Medical diagnosis with Case Based Reasoning

Final Project Report - Introduction to AI

Instructors

Kári Halldórsson
kaha@ru.is

Stephan Schiffel
stephans@ru.is

Authors

Andri Már Þórhallsson
andrit14@ru.is

Ásgeir Þór Másson
asgeirm14@ru.is

Karl Ingi Karlsson
karlk13@ru.is

Table Of Contents

Introduction

The healthcare system is one of the largest expenditures of governments in most developed countries. With longer life expectancy and new opportunities to improve health it is likely that healthcare expenditure will increase. Medical professionals are an expensive and finite resource thus it is important to make their work as efficient as possible. In most emergency departments (EDs) today only a minimal amount of pre-evaluation has been done before the patient meets a doctor. It would therefore be helpful if part of the diagnostic process could be performed while the patient waits in the waiting room.

The group is comprised of three students in software engineering at a BSc level. One has previously completed a medical degree and worked as a doctor for four years, partly in the ED. For this project we did not have extensive patient data to work with. This is partly because of the way patient data has been accumulated and stored but also because of privacy issues. Most patient data is stored as semi-structured data which would need considerable parsing before it can be utilizes and even then there will be inconsistencies and missing data that increase the complexity of the problem. This approach is possible but well out of the scope of this project.

In this project we implemented an information gathering agent that utilizes the time from which a patient enters the ED of the hospital until the patient is seen by a medical professional. The agent will ask the patient questions interactively based on initial information such as age, gender and chief complaint and end with a list of diagnoses that are relevant for the patients symptoms.
To begin with the agent uses a database built only on our domain knowledge but with each case the agent will learn and the accuracy of the results will improve.

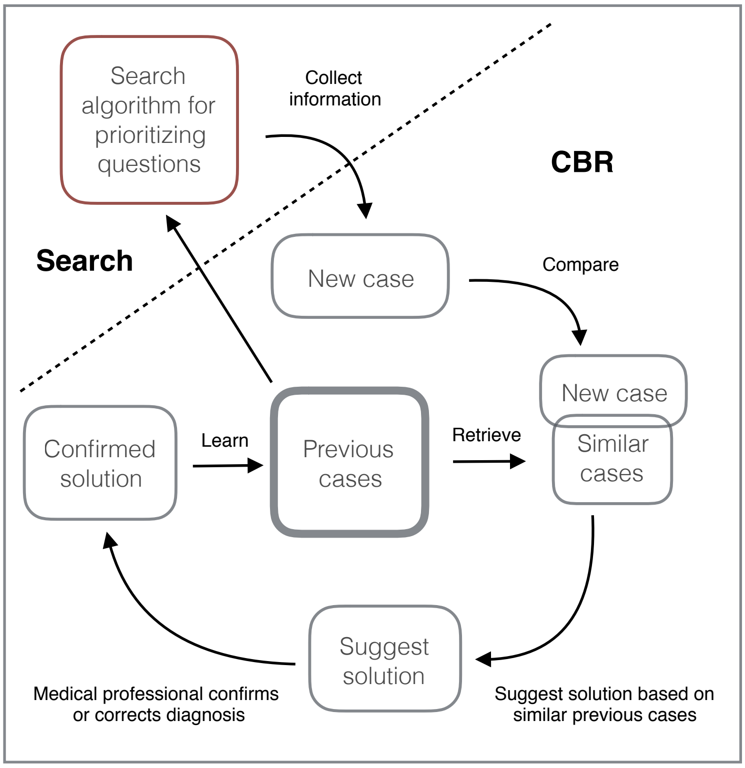
The Data

Given the time constraints of the project we decided to focus the database on one subspecialty of diseases, diseases that are typical for a general surgical resident to see at the ED. These are for instance appendicitis, diverticulitis, biliary colic and cholecystitis to name a few. We also added similar cases that are have some overlapping symptoms but are still of a different subspecialty. This was done to test whether the agent would provide a differential diagnosis list that crosses different subspecialties. Additionally we added one diagnosis from few different subspecialties to increase the amount symptoms that the agent could recognize.

We implemented a Python script that for every diagnosis, generated SQL commands to fill the database of 100 cases that roughly followed the age and gender distribution of the diagnosis in question. These insert commands were made of true values and symptoms for the given diagnosis but randomly assigned a few (10%) of the symptoms to a false value. This was done to give the data a more realistic distribution and allow the agent to use the age and gender in the search and CBR algorithms to some extent.

Using this approach it is easy to add to this implementation, for instance each subspecialty group could markup the most common diagnoses, giving typical symptoms and the age and gender distribution. These cases could then be removed when the database has sufficient real data to work with.

Implementation



After careful consideration and research we concluded that Case Based Reasoning (CBR) would be the best option for implementing an agent that can suggest possible diagnoses considering the limited data the agent will have to begin with.

That still leaves the problem of the vast amount of questions relating to the patient's symptoms. It would take quite a long time to ask all those questions and even so if the patient is presented with hundreds of questions it is likely that he will leave out some information.

Therefore the questions need to be carefully selected and prioritised so that every question is pertinent to the actual diagnosis, similar to how an actual medical professional retrieves a patient's history.

The Search Algorithm

To solve this we implemented a separate algorithm that rates each questions based on the information that has already been gathered about the patient. It uses the same database as the CBR algorithm.

To begin with the patient enters his age, gender, main symptom and whether he experiences pain. If he is in pain, he is asked to describe the pain in detail. This is done because pain is a very common symptom and usually a very strong indicator of what could be wrong. In some cases pain can also indicate a serious diagnosis and thus it is an important factor.

This initial information is submitted to the question search algorithm and it retrieves the most relevant question at each time. The question is then presented to the patient. This is repeated until the patient answers a few questions in a row with a negative response or if the patient decides that he has no more symptoms. If the question algorithm is not asking relevant questions the patient can enter symptoms of his own. This could happen for instance when the agent has not encountered a similar case.

The Case Based Reasoning

After having collected all the information needed from the patient the second algorithm uses Case Based Reasoning to find a list of possible diagnoses. It does this by first retrieving cases from the database that are similar to the current case based on age and gender values. This is done to reduce the amount of comparisons needed to build the differential diagnosis list. Then it uses a modified K-nearest neighbor algorithm which compares the current case to every retrieved case and yields a similarity index.

The similarity index is built by looking at five different types of symptoms, major and minor symptoms, pain description symptoms, pain location and referred pain location. This is our multidimensional feature space which can easily be expanded. Each type has a weight which can be manipulated. We define a minimum similarity and if the case in question exceeds that minimum it is added to the differential diagnosis list. This similarity limit can be manipulated to control the length of the differential diagnosis list and the relevance of the cases that appear.

We decided to limit the scope of the project to these five dimensions due to the time constraint of the project. However it would be interesting to add dimensions such as acuity, which could be weighted highly and thus promoting serious and time pressing diseases.

The list is outputted to the medical professional so that he can perform the necessary tests and confirm which diagnosis is correct or enter the actual diagnosis. If the diagnosis was on the differential diagnosis list the agent can revise the current case by mimicking the selected case. This has a twofold benefit. If the diagnosis can be made by having fewer symptoms then previously known by the agent it learns to be more precise. On the other hand it picks up new symptoms that are related to the disease that are possibly not currently known by modern medicine.

If the diagnosis is not know to the agent and therefore not on the differential diagnosis list the medical professional is prompted with a list of the current symptoms and can classify them into the appropriate dimensions and add a diagnosis.

The agent's ability to use previous cases to revise the current case and the option for adding new diagnoses enables the agent to learn from past successes and failures.

Test Results

It is difficult to test an agent used for aiding in medical diagnosis based on a small, manufactured dataset. Having the aim of this project in mind, namely to build a differential diagnosis list given a patients symptoms, age and gender, it is important to realize that a good differential diagnosis list includes all diagnoses that could be relevant within some abstract limit. We thus decided to test how many symptoms does the agent need so that the diagnosis in question finds it's way onto the differential diagnosis list, knowing that it is in the database.

The percentage values in the following tables represent how many of the total symptoms in each dimension were included in the test cases, for example the first value in the pain column is 39% which means that 7/18 pain symptoms were included in the test case.

Heart Attack - Typical age and gender

	Pain	Major symptoms	Minor symptoms	In differencial diagnosis
1	39%	100%	100%	✓
2	39%	100%	0%	✓
3	39%	75%	0%	✓
4	17%	75%	0%	✓
5	6%	25%	0%	✓
6	0%	75%	0%	✓
7	0%	50%	0%	✓
8	0%	25%	100%	✓
9	0%	25%	50%	✓
10	0%	25%	0%	x

Each test was run with multiple different symptoms

Heart Attack - Atypical age and gender

	Pain	Major symptoms	Minor symptoms	In differencial diagnosis
1	39%	100%	100%	✓
2	39%	100%	0%	✓
3	39%	75%	0%	✓
4	17%	75%	0%	✓
5	6%	25%	0%	✓
6	0%	75%	0%	✓
7	0%	50%	0%	✓
8	0%	25%	100%	✓
9	0%	25%	50%	80%
10	0%	25%	0%	x

Each test was run with multiple different symptoms

Cholecystitis - Typical age and gender

	Pain	Major symptoms	Minor symptoms	In differencial diagnosis
1	39%	100%	100%	✓
2	39%	100%	0%	✓
3	39%	75%	0%	✓
4	17%	75%	0%	✓
5	6%	25%	0%	✓
6	0%	75%	0%	✓
7	0%	50%	0%	✓
8	0%	25%	100%	✓
9	0%	25%	50%	✓
10	0%	25%	0%	x

Each test was run with multiple different symptoms

Cholecystitis - Atypical age and gender

	Pain	Major symptoms	Minor symptoms	In differencial diagnosis
1	39%	100%	100%	✓
2	39%	100%	0%	✓
3	39%	75%	0%	✓
4	17%	75%	0%	✓
5	6%	25%	0%	✓
6	0%	75%	0%	✓
7	0%	50%	0%	✓
8	0%	25%	100%	✓
9	0%	25%	50%	x
10	0%	25%	0%	x

Each test was run with multiple different symptoms

Another way to test the agent is to feed it with a case of a predefined diagnosis that is not in the database but that is known to be similar to diagnoses in the database. We decided to try this with the disease acute pancreatitis which is similar to a number of abdominal ailments known to the agent.

Acute Pancreatitis - Typical age and gender

	Pain	Major symptoms	Minor symptoms	Similar diagnoses
1	39%	100%	100%	✓
2	39%	100%	0%	✓
3	39%	75%	0%	✓
4	17%	75%	0%	✓
5	6%	25%	0%	✓
6	0%	75%	0%	✓
7	0%	50%	0%	✓
8	0%	25%	100%	x
9	0%	25%	50%	x
10	0%	25%	0%	x

Each test was run with multiple different symptoms

Equally important to the usability of the agent is to not insert diagnoses to the differential diagnosis list that are not relevant to the current case. This is a subjective matter and thus not easy to test. When running the above tests, it was our opinion that the differential diagnosis list was relevant. This could change with increasing the database size but that can easily be manipulated within the similarity index. For instance if the differential list becomes to large it can be pruned with increasing the similarity limit or if one aspect of cases is to dominant the weight of that dimension can be lowered. Additionally we tried to insert the full symptom list of an obscure disease not known to the database, but with minor similarities to diagnoses in the database. This did not produce a differential diagnosis list and prompted the medical professional to add a new diagnosis.

Existing Work

Artificial intelligence in medicine has been but a dream for many years. This is largely because the complexity of the problem. Multiple diseases with multiple presentations depending on environmental, genetic or even random factors that many are unknown. Medical scientific literature is growing with such velocity that the hours in a day would not allow for just reading new papers let alone act upon the new knowledge. An AI that can incorporate all of this is an huge endeavor but agents like IBM Watson might be moving in the right direction. Until then there is need to make use of AI to tackle more constraint problems.

In our opinion Bayesian networks would be a very suitable for diagnostics. This has been applied in research but usually confined to a small area in medicine. What hinders this implementation is the lack of accurate statistical data about the relations of specific symptoms and diseases.

Case Based Reasoning (CBR) is able to reason with a relatively small amount of data and will incrementally grow over time as the agent gets more experience and becomes more accurate. It uses the experience from earlier cases, estimates how similar they are and chooses a solution based on the most similar cases. CBR can be reasonably successful at solving problems at the boundaries of it’s knowledge base compared to other approaches.

This approach has been used before for medical diagnosis in research. Each of these projects have focused on one area of medicine, for instance in heart diseases (CASEY), lung disease (BOLERO) and psychiatry (SHRINK).

There are some open source libraries (e.g. freeCBR, myCBR) and even programs (e.g. Kolibri) that have CBR functionality. We decided to implement our own version because we wanted to get in-depth experience designing and improving our implementation of this algorithm and the ability to customize every aspect of it.

Continued Work

This project is a mere prototype and there are multiple aspects that need attention before the agent can be useful. The addition of a few more dimensions to the feature state space would be helpful. A dimension for acuity is essential for an ED agent, one for prior medical history as well as one for medication, to name a few. The database would need to be expanded, but this is not as daunting of a task as it might seem. The main objective of an ED is to be able to find dangerous and acute disease within a group of patients as well as sorting patients into appropriate groups, but not to yield a final diagnosis for every patient. Given that acutely ill patients usually don't wait in the waiting room, this group does need to be prominent in the initial database. Thus the initial database would need to represent common dangerous diseases as well as the most common diseases in each medical field. This would ensure that the agent would recognize the most common symptoms because each medical field focuses on different set of symptoms.

Many problems might arise when the system has matured. Factors like the size of each case will affect the memory usage of the system. In this project we ignored this fact, we focused rather on delivering a working prototype. Another important factor is deciding the number of cases to keep and how to maintain the database. There is a obvious tradeoff between accuracy given a large number of cases and a slow retrieval of cases. After the system has matured it would also be interesting to manipulate weights of different dimensions and analyze how that effects the accuracy and precision of the agent.

Because we have experience with SQL we decided to use Postgres to store the previous cases. That turned out to be a poor decision because of the object-relational impedance mismatch. It required a massive amount of work to make it functional and even then it will not scale very well. Therefore we will look into non-relational databases when proceeding with this project.

An agent like this would gather large amounts of data in logic format about symptoms. This data combined with the actual diagnosis could be used to calculate the necessary statistical data for a Bayesian Network implementation. That implementation could then give doctors and other medical professionals more accurate and detailed information about what could possibly be wrong with the patient and how likely it is as well as assisting with medical research.

Links to Supporting Materials

Case Based Reasoning References

The University of Nottingham lecture slides
http://www.cs.nott.ac.uk/~pszrq/files/9FAICBR.pdf

A Tutorial on Case-Based Reasoning by Julie Main, Tharam Dillon and Simon Shiu
http://www4.comp.polyu.edu.hk/~csckshiu/pdf/shiu01scbrb2.pdf

Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches by Agnar Aamodt and Enric Plaza
http://ibug.doc.ic.ac.uk/media/uploads/documents/courses/CBR-AamodtPlaza.pdf

Bayesian Networks in Medicine

A Bayesian Network Model for Diagnosis of Liver Disorders by Agnieszka Onisko, Marek J. Druzdzel and Hanna Wasyluk
http://www.pitt.edu/~druzdzel/psfiles/cbmi99a.pdf