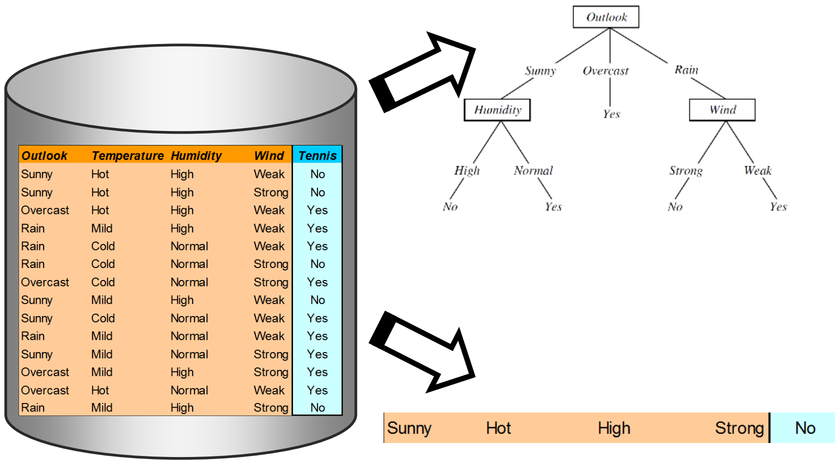
Case-Based Reasoning

Knut Hinkelmann

Literature: Bergmann, Ralph: Experience Management. Springer-Verlag 2002

Two ways of Learning from Experience



Outlook	Temperature	Humidity	Wind	Tennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cold	Normal	Weak	Yes
Rain	Cold	Normal	Strong	No
Overcast	Cold	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cold	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Sunny	Hot	High	Strong	No

- There are two ways of learning from data
  - **Machine Learning:**
    - Learn a set of rules from data
    - Apply this model for any new case
  - **Case-Based Reasoning (CBR):**
    - For a new situation find the most similar data set and take the conclusion
    - If no appropriate data set is found, solve the new case ad hoc and store it

Humans Use Cases for Problem Solving

Examples:

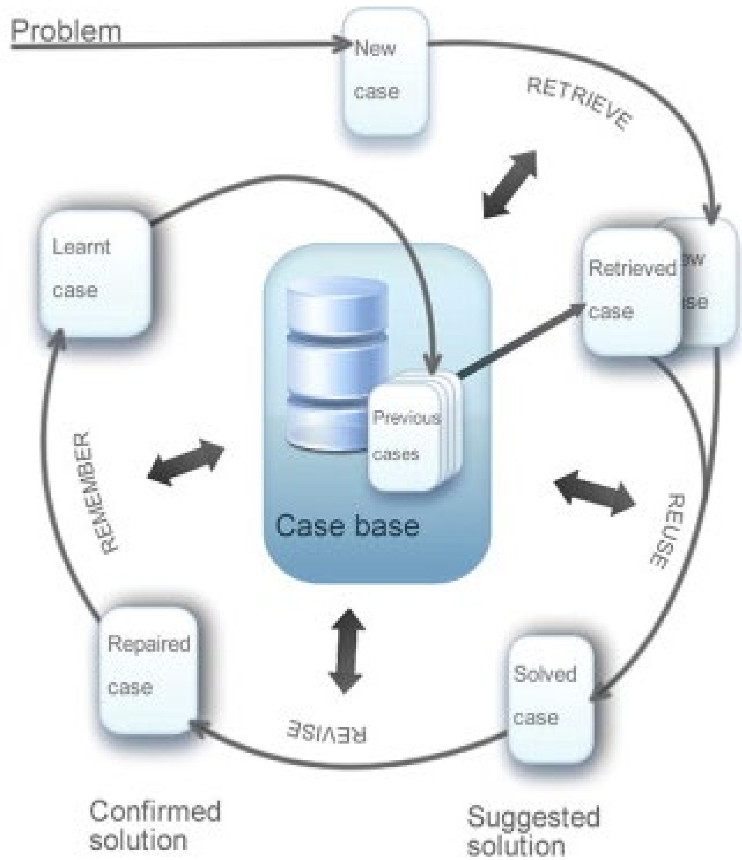
- A medical doctor remembers the case history of another patient
- A lawyer argues with similar original precedence
- An architect studies the construction of existing building
- A work scheduler remembers the construction steps of a similar workpiece (Variant)
- A mathematician tries to transfers a known proof to a new problem
- A service technician remembers a similar defect at another device

Machine Learning vs. Case Based Reasoning

- Machine Learning
  - can deliver good rules, if enough data sets are available
- Case-Based Reasoning
  - If not enough data sets are available
  - If Generalization is not possible
    - e.g. medicine, law
  - If solutions are complex or can only be vaguely described
    - e.g. therapy of a disease, repair of a machine
  - If a domain is evolving
    - e.g. law, architecture

Case-Based Reasoning

Assumption: Similar problems have similar solutions



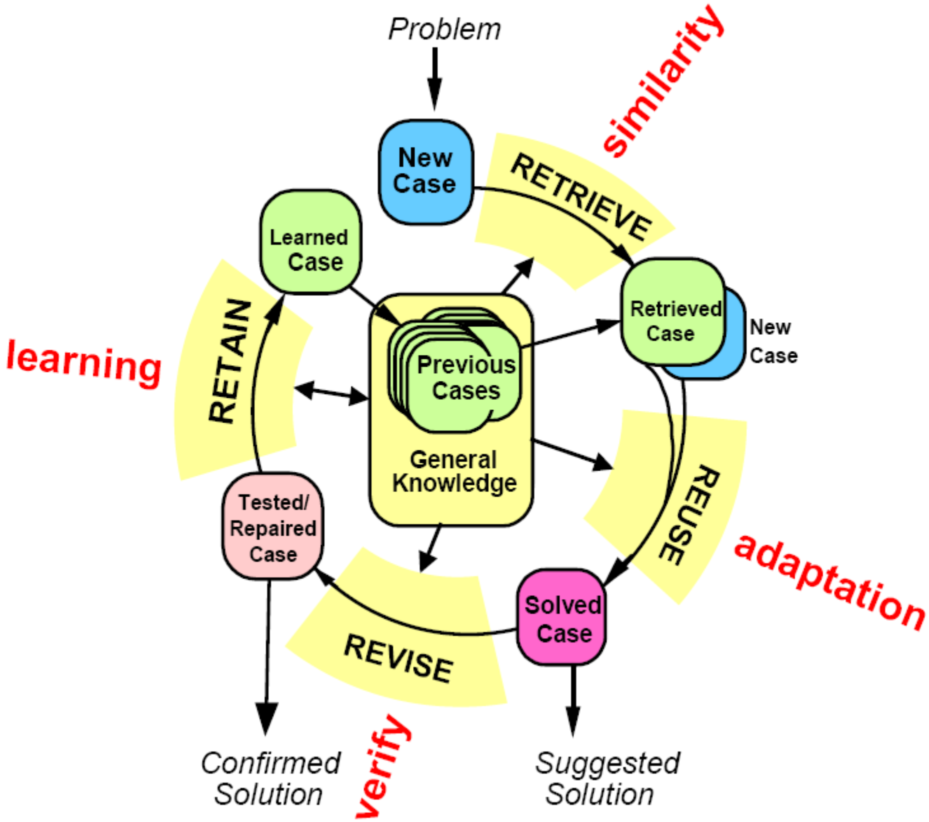
Source: A. Aamodt, E. Plaza (1994); AI Communications, IOS Press, Vol. 7: 1, pp. 39-59.

General approach:

- Experiences are stored as cases
- To solve a new problem ...
  - … retrieve similar cases
  - … solution of the most similar case is reused in the new situation
- If there is no similar case (or solution does not work) …
  - … derive a new solution
  - … store it as a new case

Quelle: Bergmann

CBR Cycle



Retrieve …

most similar case or cases

Reuse …

the information and knowledge in that case to solve the problem

Revise …

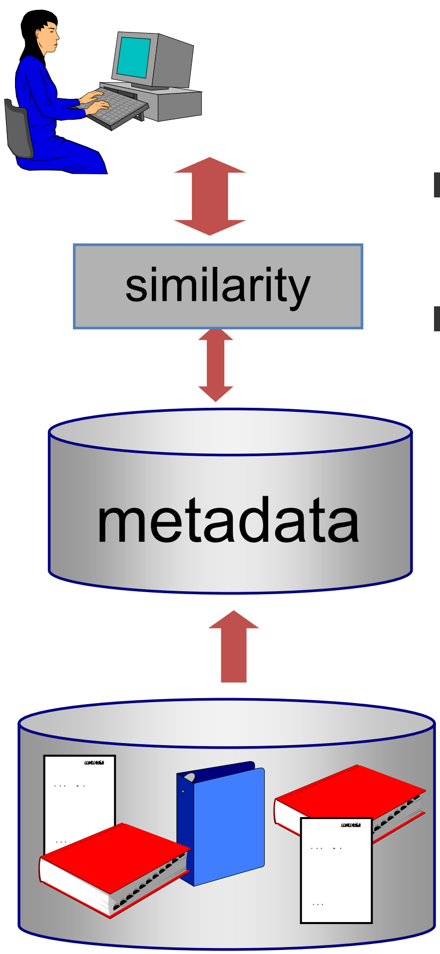
the proposed solution if necessary

Retain …

the parts of this experience likely to be useful for future problem solving

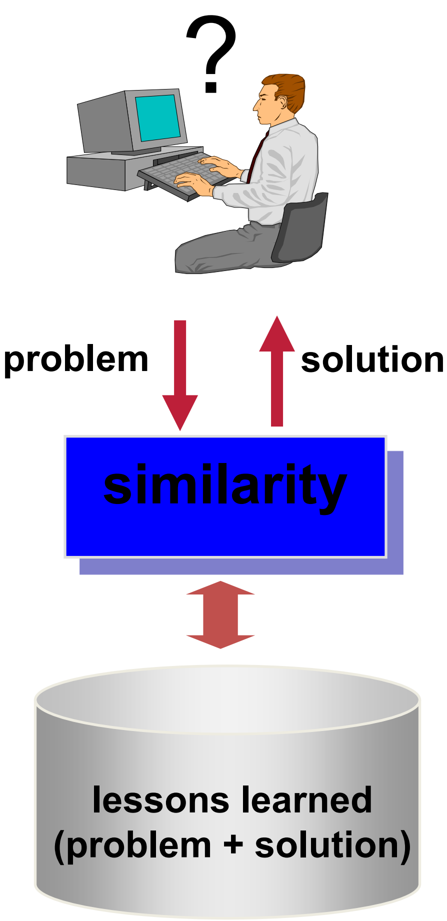
Source: K.-D. Althoff & A. Aamodt: Relating case-based problem solving and learning methods to task and domain characteristics. AI Communications 1996

Application Example: Information Retrieval



- Scenario: A database with offers of used cars
- Assume you look for the following car:
  - Audi A4, limousine, 125 PS, colour silver, automatic transmission, 2 years, bis 30'000 Fr.
- Problem: The database does not contain a car with exact this equipment
- Objective: The system should suggest those cars that are most similar to the one I look for, e.g..
  - Audi A4, limousine, 150 PS, blue, 5 gears, 2 years, € 28.000
  - Audi A4, station waggon, 125 PS, silver, automatic transmission, 3 years, € 26.000
  - BMW 320, limousine, 138 PS, silver, automatic transmission, 2 years, € 29.500
  - Volvo S40, limousine, 125 PS, silver, automatic transmission, 18 months, € 29.000
- similarity = relaxing query

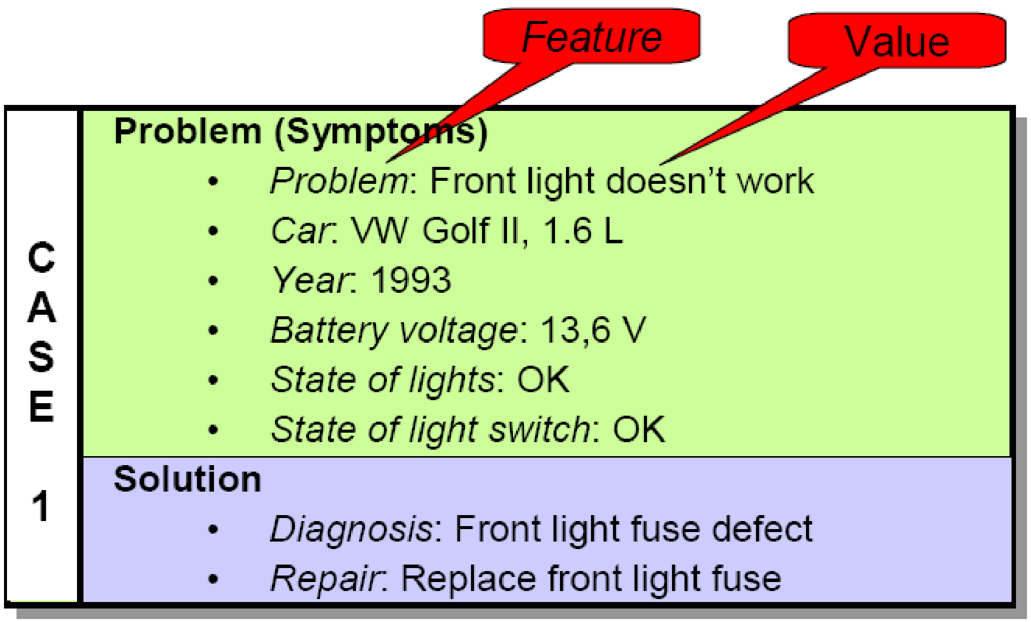
Learning from Experience: A simple Example (Overview)



- **Repairing a Car**
  - 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 a 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
  - A collection of cases is stored in a case base
  - Find case similar to current problem and reuse repair strategy

A simple Example: What's in a Case

- A case describes one particular diagnostic situation
- A case records several features and their specific values
- Assumption: All cases are independent from each other



Source: Ralph Bergmann

Example: A Case Base with 2 Cases

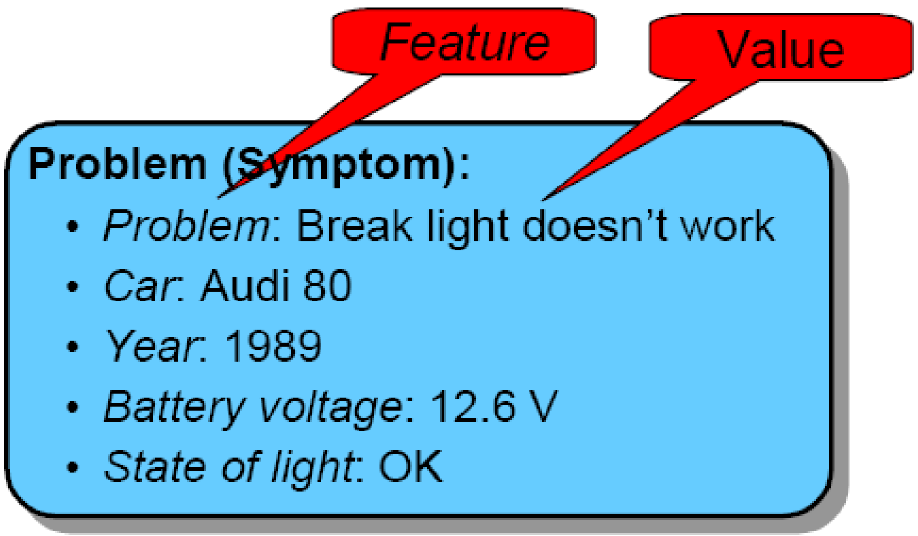
CASE	
CASE 1	**Problem (Symptoms)** • Problem: Front light doesn't work • Car: VW Golf II, 1.6 L • Year: 1993 • Battery voltage: 13,6 V • State of lights: OK • State of light switch: OK
	Solution • Diagnosis: Front light fuse defect • Repair: Replace front light fuse

CASE	
CASE 2	**Problem (Symptoms)** • Problem: Front light doesn't work • Car: Audi A6 • Year: 1995 • Battery voltage : 12,9 V • State of lights: surface damaged • State of light switch: OK
	Solution • Diagnosis: Bulb defect • Repair: Replace front light

Source: Ralph Bergmann
Adapted from Bergmann

Solving a new Diagnostic Problem

- When a new problem occures, we make several observations in the current situation
  - Observations define a new problem
  - Not all feature values may be known
- Note: The new problem is a case without solution part



Source: Ralph Bergmann

Compare the New Problem with Each Case and Select the Most Similar Case

Similarity is the most important concept in CBR !!

- When are two cases similar?
- How to rank the cases according to their similarity?
- We can assess similarity based on the similarity of each feature
  - Similarity of each feature depends on the feature **value**.

Source: Ralph Bergmann

Similarity Computation

- Assignment of similarities for features values. Not similar Very similar
- Express degree of similarity by a real number between 0 and 1

Examples:

- 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

- Different features have different importance (weights) !
  - High importance: Problem, Battery voltage, State of light, ...
  - Low importance: Car, Year, ...

Source: Ralph Bergmann

Compare New Problem and Case 1

Problem (Symptom)
- Prob.: Break light doesn't work
- Car: Audi 80
- Year: 1989
- Battery voltage: 12.6 V
- State of lights: OK

0.8

0.4

0.6

0.9

1.0

Problem (Symptoms)
- Problem: Front light doesn't work
- Car: VW Golf II, 1.6 L
- Year: 1993
- Battery voltage: 13.6 V
- State of lights: OK
- State of light switch: OK

Solution
- Diagnosis: Front light fuse defect
- Repair: Replace front light fuse

Very important feature: weight = 6

Less important feature: weight = 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**

Source: Ralph Bergmann

Compare New Problem and Case 2

Problem (Symptom)
- Prob.: Break light doesn't work
- Car: Audi 80
- Year: 1989
- Battery voltage: 12.6 V
- State of lights: OK

0.8
0.8
0.4
0.95
0

Problem (Symptoms)
- Problem: Front light doesn't work
- Car: Audi A6
- Year: 1995
- Battery voltage : 12.9 V
- State of lights: surface damaged
- State of light switch: OK

Solution
- Diagnosis: Bulb defect
- Repair: Replace front light

Very important feature: weight = 6 ⟷

Less important feature: weight = 1 ⟷

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**

Case 1 is more similar: due to feature "State of lights"

Source: Ralph Bergmann

Reuse Solution of Case 1

CASE

Problem (Symptoms):

- **Front light** doesn't work
- ...

Solution:

- Diagnosis: **Front light** fuse defect
- Repair: Replace **front light** fuse

Problem (Symptom):

- Prob.: Break light doesn't work
- Car: Audi 80
- Year: 1989
- Battery voltage: 12,6 V
- state of break light: OK

Adapt Solution:

How do differences in the problem affect the solution?

- **New Solution:**
  - Diagnosis: **Break light** fuse defect
  - Repair: Replace **break light** fuse

Source: Ralph Bergmann

Store the New Experience

- If diagnosis is correct: store new case in the memory.

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

Source: Ralph Bergmann

Example: Searching for Used Car

car 1:
- model: Audi A4
- year: 2014
- price: 15'000
- type: station waggon
- transmission: automatic
- extra: AC

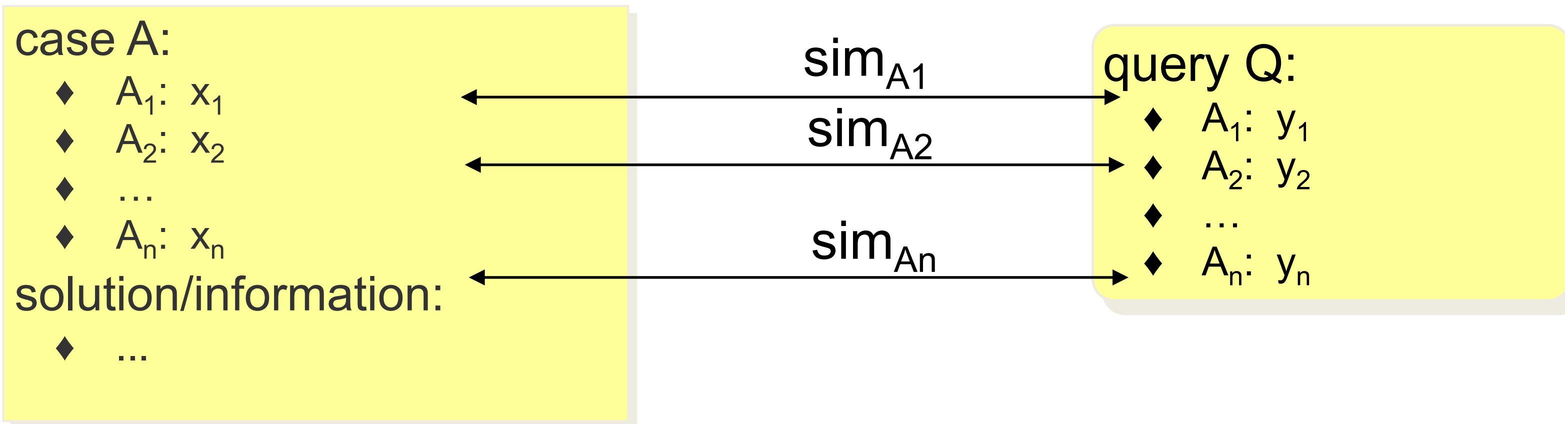
query:
- model: VW Golf
- year: 2016
- price: 18'000
- type: station waggon
- transmission: 4 gear
- extra: AC, handsfree-kit

car 2:
- model: Audi A3
- year: 2015
- price: 20'000
- type: limousine
- transmission: 5 gear
- extra: CD, handsfree-kit

Similarity Calculation for Attribute-Value Pairs

Cases resp. meta-data are represented by *n* attributes $A_1, \ldots, A_n$

- each attribute A_i has type T_i



Local similarity: for each attribute a similarity function is defined

- $sim_{Ai}(x_i, y_i): T_i \times T_i \rightarrow [0..1]$
- local similarity measures depend on the type of the attribute

Global similarity: combining values for local similarity

- $sim(A,Q) = F(sim_{A1}(x_1,y_1), sim_{A2}(x_1,y_1), \ldots, sim_{An}(x_n,y_n))$
- $F:[0..1]n \rightarrow [0..1]$ is called an **aggregation function**

Similarity Measure

Definition: A *similarity measure* is a function *sim:* M x M → [0,1]

- This definition restricts similarity to a number in the intervall [0,1].
  - It allows to express the most similar (1) und the least similar (0) situation
  - It also allows to express degrees of similarity: if *sim(x,y)* > *sim(x,z)*, then x is more similar to y than to z

(Bergmann 2002, p. 96)

CBR vs. Database Query

- Remark: Query answering in SQL can be seen as a special case of a similarity measure where the value range is a set {0,1}
  - ♦ sim(x,y) = 1 (equal) or
  - ♦ sim(x,y) = 0 (unequal)

Database System	CBR System
• simple search ⇒ "all or nothing"	• using same database but search for most similar cases
• often too many hits (underspecification) or no hits at all (overspecification)	• system can be told to show only, e.g., 10 cases by descending order
• no specific domain knowledge used for the search	• considers domain knowledge represented in similarity measures

(Bergmann 2002, p. 96)

Meanings of Similarity.

Similarity …

… always refers to a specific aspect

- Example: Two cars are similar if they
  - are of the same brand
  - have similar maximum speed

… is not necessarily transitive

- Example: For integers we can say that
  - 2 is similar to 4
  - 4 is similar to 6
  - ...
  - 99998 is similar to 100.000

  But: Is 2 similar to 100.000?

… is not necessarily symmetric

- if I look for a limousine, I probably could accept a station wagon
- if I need a station wagon because of the space, a limousine might less acceptable for me

Local Similarity Measure for Numeric Attributes

- For numeric attributes, similarity is computed as a function of distance d:

  $sim_A(x,y) = f(|d(x,y)|)$

- Typical distance functions are :
  - standard linear distance (= difference) $d(x,y) = |x - y|$
  - logarithmic distance $d(x,y) = |\log(x) - \log(y)|$

  (logarithmic distance is used for exponentially scaled value ranges. i.e. if the value range for the attribute spans several orders of magnitude)

- Examples of similarity measure for numeric attributes:

linear function

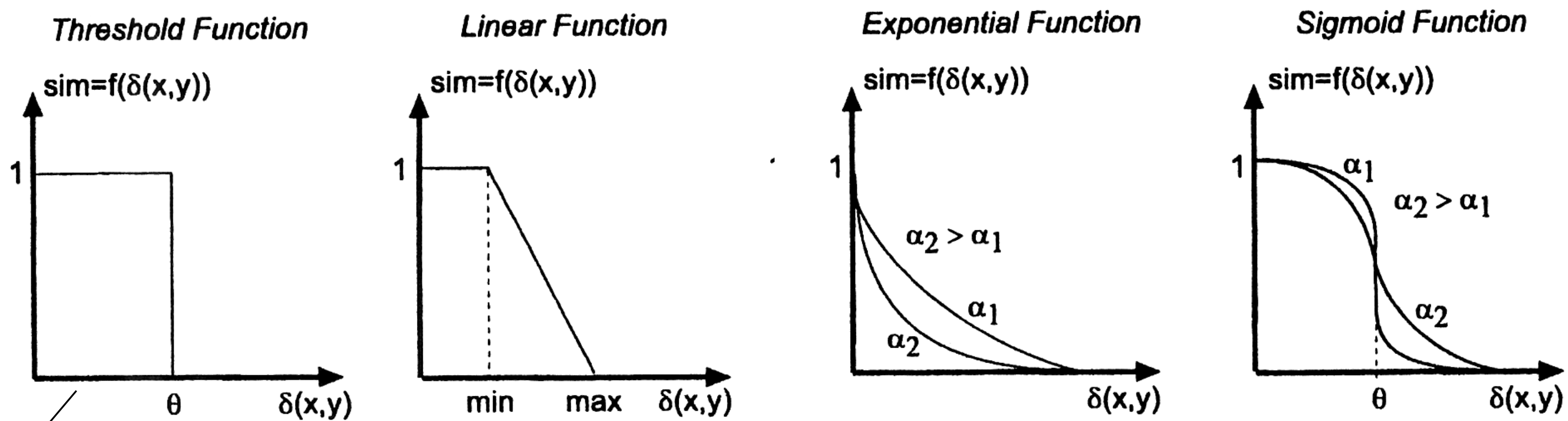
$$sim_A(x,y) = 1 - \frac{|x-y|}{\max(|x-y|)}$$

non-linear function

$$sim_A(x,y) = \frac{1}{1 + (|x-y|)}$$

(Bergmann 2002, p. 107)

Further Functions for Numeric Similarity Measures



$$f(d) = \begin{cases} 1 : & d < \theta \\ 0 : & d \geq \theta \end{cases}$$

$$f(d) = e^{d \cdot \alpha}$$

$$f(d) = \begin{cases} 1 & : \quad d < min \\ \frac{max-d}{max-min} & : \quad min \leq d \leq max \\ 0 & : \quad d > max \end{cases}$$

$$f(d) = \frac{1}{e^{\frac{d-\theta}{\alpha}} + 1}$$

(Bergmann 2002, p. 108f)

Symmetry for Numeric Values

- Symmetric:
  - It is not important for the similarity computation which attribute value belongs to the query and which one belongs to the case
  - Using the absolute distance d(x,y) = |*x-y*| *it does not matter which of the values is from case or query:* $sim(x,y) = f(d(x,y)) = f(|x-y|)$
- Asymmetric:
  - It matters, which value belongs to the query and which one belongs to the case
  - Example: For the attribute *price*, it matters whether the value in the case is higher or lower than the query value
  - Similarity can be computed with two different similarity functions

$$sim(x,y) = \begin{cases} f_1(d(x,y)) & \text{if } x>y \\ 1 & \text{if } x=y \\ f_2(d(x,y)) & \text{if } x<y \end{cases}$$

Local Similarity for Ordered Symbols

- For symbolic attributes we can distinguish approaches depending on whether there is an order defined on the symbols or not.
- Example for ordered symbols: qualitative values, e.g. {small, medium, large}
  - small < medium < large
- With such an order defined, we can determine the similarity by using the ordinal number of the symbols, e.g.
  - small --> 1
  - medium --> 2
  - large --> 3

  and applying similarity measure for numeric attributes

Local Similarity for unordered Symbols

- If there is no obvious ordering on the set of attribute values and no odering can be defined, we can apply the tabular approach
  - ♦ $sim_A(x,y) = s[x,y]$

cases y

query x

s[x,y]	limousine	convertible	station
limousine	1	0.3	0.7
convertible	0.4	1	0.2
station	0.5	0.2	1

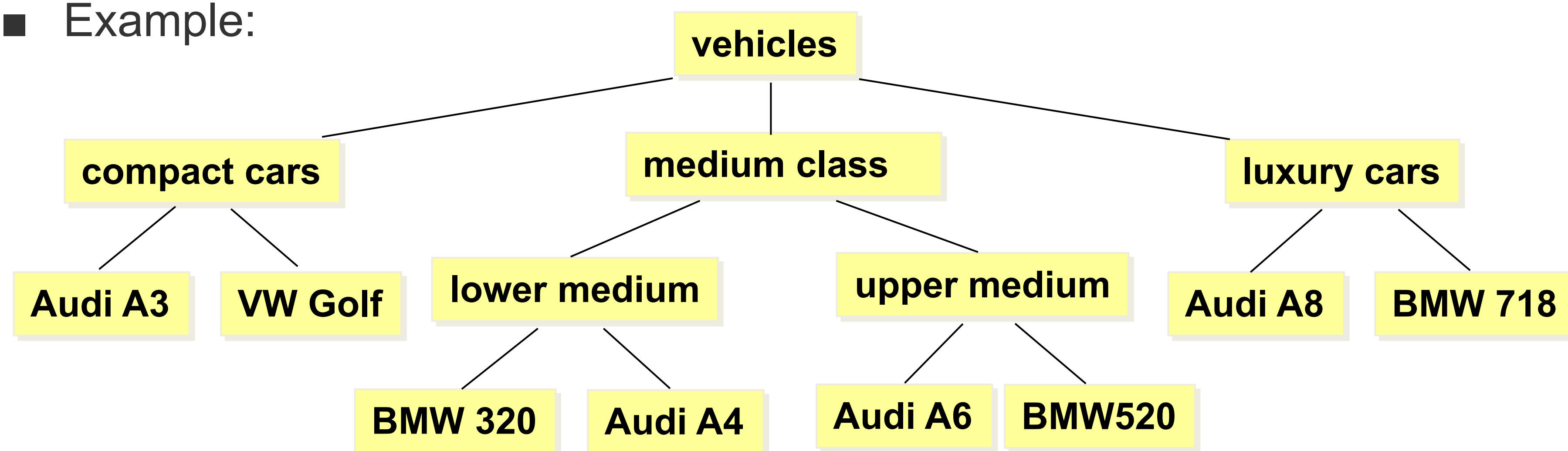
similarity of x and y

- Reflexive similarity measure: diagonal values are 1
- Symmetric similarity measure:
  upper triangular matrix = lower triangular matrix

cp. (Bergmann 2002, p. 110)

Taxonomically Ordered Symbolic Types

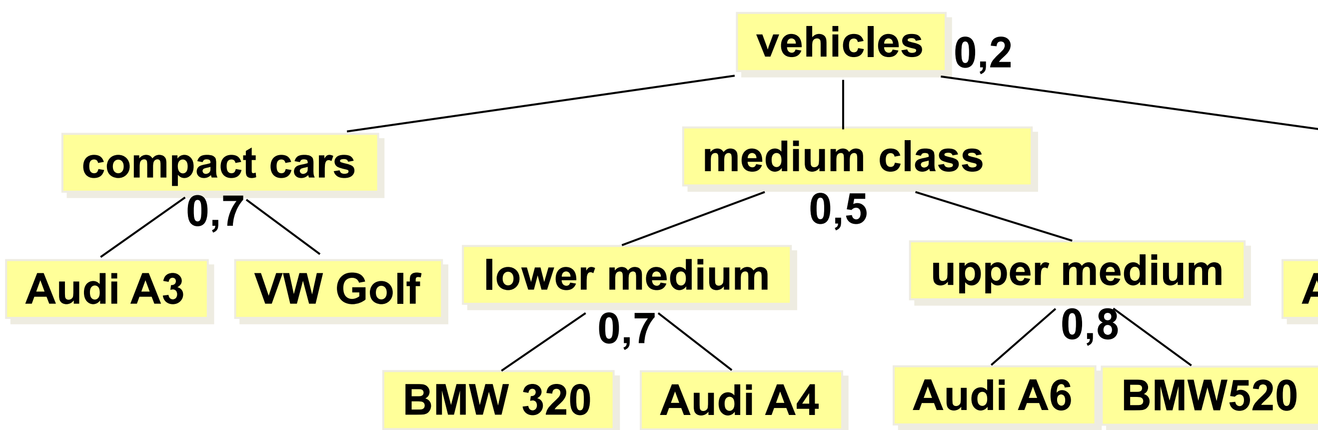
- A special variant of symbolic types are taxonomies. A taxonomy is a tree in which the nodes represent symbolic values
- A taxonomy represents an additional releationship between the symbols
  - Leaf nodes represent concrete objects of the real world
  - Inner nodes represent classes of real world objects.
  - An inner node k stands for the set of real world objects represented by leaf nodes below it
- Example:



(Bergmann 2002, p. 111ff)

Similarity Measure for Taxonomies

- Inner nodes cluster real-world objects that have some properties in common.
- The deeper we decent in the taxonomy, the more features do the objects have in common
- Similarity measures in a taxonomy
  - Every inner node K_i is annotated with a similarity value S_i
  - The deeper the nodes in the hierarchy, the larger the similarity value can become
  - Meaning: All elements belonging to a class have the similarity measure assigned to the class



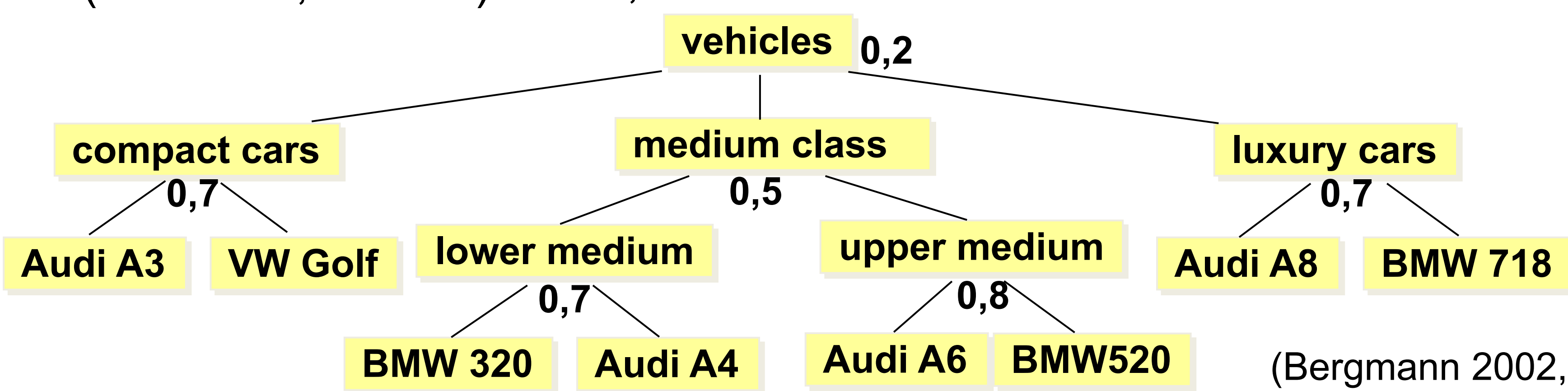
(Bergmann 2002, p. 111ff)

Computing Similarity Measure between Leaf Nodes of a Taxonomy

- Leaf nodes represent concrete element objects
- Similarity of two leaf nodes is the similarity value of the lowest common predecessor

$$sim(K_1, K_2) = \begin{cases} 1 & : K_1 = K_2 \\ S_{\langle K_1, K_2 \rangle} & : K_1 \neq K_2 \end{cases}$$

Example:
sim(BMW320, BMW320) = 1
sim(BMW320, AudiA4) = 0,7
sim(BMW320, AudiA6) = 0,5
sim(BMW320, AudiA8) = 0,2



(Bergmann 2002, p. 111ff)

Multiple Attribute Values

- Attributes can contain multiple values
  - $A_{query} = \{a_1 ,\ldots,a_n \}$
  - $A_{case} = \{b_1 ,\ldots,b_m \}$
- Similarity measure for sets:
  - Compute all pairs of similarity measrues $sim_A(a_i,b_j)$
  - Aggregate the local similarity

    $sim_A(A_{query},A_{case})$

    $= MF(sim_A(a_1,b_1),\ldots,sim_A(a_1,b_m) ,\ldots, sim_A(a_n,b_1), \ldots, sim_A(a_n,b_m))$
- There are various possible approaches for the aggregate function MF, e.g.
  - minimum
  - maximum
  - average

Unknown Attribut Values

- It often occurs that attribute values are not known (NULL):
- Strategies to deal with unknown values
  - **optimistic strategy**: Assume that unknown values are most similar: sim(NULL,x) = 1
  - **pessimistic strategy**: Assume that unknown values are least similar: sim(NULL,x) = 0.
  - **strategy of expected value**: Use an expected value, e.g. based on probability or average
  - ignore the attributes

Global Similarity

- Global similary measures are defined by applying an aggregation function $F : [0..1]^n \rightarrow [0..1]$ to the local similarity values.
  - Input: Local similarity measures $sim_{Ai}(x_i, y_i)$ for each attribute A_i
  - Global similarity:

$$sim(x,y) = F(sim_{A1}(x_1, y_1), \ldots, sim_{An}(x_n, y_n))$$

- Possible properties for F
  - F is monotone in each argument
  - $F(0,\ldots,0) = 0$
  - $F(1,\ldots,1) = 1$

Basic Aggregation Functions

- Weighted Average: $F(s_1,\ldots,s_n) = \sum_{i=1}^{n} w_i \cdot s_i$ with $\sum_{i=1}^{n} w_i = 1$
- Generalized weighted average: $F(s_1,\ldots,s_n) = \sqrt[\alpha]{\sum_{i=1}^{n} w_i \cdot s_i^{\alpha}}$ with $\alpha \in IR^+$ und $\sum_{i=1}^{n} w_i = 1$
- Maximum: $F(s_1,\ldots,s_n) = \max_{i=1}^{n}(w_i \cdot s_i)$
- Minimum: $F(s_1,\ldots,s_n) = \min_{i=1}^{n}(w_i \cdot s_i)$

(Bergmann 2002, p. 120f)