes, medical knowledge is essential. The ARBS has prepared for each situation of crisis a list of possible causes, created with the tutelage of medical experts. MyDiabetes, when asked to, proceeds to evaluate each one of the possible causes for the occurred crisis. After this evaluation, the system searches for the best advice for the conclusions obtained. These possible causes are defined in the medical rules. To better understand these particular rules, one example is represented in the listing 2.

```
possibleCause(hypoglycaemia, [(exercise,'+'), (meal,
    '-'), (insulin, '+')]).
```

Listing 2. Medical Rule that implements a list of possible causes for a Hypoglycaemia.

This particular rule stores the possible causes for an hypoglycaemia. It is constituted by a crisis type, in this case hypoglycaemia and a list of possible causes. The elements of the possible cause list are composed by a register type and a plus our minus symbol. These symbols represent the existence or absence of a recent register. In the case shown, the possible causes for an hypoglycaemia are: to have recently performed exercise, the absence of recent meals; or a recent insulin intake.

The rule that verifies the existence/absence of a register is the AQR `hasRecent(RegisterType)`. The verification process of each of the conditions by the system is not visible to the medical expert, since these are system details.

V. Data-mining system

The data-mining component’s main objective is to uncover patterns in the user’s data, in order to create rules to be added to the ARBS. This way building rules specific to the user, i.e., in a more individual way. By uncovering different patterns it is possible to reach common patterns between individuals. If able to uncover these patters, we will be able to develop rules

[1] From American Diabetes Association.

[2] Values for urgency are not totally validated and still need medical confirmation.

to guide not only a user but a group of user's associated with this pattern.

To achieve our goals, one of the main requirements for the use of data-mining is to have a large amount of data. Due to the lack of available relevant databases and information, this is an hard requirement to fulfil. In order to achieve our study goals, through a partnership with the endocrinology service of the S. João's Hospital, we started collecting volunteer's information. The application used in this test was the base MyDiabetes application. The ARBS and the data-mining components were not embedded in the test application. At this point we have thirty one volunteers registered and from these, eight sent information. From the eight, five sent information regularly.

A. Methods used

In this work two different methods were used: association rules and Bayesian networks.

1) Association rules: reveal links, and the weight of these links, between variables. By applying this algorithm to our users we were able to conclude rules such as e.g. "At Wednesday in the afternoon your usual meal results in high glycaemic values.". This connection of days and times of the week with crisis occurrences is fundamental to avoid or correct incorrect behaviours.

2) Bayesian networks: show variable probabilistic dependencies. In contrast to the last example, where we found relations between variables, with Bayesian networks its possible to approach the problem of crisis prediction in a different manner. After creating a network for a user we can now ask e.g."what is the probability of having hypoglycaemia given that today is Thursday.".

B. Creation of new rules

The data-mining and ARBS are independent systems able to work together. The data-mining system is able to uncover patterns and find relations between variables, which may be used to generate "Medical Rules". These rules would then be added to the default ones the user has.

If we take the previous example "At Wednesday in the afternoon your usual meal results in high glycaemic values." we can create a Medical Rule as represented in listing 3.

This rule triggers at the **start** of a new **meal** register when the current day is **wednesday** in the **afternoon**. By triggering the rule returns an `ID` with the value **dMrule_wed_meal** that will be converted to an advice. This advice warns the user to the conclusions from the data-mining component. The user, knowing this fact, is able to take precautions and avoid hyperglycaemia. This rule's `ID` would be connected to an advice that would warn the user for the need of taking precautions. If the user corrects this behaviour, the pattern disappears and the rule would be removed.

```
inRisk(start, meal,9,dMrule_wed_meal)
   :- isToday(wednesday, afternoon).
```

Listing 3. Medical Rule generated from Data-mining association rules

VI. Conclusion

Diabetes treatment has three pillars exercise, diet and insulin. Nonetheless, education is a crucial element in diabetes management. Considering this, we believe that automated management is not enough as diabetes requires lucid care in order to avoid crisis. Our approach aims to help the user to manage glycaemic values with alerts and educating the user (recalling the doctor's advises), to avoid crisis. We plan to include this system on the MyDiabetes application as soon as the inference component is available. This will allow further tests to conclude the usefulness of this system to the users.

By using data-mining we are able to individualise our advice, this way adapting to our user's needs. Our approach still depends of the user's input to function. A great contribution would be for the application to be able to communicate with other devices like glucometers or even with continuous monitoring devices. In the immediate future it will be impossible to implement these communications between devices as companies are cautious of their trademarks and protective of their communication protocols. Cooperating at this level could be a major benefit for the user as most of the user input would be done automatically.

Acknowledgment

This article is a result of the project NanoSTIMA Macro-to-Nano Human Sensing: Towards Integrated Multimodal Health Monitoring and Analytics, NORTE-01-0145-FEDER-000016, supported by Norte Portugal Regional Operational Programme (NORTE 2020), through Portugal 2020 and the European Regional Development Fund.

References

[1] International Diabetes Federation, "IDF diabetes atlas," 2015. [Online]. Available: www.diabetesatlas.org

[2] A. P. Demidowich, K. Lu, R. Tamler, and Z. Bloomgarden, "An evaluation of diabetes self-management applications for Android smartphones," *Journal of Telemedicine and Telecare*, vol. 18, May 2012.

[3] K.-W. P. Byoung-Ho Song and T. Y. Kim, "U-health expert system with statistical neural network," *Advances on Information Sciences and Service Sciences*, vol. 3, pp. 54–61, 2011.

[4] T. S. Zeki, M. V. Malakooti, Y. Ataeipoor, and S. Tabibi, "An expert system for diabetes diagnosis," *American Academic & Scholarly Research Journal*, vol. 4, September 2012.

[5] Y. A. Seyedeh Talayeh Tabibi, Tawfik Saeed Zaki, "Developing an expert system for diabetics treatment advices," *International Journal of Hospital Research*, vol. 2, pp. 155–162, September 2013.

[6] A. Mbogho, J. Dave, and K. Makhubele, "Diabetes advisor–a medical expert system for diabetes management," in *International Conference on e-Infrastructure and e-Services for Developing Countries*. Springer, 2013, pp. 140–144.

[7] P. Ferreira, N. A. Fonseca, I. Dutra, R. Woods, and E. Burnside, "Predicting malignancy from mammography findings and surgical biopsies," *IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2011)*, November 2011.

[8] L. Han, S. Luo, J. Yu, L. Pan, and S. Chen, "Rule extraction from support vector machines using ensemble learning approach: An application for diagnosis of diabetes," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, pp. 728–734, May 2014.

[9] M. Lee, T. M. Gatton, and K. K. Lee, "A monitoring and advisory system for diabetes patient management using a rule-based method and KNN," *Sensors (Basel)*, vol. 10, pp. 3934–3953, April 2010.

[10] V. S. Costa, R. Rocha, and L. Damas, "The YAP prolog system," *TPLP*, vol. 12, no. 1-2, pp. 5–34, 2012.

[11] T. Soares, M. Ferreira, and R. Rocha, "The MYDDAS programmer's manual," Faculdade de Ciências da Universidade do Porto, Tech. Rep., 2005.