Exercise 2
TDT4173

Simon Borøy-Johnsen
MTDT

February 25, 2016

1 Theory

1.1

Case based reasoning (CBR) solves problems based on known solutions to similar problems (cases). A case contains a problem, its solution, and usually a description on how the solution was derived. The known problems are compared to the current problem in order to calculate the similarity between them. The solutions for the known cases are then modified in order to fit the problem that is being solved.

The CBR cycle has four main parts:

1. Retrieve:

   Given a problem, **retrieve** similar cases from the memory. These cases should be relevant for solving the problem.

2. Reuse:

   The solutions for the known cases are then **reused** in order to solve the current problem. The solutions are mapped onto the current problem, suggesting how to solve it.

3. Revise:

   The suggested solution is then tested (in the real world, or as a simulation).

If necessary, the solution is **revised** to fit the problem even better.

4. Retain:

   After a successful solution has been found, the case is **retained** in memory. By doing this, the case can be used for later problem solving.

The main difference between CBR and other machine learning methods (like rule based ones) lies in when the training examples are generalized. In a rule based algorithm, the training examples are generalized at training time, with no regard to the testing problem. CBR generalized the training examples at testing time. It fetches the relevant known cases according to the problem it is solving.

1.2

Case based reasoning solves problems much in the same way as humans do; it is based on experience, and we adapt known cases onto the problems we try to solve. CBR reasons by remembering. By understanding the cognitive procedures behind our problem solving, we formed the case based reasoning methods.

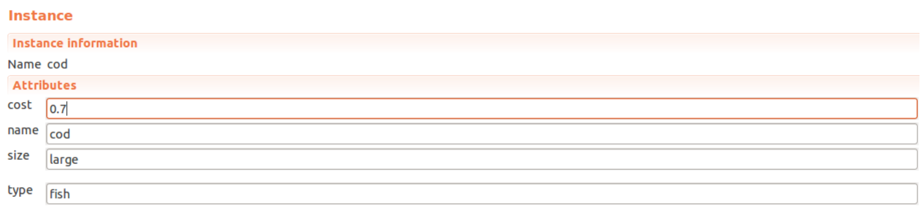
1.3

Surface similarity is based on high level (surface) features. Usually includes standard value types (strings, symbold, integers, etc.) for measuring similarity. May for example use the ”k nearest neighbours” method for determining similarity.

Structural similarity uses domain knowledge, and is highly dependent on the representation of the case. Has the possibility to retrieve cases more relevant to the problem than the similarity measure can.

2 Practical

2.1 Case modelling



Instance

Instance information

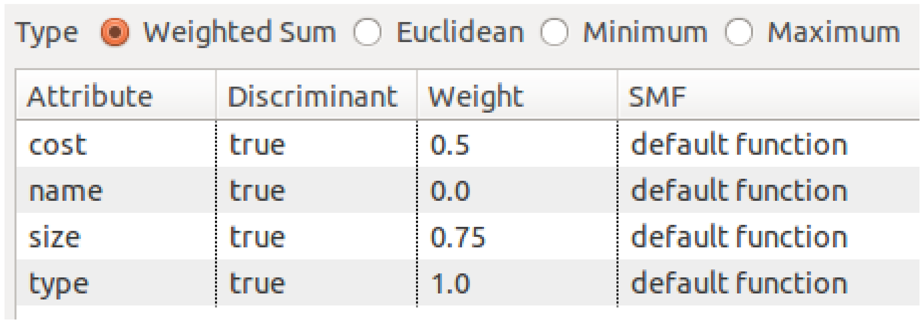
Name cod

Attributes

cost	0.7
name	cod
size	large
type	fish

Figure 1: The cod instance

2.2 Case retrieval



Type [✓ Weighted Sum] Euclidean Minimum Maximum

Attribute	Discriminant	Weight	SMF
cost	true	0.5	default function
name	true	0.0	default function
size	true	0.75	default function
type	true	1.0	default function

Figure 2: The global similarity measure

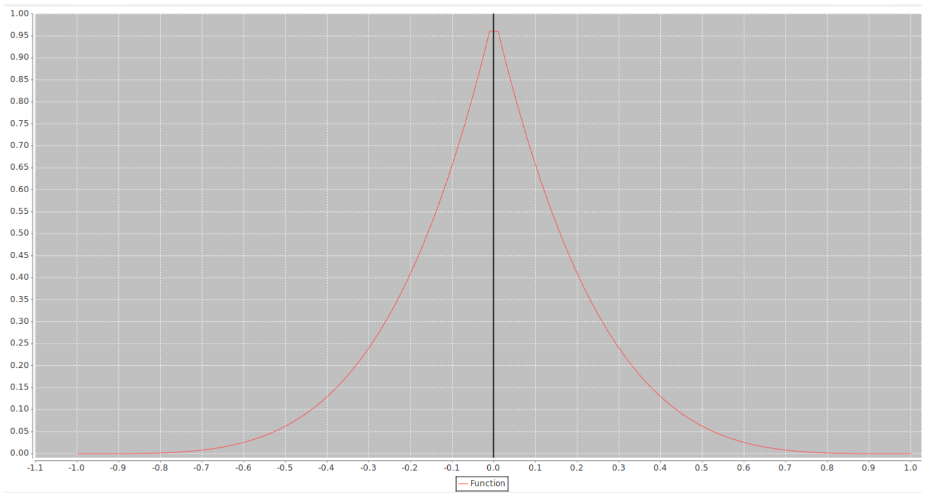
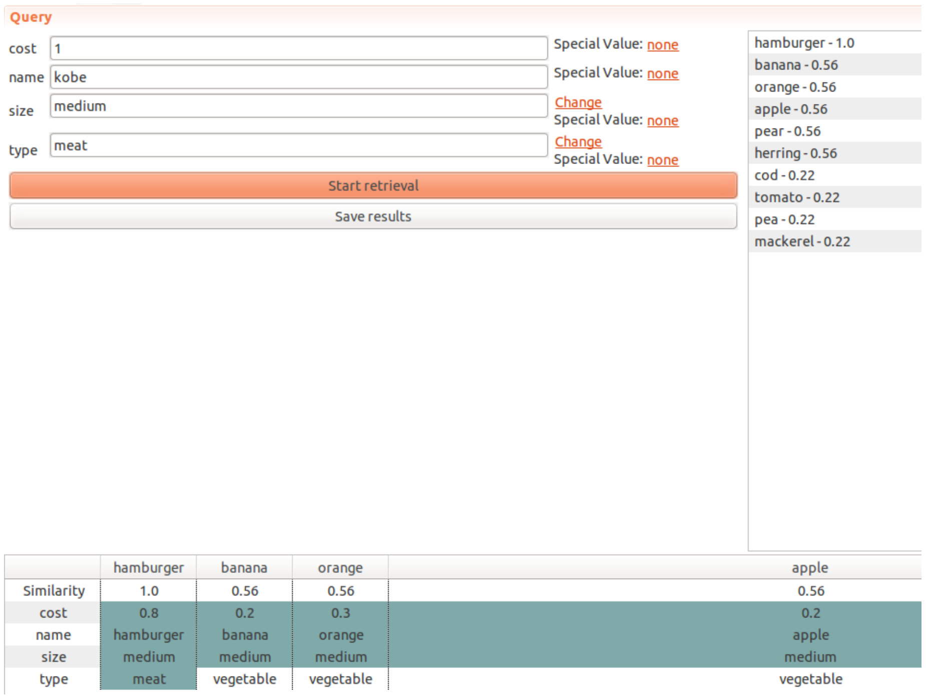


Figure 3: The default similarity measure for the cost attribute



cost	1	Special Value: none
name	kobe	Special Value: none
size	medium	Special Value: none
type	meat	Special Value: none

hamburger - 1.0
banana - 0.56
orange - 0.56
apple - 0.56
pear - 0.56
herring - 0.56
cod - 0.22
tomato - 0.22
pea - 0.22
mackerel - 0.22

	hamburger	banana	orange	apple
Similarity	1.0	0.56	0.56	0.56
cost	0.8	0.2	0.3	0.2
name	hamburger	banana	orange	apple
size	medium	medium	medium	medium
type	meat	vegetable	vegetable	vegetable

Figure 4: Results when querying Kobe beef

My most interesting query was the query for Kobe beef. The results can be seen in Figure 4.

The hamburger is naturally the clear winner. Both *size* and *type* are exact matches, and the *costs* are almost equal.

I am a bit surprised the banana and orange are second and third, since the *size* is the only match. I would expect the cod to be more similar, but the weight of the *size* outweighs the *type* similarity.

There were no other strange results. Something that can cause strange results is having too few attributes and value spans for the attributes. Take the *size* attribute, for example. How different are the sizes of a hamburger and a beef? It is hard to model exact values for such varying values. Also, so much more can separate a hamburger from a beef. Type of meat, animal, etc.