Klara Schlüter
Machine Learning
NTNU - Department of Computer Science

October 15, 2019

ASSIGNMENT 4 — Case-Based Reasoning

1 Theory

1. Case-Based Reasoning (CBR) is a machine learning approach that is based on one model of how problem solving in a human mind works. As opposed to many other machine learning methods which use the training set to compute a target function that is then used to classify new examples (eager learning), in CBR, the training examples are each stored as a case. For classification, these "memories of previous experiences", the case base, are searched for experiences similar to the new case, and solution is computed reusing (adapting) the retrieved instances (lazy learning).

2. First of all, CBR is both a model for human problem solving (cognitive approach), that was introduced by cognitive science, and a machine learning method based on this understanding of intelligence (engineering approach). By remembering earlier experiences (problems and solutions), we humans find similar experiences and try to adapt the solutions to the current problem. CBR implements this model for intelligent problem solving with earlier cases. So CBR is not only influenced by, but, among other disciplines, based on cognitive science.

   In the following, I will discuss two specific aspects of the cognitive understanding that have been picked up in CBR: episodic memories and learning. Firstly, intelligence is not only based on a semantic understanding of the world and a memory storing it, but also on a so-called *episodic memory*. We remember previous experiences and how those turned out. This episodic memory is implemented in CBR as the *case base*. In the case base, known cases are stored along with everything know about. Secondly, an intelligent mind *learns* from their experiences. Having experienced that we burn our finger when touching a hot mug, we hopefully do not touch it next time, and therefore have learned not to touch hot things. In CBR, this learning from experiences is implemented as the revise (and even more importantly) *retain* part of the approach: with every newly developped solution, that is revised and judged as good enough, the corresponding problem can be stored as a new case in the case base (or adaptions can be made to existing cases).

3. *Surface similarity* depends on surface features, for example the color or the height or the weight of an object, described by standard value types like `symbol` or `float`. The surface similarity is therefore flexible to different case representations. Examples are determining the similarity of two cars by their color, size, weight and speed, or the similarity of two trees by their height and their age.

   Structural similarity takes the structure of the cases that are to be compared into account and therefore is specialised for a certain representation. The similarity is determined by very domain specific knowledge. Structural similarity for example is measuring the similarity of two humans by their way of understanding and learning as well as their social behaviour, or similarity of tectonic and geologic situation to predict earthquakes.

4. If the cases have attributes of different types, we can use a specific local similarity function for each attribute and then combine the results by using a suitable amalgamation

function, the global similarity. More concret: For each attribute $i \in \{1, ..., n\}$ of type T_i we use a local similarity

$$sim_i : T_i \times T_i \to [0, 1]$$

to compare the attribute values. A global similarity measure

$$F : [0, 1]^n \to [0, 1]$$

accepts the results of the local similarity measures as input and combines them to a single similarity (amalgamation).

For example let a case describe a tree with the attributes *height* of type `float` and *leaves* of type `symbol`, where $height \in [0,\ 50]$ and $type \in \{leaves,\ needles\}$. A possible local similarity for the *height* would be the normalized difference, while a simple lookup table is fitting for the leaves.

$$sim_{height} : \texttt{float} \times \texttt{float} \to [0, 1],\ sim_{height}(x,\ y) = \frac{|x - y|}{50}$$

sim_{leaves}	**leaves**	**needles**
leaves	1	0
needles	0	1

The global similarity could be a weighted sum, weighting the leaves higher since they are more important for similarity:

$$F : [0, 1] \times [0, 1] \to [0, 1],\ F(l,\ h) = 0.8 \cdot l + 0.2 \cdot h$$

Using this similarity measure, the comparison of a query tree with $height = 42.5$ and *leaves* and a case tree with $height = 32.5$ and *needles* yields

$$0.8 \cdot\ sim_{leaves}(leaves,\ needles) + 0.2 \cdot\ sim_{height} = 0.8 \cdot\ 0 + 0.2 \cdot \frac{|42.5 - 32.5|}{50} = 0.04$$

5. The collection of knowledge needed to perform CBR is split up in four containers, that group a specific kind of knowledge. The four knowledge containers are

   - **Case Base**: all previous experiences are stored as cases in a special data base called case base.
   - **Similarity Measures**: the functions that assign a query case and a case from the case base to a certain value representing their similarity. We need this for the retrieval of similar cases.
   - **Adaptation Knowledge**: the knowlegde we need to adapt the retrieved cases solutions to get a solution for the query case. It is expressed in rules.
   - **Vocabulary**: The vocabulary needed to express cases, attributes and rules that are contained in the previous containers, for example the concepts and terms.

2 Practical

Case Modelling

Imagining a clinic measuring temperature, heart rate and weight of all new patients, I created the concept `PATIENT` with 6 attributes.
As required for the exercise:

- `name` – a purely descriptive string.

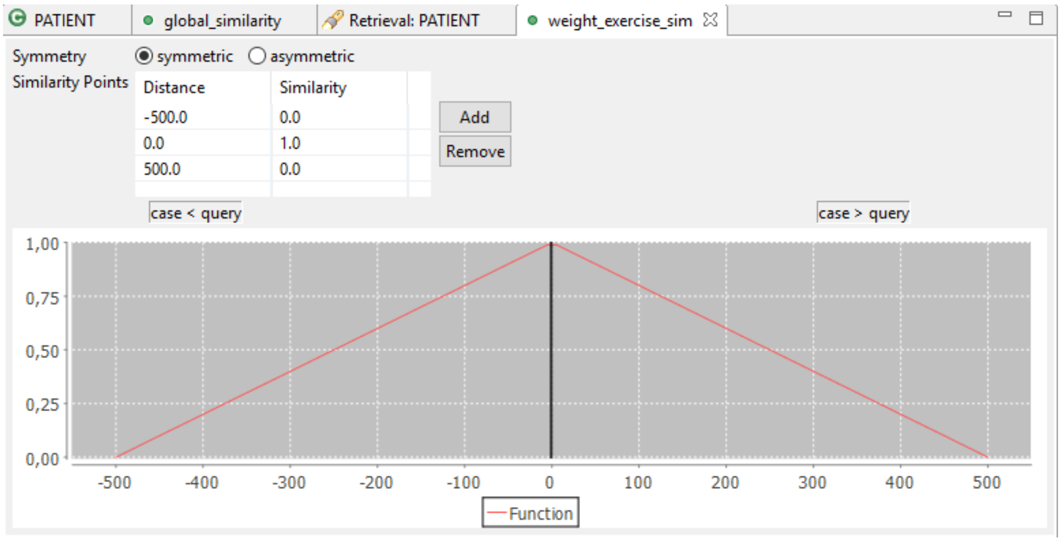


Figure 1: The similarity function for the attribute `weight`.

- `weight` – a float value representing the weight of the patient.
- `sleep_quality` – a symbol type with *low*, *medium* and *high* as allowed values.

Additionally:

- `birthday` – a date representing the patients birthday.
- `heart_rate` – an integer representing the patient's heart rate.
- `temperature` – a float value representing the patient's weight.

For this concept, I created a case base `Patienten` with 10 instances of a `PATIENT`. Screenshots of three instances can be found in the appendix.

2.1 Case Retrieval

Similarity Measure

In the project file, you will find one similarity measure each for the attributes `sleep_quality`, `heart_rate` and `temperature`. For the attribute `birthday`, which is of type `date`, *myCBR* wasn't able to create a similarity measure (when choosing the type of the function, no options were displayed). So unfortunately, there is no similarity measure for the birthday (the age) of patients. For the attribute `weight`, I provide several similarity measures. The one named `weight_exercise_sim` (see figure 1) is the one the assignment sheet asks for and the one included in the global similarity measure.

The global similarity measure is a weighted sum of the single similarities (figure 2). The sleep quality has the most influence with a weight of 2.0, followed by weight and temperature with a weight of 1.0. The attribute that is taken into account the least is the heart rate with a weight of 0.5.

Retrieval Queries

In this section, I am going to show the results of five retrievals. Unfortunately, maybe due to a bug in *myCBR*, the retrieval results for my retrievals weren't displayed in the program as nicely as in the demo video: the output shown in the program can be seen in figure 3. So I saved the query results as a csv-file and will include screenshots of the resulting table together with a screenshot of the query and the similarity order of the cases.

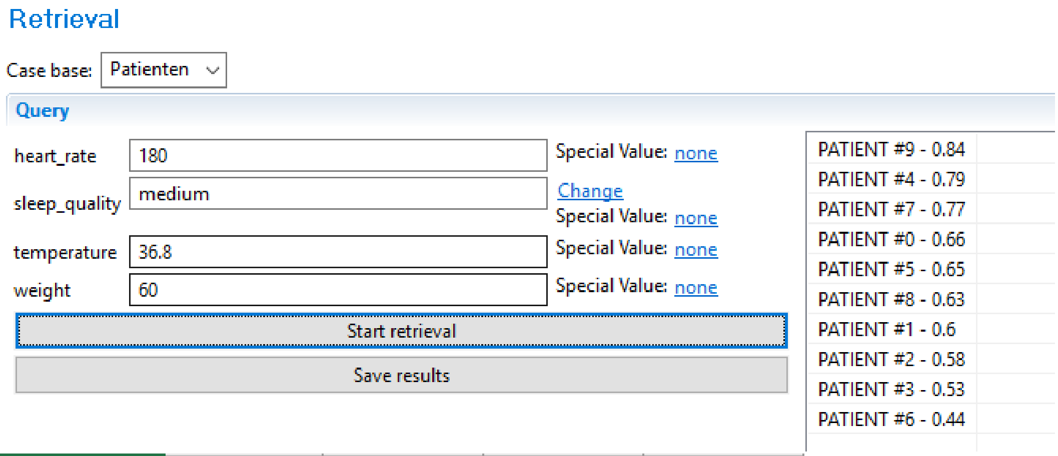
Type (•) Weighted Sum ( ) Euclidean ( ) Minimum ( ) Maximum

Attribute	Discriminant	Weight	SMF
birthday	false	0.0	default function
heart_rate	true	0.5	heart_sim
name	false	0.0	default function
sleep_quality	true	2.0	sleep_sim
temperature	true	1.0	temp_sim
weight	true	1.0	weight_exercis...

Figure 2: The global similarity measure.

	PATIENT #0	PATIENT #9	PATIENT #4	PATIENT #7
Similarity	0.75	0.66	0.64	0.64
birthday	Thu Nov 20 00...			

Figure 3: *myCBR* diyplay of the query results.



	PATIENT #9	PATIENT #4	PATIENT #7	PATIENT #0
Similarity	0.84	0.79	0.77	0.66
birthday	Mon Apr 10 (	Wed Feb 02 (	Fri Apr 09 00:	Thu Nov 20 0
heart_rate	67	58	95	75
name	Ingrid Ola	Cédric Machi	Harald Schm	Antonia Fehı
sleep_qualit	medium	medium	medium	medium
temperature	36.4	36.0	35.9	39.8
weight	49.1	90.0	95.3	65.0

Figure 4: A query with high heart rate.

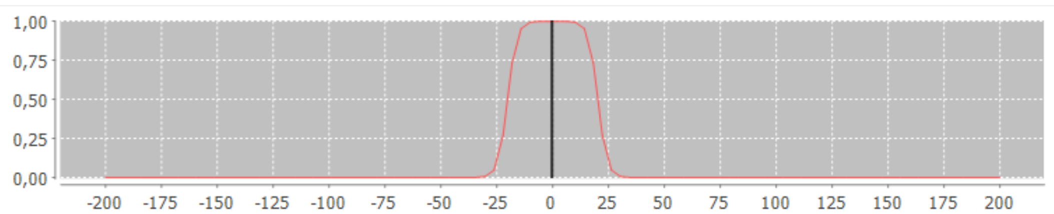


Figure 5: The similarity function for the patient's heart rate.

The most surprising query in my opinion is one where the heart rate is set to $180bpm$ for the query, which is quite high. All other query parameters are set to somewhat average values. Both query and results can be seen in figure 4. There is one other instance, `Patient #5`, with a heart rate quite fast, $170bpm$. The similarity function for the heart rate (figure 5) is designed to give a similarity close to 1 for only the closest results (like the heart rate of `Patient #5`), and similarities close to 0 for a difference greater than $25bpm$. As a result, we would expect that `Patient #5` is under the top results - instead, he is on place five with a similarity of only 0.65. The three top results all have heart rates smaller than $100bpm$, so the similarity measure is close to 0. Despite this, the top result even has an overall similarity of 0.84.

This can be explained by the global similarity function (see figure 2): it is a weighted sum of the single similarities. The weight of the heart rate similarity is only 0.5, of an overall weight of $0.5 + 2.0 + 1.0 + 1.0 = 4.5$. So the heart rate has really limited influence on the overall similarity. The top three results can instead be explained by the sleep quality: with a weight of 2.0 it represents nearly half of the overall similarity. All of the top three results have a sleep quality equal to the query value of *medium*, so the similarity of sleep quality is 1. All three top results also have a quite similar temperature. The weight of the patient, weighted by 1.0 in the global similarity, decides the final order: with a weight of $49.1kg$, `Patient #9` is closest to the query patient and therefore on place one, while the patients on place two and three with weights around $90kg$ are significantly less similar.

Four more retrieval queries and the results can be seen the appendix. In all queries, the big influence of the sleep quality is visible in the results.

The CBR cycle

The concept PATIENT could for example be used for triage in an emergency room: after measuring temperature, heart rate and weight (probably some more attributes have to be added for a system classifying well in practice, but domain knowledge is needed to do that), CBR could be used to suggest in which section of the emergency room the patient should go for further treatment (internistic, surgical, psychological, ...). In the following, the 4 steps in the CBR cycle are explained with this example. Additionally, for each step a possible usage of *myCBR* is given.

- *retrieve*: from all cases in the case base (all previous patients), the ones that are the most similar to the new case have to be retrieved. For this task, the RETRIEVAL functionality of *myCBR* comes in handy: by opening the retrieval tab and making a query with the attributes for the new patient, the retrieval can be easily performed using *myCBR*.

- *reuse*: The k most similar cases can now be used to solve the new task. In the emergency room, we could for example just send the new patient to the section to which the most similar previous patient (case) has been sent to. *myCBR* prints the top results for each retrieval with all attributes, so the most similar case is shown with its assignment to a section. The assignment can therefore be looked up and the new patient be assigned to the same section.

- *revise*: Now, we have to decide, if the classification was right. For this, we could for example track if the patient is for the treatment sent to other sections as assigned to by the doctor examining him in the assigned section. If he is treated in the assigned section, the solution was right. If he is sent somewhere else, the solution wasn't right.

- *retain*: After we know the best solution for the current case, this case should be added to the case base. With *myCBR*, this can easily be done by creating a new instance in the case base.

Appendices

A Three instances in the case base

Instance

Instance information

Name	PATIENT #0

Attributes

birthday	Thu Nov 20 00:00:00 CET 1947	Special Value: none
heart_rate	75	Special Value: none
name	Antonia Fehnker	Special Value: none
sleep_quality	medium	Change Special Value: none
temperature	39.8	Special Value: none
weight	65.0	Special Value: none

Instance

Instance information

Name	PATIENT #4

Attributes

birthday	Wed Feb 02 00:00:00 CET 2000	Special Value: none
heart_rate	58	Special Value: none
name	Cédric Machine	Special Value: none
sleep_quality	medium	Change Special Value: none
temperature	36.0	Special Value: none
weight	90.0	Special Value: none

Instance

Instance information

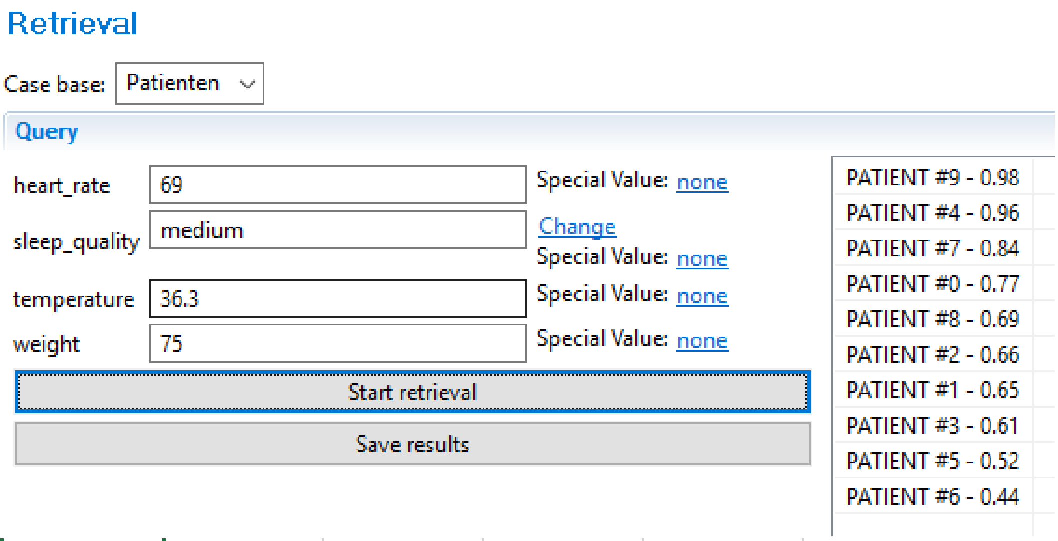
Name	PATIENT #8

Attributes

birthday	Wed Jan 04 00:00:00 CET 1950	Special Value: none
heart_rate	89	Special Value: none
name	Karlsson Sams	Special Value: none
sleep_quality	high	Change Special Value: none
temperature	36.5	Special Value: none
weight	59.8	Special Value: none

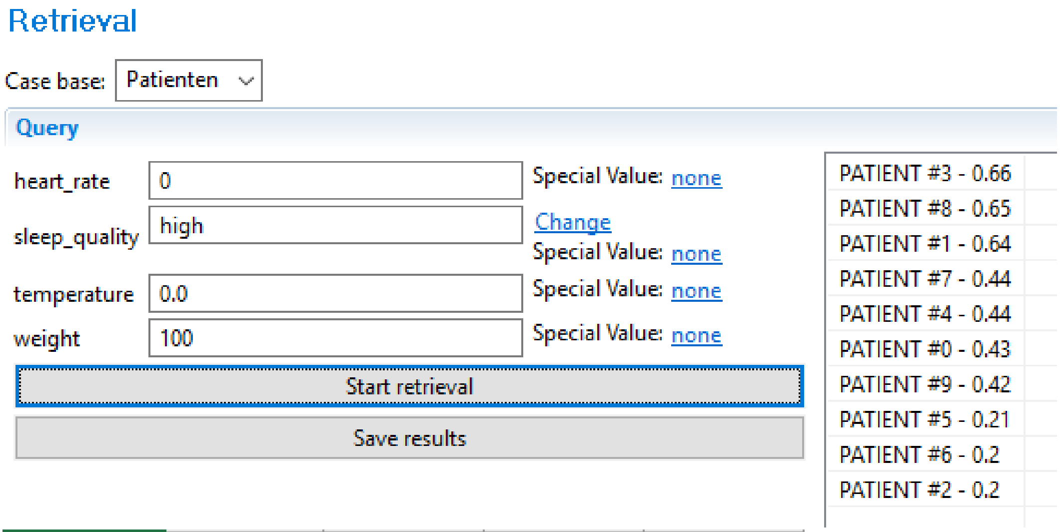
B Example Retrievals

A query with average values:



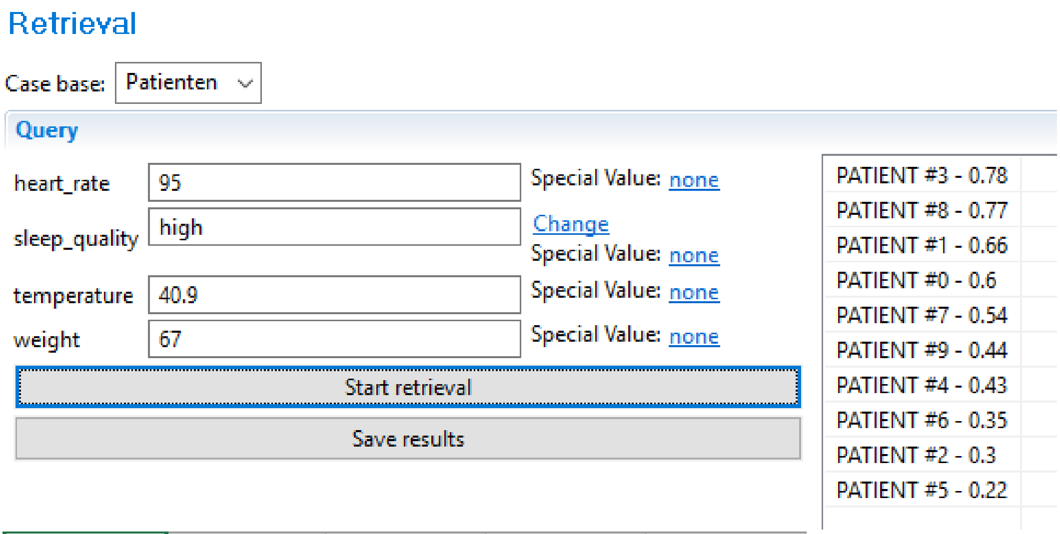
	PATIENT #9	PATIENT #4	PATIENT #7	PATIENT #0
Similarity	0.98	0.96	0.84	0.77
birthday	Mon Apr 10 (	Wed Feb 02 (	Fri Apr 09 00:	Thu Nov 20 0
heart_rate	67	58	95	75
name	Ingrid Ola	Cédric Machi	Harald Schm	Antonia Fehı
sleep_qualit	medium	medium	medium	medium
temperature	36.4	36.0	35.9	39.8
weight	49.1	90.0	95.3	65.0

A query with non-human values (far away from all instances in the case base):



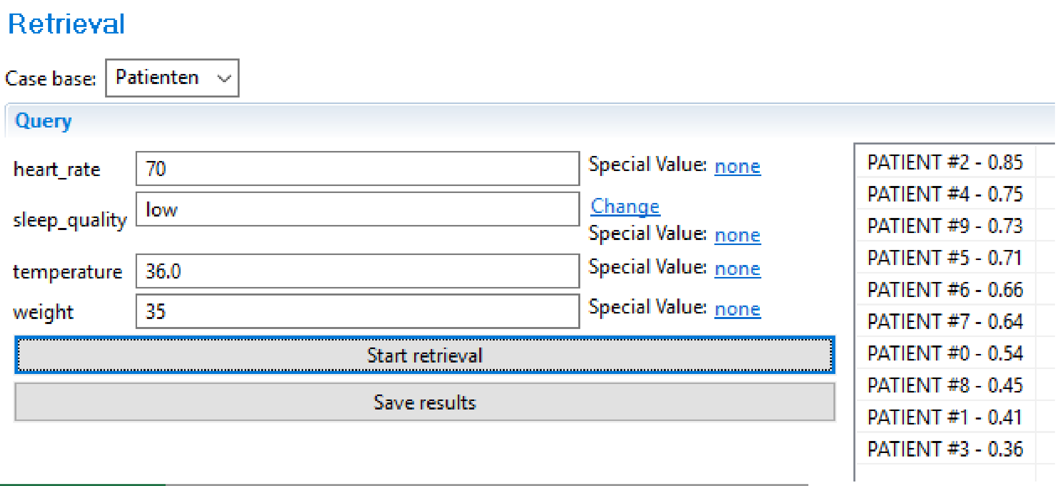
	PATIENT #3	PATIENT #8	PATIENT #1	PATIENT #7
Similarity	0.66	0.65	0.64	0.44
birthday	Sat Nov 28 00	Wed Jan 04 0	Mon Mar 16	Fri Apr 09 00:
heart_rate	83	89	120	95
name	Claus Hanser	Karlsson Sam	Louise Schill	Harald Schm
sleep_qualit	high	high	high	medium
temperature	38.1	36.5	36.3	35.9
weight	80.0	59.8	40.0	95.3

A query for a patient with fever:



	PATIENT #3	PATIENT #8	PATIENT #1	PATIENT #0
Similarity	0.78	0.77	0.66	0.6
birthday	Sat Nov 28 00	Wed Jan 04 0	Mon Mar 16	Thu Nov 20 0
heart_rate	83	89	120	75
name	Claus Hanser	Karlsson Sam	Louise Schill	Antonia Fehi
sleep_qualit	high	high	high	medium
temperature	38.1	36.5	36.3	39.8
weight	80.0	59.8	40.0	65.0

A query for an underweight patient:



	PATIENT #2	PATIENT #4	PATIENT #9	PATIENT #5
Similarity	0.85	0.75	0.73	0.71
birthday	Sun Aug 07 0	Wed Feb 02 (	Mon Apr 10 (	Sat Aug 31 0(
heart_rate	90	58	67	170
name	Mary Flasche	Cédric Machi	Ingrid Ola	Felix John
sleep_qualit	low	medium	medium	low
temperature	36.4	36.0	36.4	37.9
weight	150.0	90.0	49.1	78.0

Submitted by Klara Schlüter on October 15, 2019.