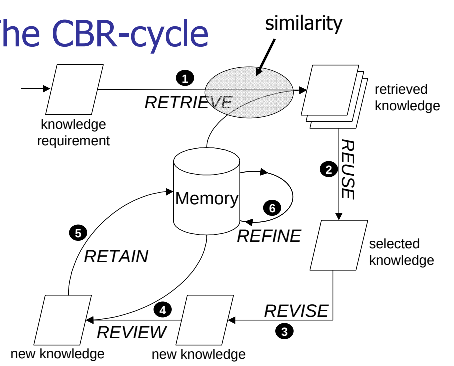
Case-Based Reasoning

Similarity / Diagnosis
Prof. Ian Watson

The CBR-cycle



Similarity ???

- Purpose of similarity, either:
  - Select cases that can be adapted easily to solve the current problem
  - Select cases that have (nearly) the same solution to the current problem
- Basic assumption:
  - <u>similar problems have similar solutions</u>

4

Similarity ???

- Degree of similarity = utility or reusability of the solution
- Similarity is an *apriori* approximation of reusability
- Goal of similarity modelling:
  - provide a good approximation
  - close to real reusability
  - And easy to compute

5

Similarity ???

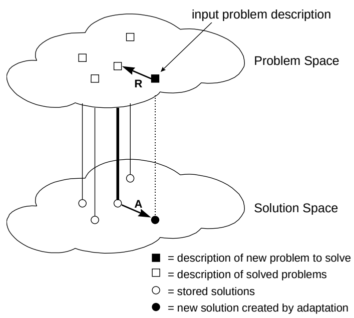
- Assumptions
  - 2 similar problem descriptions have similar solution descriptions
  - It is easier to adapt the solution of a similar problem than the solution of a less similar problem

6

Similarity ???

- Assumptions
  - 2 similar problem descriptions have similar solution descriptions
  - It is easier to adapt the solution of a similar problem than the solution of a less similar problem

Similarity ???



Modeling similarity

- Different approaches depending on case representation
- Similarity measures (metrics):
  - Functions to compare two cases: *sim*($Case_1$,$Case_2$) = [0..1]
  - Local similarity measure: similarity on feature level
  - Global similarity measure: similarity on case or object level
    - combines local similarity measures
    - takes care of different importance of attributes (weights)

Hamming distance

- A measure of the difference between two messages, each consisting of a finite string of characterS, expressed by the number of characters that need to be changed to obtain one from the other.
- E.g., 0101 and 0110 has a Hamming distance of two

Hamming distance

- In CBR each case feature in target case is compared to features of cases in entire case-base

Target case	case-base		
Apples	Apples	Apples	
Bread	Bread	Bread	
Eggs	Butter	Butter	case-base
Beer	Beer	Wine	
Steak	Steak	Steak	
Hd	4/5 = 0.8	3/5 = 0.6	

Hamming distance

- Problem - no attempt to indicate the importance or value of individual features

Target case	case-base		
Apples	Apples	Apples	
Bread	Bread	Bread	
Eggs	Butter	Butter	
Beer	Beer	Lager	lager is a type of beer
Steak	Steak	Steak	
Hd	4/5 = 0.8	3/5 = 0.6	

Similarity

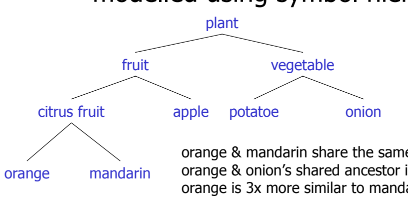
- We therefore need more complex local similarity metrics
- 4 == 10? $\Delta = 6$
- 10 == 20? $\Delta = 10$
- 20 == 100? $\Delta = 80$
- Therefore the range is important & *outliers* are bad

Similarity

- We therefore need more complex local similarity metrics
- Sim = R - Δ
- 4 == 10? Δ = 100 - 6 Sim = 94%
- 10 == 20? Δ = 100 - 10 Sim = 90%
- 20 == 100? Δ = 100 - 80 Sim = 20%

Similarity

- Symbolic attributes are commonly modelled using symbol hierarchies



orange & mandarin share the same parent (1 node higher)
orange & onion's shared ancestor is 3 nodes higher
orange is 3x more similar to mandarin than onion

orange == mandarin *Sim* = 75%
orange == onion *Sim* = 25%

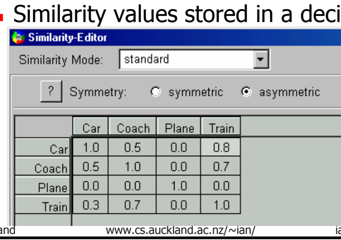
Similarity Metrics

- We'll use CBR-Works as an exemplar
  - Symbols
  - Numbers
  - Intervals
  - Sets
  - Ordered sets
  - Strings
  - Taxonomy

16

Similarity

- Symbols (unordered)
  - Similarity defined by developer
  - Similarity values stored in a decision table



	Car	Coach	Plane	Train
Car	1.0	0.5	0.0	0.8
Coach	0.5	1.0	0.0	0.7
Plane	0.0	0.0	1.0	0.0
Train	0.3	0.7	0.0	1.0

17

Similarity

- Symmetric vs asymmetric similarity
  - In symmetric similarity the result is independent of the role of the values being compared
    - Sim(Car, Train) = 0.8
    - Sim(Train,Car) = 0.8
  - In asymmetric similarity the role is important
    - Sim(Car,Train) = 0.8
    - Sim(Train,Car) = 0.3

18

Similarity

- Numbers
  - Step function
    - Computes a *sim* of 1 if the difference is < step point otherwise a *sim* of 0
  - Polynomial function
    - Computes a *sim* between 0 & 1 using a polynomial curve (ie the closer to the query the greater the sim)
  - Function with a smooth step point
    - *sim* will be in (0.0; 1.0) in respect to the distance between query- and case-value.

Similarity

- Perfect options
  - *Greater is perfect*
    - More horsepower
    - More days holiday
  - *Less is perfect*
    - Less money
    - Less time

Similarity

- Intervals
  - if the intervals in query and case do not intersect the similarity is higher the closer the gap
  - if the intervals intersect the similarity is higher the closer the bounds
  - if the case completely covers the query the
  - similarity is 1
  - if the query completely covers the case the similarity is higher the closer the bounds

Similarity

- Sets – three similarity methods

Similarity	Description
intersection	The similarity computes to the proportion of the intersection of two SymbolSets to its union.
case-inclusion	The similarity equals to **1**, if the case is included in the query. Entries included in case but not in query lower the similarity.
query-inclusion	The similarity equals to **1**, if the query is included in the case. Entries included in query but not in case lower the similarity.

22

Similarity

- Ordered Symbols
  - Similar to numbers
  - The symbols are mapped to a numeric range

23

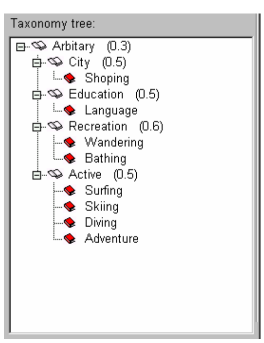
Similarity

- Strings
  - exact match - two strings are similar if they are spelled the same way
  - spelling check - compares the number of letters in two strings which are the same (Useful for strings consisting of one word only)
  - word-count - counts the number of matching words of two cases. (Useful for strings consisting of several words).

24

Similarity

- Taxonomy
  - A classification hierarchy defines similarity for concepts
  - Inner nodes of the tree are assigned similarity values
  - Leaves under a node will share the nodes similarity



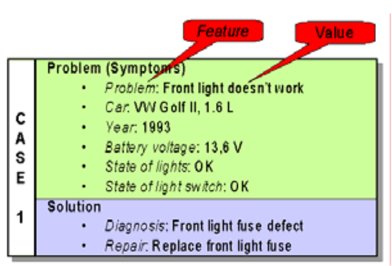
Retrieve, but efficiently ...

- Efficient case retrieval is essential for large case bases
- Different approaches depending on the:
  - case representation
  - size of the case base
- Organization of the case base:
  - Linear lists, only for small case bases
  - Index structures for large case bases
    - Kd-trees: index structure for large case bases (Wess)
    - Retrieval nets: index structure for textual (conversational) CBR (Lenz)

Technical diagnosis

- Simple example: Car Faults
  - Symptoms are observed (e.g. engine doesn't start) and values are measured (e.g. battery voltage = 6.3V)
  - Goal: Find the cause for the failure (e.g. battery empty) and repair strategy (e.g. charge battery)
- Case-Based Diagnosis:
  - A case describes a diagnostic situation and contains:
  - description of the symptoms
  - description of the failure and the cause
  - description of a repair strategy

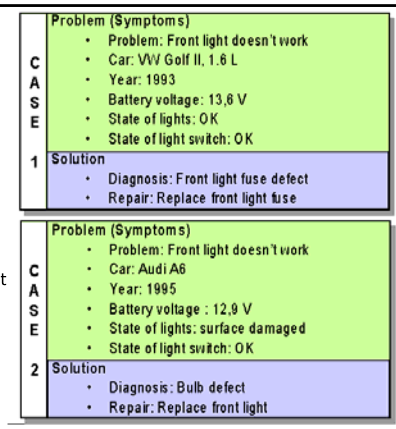
Technical diagnosis



Each case describes one situation

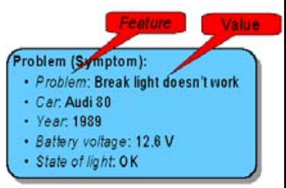
Cases are independent of each other

Case are not rules



Solving a diagnostic problem

- Make several observations about new problem
- Not all features must be known
- The new problem is a case without the solution part



Solving a diagnostic problem

- Compare the new problem with each case & select most similar case
- *Similarity* is the most important concept in CBR

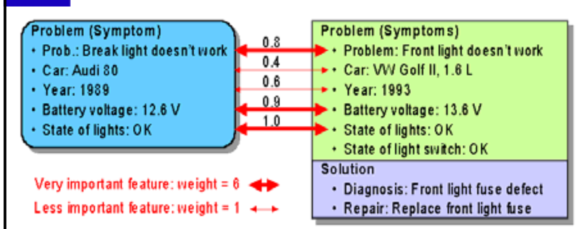
Similarity

- Similarity is assessed for each feature
- Depends on the feature value
- Features can have different weights (importance)
  - Feature: *Problem*
    - Front light doesn't work ←0.8→ Break light doesn't work
    - Front light doesn't work ←0.4→ Engine doesn't start
  - Feature: *Battery voltage* (similarity depends on the difference)
    - 12.6 V ←0.9→ 13.6 V
    - 12.6 V ←0.1→ 6.7 V

Similarity

- Different features have different importance
  - High importance:
    Problem: Battery_voltage: State_of_light:
  - Low importance:
    Make: Model: Year: Colour:

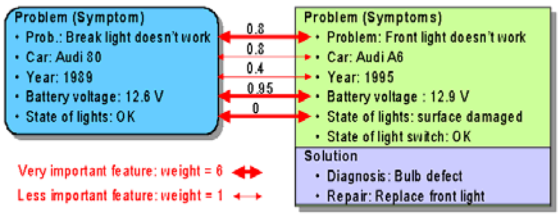
Compare new problem with case 1



Similarity Computation by Weighted Average

similarity(new,case 1) = 1/20 * [6*0.8 + 1*0.4 + 1*0.6 + 6*0.9 + 6* 1.0] = **0.86**

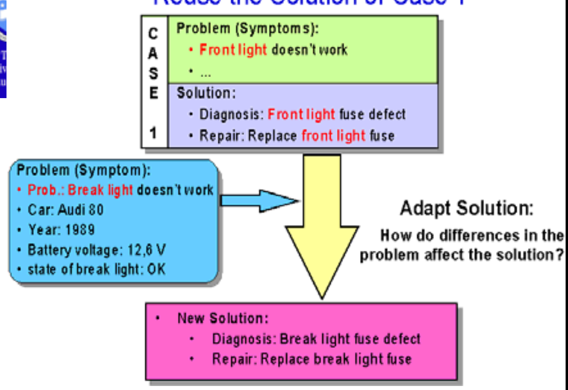
Compare new problem with case 2



Similarity Computation by Weighted Average

similarity(new,case 2) = 1/20 * [6*0.8 + 1*0.8 + 1*0.4 + 6*0.95 + 6*0] = **0.585**

Reuse the Solution of Case 1



Store the New Experience

If diagnosis is correct:

store new case in the memory.
(retain)

CASE 3

Problem (Symptoms):
- Problem: Break light doesn't work
- Car: Audi 80
- Year: 1989
- Battery voltage: 12.6 V
- State of break lights: OK
- light switch clicking: OK

Solution:
- Diagnosis: break light fuse defect
- Repair: replace break light fuse