Problem Solving by

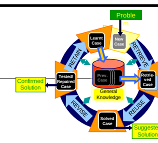
Case-Based Reasoning

PART 2

Machine Learning
Sommersemester 2009
23.06.2009

Thomas Gabel
Machine Learning Lab

Trailer



CBR Noir

(c) David Wilson,
University of North Carolina

Agenda

PART 1

1. Introduction to CBR
2. Knowledge and Case Representation
3. Similarity
4. Similarity-Based Retrieval

PART 2

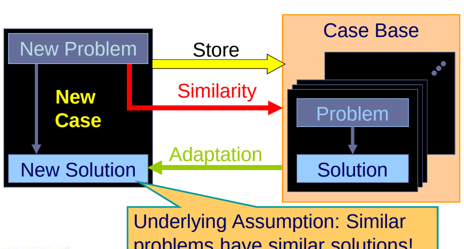
5. Solution Adaptation
6. Learning in Case-Based Reasoning
7. Applications
8. References

SUMMARY OF PART 1

What is Case-Based Reasoning?
How can cases be represented?
When is a new problem (query) similar to a case's problem part?
How to retrieve a query's nearest neighbour(s)?

Simplified CBR Model

Solve new problems by selecting cases used for similar problems and by (eventually) adapting the belonging solution.



A Simple Example Scenario: Solving a New Diagnostic Problem

Case Base with Two Cases
➔ each case describes one particular situation
➔ all cases are independent of one another

A New Problem (Query) Has to Be Solved
➔ we make several observations in the current situation
➔ observations define a new problem
➔ not all attribute values have to be known
➔ Note: The new problem is a ``case" without solution part.

Case 1

Problem (Symptoms):
- Problem : front light does not work
- Car : VW Golf IV, 1.6l
- Year : 1998
- Battery Voltage : 13.6V
- State of Light : ok
- State of Light Switch : ok

Solution:
- Diagnosis : front light fuse defect
- Repair : repair front light fuse

Case 2

Problem (Symptoms):
- Problem : front light does not work
- Car : Audi A4
- Year : 2002
- Battery Voltage : 12.9V
- State of Light : surface damaged
- State of Light Switch : ok

Solution:
- Diagnosis : bulb defect
- Repair : replace front light

Problem (Symptoms):
- Problem : break light does not work
- Car : Audi 80
- Year : 1990
- Battery Voltage : 12.6V
- State of Lights : ok
- State of Light Switch :

Compare the new problem with each case and select the most similar one!
➔ **CASE 1**

Questions:
When are two cases similar?
How to rank the cases according to their similarity?
How to reuse the solution of the corresponding case?

Note:
Similarity is the most important concept in CBR.
Similarity may be assessed based on the similarity of each feature, while the importance of different features may vary (feature weighting).

A Simple Example Scenario: Reuse and Retain

- Reuse
  - adapt the solution
  - how do differences in the problem affect the solution

Case 1

Problem (Symptoms):
- Problem : **front light** does not work
- …

Solution:
- Diagnosis : **front light** fuse defect
- Repair : repair **front light** fuse

Problem (Symptoms):
- Problem : **break light** does not work
- Car : Audi 80
- Year : 1990
- Battery Voltage : 12.6V
- State of Lights : ok
- State of Light Switch :

New Solution:
- Diagnosis : **break light** fuse defect
- Repair : repair **break light** fuse

- Retain
  - if diagnosis is correct: store new case
  - add case to case base

Case 3

Problem (Symptoms):
- Problem : break light does not work
- Car : Audi A80
- Year : 1990
- Battery Voltage : 12.6V
- State of Light : ok
- State of Light Switch :

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

Knowledge Container Model [Richter, 1989]

- „In order to solve problems, one needs knowledge.“
- Knowledge of a CBR System
  - vocabulary: knowledge representation
  - retrieval: similarity assessment (measures)
  - solution transformation: rules
  - cases
- Knowledge Management
  - as the environment may change, maintenance of the containers' contents over the lifetime of the CBR system is crucial to guarantee its continued usabilty

Similarity

- Similarity is the central notion in Case-Based Reasoning.
- Similarity is always considered between problems (not solutions of cases).
- Selection of cases during the ``Retrieve“ phase is based on the similarity of cases to a given query.

➔ Observation I: There is no universal similarity; similarity always relates to certain purpose.
➔ Observation II: Similarity is nor necessarily transitive.
➔ Observation III: Similarity does not have to be symmetric.

- Purpose of Similarity: Selection of solutions that can be easily transferred / adapted to the problem at hand.
- Similarity = Utility for Solving a (new) Problem
- **Goal:** Similarity must approximate utility as accurately as possible.

Local-Global Principle

- case description by n attributes $A_1,\dots,A_n$
- each attribute has a certain type T_i (e.g. numeric)
- **Local Similarity**
  - a seperate similarity function is used for each attribute: $sim_{A_i}: T_i \times T_i \rightarrow [0,1]$
  - local measures are depending on the respective type T_i of the attribute A_i
- **Global Similarity**
  - $sim(x,y) = sim((x_1,\dots,x_n),(y_1,\dots,y_n)) = F(sim_{A_1}(x_1, y_1),\dots, sim_{A_n}(x_n, y_n))$
  - $F:[0,1]^n \rightarrow [0,1]$ ➔ **Amalgamation Function**
  - requirements on F:
    - F is monotonous in each of its arguments
    - $F(0,\dots,0) = 0$ and $F(1,\dots,1) = 1$

Very frequently used in practice!

Examples:
- Weighted Average
  $F(s_1,\dots,s_n)= \sum_{i=1}^{n} w_i s_i$
- Maximum
  $F(s_1,\dots,s_n) = \max\{s_1,\dots,s_n\}$
- k-Minimum
  $F(s_1,\dots,s_n) = s_{i_k}$ with $s_{i1} \le s_{i2} \le \dots \le s_{in}$
- etc.

Local Similarity Measures

(for unordered symbolic and integer/real-valued attribute types)

Similarity Tables

- for attributes with symbolic type $T_A=\{v_1,\dots,v_k\}$
- sim table/matrix $sim_A(x,y)=s[x,y]$
- example: attribute ``RAM-Type“ with T_A={SD,DDR,RD}

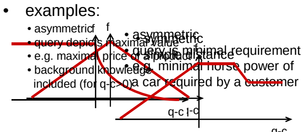
q \ c	SD	DDR	RD
SD	1.0	0.9	0.75
DDR	0.5	1.0	0.75
RD	0.25	0.5	1.0

RAM-Type

- reflexive similarity measure: diagonal elements s[k,k]=1
- symmetric similarity measure: $s=s^T$

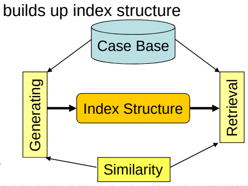
Difference-Based Similarity Functions

- for attributes with numeric type (e.g. integer or real-valued)
- similarity is based on the numerical difference between case and query value
  - linearly scaled domains: $sim_A(x,y)=f(x-y)$
  - exponentially scaled domains: $sim_A(x,y)=f(\log(x)-\log(y))$
- examples:



Case Retrieval Approaches

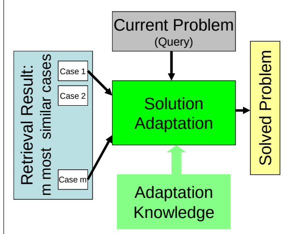
- Retrieval Task
  - Input
    - case base $CB=\{c_1,\dots,c_n\}$
    - similarity measure sim
    - query (new problem) q
  - Output
    1. most similar case c_i
       or
    2. m most similar cases $\{c_{i_1},\dots,c_{i_m}\}$
       or
    3. all cases $\{c_{i_1},\dots,c_{i_l}\}$ which have at least a similarity of sim_{min} to q
  - Main Problem: Efficiency
  - Question: How can the case base be organised in such a way to support an efficient retrieval?
- Sequential Retrieval
  - iterates over all $c \in CB$ and calculates sim(c,q)
- Two-Stage Retrieval
  - MAC/FAC approach
- kd-Tree Retrieval
  - builds up index structure



5. SOLUTION ADAPTATION

How to adapt existing solutions to be applicable for the problem at hand?

Overview on Solution Adaptation

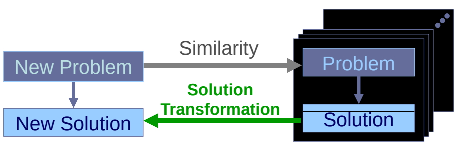


- Types of Adaptation
  - null adaptation
  - transformational analogy (transformation-based)
  - generative adaptation (derivational analogy)
  - compositional adaptation
    → combine several cases' solutions to gain a solution to the new problem
  - hierarchical adaptation

Thomas Gabel --- Problem Solving by Case-Based Reasoning --- 23.06.2009

Transformational Analogy

- **Idea:** Transform the Solution of a Similar Problem



- Representation of Adaptation Knowledge
  - by a fixed set of adaptation rules
  - by a fixed set of adaptation operators
- Types of Transformations
  - **substitutional:** change, insert, remove of parts of the solution or parameters
  - **structural:** reorganisation of the solution, insertion or deletion of objects

Thomas Gabel --- Problem Solving by Case-Based Reasoning --- 23.06.2009

Example:

Adaptation Rules for Transformational Analogy

- Product Recommendation System for PCs
  - CBR system with case base of pre-configured PC systems
  - user specifies inquiry / claims
  - system chooses the best PC system and eventually modifies some components appropriately
- Examplary Rule for Substitutional Adaptation

```
if query.database_applications==true
   and retrieved.diskspace<100GB
then
  target.diskspace:=160GB
```

- Examplary Rule for Structural Adaptation

```
if query.games==true
   and retrieved.games==false
then
  addObject target.gamecontroller
  addObject target.soundcard
```

Thomas Gabel --- Problem Solving by Case-Based Reasoning --- 23.06.2009

Generative Solution Adaptation

- Assumption: Availability of a (knowledge-based) Problem Solver
  - capable of solving any problem without cases
  - i.e. comprising sufficient domain knowledge
  - disposing of an inference mechanism
- Why use cases at all?
  - to find the solution faster
  - to find a solution similar to existing solutions
  - to obtain a solution by adaptaing an existing solution as little as possible
- Derivational Analogy (most prominent form of GSA)
  - Case = Problem + Solution + Inference (path to the solution)
  - Idea
    - reuse parts of the path to the solution for the current problem
    - combine (eventually) solution paths from several cases
    - solve the remaining gaps in the solution by using the generative problem solver
  - Algorithmic Realisations: One-Shot Replay and Interleaved Replay

Thomas Gabel --- Problem Solving by Case-Based Reasoning --- 23.06.2009

Comparison of Transformation-Based and Generative Adaptation

Generative Adaptation

- explicit problem-solving knowledge is necessary
- cases must contain the steps towards the solution (solution path)
- if problem-solving knowledge is correct, correctness of the final solution is guaranteed
- usually: usage of experience (cases) yields improved efficiency

Transformation-Based Adaptation

- **no** explicit problem-solving knowledge is necessary
- explicit knowledge about solution adaptation is required
- cases need **not** contain the steps towards the solution
- correctness of the final solution is **not** guaranteed
- usually: usage of experience (cases) yields increased competency (e.g. improvement of classification accuracy)

➔ Is a rather heuristic/conservative method – much knowledge has to be incorporated (e.g. in order to transform one PC into a very different one).

Thomas Gabel --- Problem Solving by Case-Based Reasoning --- 23.06.2009

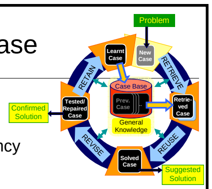
Relation between Solution Adaptation and Retrieval

- Possible Purpose of Similarity
  ➔ Selection of Solutions that are Easily Adaptable to the Current Problem
- **Goal:**
  Similarity = Adaptability of the Solution to the New Problem
- Note:
  - Adaptability is an a-posteriori criterion. The adaptability of a problem can usually be assessed only **after** having solved the problem.
  - Similarity regarding the cases' problem parts is an a-priori criterion. Similarity must be determined **before** having solved the problem.
- Goal:
  - Similarity should predict/approximate adaptability as good as possible.
  - Possible solution: Use machine learning to obtain (learn) a similarity measure that explicitly considers the adaptability of cases, too.
    (cf. next chapter)

6. LEARNING IN CASE-BASED REASONING

How are other Machine Learning approaches utilized in the context of CBR systems?

Learning and the Retain Phase



- Tasks of Learning in the Retain Phase
  - improvement of competence of the system
  - improving/maintaining the system's efficiency
- Processing of New Experience
  - may imply reorganisation of the case base (rebuilding of index structures)
  - adding new cases to the case base
  - removal of old cases from the case base
  - improvement of similarity assessment
    ➔ learning similarity measures
  - improvement of solution adaptation

Instance-Based Learning

- IBL Algorithms
  - type of case-based learning where only the case base is being adapted (similarity measures remain constant)
  - Here: focus on
    - attribute value-based case representation
    - real-valued attributes
    - classification tasks
- Input
  - some similarity (or distance) measure (e.g. Euclidean distance)
  - sequence $T=(c_1,\ldots,c_n)$ of training cases $c_1,\ldots,c_n$
- Output
  - (final) case base $CB=CB_n$
- Variants
  - IB1: addition of each case to the case base
  - IB2: addition of case c if it is being classified wrongly by the current case base
  - IB3: addition of case c if it is being classified wrongly by the current case base and removal of ``bad" cases

IB1 and IB2

- IB1

```
CB := {}
for i:=1 to n do
  CB := CB ∪ {c_i}
```

  - Problem: Case base may be growing extremely. If IB1 was used in the CBR cycle's retain phase, a case would be added to the case base after each problem-solving.

- IB2

```
CB := {}
for i:=1 to n do
  c_i := (p_i,s_i)       //p=problem, s=class
  retrieve nearest neighbour c'=(p',s') of c as
    c' := argmax_{c∈CB} sim(c_i,c)
  if s_i!=s'           //misclassification
  then CB := CB ∪ {c_i}
```


Remarks on IB2

- Properties
  - IB2 is dependent on the sequence of presentation of the cases
    ➔ Even though a training case c might be classified correctly during the training (and for that reason is not added to the case base), it may happen that c will be classified wrongly by the final case base CB_n.
  - IB2 stores much less cases than IB1.
  - Empirically: IB2 yields only slightly worse classification accuracies than IB1.
- Drawbacks
  - difficulties in handling noisy data
    → noisy data points are (usually) classified incorrectly
    → hence, they are added to the case base
    ➔ overall classification accuracy decreases
  - Goal: IB3 ought to be capable of handling noisy data.
    ➔ Extension introduced in IB3: For each case c a statistic concerning the classification accuracy CA(c) is maintained.

IB3 (I)

- Algorithm IB3

$$CA(c) = \frac{\text{number of instances classified correctly with c}}{\text{number of instances classified in total with c}}$$

```
CB := {}
for i:=1 to n do
  ci := (pi,si)      //p=problem, s=class
  CBacc := { c∈CB | acceptable(c) }
  if CBacc != {}
  then
    c' := argmax_{c∈CB} sim(ci,c) with c'=(p',s')
  else
    j := random number from {1,…,|CB|}
    c' := case from CB that has the j-highest similarity to ci
  if si!=s' then  //we have a wrong classification
    CB := CB ∪ {ci}
  for all c*=(p*,s*)∈ CB with sim(pi,p*)>=sim(pi,p') do
    update statistic CA(c*) w.r.t. si and s*
    //correct classification (si=s*) → increment numerator and denominator of CA(c*)
    //wrong classification (si!=s*) → increment denominator of CA(c*) only
    if significantlyBad(c*) then
      CB := CB \ {c*}
```


IB3 (II)

- different statistical measures are usable to implement the predicates concerning the classification accuracy
- simple variant
  - acceptable(c) iff. $CA(c) > \theta_1$
  - significantlyBad(c) iff. $CA(c) < \theta_2$
- θ_1 and θ_2 are parameters to control the algorithm

Remarks on IBx

- Discussion of IB3
  - strong sensitivity to irrelevant features
    - storage requirements scale exponentially with the number of irrelevant features
  - higher classification accuracies and lower storage requirements than IB1 and IB2
- Extension to IB4: Tolerating Attributes with Uncertain Relevance [for details, see Aha, 1992]
  - feature-specific weights used in the similarity computation
  - weights are adjusted using a simple performance feedback algorithm (weight learning) to reflect the relevance of the attributes used to describe the training instances
  - storage requirements are nearly constant with respect to the number of irrelevant features

Formal Considerations

- There are numerous formal analyses of IBx algorithms.
  ➔ Here: Focus on convergence proof of IB1 for the two-dimensional case ($D=[0,1]\times[0,1]$).
- **Goal:** How many training instances are needed for IB1, to obtain with high probability ($1-\gamma$) a nearly correct (ε-correct) characterisation of the searched concept.

- **Definition:** Let $X\subseteq\Re\times\Re$, P a probability distribution on X, and $\varepsilon>0$, $\gamma>0$. A subset $S\subseteq X$ is a *(ε,γ)-Net for X* iff. there is an exception set $Y\subseteq X$ with
  - $P(Y) \le \gamma$
  - $\forall x\in X\setminus Y\ \exists s\in S$ with $|x-s|<\varepsilon$

Coverage Lemma

- Idea: Show that a sufficiently large subset S from $[0,1]\times[0,1]$ is with ``high" probability a (ε,γ)-Net.
- **Lemma:** Let $X=[0,1]\times[0,1]$ and $\varepsilon,\gamma,\delta\in(0,1)$. Then, there is a $n_0>0$ so that it holds for each $S\subseteq X$ with $|S|=n>n_0$:
  P (S is an (ε,γ)-Net) $\ge 1-\delta$
- *Interpretation: If we are in possession of many cases (more than n_0), then we have with high probability ($\ge 1-\delta$) a classifier, for which there is with high probability ($\ge 1-\gamma$) for each problem a case with very small distance ($<\varepsilon$) in the case base.*
- Extension to m Dimensions
  - number of necessary instances grows exponentially with the number of attributes

$$n_0 = \frac{\left(\frac{\sqrt{m}}{\varepsilon}\right)^m}{\gamma} \cdot \ln \frac{\left(\frac{\sqrt{m}}{\varepsilon}\right)^m}{\delta}$$

Convergence Theorem (I)

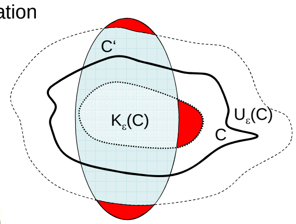
- Proof (of Coverage Lemma): *see Blackboard*
  - It can be shown that $n_0=(k^2/\gamma)\ln(k^2/\delta)$

- Note
  - So far, only statement on the density by which the problem space is covered.
  - So far, no statement on the learned concepts.

- **Definition:** Let $C\subseteq\Re\times\Re$. The *ε-Kernel of C* is defined as $K_\varepsilon(C)=\{\, x\in C \mid \text{for all } y\in\Re\times\Re \text{ with } |x-y|<\varepsilon \text{ it holds } y\in C \,\}$.
- **Definition:** Let $C\subseteq\Re\times\Re$. The *ε-Environment of C* is defined as $U_\varepsilon(C)=\{\, x\in\Re\times\Re \mid \text{there is a } y\in C \text{ with } |x-y|<\varepsilon \,\}$.

Convergence Theorem (II)

- **Definition:** Let $C \subseteq \Re \times \Re$, $\varepsilon > 0$, $\gamma > 0$, P be a probability distribution on $\Re \times \Re$. Then, $C' \subseteq \Re \times \Re$ is called an *(ε,γ)-Approximation* of C iff. there is an exception set $Y \subseteq \Re \times \Re$ with
  - $P(Y) \leq \gamma$
  - $K_\varepsilon(C) \setminus Y \subseteq C' \setminus Y \subseteq U_\varepsilon(C) \setminus Y$
- Visualisation



Convergence Theorem (III)

- Idea: Show, that IB1 approximates a searched target concept C by C' with high probability, if a sufficient amount of training cases is available.
- **Theorem:** Let $\varepsilon,\gamma,\delta \in (0,1)$, $C \subseteq \Re \times \Re$ be a bounded set, T a set of training cases and n_0 given as determined by the Coverage Lemma. It holds: If $T > n_0$, then the concept determined by the IB1 algorithm, is an (ε,γ)-Approximation of C with probability $p > 1-\delta$.
- Remarks
  - IB1 always converges to a correct classifier, given a sufficient number of training data.
  - If the ε-kernel is empty, the concept learnt may be empty, too.
  - The smaller the concept to be learnt, the worse the classification accuracy may become.

Learning Attribute Weights

- **Goal:** Improving the Similarity Assessment by Adapting the Feature Weights
- Weight Models:
  - global weights: : $sim(q,c) = \Sigma\, w_i\, sim(q_i,c_i)$
  - local weights (class-specific weights) : $sim(q,c) = \Sigma\, w_{i,class(c)}\, sim(q_i,c_i)$
    (w_{ij} as relevance matrix)
  - case-specific weights : $sim(q,c) = \Sigma\, w_{ic}\, sim(q_i,c_i)$
- Weight Adaptation
  - changing the relevance of certain features for the solution
  - numerous learning approaches exist
  - learning with or without feedback provided by retrieval/reuse
    - without: exploitation of the distribution of the cases in the case base to determine the relevance of features
    - with: correct / incorrect case selection or classification result implies an adaptatation of the weights (e.g. IB4)

Learning Local Similarity Measures

- Remember: Local vs. Global Similarity Measures
  - compare query and case values of single attributes

$$Sim(Q,C) = \sum_{i=1}^{n} w_i \cdot sim_i(q_i,c_i)$$

- Similarity Measures: Heuristics to select ***useful*** cases

traditional — approach — alternative

Knowledge-Poor Similarity Measures
- e.g. Hamming Distance
- mainly based on syntactical differences
- consider no or only little domain knowledge

+ easy to define
– lead often to poor retrieval results

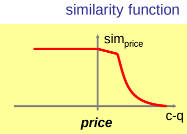
Knowledge-Intensive Similarity Measures
- e.g. use of sophisticated ***local similarity measures***
- based on knowledge about influences on the utility of cases

+ better retrieval results
– require deeper analysis of the domain and more modelling effort

Local Similarity Measures (rep.)

- Representation Depends on Attribute Type

numeric: difference-based similarity function



price

„lower and slightly higher prices will be tolerated"

symbolic: similarity table

q \ c	ROM	RW	DVD
ROM	1.0	1.0	0.9
RW	0.0	1.0	0.3
DVD	0.0	0.3	1.0

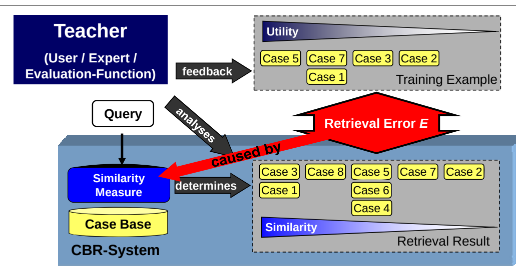
CD-Drive

encodes knowledge about the functionality of CD-Drives

Problems:
- modelling of local similarity measures is costly
- necessary domain knowledge is usually difficult to acquire

➡ **Idea**: Application of Machine Learning techniques

Learning Similarity Measures from Utility Feedback



➡ **Goal**: Finding a similarity measure that minimises *E*

Evolutionary Similarity Measure Optimisation

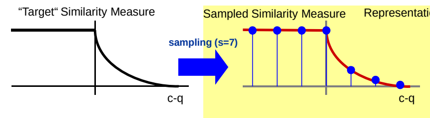
- Evolutionary Algorithms
  - search algorithms based on the mechanics of natural genetics, selection, and the principle “survival of the fittest“
  - reproduction via crossover and mutation of individuals
  - differentiation from (standard) genetic algorithms
    1. representation of individuals (example)

GA	0 1 1 0 1 0 0 0 1 1 0 0 1 1 1 1 0 ...
EA	0.3 1.3 2.6 -0.1 1.4 0.7 4.1 7.6 -2.3 4.0 0.0 0.1 -0.7 8.3 2.4 6.2 0.1 ...

    2. specialised genetic operators
- Advantages
  - robust and powerful search strategy
  - ability to handle complex entities such as local similarity mesaures
  - adequate representation of local measures as individuals

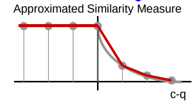
Representing Individual

- Similarity Functions per Sampling



index	sim value
1	1.0
2	1.0
3	1.0
4	1.0
5	0.2
6	0.05
7	0.0

Approximated Similarity Measure



- Similarity Tables as Matrices

q \ c	SD	DDR	RD
SD	1.0	0.9	0.75
DDR	0.5	1.0	0.75
RD	0.25	0.5	1.0

RAM-Type

similarity table individual represented as matrix

1.0	0.9	0.75
0.5	1.0	0.75
0.25	0.5	1.0

Specialised Genetic Operators & Evaluation

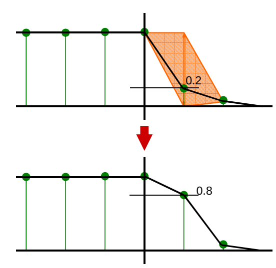
[for details, see Stahl & Gabel, 2006]

- Several Specialized Operators Implemented
  - simple mutation
  - in-/decreasing mutation
  - simple crossover
  - arithmetical crossover
  - line/row crossover
- Standard Control Algorithms of the EA
  - speciality: simultaneous learning of several local similarity measures is possible vie round robin optimisation
- Experimental Evaluation
  - learn a similarity measure that considers provided case adaptation possibilities during case retrieval
  - Scenario: product recommendation system for PCs with adaptation rules for customisation
  - Results: retrieval quality improved by approximately 50%

Specialised Genetic Operators

<u>Exemplary Operators:</u>

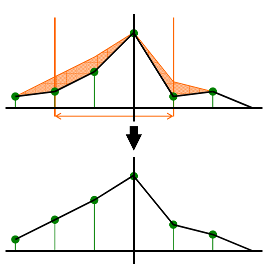
- simple mutation



Specialised Genetic Operators

<u>Exemplary Operators:</u>

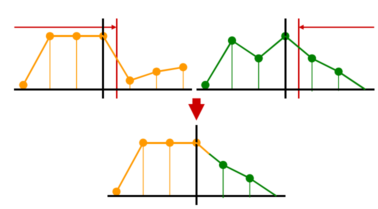
- simple mutation
- in-/decreasing mutation



Specialised Genetic Operators

<u>Exemplary Operators:</u>

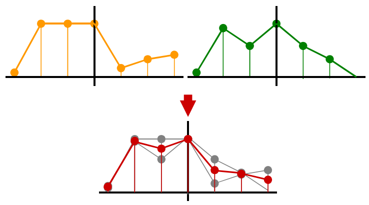
- simple mutation
- in-/decreasing mutation
- simple crossover



Specialised Genetic Operators

Exemplary Operators:

- simple mutation
- in-/decreasing mutation
- simple crossover
- arithmetical crossover

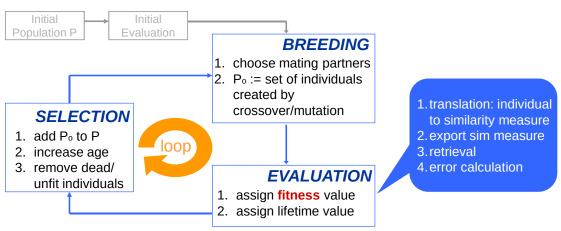


Specialised Genetic Operators

Exemplary Operators:

- simple mutation
- in-/decreasing mutation
- simple crossover
- arithmetical crossover
- line/row crossover

Control Algorithm



- simultaneous learning of several local similarity measures: round robin optimisation

Experimental Evaluation (I)

- Idea: learn a similarity measure that considers provided case adaptation possibilities during case retrieval
- Scenario: product recommendation system for PCs with adaptation rules for customisation
- Example:

Semantic 1:

Utility with respect to performance

q \ c	SD	DDR	RD
SD	1.0	0.9	0.75
DDR	0.5	1.0	0.75
RD	0.25	0.5	1.0

RAM-Type

Semantic 2:

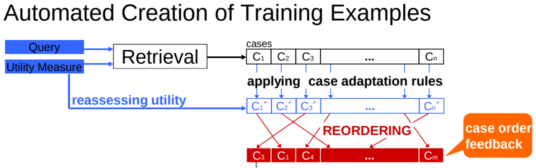
Utility with respect to performance but under consideration of possibilities to **adapt cases**

q \ c	SD	DDR	RD
SD	1.0	**1.0**	0.75
DDR	**1.0**	1.0	0.75
RD	0.25	0.5	1.0

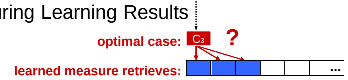
RAM-Type

Experimental Evaluation (II)

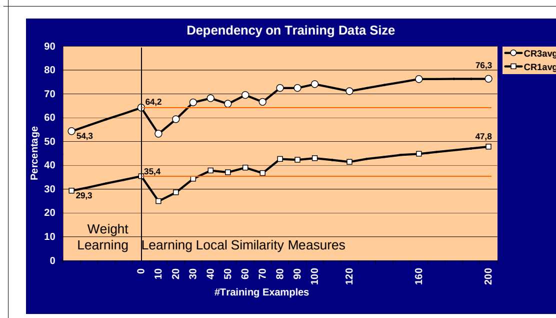
- Automated Creation of Training Examples



- Measuring Learning Results
  - CR1
  - CR3



Experimental Evaluation (III)



Discussion

- Learning Knowledge-Intensive Local Similarity Measures
  - simplified definition of accurate similarity measures
  - overcome the problems of knowledge acquisition
  - better approximation of the underlying utility function
- Necessary Precondition
  - sufficient amount of easily acquirable training data

7. APPLICATIONS AND TOOLS

Is CBR actually employed in practice?
Are there tools available I may use for trying out some of the things introduced in this talk?

Typical Applications …

- … are to be found (among others!) in the areas of
  - medicine (medical diagnosis)
  - law (textual case-based reasoning)
  - banking (credit-risk assignment)
  - engineering, engine-construction, plant construction (gas turbines)
  - biology and environment protection (waste-water treatment)
  - e-commerce (product recommendation, configuration)
- Demonstration video on …
  - imitating a RoboCup soccer player using case-based reasoning.
  - description of how the imitating agent learns through observation and of how it imitates
  - behavior of the imitating agents at different stages of the learning process

e-Commerce

- Numerous Applications with Focus on Retrieval and/or Configuration
- WebSell Project for Online Shops
  → Case-Based Retrieval (and Collaborative Recommendation / Customization)
  - tourism
    - Check Out Touristik (www.reiseboerse.com)
    - Müritz Online (www.mueritz.de)
  - residential letting
    - Hooke & MacDonald (www.hookemacdonald.com)
  - used cars
    - Quoka (www.autoaktuell.de)
  - search in online warehouses
    - Shopping24 (www.shopping24.de)
  - online shopping
    - special switches (www.jola-info.de)
    - music (www.intercox.de)
    - general (www.otto.de)
  - product advisor
    - plastics products (www.plastics.bayer.com)
    - white products (www.neck.nl)
  - after sales support
    - Software AG knowledge center (www.sag.de)

Diagnosis of Gas Turbines

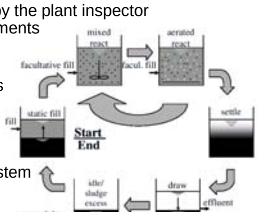
- application at General Electrics (Atlanta)
- Goal: CBR-based diagnosis of faults

- System Specifics
  - time series of sensor values read from some time prior to failure
  - higher-level representation via knowledge-based feature extraction
  - CBR system is configured via XML files (plug and play)
    - similarity measures, weights, specific case bases to be used
  - retrieve: number of nearest neighbours from selected case base
  - weighted voting to decide for one particular diagnosis (single answer)
  - recommendation provided by the system is enhanced by a confidence level statement

→ for more details see: [Devaney & Cheetham, 2005]

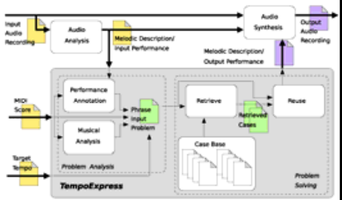
Waste-Water Treatment

- Sequencing Batch Reactor Plants
  - no continuous inflow of treatment chemicals
  - treatment processes in one single reactor
  - cyclic treatment procedure (cycle duration is 3-8h)
- Treatment in Practice
  - much habitual knowledge applied by the plant inspector
    → often applying sub-optimal treatments
  - case-knowledge is intuitive and self-explaining
    → easier to overcome ``bad" habits
- Zerberus Project
  - preliminary stage: case collection by questionnaires
  - 70 actual cases, online retrieval system
  - applied in Rhineland-Palatinate
  - www.zerberus-online.de



Expressivity-Preserving Music Transformations

- expressivity-aware tempo transformations of monophonic audio recordings (saxophone jazz)
- rendering at another tempo, while preserving naturally sounding expressivity
- CBR-based approach: TempoExpress
  - audio examples
    - original
    - uniform transform
    - CBR-based tranformation



CBR in the Health Sciences

- CBR is frequently used for medical diagnosis and support
  - diabetes treatment
    - real-time monitoring of glucose level
    - insuline dosage planning using CBR
  - dialysis medicine
  - cancer treatment
    - dose planning in radiotherapy
    - deriving treatment plans including adaptation
- References
  - www.cbr-health.org
  - www.cbr-biomed.org

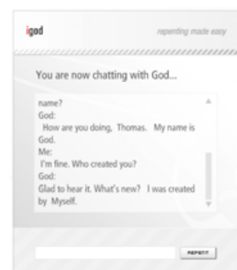
Poker Playing

- A CBR Approach to Create an Intelligent and Adaptive Poker Player (demo video)

(c) Ian Watson,
University of Auckland

Other Applications

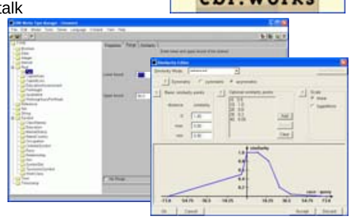
- Chat/Fun Applications (chatting bots)
  - iGod (www.titane.ca/concordia/dfar251/igod)



Tools

- CBR Works
  - very comprehensive
  - developed as research prototype
  - implemented in Smalltalk
  - applied in many practical applications
- orenge
  - commercial product by empolis (www.empolis.com)
  - implemented in Java
  - free version for students available
  - in use by many large-scale customers (Siemens, Otto, Bayer, etc.)

Company has recently been merged with Attensity.



Summary as a Video

- CBR Noir
  - „Sleuthy introduction to case-based reasoning"
    ➔ A Murder Mystery Cycle in Five Parts
  - winner of best video narration award (AAAI 2008)
  - mentions lots of names of CBR tools (for insiders)

(c) David Wilson,
University of North Carolina

8. REFERENCES

Where can I find more about CBR?

General References

- General Links to the Topic of Case-Based Reasoning
  - short characterization of CBR (in German, http://www.dfki.de/web/km/kompetenz/forschung/fallbasiertes-schliessen
  - CBR at Wikipedia: http://de.wikipedia.org/wiki/Fallbasiertes_Schlie%C3%9Fen
  - International and European Conferences on CBR: ICCBR (http://www.iccbr.org) and ECCBR (http://www.eccbr.org)

- Books and Book Chapters (sorted by author)

  - Althoff, K.-D. (2001): Case-Based Reasoning. In: S.K. Chang (Ed.), *Handbook on Software Engineering and Knowledge Engineering.* Vol. 1 „Fundamentals“, World Scientific, pages 549-588.
  - Bergmann, R. (2002): *Experience Management – Foundations, Development Methodology and Internet-Based Applications.* Lecture Notes in Art.Intelligence, Vol. 2432, Springer, Berlin.

More Books

- Bergmann, R., Althoff, K.-D., Breen, S., Göker, M., Manago, M., Traphöner, R., Wess, S. (2003): *Developing Industrial Case-Based Reasoning Applications – The Inreca Methodology.* Lecture Notes in Artificial Intelligence, Vol. 1612, 2nd Edition, Springer, Berlin.
- Kolodner, J.(1993): *Case-Based Reasoning*. San Mateo, Morgan Kaufmann.
- Lenz, M., Bartsch-Spörl, B., Burkhardt, H.-D., Wess, S. (1998): *Case-Based Reasoning Technology: From Foundations to Applications*. Lecture Notes in Artificial Intelligence, Vol. 1400, Springer, Berlin.
- Richter, M.M. (2003): Fallbasiertes Schließen. In: Görz, G., Rollinger, C., Schneeberger, J. (Eds.): *Handbuch der Künstlichen Intelligenz*, 4th Edition, Munich/Vienna, pages 407-430.
- Stahl, A. (2003): Learning of Knowledge-Intensive Similarity Measures in Case-Based Reasoning, Volume 986, dissertation.de

More Articles (I)

- Overview of CBR

  - Aamodt, A., Plaza, E. (1994): *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches.* In: AICom – Artificial Intelligence Communications, Vol. 7:1, IOS Press.
  - Leake, D. (1996): CBR in Context: The Present and Future. In: Leake, D. (Ed.), *Case-Based Reasoning, Experiences, Lessons & Future Directions*, AAAI Press.

- Articles for Specific Topics

  - Smyth, B., Keane, M. (1995): Remembering to Forget. Proceedings of IJCAI 1995, pages 377ff.
  - Wettschereck, D., Aha, D. (1995): *Weighting Features.* Proceedings of ICCBR'95, pages 347-361.
  - Stahl, A., Gabel, T. (2006): Optimizing Similarity Assesment in CBR, Proceedings of AAAI-06.

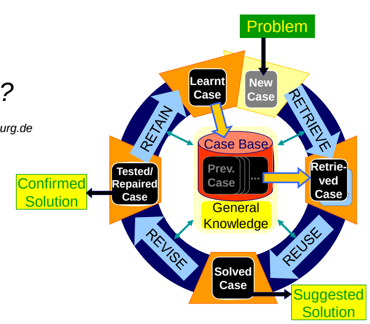
More Articles (II)

- Richter, M.M. (1995): *The Knowledge Contained in Similarity Measures.* In Proceedings of ICCBR 1995.
- Lenz, M., Burkhardt, H.-D. (1996): *Case Retrieval Nets: Basic Ideas and Extensions.* In KI 1996, pages 227-239.
- Schaaf, J.W. (1996): *Fish and Shrink: A next Step Towards Efficient Case Retrieval in Large-Scale Case Bases.* In Proceedings of EWCBR 1996, pages 362-376.
- Schumacher, J., Bergmann, R. (2000): *Similarity-Based Retrieval on Top of Relational Databases.* In Proceedings of GWCBR 2000, pages 75-86.
- Aha, D. (1992): *Tolerating Noisy, Irrelevant, and Novel Attributes in Instance-Based Learning Algorithms.* In Journal of Man-Machine Studies, Volume 36, pages 267-287.
- Stahl, A., Gabel, T. (2003): *Learning of Knowledge-Intensive Similarity Measures in Case-Based Reasoning.* In Proceedings of ICCBR 2003, pages 537-551.
- Devaney, M., Cheetham, B. (2005): Case-Based Reasoning for Gas Turbine Diagnostics. In Proceedings of FLAIRS-05.

Thanks!

Questions?

tgabel@informatik.uni-freiburg.de



END