Expert System to Diagnose Multiple Diseases using Association Factor

Imran M. R.[1], Moez ur Rehman[2], Zia-ul Qayyum[3], Aslam Muhammad[4], Martinez-Enriquez A. M.[5] , Afraz Z. Syed[6]

[1,4,6]Department of CS & E, UET, Lahore, Pakistan
[2]Department of CS, GC University, Lahore, Pakistan
[3]Department of Computing and Technology, IQRA University, Islamabad, Pakistan
[5]Department of CS, CINVESTAV-IPN, D.F., Mexico
(Corresponding author email: maslam@uet,edu,pk)

Abstract: Many learning algorithms exist that are routinely used as commercial system. However, given knowledge in health domain, it is difficult to train computers for the decision making and learning. The problem becomes complex when some common symptoms of multiple diseases are present. Some knowledge based systems are available to find a particular disease but cases exist where patient may have more than one illness. We focus on this issue and develop an expert system which not only finds certain disease specifically, but also diagnoses the probability of other diseases to support in prescribing enhanced treatment. The proposed system learns based on a given knowledge, creating rules for making probable decisions and finds association among symptoms occurred mutually in previous assessments. The tested results are quite satisfied and it works accordingly. The system is flexible for new rule generation and association symptoms.
[Imran M. R., Moez ur Rehman, Zia-ul Qayyum, Aslam Muhammad, Martinez-Enriquez A. M., Afraz Z. Syed: **Expert System to Diagnose Multiple Diseases using Association Factor.** Life Science Journal. 2012;9(1):542-547] (ISSN:1097-8135). http://www.lifesciencesite.com. 81

Keywords: Rule based system, Disease diagnostic, Association factor

1. Introduction:

The expert systems are one of those programs that work on intelligence grounds and produce results on reasoning. Expert systems have two basic components: knowledge base, inference engine [1] and some have also the third component as working memory [2]. These systems are used in many applications of daily routine but are also used in health domain.

MYCIN [3] was designed to diagnose a "Blood Infection" disease and to recommend treatment. Quick Medical Reference QMR [4] diagnoses adult disease. These systems do not provide other patient diseases. Normally, disease itself does not exist; rather patients can be affected by a particular one. Thus, a real Knowledge Base System (KBS) should be exist to diagnose and to associate specific probability for possible diseases.

In addition, it is difficult to produce a differential diagnosis for diseases like Malaria, Dengue Fever and Bone Cancer having some common symptoms (e.g. vomiting, headache, nausea, fever and cough which can be found on http://anytestkits.com). Consequently, not discover on time a hidden disease, it may prolong and becomes severe, giving rise to loss of time, increase cost, and fatal result for patients.

In this paper we tackle this issue and come up with a system which diagnoses different diseases having common symptoms. Our Medical Diagnostic Expert System (MDES) uses associations among the symptoms, trying to find it on probabilistic grounds. The system infers knowledge taking into account previous results since stores previous result for future decision. MDES uses inference rules and precedent decisions available. MDES increases all new knowledge. The system not only finds certain diseases specifically, but also diagnoses the probability of others which may help for enhancing treatment.

Many KBSs have the facility to diagnose by acquiring certain parameters, like MYCIN that diagnoses diseases, its inference rules are represented by IF THEN with a confidence factors. MYCIN uses basic backward chaining reasoning, i.e. is goal oriented.

Dxplain [5] online system assists doctors and paramedical staffs to process differential diagnoses on patients, given some signs, symptoms and laboratory test, the produced results are present and evaluated accordingly. Dxplain starts by acquiring different information about the patient or case in the form of answers binary questions and tries to close possible list of disease candidates.

QMR basically diagnoses adult diseases and also provides information about more than 700 diseases representing the enormous majority of disorder seen by internists in the routine practice along with compendium of less familiar disease.

Similarly GIDEON [6] uses symptoms, laboratory test results, signs and country region to generate a Bayesian ranked differential diagnosis

which is used for diagnostic support and simulation of all infectious diseases in all countries. The Knowledge repository of GIDEON was collected from famous resources from all over the world and the system also provides a monthly electronic literature.

HELP [7] is a hospital information system which provides function to the administration of the hospital in all aspects and also provides decision support system functionalities to the medical doctors to make their decision in efficient way while diagnosing or a performing a treatment. The decision support system part of the system provides important alerts and reminders, data interpretation, diagnosing a patient disease, patient management for suggestions and clinical protocols.

Decision making through association rules mining among large item sets available in different domains to fulfill the required result. There are number of algorithms available for decision making and support systems which works excellent in the relative environment. The algorithms are trying to find the frequency of association among items sets by applying different techniques like association [8], grid analysis [9], consensus theory and decision tables [10, 11]. For instance, Prutax [12] algorithm converts the problem into an acyclic directed graph to make decisions; Partition [13, 14] divides the large itemsets into small chunks for making a decision. Apriori [13] uses association among item sets to draw a final decision. Some extensions of apriori algorithms are AprioriTid [15], Apriori-C [8]

There is no such system to diagnose more than one disease at a time in a patient according to our knowledge but we done it using the Apriori [13, 16] type algorithm named Medical Diagnostic Expert System (MDES) described in detail in next section.

2. Materials And Methods

In order to ease the decision process even in complex situation, we designed, implemented, and tested the Apriori algorithm. Apriori mines a large database, providing relationships and discovering connections through different attributes that characterize diseases. Apriori algorithm is based on prior frequent item set properties for learning association rules [16].

The diagnosis of different possible maladies take into account different parameters associated to certain questions regarding internal and external patient structure, and examination systems. Each symptom consists of a set of characteristics like type, description. Similarly, while diagnosing, requires diagnosis to hit upon. All diagnosis like Kidney Failure, UTI, Blood Cancer or Heart Failure etc. can also have different parameters, serving into the diagnosis process.

Each query in MDES carries equal strength towards the diagnosing process. The criterion is pronounced as normal or abnormal. When the answer is under abnormal value category, this value is recorded and takes part into the process of diagnosing by providing association for the upcoming forecaster symptoms and diagnosis. The Apriori algorithm with the slightly change, takes into the probabilistic grounds. So the solution comes with more than one answer having different probabilities. For this we need the following information.

Categorization of Symptoms

A disease is characterized by different in particular one apparent to the patient and it can be classified by:

***Boolean*:** symptoms with two values {yes, no} or {true, false} e.g. “Feet Swelling” and “Urine is irritable”.

Enumerate: Symptoms whose value is a set of enumeration values at a time like and also one choice taking part in more than one symptom. Always, one was selected for future processing. For instance, “Urine Color” takes values into the set: {Normal, Dark, Yellowish, Brown}; “Amount of Urine” value can be {Normal, More Often, Less Often}.

Range of values: certain range values split into {min-max}, e.g. “Blood Pressure Diastolic” normal-value is into [50 to 100] and other than that value are consider to be abnormal.

Definition of the problem

The process of diagnosis can be defined by the followings tuple:

$$(P, s, d, p) \quad (1)$$

Where

P: Patient to be treated

s: Symptom available = $\{S_1, S_2, \ldots S_n\}$

p= probability of proposed diagnose

d: Diagnosis = $(A, S`)$ (2)

Where

A: the association factor for the symptoms appearing for a given disease

S`: the previously inquired symptom

We consider the following two cases about the selection of the next symptom to be inquired based on the maximum value of associated factor (A).

***The start process*:** When the previously inquired symptom does not exist, *i.e.* S` = Nil and A = 0 (an inexistent previous symptom)

The start of process, when the first symptom begins to be asked, S` depends upon the choice of use. The selection starts the search by exploring the list of symptoms of different diseases.

Previous knowledge exists: Symptoms depend upon the previous knowledge

S` exists, then the choice of next symptom ($S_i$) takes place. $S_i$ depends upon the highest value of association factor $A_i$ *w.r.t.* S` & S_i

The association factor is the occurrence of symptoms that took place together in a particular disease as shown in Figure 1.

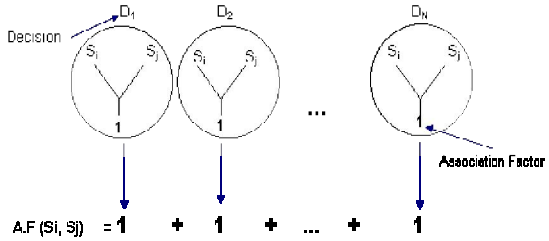

Figure 1 Calculating Association Factor

To calculate the total value of association factor of a certain symptom with respect to other, we count the occurrence of relative symptoms with each other in all decisions. So, to illustrate the functionality, we take some decisions with its symptoms as follow.

$D_1 = S_1, S_5, S_7, S_{13}$

$D_2 = S_5, S_7, S_{10}, S_{12}$

$D_3 = S_1, S_5, S_{10}, S_{15}$

Giving D_i decision, all mutual occurrence of S_i with respect to other S_j is added into the cell (S_i, S_j) (see Table 1). This process is repeated for all i from 1 to 15 symptoms in the present case. For instance, the association between symptoms S_5 and S_7, and then we follow the guidelines from Figure 1 and occurrences are counted: 2 in this case and represented in Table 1, cell (2, 3).

Table 1 Association of symptoms

Sympto	S_1	S_5	S_7	S_{10}	S_{12}	S_{13}	S_{15}
S_1	---	2	1	1	0	1	1
S_5	2	---	2	2	1	1	1
S_7	1	2	---	1	1	1	0
S_{10}	1	2	1	---	1	0	1
S_{12}	0	1	1	1	---	0	0
S_{13}	1	1	1	0	0	---	0
S_{15}	1	1	0	1	0	0	---

The values named association factor are used to rule the decision process. The support of the symptom towards a particular disease can be generated using simple percentage formula which can be derived as

$$\text{Presence of 'd' (\%)} = \frac{\text{Symptoms matched}}{\text{Total Symptoms Associated with 'd'}} \times 100$$

Considering the example, when there are seven symptoms associated with a disease "d" and some instance, only four symptoms qualify the required criteria, and then by using the above formula, the disease "d" has occurred as under

$$d = 4*100 / 7 = 57.14\ \%$$

So, the algorithm predicts that 57.14 percent is the chance that of the disease may occur in the patient and keeping in view that all the

"*Symptoms have same support value for diagnosing a disease*".

Implementation of MDES

The important objects of the algorithm are Symptom and Diagnose which have some attributes like its number of identification, name, and other related fields that may help in the processing of the algorithm. The process of diagnosing is described by the following pseudo code-

```
1. Procedure MDES ()
2.  Symptom S1, S2 … Sk
3.  Diagnose D1, D2 … Dk
4.  Select Symptom s
5.  While (Not Diagnose D)
6.        Result R ← Result.Abnormal
7.        Load Symptom Detail
8.  //  Check the type of the symptom s
9.        if ( value is not valid)
10.           R ← Result.Abnormal
11.           Confidence c ← Confidence [i]
12.       if (c < s.Confidence)
13.           R ← Result.Normal
14.       Process.Add (s,value,R,patient)
15.       if (R = Result.Abnormal )
16.          Add(s)
17.//loading the next proposed symptoms
18.
          Symptom[]symptoms←ProposedSymptoms(s
)
19.       Diagnose[]    diagnose ←
             ProposedDiagnosis(lstSymptoms)
20.       s ← symptoms [i]//choice of selection
21.       D ← diagnose [i]
22.       End
```

The above algorithm is a complete solution for acquiring the disease when different types of symptoms were available in the process. Symptoms are available in a list: {S_1, S_2 … S_k} which is a structure or class used to contain information about the symptoms used in the algorithm.

The recorded symptoms are added into a list for future for-casting further processes. The actual functionality of the algorithm starts from line no 4 where symptom s has been selected to run the process of learning and diagnosing certain disease. The choice of first symptom is most important as it will initiate the process and algorithm molds itself according to that symptom which has been selected so far. This process going on until some disease will

be identified with the "while" loop at line no 5 and ends at the line 21.

The type of symptom is checked at line no 8 and mode selection made by the user of which choice has been selected from the available symptom types and their relative results, the next option is to mark the confidence level, which starts from the min value of 10 and goes to max value of 100. The confidence level indicates the degree of authenticity over the selection made or answered the choices. The algorithm checks whether the value of confidence is greater than or equal to value of confidence associated with the symptom initially or not, if less than the defined percentage then the Result "R" value again set to "Normal" value because of its low degree of confidence otherwise the value will remain its value if it has selected as "Abnormal".

At line 14, all process which has been carried out so far will be saved / added in the "process" list along with symptom asked, value to be answered, Result "R" and the patient to whom symptom is being asked. When the symptom result is abnormal, it is recorded into the list of Symptoms to note all the symptoms which were caused as abnormal values and used to perform the diagnosing process in the lines.

At line no 18, the proposed symptoms are loaded against the last symptom asked. The proposedSymptoms function provides sorted array of symptoms associated with the last symptoms in term of their occurrences (having maximum association first). For this, it uses a structure with three attributes {FirstSymptom, SecondSymotom and Association Factor}. The "Association Factor" holds the value of association as integer between the two symptoms.

Similarly, the ProposedDiagnosis function provides the array of proposed diagnoses which may be intimation to actual diagnose as all the provided diagnoses are based on the probability of occurrence. The list of asked symptoms is provided and it checks all symptoms one by one in the available diagnoses whether it contain the particular symptom or not and calculates its probability accordingly. At line 19, the provided diagnoses are stored in the array called "diagnose".

After maintaining the proposed symptoms and diagnosis, one of the symptom is selected in the variable "s" on line no 20 for continuing the further process of diagnosing certain disease and a diagnose in variable "D" which is used in the while loop at line no 5 to terminate the process of diagnosing a disease on probability grounds. The process of determining carries on within the line of codes from 5 – 21 until unless some expected diagnose comes from the knowledge at line no 19.

The basic elements of the MDES are described in Figure 2. *'Patient'* element represents to which certain disease is to be diagnosed and it has some signs in the form of '*symptoms'*.

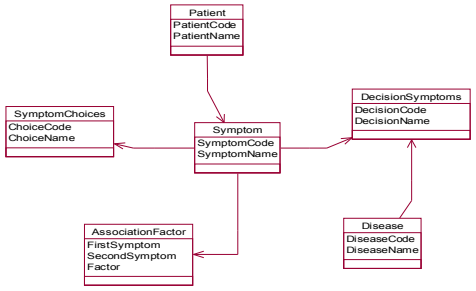

Figure. 2 MDES Structure

The person has some symptoms which show some disorder of internal and external examination systems about it. The symptom has different choices associated with it and Disease object contains all the diseases in its repository and it is associated with "DecisionSymptoms" to show their relation with the list of symptoms. The set of symptoms which are associated with a particular disease is stored in this object. The object "AssociationFactor" used to store the association of one symptom with respect to all symptoms available in Knowledge base (KB) as a list. In the process of diagnosing a probable disease in a patient, the system gone through a starting symptom, the system shows all the possible choices attached with it like Boolean, enumeration as described in previous section. The user selects one of his / her best choice with confidence level, the system proceed to next step by verifying the defined criteria and then, proposing the best symptoms associated with the symptom asked. It also creates the probability of diseases found and display accordingly.

Here we present two major panels for the description and discusses functionality in detail with the help of figures as shown under.

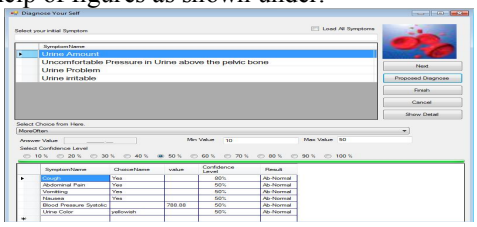

Figure. 3 MDES System

When a symptom has been selected, the system loads its complete detail regarding its type, range value or a choice selection along with the confidence level. The confidence level ranges from 10 % to 100% with the provision of the answer, if this level is below than 50%, the system will not

marked it as "abnormal" value while the status shows it as an abnormal value. All the symptoms having confidence level more than 50% and choice of answer value not in normal range, the system marked these symptoms as abnormal and includes them in the process of decision making. The first text field is used to write the different symptom The 'Next' button performs all the necessary process of selecting the next symptoms by calculating the association factor among the symptoms available in the KB. When range value symptom has been selected and its value lied in the certain range. So the field allowed the user to enter the value within the range specified by the min and max values besides field. The 'Proposed Diagnose' button allows checking the formed diagnosis so far during the process of decision making while answering the different symptoms towards the actual. The list in the bottom part of screen contains the resulted value entered by the user cum operator.

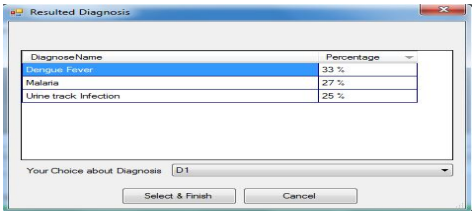

Figure. 4 Proposed Diagnoses

The Proposed Diagnose screen displays the list of diagnoses which are produced in the decision making process. The list shows two columns, one is 'Diagnose Name' to indicate the actual name of finding disease and the other column is 'Percentage' which indicates the probability in numeric form. E.g. "Urine Track Infection" has the value of 25 % in the list at number 3 (row wise).

3. Discussion

MDES system identifies diseases in the presence of symptoms. In order to illustrate its functionality, consider the following diseases with their relative symptoms.

Disease D_1 refers to Malaria and its symptoms are
S_1 →Vomiting, S_2 →Fever, S_3 → Cough,
S_4→Abdominal Pain, S_5→ Headache

Disease D_2 refers to Dengue Fever
S_1 → Vomiting, S_2 → Fever, S_3 →Cough,
S_6 →Nausea, S_7 → Muscle Pain

Disease D_3 refers to Bone Cancer
S_1→ Vomiting, S_3→ Cough, S_6→ Nausea,
S_8→ Fatigue, S_9→ Constipation

The above symptoms are registered in Base of Fact and they have different degree in all foresaid patient.

Scenario: D_1 ➔ S_1, S_2, S_3, S_4, S_5
D_2 ➔ S_1, S_2, S_3, S_6, S_7
D_3 ➔ S_1, S_3, S_6, S_8, S_9

The association factor among the symptoms with respect to each other for diagnosing a particular disease is described in the following table.

Table 3 Association among scenario symptoms

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	-	2	3	1	1	2	1	1	1
S_2	2	-	2	1	1	1	1	0	0
S_3	3	2	-	1	1	2	1	1	1
S_4	1	1	1	-	1	0	0	0	0
S_5	1	1	1	1	-	0	0	0	0
S_6	2	1	2	0	0	-	1	1	1
S_7	1	1	1	0	0	1	-	0	0
S_8	1	0	1	0	0	1	0	-	1
S_9	1	0	1	0	0	1	0	1	-

The association for each symptom is defined with all other symptoms available in KB and calculated by using simply the occurrence together in the past in any order during the process.

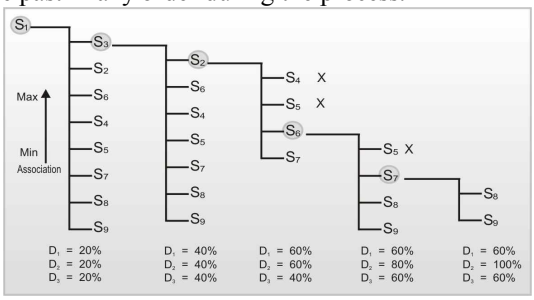

Figure. 5 Decision Process using association facto

Starting from S_1 symptom and after satisfaction of fulfilling the required criteria, the proposed algorithm loads all the symptoms which are associated with it in ascending order (having higher value first). One by one, the user selects its required option and gone through the process as listed in Figure 5. When the user progresses from one step to another, the systems records all the successful symptoms in its repository and tries to propose a disease list based on these symptoms so far. The support of each symptom in a disease is same and its value can be calculated as shown in lower part of the each list of the symptoms (see Figure .5). Similar case is repeated until required disease is found. At every step, it produces the associated symptom list in ascending order and calculates the probability for each disease based on asked symptoms. So, the successful sequence of symptoms asked so far is S_1, S_3, S_2, S_6 and S_7 and at this stage Disease D_1 has its max value of 100 and we can stop our procedure. The other minor diseases are also available with 60% value each. So proposed algorithm MDES is able to find the required disease by locating the symptoms

which had been asked through the process and other diseases as well.

4. Conclusion

MDES is a KBS which has been developed to ease the work of medical domain to make decision efficiently and reduce workload as well. The system allows them to start from a scratch but as long as its usage made, it became more efficient and learns accordingly.

In future, the system will able to produce treatments based on diagnosed diseases accordingly, to facilitate the paramedical staff in more efficient approach.

Acknowledgement

I am heartily thankful to my supervisors, especially from Dept. of Computer Science, Govt. College University, Lahore and Dept. of CS & E, University of Engineering & Technology, Lahore whose encouragement, guidance and support from the initial to the final level enabled me to develop this system.

Corresponding Authors:
Dr. Aslam Muhammad, Imran M. R
Department of CS & E, UET, Lahore, Pakistan.
Email: maslam@uet.edu.pk,
imranmrabbani@gmail.com

REFERENCES

1. Karl Anders Ericsson, Neil Charness, Paul J. Feltovich, Robert R. Hoffman, "The Cambridge handbook of expertise and expert performance", 2006, ISBN 052184097X, 9780521840972
2. Robert S. Engelmore, Edward Feigenbaum, "Expert Systems and Artificial Intelligence", Chap #1, Published: May 1993; WTEC Hyper-Librarian
3. Buchanan, B.G.; Shortliffe, E.H. (1984). "Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project", Addison-Wesley. ISBN 9780201101720.
4. Randolph A. Miller, "Quick Medical Reference QMR", University of Pittsburgh, CA, 1980
5. G. Octo Barnett, MD, Kathleen T. Famiglietti, Richard J. Kim, Edward P. Hoffer, MD, Mitchell J. Feldman, "DXplain on the Internet", MD Harvard Medical School, Laboratory of Computer Science, Massachusetts General Hospital, Boston, Massachusetts, 1998
6. GIDEON online, Medical Expert System, Retrieved From http://www.gideononline.com on Dated 21-05-2011, ISSN: 1938-6508
7. Pryor R.A, Gardner RM, Clayton PD, Warner H.R, "The HELP System", Department Of Medical Informatics, University of Utah, Salt Lake City. 1983, 7: 87-102
8. J Viktor and L Nada, "Classification Rule Learning with APRioRI-C", J. Stefan institute, jamova 39, 1000 ljubijana, Slovenia {Viktor. Jovanoski, Nada, Lavrac} ijw.wi, 2001
9. Ruth Mcguire, "Decision Making", Continuing Professional Development, The Pharmaceutical Journal (Vol. 269), 2007
10. Kamila Aftarczuk, Adrianna Kozierkiewicz and Ngoc Thanh Nguyen, "Using Representation Choice Methods for Medical Diagnosis Problem", Institute of Information Science and Engineering, Wroclaw University of Technology, Poland, 2006
11. János Fülöp, "Introduction to Decision Making Methods", Laboratory of Operations Research and Decision Systems, Hungarian Academy of Sciences, 2005
12. H Jochen, M Andreas, W Rudiger and G Ulrich, "A New Algorithm for Faster Mining of Generalized Association rules", Wilhelm Schickard institute, University of Tubingen, 72076 Tubingen, Germany, {hippj, myka, gucntzer} @ informatik.uni-tuebingen.de, 1998
13. Javeed M Zaki, P Srinivasan, O Mitsunori and Wei Li, "New Algorithms For Fast Discovery Of Association Rules", {Zaki, Srini, ogihara, li}@cs.rochester.edu, Computer Science Department, Rochester, New York 14627, Technical Report 651, July 1997
14. H Jochen, G Ulrich, and N Gholamreza, "Mining Association Rules; Deriving a Superior Algorithm by Analyzing Today's approaches", Wilhelm Schickard-Institute, University of Tubingen, Germany, {jochen. hippj, guentzer}@informatik.uni-tuebingen.de, C Daimler AG, Research & Technology FT3/AD, 89081 Ulm, Germany, 2000
15. A Rakesh, S Ramakrishnan, "Fast Algorithms for Mining Association Rules", IBM Almaden Research Center, 1994
16. Nevcihan Duru, "An Application of Apriori Algorithm on a Diabetic Database", Department of Computer Eng. University of Kocaeli, 41440, Izmit, Kocaeli, Turkey, nduru@kou.edu.tr, 2005

2/20/12