MediQuery-An Automated Decision Support System

Rebeck Carvalho[#1], Rahul Isola[#2], Amiya Kumar Tripathy [#3]
Don Bosco Institute of Technology
Mumbai - 400078, Maharashtra, India
[1] *maaask3@gmail.com,*[2]*rahulisola@gmail.com,*[3]*aktripathy@iitb.ac.in*

Abstract

Traditionally the enormous quantities of medical data are utilized only for clinical and short term use. MediQuery puts to use this vast storage of information so that diagnosis based on this historical data can be made. There are systems to predict diseases of the heart, brain and lungs based on past data collected from the patients. We focus on computing the probability of occurrence of a particular ailment from the medical data by mining it using a unique algorithm which increases accuracy of such diagnosis by combining Neural Networks, Bayesian Classification and Differential Diagnosis all integrated into one single approach. The system uses a Service Oriented Architecture (SOA) wherein the system components of diagnosis, information portal and other miscellaneous services provided are coupled.

1. Introduction

Since the advent of advanced computing, doctors have always made use of technology to help them in various possible ways, from surgical imagery to x-ray photography. But technology has always stayed behind when it came to diagnosis, a process, that still requires a doctor's knowledge and experience to process the sheer number of variables involved, ranging from medical history to climatic conditions, blood pressure, environment and lots more. The number of variables counts to the total variables that are required to understand the complete working of nature itself, which no model has successfully analyzed yet. To overcome this problem, Medical decision support systems [1] [2] [3] are becoming more and more essential, which will assist the doctors in taking correct decisions. The doctors or any medical personnel have to enter the symptoms of the disease. The system, making use of various techniques including Bayesian Theorem and Neural Networks, will in turn output the root disease along with the set of most probable diseases which have similar symptoms. Conventional methods completely overlook various variables involved such as prevailing conditions, the build-ups resulting in the symptoms etc., due to sheer magnitude of available unknown variables.

To solve this problem, doctors generally perform the process of Differential Diagnosis [4] i.e. get the probable disease by listing down all diseases that show symptoms similar to the one under consideration. This requires a lot of research and a lot of prior medical experience and is a daunting task even for the doctors.

The process of Differential Diagnosis has been emulated in this system, thus making this rather tough task a lot easier. This method is further modified and enhanced to reduce the huge number of underlying variables to just one by finding the root disease, or the most probable disease, using smart pattern matching of symptoms in dynamic classes generated by Bayesian Classification technique [5] and the next probable diseases by performing Differential Diagnosis, using the Hebbian Neural Networks Theory [6]. Using these, and by utilizing a database having a comprehensive list of medical history at the disposal of this system, the probability of occurrence of a disease may be calculated, regardless of the various unknown variables. The algorithm will output the disease from the symptoms entered and also gives the next highly probable disease and thus, the most effective course of action to be performed can be determined.

Medical decision is a highly specialized and challenging job due to various factors, especially in case of diseases that show similar symptoms, or rare diseases. These factors may vary from inexperience, or misdiagnosis due to environmental factors like stress etc. Also, the latest findings and developments by the researchers are not quickly spread to all doctors, which could help in delay in diagnosis and treatment of patients. Experienced doctors generally classify such diseases based on differential diagnosis method. This involves specialized doctors narrowing down the diseases to the root cause by using their knowledge and experience and confirming it by performing various tests. In case of rare diseases or diseases with similar symptoms, due to the number of tests involved, it

might not be always feasible. Especially in developing countries, the problem of lack of trained and experienced doctors leads to intensification of this problem [7]. To tackle this, medical decision support systems were introduced that can be accessed by anyone, anywhere. Thus, the aim of this system is to provide a centralized Medical Decision Support System accessible by all doctors anywhere and help them overcome all these shortcomings and provide more effective diagnosis.

2. Similar Works

A simple search on any search engine gives every possible list of symptoms with even the required medication for any disease. The search results also consist of residential remedies arising mostly from regional superstition and beliefs to extracts from latest medical journals. However, this problem of non-verified over-information may also prove fatal. The authenticity of such results cannot be validated. Even if it can be, the efficiency of the remedy cannot be calculated, as the diagnosis itself might be wrong. Some sites give the feature to diagnose the disease based purely on the input of symptoms [8].

Considering a small subset of medical datasets, algorithms have been formed [9]; and accurate results have been achieved by some of them. But on a much larger generalized data set for every medical field, obtaining accurate results has yet been very difficult.

At the University Of Utah School Of Medicine's Dept. of Medical Informatics, an Expert System program called Iliad [10] has been under development for several years. Iliad uses Bayesian reasoning to calculate the posterior probabilities of various diagnoses under consideration, given the findings present in a case. It is expert medical software that provides expert diagnostic consultations and diseases manifestations. The version 4.5 covers more than 930 diseases and 1500 syndromes. Its current use: primarily as a teaching tool for medical students.

Similarly, DXplain [11] is a decision support system which acts on a set of clinical findings (signs, symptoms, laboratory data) to produce a ranked list of diagnoses which might explain (or be associated with) the clinical manifestations. DXplain takes advantage of a large data base of the crude probabilities of over 4500 clinical manifestations associated with over 2000 different diseases, which is very limited considering more than 20000 diseases that actually exist. Also, new diseases are not considered at all, and the system leaves no scope to add them subsequently. The system uses a modified form of Bayesian logic. DXplain is in routine use at a number of hospitals and medical schools mostly for clinical education but also for clinical consultation.

The disadvantages of both these softwares are that both use only Bayesian Classification to perform differential diagnosis, which is performed on a fixed set. Hence, the softwares become redundant over time. They run on the assumption that the classes pre-generated in the systems is fixed, and will not change, but in real life, it is not so. The symptoms vary based on various conditions, including climatic conditions like season, average temperature, humidity and rainfall, to prevailing medical conditions that the patient is suffering from. These softwares have a static database, which is not self-learning, hence catching trends, and adapting according to the conditions becomes impossible.

3. Functional Aspects: Relevant Data Fetching (Data Mining)

Existing medical systems focus on collecting and mining the entire data. The entire patient records are loaded and all factors are considered. The medical field cannot be easily analyzed because for generating a probabilistic rating, not only symptoms but factors like test results and external climate conditions are also required, which may or may not be present in the report. Existing system have failed to understand one of the most important attributes, Misdiagnosis, which interconnects and addresses all these issues. If we mine Misdiagnosis and store it as an attribute, it will solve the problem, because doctors' misdiagnosis only in case of presence of ambiguities, and in similar cases, there is a high chance of other doctors also performing the same misdiagnosis. The misdiagnosis attribute is very important in mining because it will directly lead to correct diagnosis, and in turn, it will eliminate all the underlying variables as they would already have been covered in the diagnosed ailment.

Retrieving relevant medical data for this system is a major task, with various issues rising about data confidentiality, data security, ambiguity etc. To tackle these issues, we came up with the idea of implementing a complete Hospital management system, with the MediQuery system as an integral part. This way, the issue of data confidentiality is addressed, as the system will be accessible only by the hospital representatives, but also have access to all the records generated by the doctors. Also, since the MediQuery system will be hosted on the internet, the database will be accessed and updated by every user of the system from all the hospitals using the hospital management system, resulting in an updated and complete database.

More users will result in a more complete and accurate database.

Once the medical records are obtained and are in place, the system uses NLP (natural language processing) [12] technique to determine the report results. MontyLingua, a NLP tool, was used in the system. Text mining using NLP obtains two results from the reports generated by the doctors, the correct and wrong diagnosis, and the symptoms resulting in the diagnosis. Other aspects like personal patient information, like name etc., or any information that can identify the patient in any way, are completely ignored, hence keeping the confidentiality intact. These results are stored in the database as: Disease Diagnosed & Actual Disease attributes. The symptoms will result in change in weightage of the symptoms for the corresponding disease (explained later in the algorithm section).

4. Functional Aspects: Algorithm

Various algorithms utilized in this method are explained below. Fig. 1 shows a 3 tier workflow pattern of the system. The system uses a 3 tier workflow method for triple precision diagnosis. Each case is explained with the help of a practical case study given below.

Fig. 1: Workflow Pattern

A. Iterative pattern search

TABLE I
Database for Step 1: Iterative Pattern Search

Diseases	**Symptoms**	**Weight**
Diabetes	Headache, increased in blood sugar,insulin low	W1,W2, W3,W4
Pericarditis	Chest pain, Fever, weakness, malaria, shortness of breath, Syncope	W5,W6, W7,W8, W9,W11
Viral Fever	Head Ache, Cold, Fever, Running Nose, Weakness	W1,W12, W6,W13, W8
…	…	

Iterative pattern search utilizes data that is stored as given in table 1. The first step of the algorithm involves selecting the symptoms shown by the patient. As an output, the algorithm gives the list of all possible diseases ranked according to the number of symptoms matched in the database. The list is generated after input of every symptom. After the first iteration, for the second iteration, the next list of symptoms will be shortlisted according to the disease list that was obtained in the previous iteration. i.e. the new symptom list will contain symptoms of only those diseases that were obtained in the previous list. These related symptoms will then be shown to the user who shortlists another symptom from the new list. The new disease list will be listed, ranked according to the number of symptoms matched. The ranking is generated according to the percentage match of the total number of symptoms entered. This procedure goes on iteratively, with diseases being placed in the ranks according to its probabilities. After a few initial iterations, top diseases in the list gain highly in ranking, allowing one to identify the ailment. i.e. ranking of the disease varies at a much greater precision as more and more symptoms are given. e.g. on the database given in Table I, on entering Headache as a symptom, Diabetes and Viral fever will have 100% match, giving it equal ranking, whereas Pericarditis will be excluded. On entering the next symptom, say Fever, Viral Fever will have both the symptoms matching, giving it 100% match, while the other two will have a match of only one out of the given two symptoms, thus 50% match, resulting in a drop in their ranking. But this has to search a record database of more than 20000 diseases and even more symptoms, which is very time consuming, so we apply Bayes Classification to classify diseases into subgroups and if a group of symptoms match we give higher preference to that subgroup, hence searching in that subgroup thus reducing database access.

The working of this is as follows: Let X be the symptom whose class label is unknown. Let H be some hypothesis, such as "symptom X belongs to a specified class C." For classification, we want to determine P (H|X) -- the probability that the hypothesis H (which corresponds to a group of diseases) holds, given the observed symptom X.

P(H/X) = (P(X/H) * P(H))/P(X)……I

The standard algorithms compute symptoms probability with individual diseases, but we compute symptoms probability with a group of diseases. We have computed support and confidence levels (eq II and III) for each set of symptoms with a particular group using statistical record. So if the required support and confidence is reached [13]; we directly search the Group of disease resulting in faster Pattern Matching.

Support = $(X_1 U X_2 \ldots U X_N)$ / Total Records..……...II

Confidence = $(X_1 U X_2 \ldots X_N)$ / Total Records of C_1, C_2 or...C_N ……..…….....III

Practically, when the maximum symptoms entered show highest weights for diseases falling in a few

particular subsets, the diseases in those subsets will be searched. E.g. if the symptoms include Chest pain, palpitation, high breath rate (ref. Table II), and these symptoms have maximum weightage for the diseases Heart Attack and Coronary Artery Disease, that have maximum probability of falling in the cardiovascular subset, then the algorithm will automatically switch to Cardiovascular subset, as the symptoms have maximum probability of pointing towards Heart attack, or other related cardiovascular diseases. The logic of assigning weights is explained later. *(Ref. part D of this section).*

TABLE II
Database for Classified Iterative Pattern Search

General • Weight Gain • Dry Mouth • Fatigue **Ocular** • Amaurosisfugax • Cataract • Blurred Vision **Gastrointestinal** • Anorexia • Bloating • Belching • Blood In Stool **Obstetric / Gynaecological** • Pelvic Pain • Infertility • Labour Pains	**Cardiovascular** • Chest Pain • Palpitation **Neurological/Psychological** • Phobias • Hyperactivity **Urologic** • Impotence • Polyuria **Pulmonary** • Hypoventilation • Hyperventilation • Bradypnea **Integumentary** • Abrasion • Alopecia • Anasarca

B. *Mining medical records*

In the previous case, if multiple diseases are found with similar ranking, it becomes difficult to pinpoint to one of them, when no more symptom is unique to any single disease affecting its ranking. This is especially the case in case of some epidemic in the area, or some rare disease, or disease arising due to localized conditions etc. e.g. Swine flu epidemic initially showed the same symptoms as that of viral fever, resulting in rising the ranking of both viral fever and swine flu. In such cases, it becomes very difficult to point at one disease using the iterative pattern search method. In such cases, we use recent medical history with a time period of 3 months to rank the diseases on basis of the probability of their occurrence in the review period. 3 months provided accurate diagnosis with less noise. If the interpreted diagnosis is still vague, then we proceed to point C.

C. *Differential Diagnosis:*

Hebbian Learning [14] involves two neurons in the brain may well have a connection between them. The neurons can be activate (i.e. firing on all cylinders) or inactive (asleep). If both the neurons are active at the same time, then the strength of the connection between the two should increase. If the neurons are not both active at the same time (i.e. one or both of them are inactive), then the strength of the connection does not increase. This idea of a connection between neurons strengthening is referred to as Long Term Potentiation (or LTP). The Hebbian network model has a 2-node input layer $x=[x_1, x_2]^T$ and a single node output layer $y=[y_1]^T$. Each output node is connected to both input nodes: $y_1=w_{11}x_1+w_{12}x_2$ or in matrix form, we have $y=\mathbf{W}x$

TABLE II
Sample Sorted Database for Differential Diagnosis

Patient	Disease Diagnosed	Actual Disease
A	Diabetes	Diabetes
B	Diabetes	Diabetes
C	Diabetes	Diabetes
D	Diabetes	Diabetes
E	Diabetes	Diabetes
F	Diabetes	Hypertension
G	Diabetes	Hypertension
H	Diabetes	Hypertension
I	Diabetes	Arthritis
J	Diabetes	Arthritis

The learning rule is: $w_{ij}^{new}=w_{ij}^{old}+\eta\, x_j y_i$ (i=1,2; j=1) or in matrix form: $W^{new}= W^{old} + \eta\, y\, x^T$

Here W denotes weights and η is the learning rate, a parameter controlling how fast the weights get modified.

$$\mathbf{\Delta w_{ij}=w_{ij}^{new}-w_{ij}^{old}= \eta.(2a_i -1).a_j} \quad \text{.....IV}$$

Δw is the change in the weight connecting input element i to output element j. a_i is the activation of input element i and a_j is the activation of output element j. η is constant term that prevents the changes in the weight from being too extreme. For the data given in Table III, if we apply Hebbain rule (equation IV) we find that Δw=Relative frequencies.Therefore the individual weights are as follows:

$W(C_1)=5/10$, $W(C_2)=3/10$, $W(C_3)=2/10$

Where,

$C_{1=}$ Diabetes, $C_{2=}$ Hypertension & $C_{3=}$Arthritis

So the Differential Diagnosis would be: *Diabetes => Hypertension =>Arthritis*

D. Weight assigning using Back propagation:

Using Step C, the correct disease shortlisted by the doctor is obtained who confirms it by taking the necessary tests. The final report is then mined using NLP processor Montylingua and the correct symptoms are compared with the original symptoms entered. Initially, all weights are assigned the value zero. For each correct matched symptom the weight increases +1 and for each unmatched symptom the weights are kept constant. The reason for keeping weights constant for unmatched symptoms is that if the unmatched symptoms were assigned negative weights, then certain symptoms would be repeatedly degraded and when they would actually surface in some diagnosis, because of too much negative weight, the change in the ratio of weight of the symptom for that particular disease to the total weight of all symptoms for the disease will be more significant. This will lead small changes in trend to result in bigger change in the ratio as compared to not subtracting negative weight. To keep the weight ratio to be as stable and precise as possible, the fluctuations should not be much. Hence, only positive weights are considered.

The weights assigned here have been found out from the rigorous three tier process. It very important in pattern matching in the sense that they incorporate the misdiagnosis factor and the doctor may get 100 % result in the first step itself.

5. IMPLEMENTATION

The rapid rise of technology and its adoption into the healthcare field has caused healthcare organizations to collect an accumulation of non-interoperable systems that not only need to work together within the organization, but are also accessed from outside. The burden of integration usually falls on the users of the system, who are forced often to access many different systems to complete one task. The use of Service Oriented Architecture (SOA), however, can improve the delivery of important information and make the sharing of data across a community of care practical in cost, security, and risk of deployment. [15]

For implementation on the SOA architecture, we make use of the following web services in our MediQuery system: Pattern Matching, Recent Trends, Differential Diagnosis and Recent DFD. Two main databases used are: Disease / Symptoms Database and the Records Database. Disease / Symptom database is a centralized database and it contains the list of the existing known diseases and their corresponding symptoms, with their weights. The Records Database maintains records of the patients from all the hospitals in the distributed network. These databases are replicated across various servers to achieve fault tolerance with the standard concurrency protocols to achieve atomic transactions.

The Doctor first Queries our system by entering a set of symptoms. Pattern Matching service is then activated which fires the query and presents the result to the Doctor. If the doctor is not satisfied with the results he invokes the Recent Trend Service. Recent Trend Service makes use of the Disease/Symptoms Database and the Record Database and the result obtained from Pattern Matching Service to get results. Lastly to avoid ambiguity in decisions the doctor make use of Differential diagnosis which invokes Differential Diagnosis and Recent Diagnosis Feature which make use of the Disease/Symptoms Database and the Record Database and result obtained from Recent Trend service to get results.

6. Results

We have developed a sample system with limited database to test the above theory using web tools & technologies like PHP, MySQL and AJAX, implemented on simple client-server architecture. The data has been obtained from the following sites medicine.net, wrongdiagnosis.com& webmd.com. We have applied our system on a sample dataset for malaria. Specific test cases were run, and the following results were obtained.

The 1st chart in fig. 2 depicts the number of symptoms matching and their probabilities considering the ranking adjusted according to the weights. The first step resulted in accurate prediction of diseases based on the symptoms entered.

The table in fig.2 shows the list of diseases found by matching symptoms and its probabilities of occurring calculated based on its occurrence in the past three months. The second step resulted in very accurate prediction of diseases based on the recent trends. E.g. it accurately caught Malaria and Dengue for the given symptoms, during the monsoon season, where mosquitoes were a menace in the test locality. This matched very accurately with the patient's actual ailment.

The last chart in fig. 2 shows the diseases similar to the root disease selected and probabilities of the disease occurring calculated based on the differential diagnosis technique.

This system aims to provide essential medical services with clinical precision to everyone. But to do that, decent accuracy was required. Even though the system is to be used by doctors only, and the doctors have the final say, the accuracy of the system should

not be compromised. To verify this, we compared the results obtained by our system with various other medical systems, and these were also verified with a few doctors. The results obtained were fairly accurate, and since the system is self-learning, with time, as the database grows, the accuracy of the system improves.

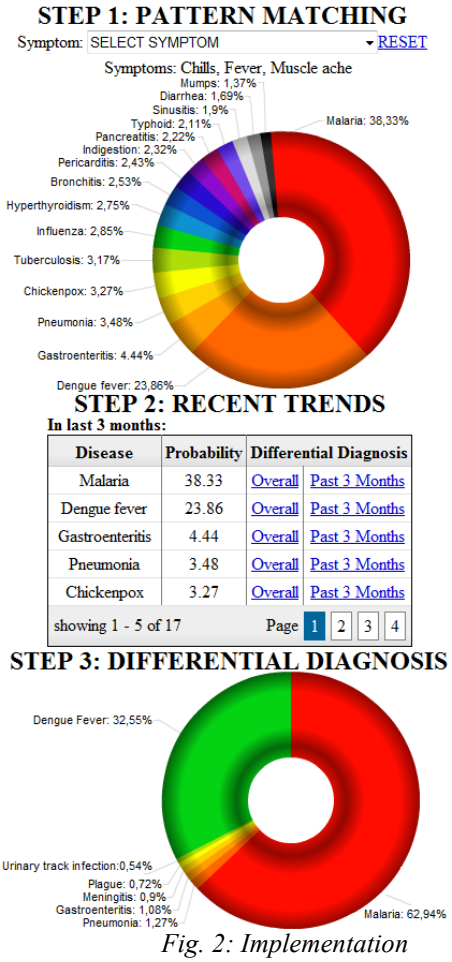

Fig. 2: Implementation

7. Conclusion

With the support of various medicinal practitioners and hospitals, higher probability of getting the diagnosis right can be obtained, compared to what individual doctors can do alone. The system does not give 100% accurate results, which even the doctors cannot do. So it cannot be used as a substitute or a shortcut to diagnosis, as each patient is different. But it can definitely complement the doctors' knowledge and assist them to reach a conclusion. The doctor always has the upper hand to decide whether to use the diagnosis given by the algorithm or not. After sufficient self-learning, with an extensive database of medical records to mine from, this can be used to build formidable medical assistance software that can be of great use to all doctors, and specially the new practitioners and students. It will also help the medical fraternity in the long run by helping them in getting accurate diagnosis and sharing of medical practices which will facilitate faster research and save many lives.

10. References

[1] Berner, Eta S., ed. Clinical Decision Support Systems. New York, NY: Springer, 2007

[2] Kensaku Kawamoto, Caitlin A Houlihan, E Andrew Balas and David F Lobach , "Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success", BMJ 330 : 765 doi: 10.1136/bmj.38398.500764.8F (Published 14 March 2005)

[3] Randolph A Miller, "Medical Diagnostic Decision Support Systems—Past, Present, And Future - A Threaded Bibliography and Brief Commentary", JAMIA 1994;1:8-27 doi:10.1136/jamia.1994.95236141

[4] Walter Siegenthaler , Differential diagnosis in internal medicine: from symptom to diagnosis, 2011 Edition, APPL, aprinta druck, Wemding, Germany.

[5] Jiawei Han, Micheline Kamber, Data Mining Concepts and Techniques, 2011 edition, Morgan Kaufmann Publications.

[6] Richard Bowles, Neural networks and Hebbian learning. available at: http://richardbowles.tripod.com/neural/neural.htm (24th Dec 2010, 5pm GMT)

[7] Susan F. Murray, Stephen C. Pearson, "Maternity referral systems in developing countries: Current knowledge and future research needs", Social Science & Medicine 62 (2006) 2205–2215

[8] Misdiagnosis: Symptom and heath diagnosis checker.Available at: http://www.misdiagnosis.com.(4th Feb 2011,7.30pm GMT)

[9] Michele Berlingerio, Francesco Bonchi Fosca Giannotti, Franco Turini, "Mining Clinical Data with a Temporal Dimension: a Case Study", 2007 IEEE International Conference on Bioinformatics and Biomedicine

[10] Warner HR, Bouhaddou O (1994). "Innovation review: Iliad--a medical diagnostic support program.". Top Health Inf Manage 14 (4): 51–8. PMID 10134761.

[11] Department of Medicine Massachusetts Hospital, Boston, DXplain System (2011). Available at :http://dxplain.org/dxpdemopp/dxpdemo-brief_files/frame.htm

[12] Maurice HT Ling, "An Anthological Review of Research Utilizing MontyLingua, a Python-Based End-to-End Text Processor.", The Python Papers, Vol 1, No 1 (2006).

[13] Roman Slowinski, Izabela Brzezinska1, and Salvatore Greco, "Application of Bayesian confirmation measures for mining rules from support-confidence Pareto-optimal set", ICAISC 2006, Zakopane

[14] Wulfram Gerstner and Werner M. Kistler, "Mathematical formulations of Hebbian learning", Biological Cybernetics, Volume 87, Numbers 5-6, 404-415, DOI: 10.1007/s00422-002-0353-y

[15] Bell, Michael (2010). SOA Modeling Patterns for Service-Oriented Discovery and Analysis. Wiley & Sons. pp. 390. ISBN 978-0470481974.