



NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY  
DEPARTMENT OF COMPUTER SCIENCE

## Assignment 4

**TDT4173: Machine Learning and Case-Based Reasoning**

*Submitted by*  
**Vittorio Triassi**

October 2019

# 1

## Theory

### 1.1 What characterises case-based reasoning (CBR) methods? How are they different from other machine learning approaches?

*Case-Based Reasoning* (CBR) can be defined as both a “cognitive approach” and an “engineering approach” that aims at solving problems by using previous knowledge. CBR has been formally defined through four steps that can be summarized as follows:

1. *Retrieve*: given a problem to be solved, the most similar problems (previously solved) are retrieved.
2. *Reuse*: starting from the retrieved solutions, we try to reuse them to solve our problem. Since there might be differences, adaptation is necessary most of the times.
3. *Revise*: we test the solution and understand its effectiveness on the target problem.
4. *Retain*: if we came up with the right solution, we store it in the case base memory.

The main difference between CBR approach and machine learning approach finds its roots in the way they generalize the model. In fact, in CBR we have what is called a “lazy learner”. In contrast to that, in machine learning we have an “eager learner”. More specifically, in ML we have to be able to predict something after the model is already trained. A neural network for example, is trained in order to perform a specific classification task. This means that if we provide it test examples from another domain, it will not be able to correctly generalize. On the other hand in CBR, the solution is adjusted directly on the test examples. By doing so, it is possible to consider richer domains.

## 1.2 Discuss (some of) the ways in which cognitive science has influenced CBR.

Concepts such as *experience*, *memory* and *analogy* are really important when talking about CBR [3]. Humans have something called “episodic memory” that is nothing but the collection of past experiences that took place at a particular time [5]. In fact, thanks to such experiences, humans are able to take decisions, hopefully with better results than if they would not have any previous knowledge. CBR exploits this cognitive aspect to address new problems. In this way though, it almost seems that reasoning strictly depends on the ability of remembering the past and the action of *learning* seems directly related to it. In any case, it is really interesting to see how much interdisciplinarity there is when dealing with CBR. There have been a lot of theories regarding the use of experiences in understanding, problem solving and learning [3]. Some of them lead us to consider what is called *analogical reasoning*, that is a way of thinking based on the fact that since two concepts share something in some respects, they will eventually share something else in other respect [2]. Other works have shown how analogical reasoning has been particularly interesting in cognitive science and the big impact a few models have had on CBR.

## 1.3 Methods to evaluate the degree of similarity between two cases are essential in CBR. What is the difference between surface similarity and structural similarity? Give some examples for each approach.

In CBR, when we need to evaluate the similarity between two cases, we can start considering a case in terms of its surface features. These, are generally provided with the case itself and are represented in the form of *attribute-value* pairs. If we have more complex structures instead, we are interested in their structural similarity.

In terms of computation, structural similarity is more expensive but it is also the one that retrieves more relevant cases. In the case of surface similarity, this is computed by using a similarity measure and is represented through a real number in  $[0, 1]$ . A typical approach would be retrieving the  $k$  most similar cases (e.g.  $k$ -NN) [1].

It is worth mentioning that in surface similarity, a CBR system can retrieve the  $k$  most similar cases with a complexity  $O(n)$ , with  $n = \#cases$ . This might not be good in practice though. Another approach to reduce computational time is when cases are stored in memory based on similarity among each other. In this case, a binary tree is used and it is called *k-d tree*. The idea is that we perform divisions of the dataset in smaller sets. On the other side, with structural similarity we can retrieve more cases at the expenses of computational time. In order to narrow the extra time down, we can combine the surface and the structural similarity in a 2-step retrieval as proposed in MAC/FAC.

## 1.4 Explain how the similarity between cases can be measured when cases are made up of attributes with different data types. Give an example of how this can be done.

When interested in measuring the similarity between two cases, we need to take into account the different data types of the several attributes we have. In fact, each attribute can be made up of *unordered* symbols, *ordered* symbols, *taxonomies* or *integer/real* values. In such cases, we talk about heterogeneous spaces and these have to be carefully modelled in order to get any useful insight. Thanks to the *Local-Global* principle, it is actually possible to combine *local similarities* by using an *amalgamation* function to compute a *global* similarity. After we have modelled the local measures according to our preferences we can use an amalgamation function such as the *weighted sum*. More specifically, we can assign different weights to our features depending on their importance and then we compute the similarity. Something that sometimes is also useful to perform is the normalization of our values to compute the weighting more precisely. We should also pay attention to possible missing or null attribute values. In the event that we have attributes that are not contributing to the computation, we just pretend not to have them and for this reason they will be excluded.

## 1.5 What are knowledge containers in the context of CBR? Give a brief explanation of the different containers.

The *knowledge containers* emphasize the idea that CBR is a knowledge-based system that divides knowledge in several parts. In CBR we have four knowledge containers [4] and they can be described as follows:

1. *Vocabulary container*: it is necessary for any system. Without this container, we could not talk about any concept because its abstraction would not be defined otherwise. Thanks to the vocabulary, we are actually able to describe the knowledge.
2. *Similarity container*: it consists of all the information we need in order to perform any similarity measure that allows us to understand the degree of similarity between two cases. The similarity is very important because it is the means by which the retrieval becomes possible.
3. *Case Base container*: in the following container, there are all the cases previously solved. They may have been designed ad-hoc, or they might have been the result of variations of other cases. In any case, it is here that we check when trying to address new problems.
4. *Adaptation container*: here the knowledge can be used to adjust cases already solved to address new problems. In fact, it might happen that two cases are not equal and adaptation has to be performed. Adaptation is the process of modifying an existing solution trying to adjust it to a new context.

# 2

## Practical

### 2.1 Case Modelling

a) Creation of the concept *patient*

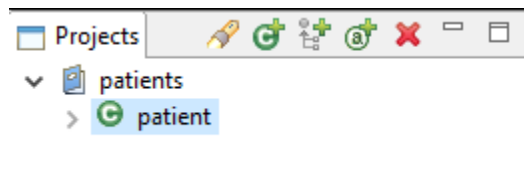


Figure 2.1: Patient concept

b) “Patient” has the following attributes:

1. *age*: Integer (*0-110*)
2. *diagnosis*: Symbol (allowed values: *ashtma*, *collapses*, *cough*, *fever*, *thoracic pain*)
3. *name*: String
4. *sex*: Symbol (allowed values: *M*, *F*)
5. *sleep\_quality*: Symbol (allowed values: *low*, *medium*, *high*)
6. *treatment*: Symbol (allowed values: *blood test*, *bronchodilator*, *cough medicine*, *paracetamol*, *radiography*, *sugar*). Multiple options are allowed.
7. *weight*: Float (*0.0 - 180.0*)

- c) For the purpose of the exercise, 11 instances of the patient concept have been created.

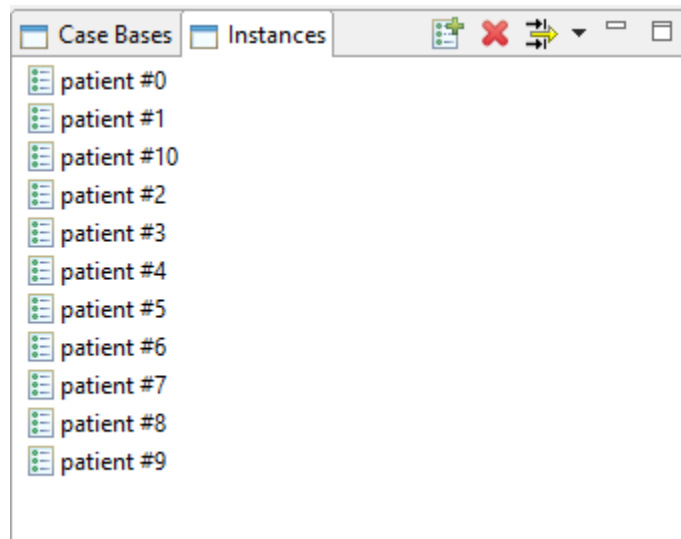


Figure 2.2: Patient instances

- d) Here, two of them are shown.

Instance information		
Name	patient #0	
Attributes		
age	31	Special Value: <a href="#">none</a>
diagnosis	thoracic pain	<a href="#">Change</a> Special Value: <a href="#">none</a>
name	Delphine Geist	Special Value: <a href="#">none</a>
sex	F	<a href="#">Change</a> Special Value: <a href="#">none</a>
sleep_quality	medium	<a href="#">Change</a> Special Value: <a href="#">none</a>
treatment	blood test radiography	<a href="#">Add</a> <a href="#">Remove</a> Special Value: <a href="#">none</a>
weight	58.5	Special Value: <a href="#">none</a>

Instance information		
Name	patient #4	
Attributes		
age	23	Special Value: <a href="#">none</a>
diagnosis	ashtma	<a href="#">Change</a> Special Value: <a href="#">none</a>
name	Luke Brumer	Special Value: <a href="#">none</a>
sex	M	<a href="#">Change</a> Special Value: <a href="#">none</a>
sleep_quality	low	<a href="#">Change</a> Special Value: <a href="#">none</a>
treatment	bronchodilator	<a href="#">Add</a> <a href="#">Remove</a> Special Value: <a href="#">none</a>
weight	75.5	Special Value: <a href="#">none</a>

Figure 2.3: Examples of instances

## 2.2 Case Retrieval

- e) In Figure 2.4 is shown the global similarity measure created for the *patient* concept. The attributes *name* and *treatment* have been set to *false*, while the others have been properly weighted. The reason why they both are false is that they do not have to contribute to the weighted sum since they simply provide information that will be useful later for the retrieval.

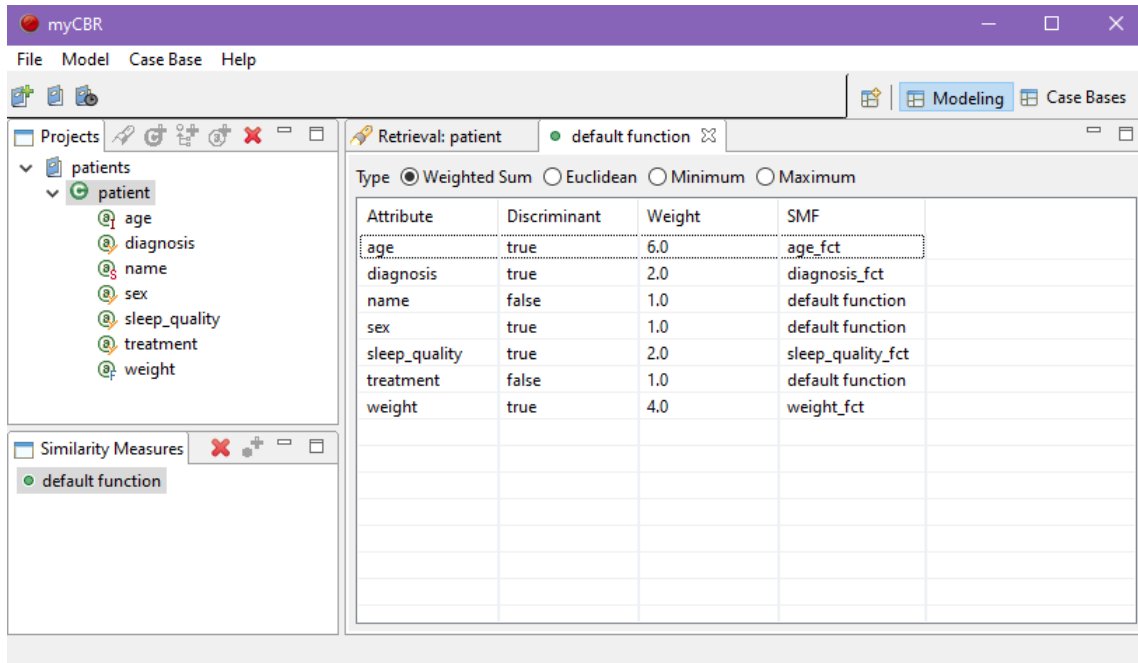


Figure 2.4: Global similarity measure for *patient*

In addition to that, a few other local similarity measures have been created. They aim at setting the sensitivity of the attribute *weight* when considering new cases or the relations among different diagnoses as shown in Figure 2.5.

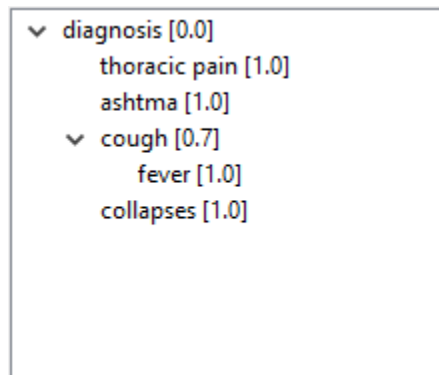


Figure 2.5: Relations among diagnoses

In Figure 2.6 is shown the similarity function for the attribute *weight*. The similarity in this case would be equal to 1 if we have an exact match, otherwise it linearly goes towards 0. To do so, we have set the *polynomial* with a value of 1. To model different behaviours, other two functions have been created and they are respectively: *weight\_fct2* and *weight\_fct3*. These functions are nothing but polynomial of degrees two and three. As far as the attribute *age*, this has been properly modelled with a polynomial of degree two in *age\_fct*. The reason is that we make the assumption that there is a relation between symptoms and age more than just linear.

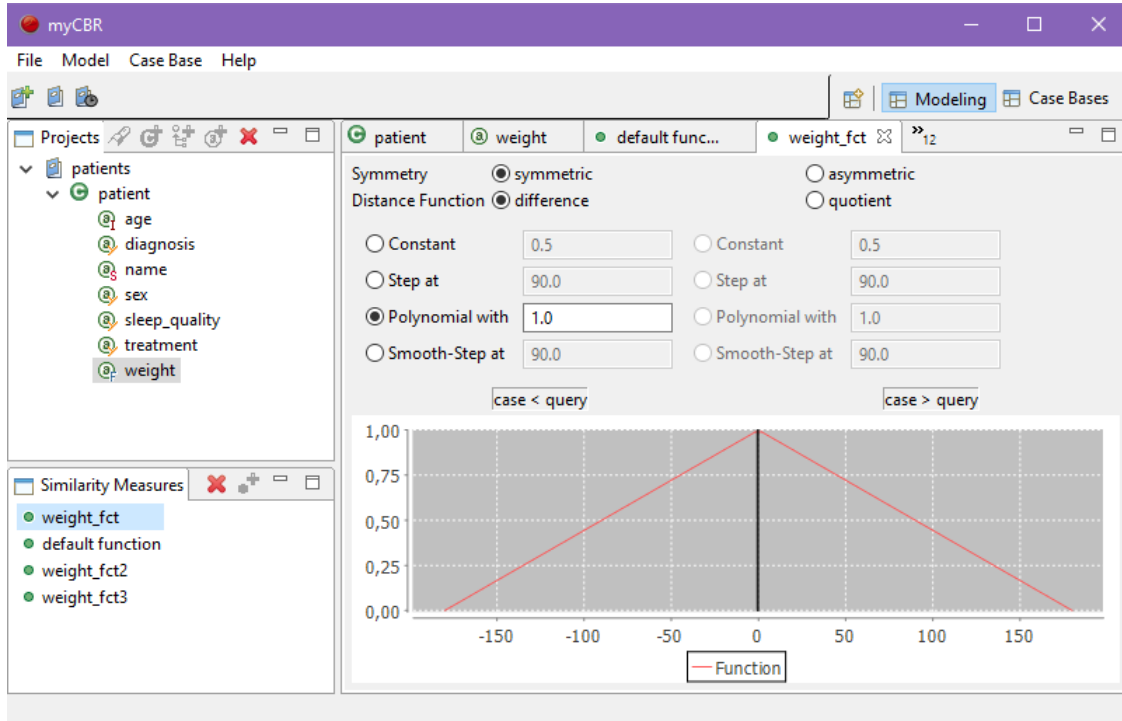


Figure 2.6: Local similarity measures for the attribute *weight*

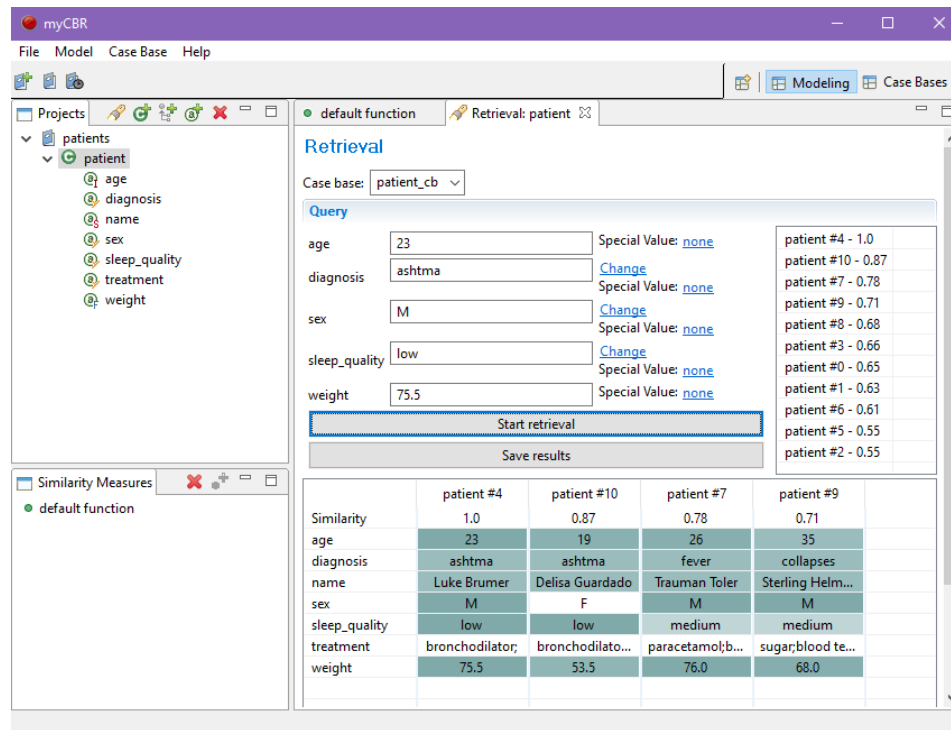
As regards to the attribute *sleep\_quality*, this has been adjusted in the following way:

● default function	Ⓐ sleep_quality	● sleep_quality_fct	⌘
Symmetry ● symmetric ○ asymmetric			
	high	low	medium
high	1.0	0.25	0.5
low	0.25	1.0	0.5
medium	0.5	0.5	1.0

Figure 2.7: Sleep quality function



f) Results obtained after having issued 5 queries:



myCBR

File Model Case Base Help

Projects

- patients
  - patient
    - age
    - diagnosis
    - name
    - sex
    - sleep\_quality
    - treatment
    - weight

Similarity Measures

- default function

Retrieval: patient

Retrieval

Case base: patient\_cb

Query

age: 23 Special Value: none

diagnosis: ashtma Change Special Value: none

sex: M Change Special Value: none

sleep\_quality: low Change Special Value: none

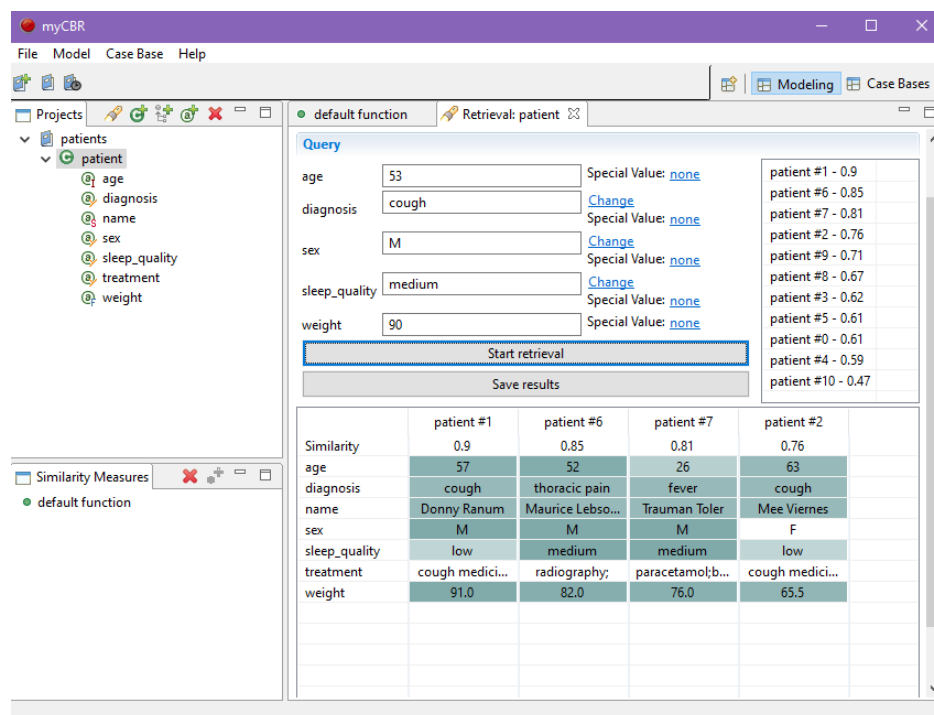
weight: 75.5 Special Value: none

Start retrieval

Save results

	patient #4	patient #10	patient #7	patient #9
Similarity	1.0	0.87	0.78	0.71
age	23	19	26	35
diagnosis	ashtma	ashtma	fever	collapses
name	Luke Brumer	Delisa Guardado	Trauman Toler	Sterling Helm...
sex	M	F	M	M
sleep_quality	low	low	medium	medium
treatment	bronchodilator;	bronchodilato...	paracetamol;b...	sugar;blood te...
weight	75.5	53.5	76.0	68.0

Figure 2.8: First query



myCBR

File Model Case Base Help

Projects

- patients
  - patient
    - age
    - diagnosis
    - name
    - sex
    - sleep\_quality
    - treatment
    - weight

Similarity Measures

- default function

Retrieval: patient

Retrieval

Case base: patient\_cb

Query

age: 53 Special Value: none

diagnosis: cough Change Special Value: none

sex: M Change Special Value: none

sleep\_quality: medium Change Special Value: none

weight: 90 Special Value: none

Start retrieval

Save results

	patient #1	patient #6	patient #7	patient #2
Similarity	0.9	0.85	0.81	0.76
age	57	52	26	63
diagnosis	cough	thoracic pain	fever	cough
name	Donny Ranum	Maurice Lebso...	Trauman Toler	Mee Viernes
sex	M	M	M	F
sleep_quality	low	medium	medium	low
treatment	cough medici...	radiography;	paracetamol;b...	cough medici...
weight	91.0	82.0	76.0	65.5

Figure 2.9: Second query

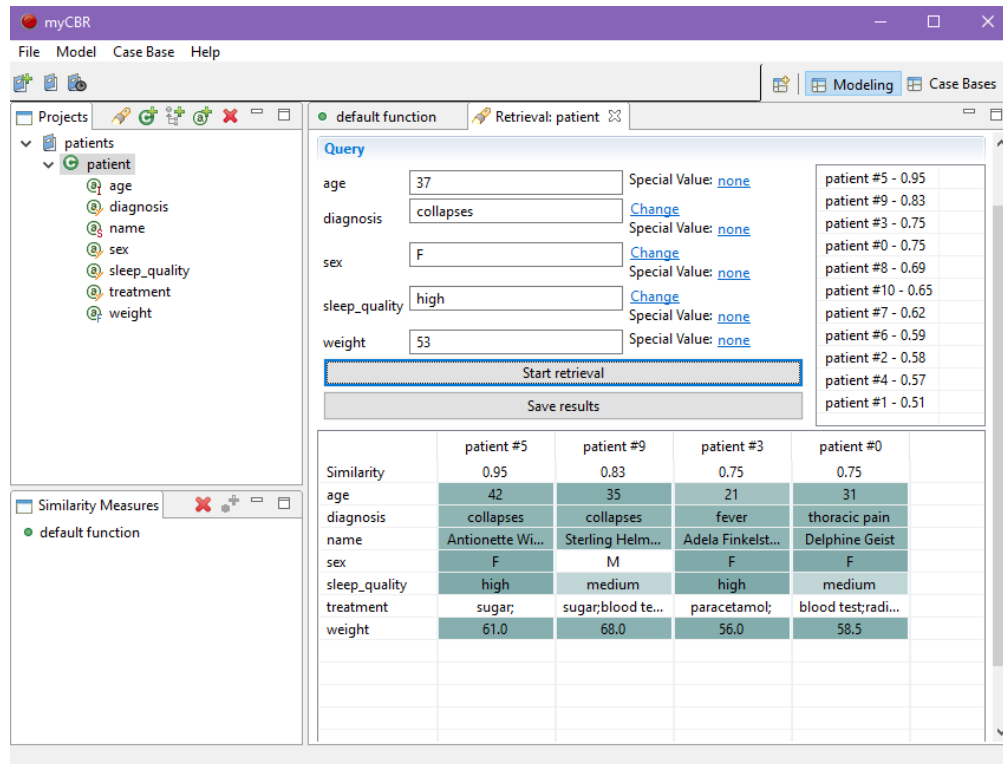


Figure 2.10: Third query

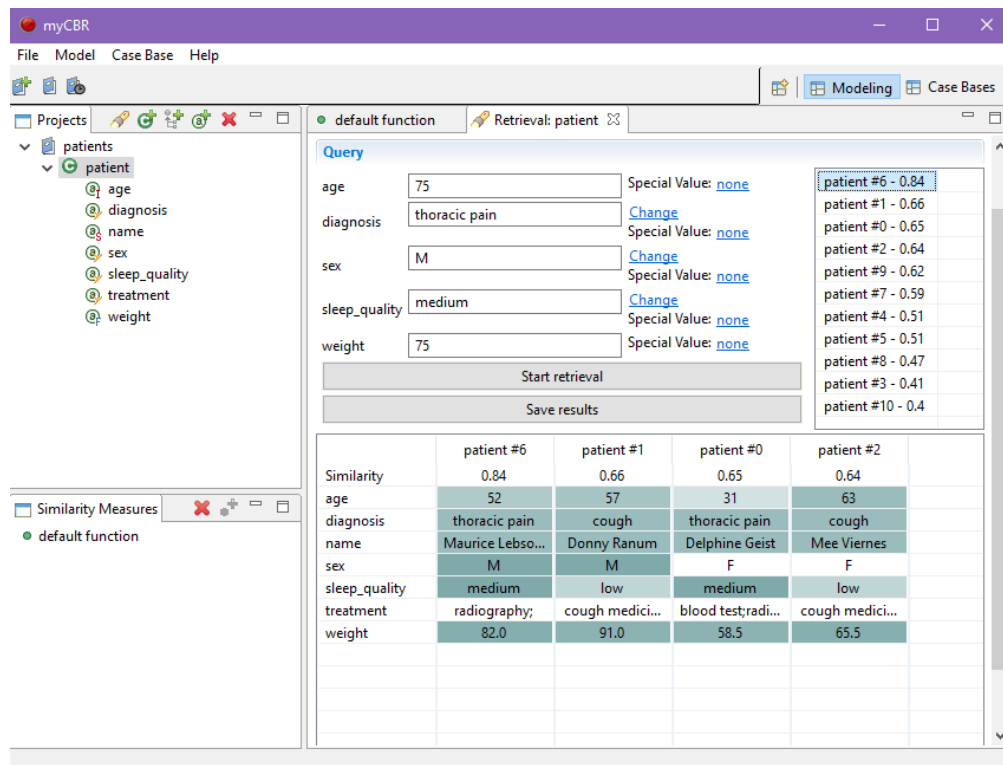


Figure 2.11: Fourth query

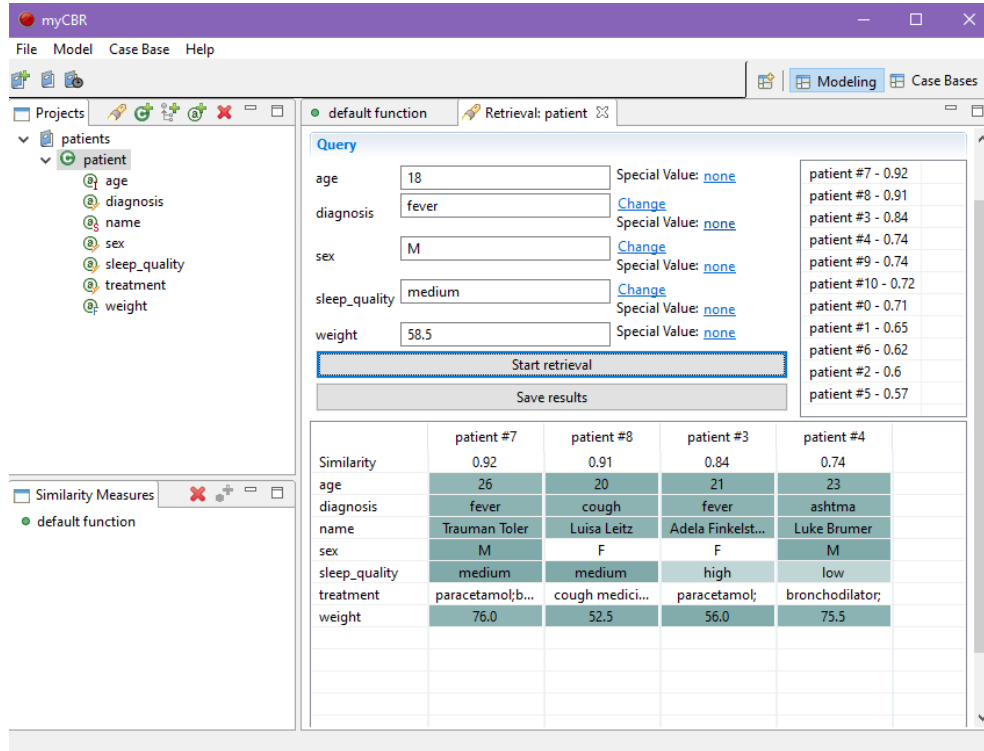


Figure 2.12: Fifth query

The queries issued and their results do not disagree with what we could expect. This means that the way we have weighted the attributes, reflects quite closely the instances previously stored in the CBR system and as a consequence, the real world scenario. It is worth to point out that 11 instances are probably too little to have a very performing system. In fact, the more cases we have access to, the more precise we can be when proposing a treatment to the new patients. Also, it is quite important to store the cases with consistent information. For the sake of this task, we have tried to list a few symptoms followed by a couple of possible treatments. Since we are not domain experts, the proposed matches (diagnosis - treatment) shall not be held reliable.

As far as an interesting query, we would like to show something that really points out the sensitivity of the CBR system. As shown in Figure 2.13, we get very low values for the similarity measure. That is because we tried to test as a new case, something that the CBR system had never seen before. Something that is quite uncommon. In fact, it can not provide any useful feedback and this is quite clear because the results proposed, show all different treatments.

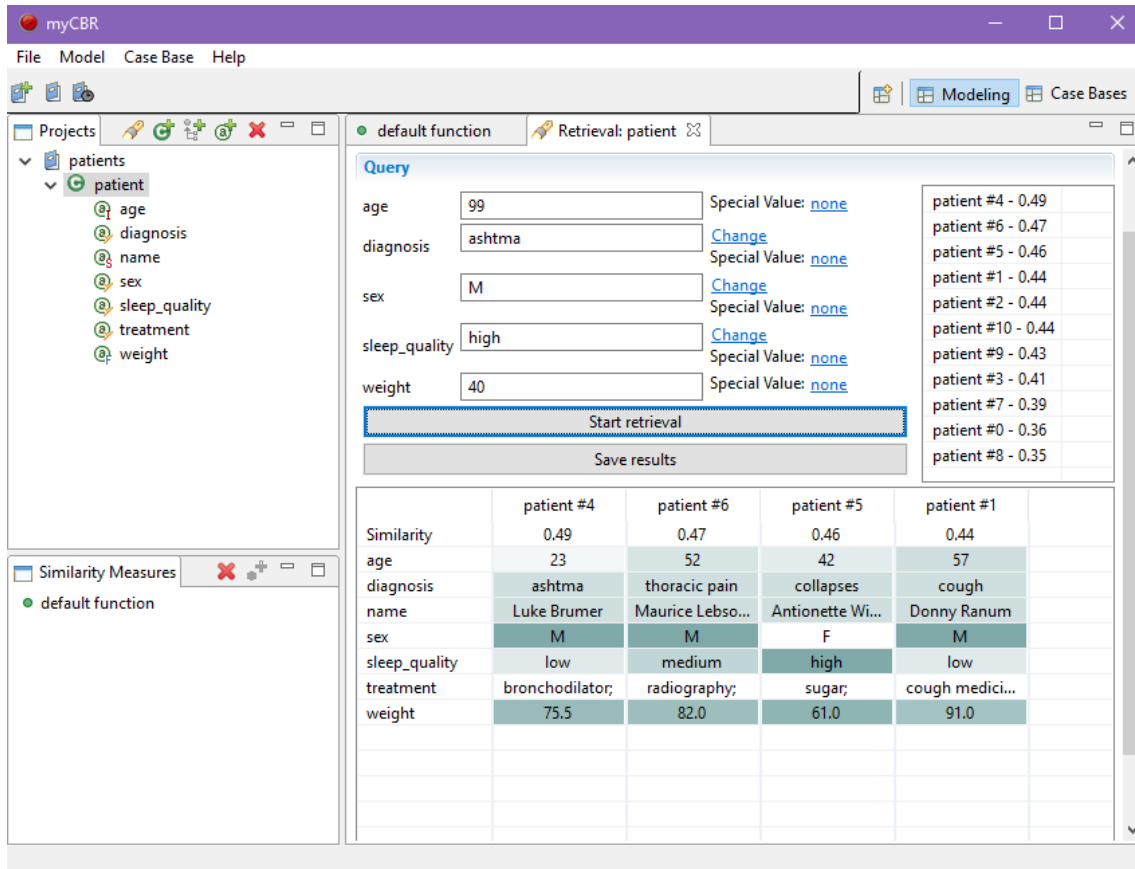


Figure 2.13: Interesting query issued

g) In the previous paragraph we have performed the retrieval step by using the *patient* concept. Thanks to CBR, it is actually possible to enrich our knowledge about a specific problem. If we want to schematically list the whole cycle, we would come up with something like:

1. *Retrieval*: we retrieve the cases of other patients stored in the CBR system based on our previous knowledge on patients.
2. *Reuse*: depending on the results we obtain after the query is issued, we try to use the treatment that is *more likely* to be good for our new patient (taking care of possible adaptations).
3. *Revise*: the solution is then evaluated by domain experts and eventually accepted.
4. *Retain*: the solution (treatment) proposed is defined good (or not) to solve the problem and the case is added to the other cases. This means that the system has learned a new problem.

There are several applications that might use the concept *patient* for further analyses. For instance, we might collect data of patients to understand which are the most common diagnoses at the emergency room. So the hospital can allocate the

right number of specialized doctors and nurses. Or we might think about performing analyses to understand the healthcare costs for a specific region/country based on the treatments they provide. In this and many other cases, the important thing is just to have the right data. After having collected data, CBR systems are able to exploit their knowledge thanks to tools like myCBR, allowing us to make quite accurate recommendations.

## References

- [1] Ramon Lopez De Mantaras, David McSherry, Derek Bridge, David Leake, Barry Smyth, Susan Craw, Boi Faltings, Mary Lou Maher, MICHAEL T COX, Kenneth Forbus, et al. Retrieval, reuse, revision and retention in case-based reasoning. *The Knowledge Engineering Review*, 20(3):215–240, 2005.
- [2] Wendelin Küpers. *Analogical Reasoning*, pages 222–225. Springer US, Boston, MA, 2012.
- [3] Michael M Richter and Agnar Aamodt. Case-based reasoning foundations. *The Knowledge Engineering Review*, 20(3):203–207, 2005.
- [4] Michael M Richter and Rosina O Weber. *Case-based reasoning*. Springer, 2016.
- [5] Wikipedia. Episodic memory.