

# Exploiting Background Knowledge when Learning Similarity Measures



**Thomas Gabel**



**Armin Stahl**

European Conference on Case-Based Reasoning  
*September, 2nd 2004*

# Overview

1. Knowledge-Intensive Similarity Measures
2. Using Evolutionary Algorithms for Learning Similarity Measures
3. Incorporation of Background Knowledge
4. Experimental Evaluation
5. Conclusions

# Knowledge-Intensive Similarity Measures

- Similarity Measures: Heuristics for selecting *useful Cases*
- Traditional Similarity Measures:
  - usually based on simple geometric distances
  - mainly estimate syntactical differences only
- Knowledge-Intensive Similarity Measures (kiSM):
  - encode specific knowledge about the application domain
  - allow a much more accurate estimation of the cases' utility
  - basic structure:

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

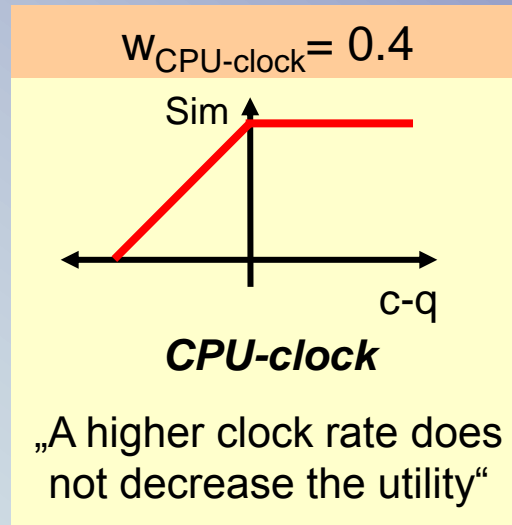
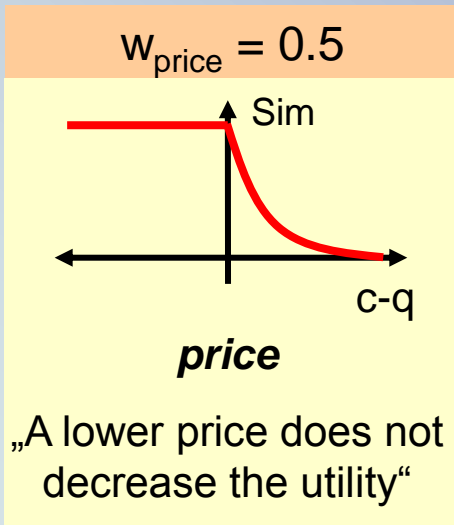
global similarity

local similarity measures

attribute weights

# Examples of kiSM

- CBR-System used for recommending PCs
  - kiSM encode knowledge about customer preferences
- Local Similarity Measures
  - difference-based similarity functions for numeric attributes
  - similarity tables for symbolic attributes
- Attribute Weights



$w_{\text{CD-Drive}} = 0.1$

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

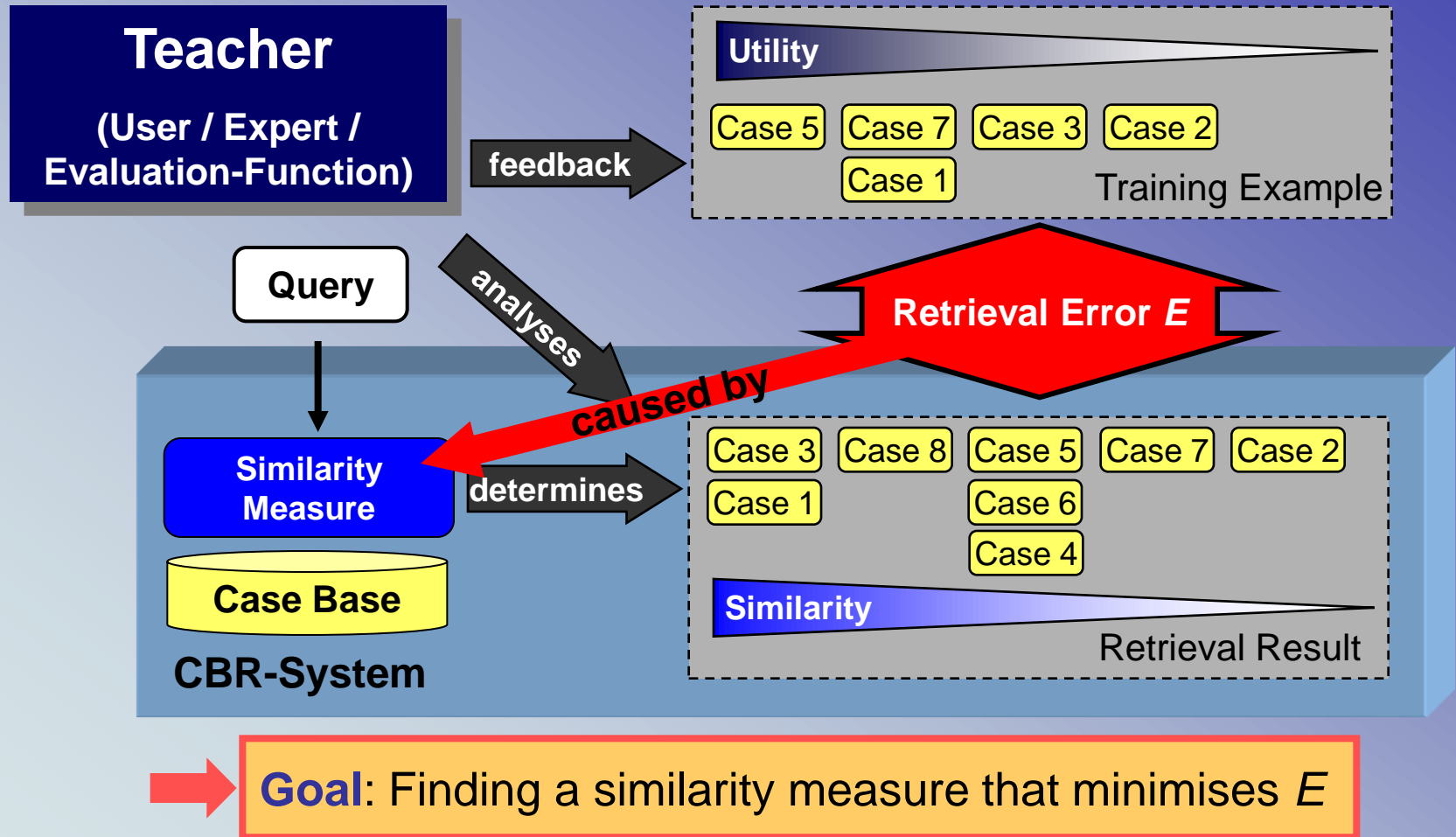
CD-Drive

The measure encodes knowledge about functionality of CD-Drives

# Modelling kiSM

- Manual Modelling of kiSM is coupled with Problems:
  - procedure is very time consuming
  - required low-level knowledge is not or only partially available
  - domain experts are not familiar with the representation formalisms
  - actual utility of cases is not considered explicitly
- Alternative Approach: Learning
  - **acquire high-level knowledge** about the actual utility of certain cases for given queries
  - **apply machine learning algorithms** for generating accurate similarity measure leading to the desired retrieval results

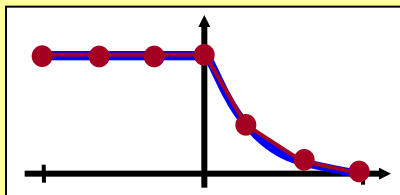
# Learning Similarity Measures from Utility Feedback



# Applying Evolutionary Algorithms

- Idea:
  - encode attribute weights and local similarity measures as individuals to be optimised by a GA
  - define corresponding mutation/crossover operators

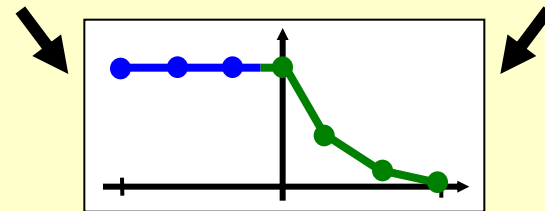
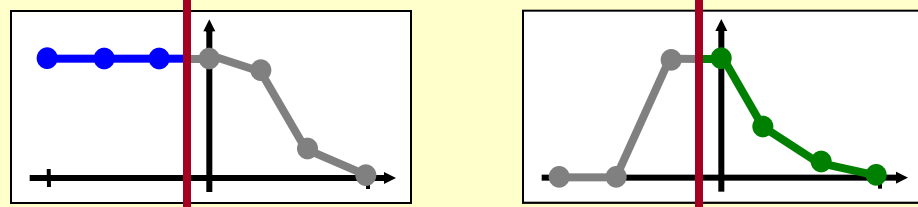
## Representation



1.0 1.0 1.0 1.0 0.4 0.1 0.0

similarity function  
as **vector** of  
sampling points

## Crossover and Mutation-Operators



# Problems

- Learning from Utility Feedback only may be critical:
  - underlying hypothesis space is huge
    - given only few training data, learning tends to overfitting
  - some certain low-level knowledge is often available
    - learning this knowledge is needless and counterproductive
  - similarity measures have typical properties, e.g. monotony
    - learning algorithms should ensure compliance with these properties
  - given utility feedback and case bases usually provide only limited information about certain value combinations
    - trying to learn kiSM for other value combinations is useless

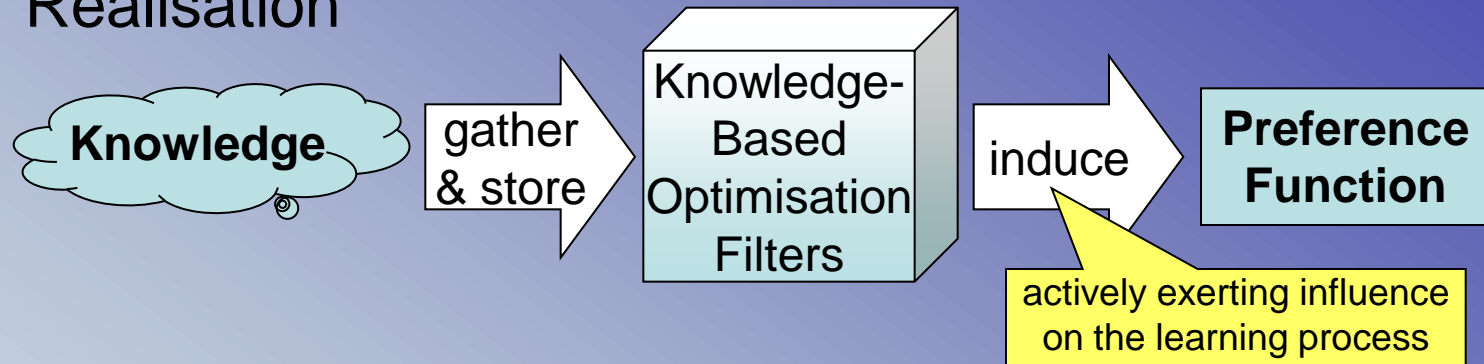


**Goal:** Restricting the Search Space and biasing the Learner by exploiting available Background Knowledge

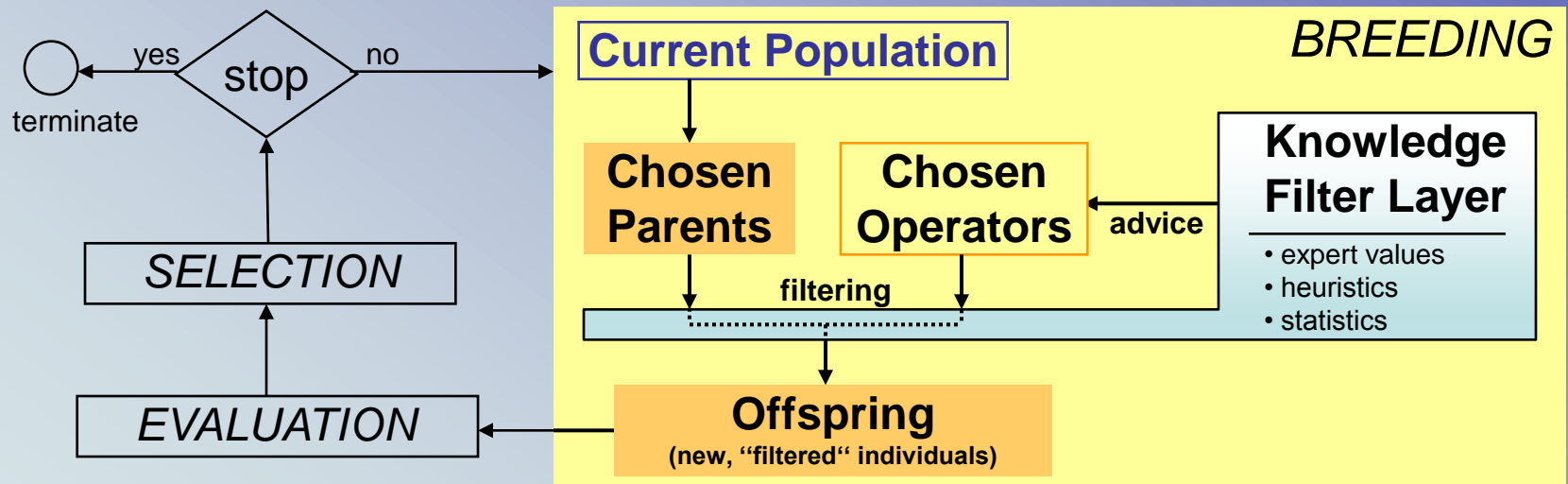


# Incorporating Background Knowledge

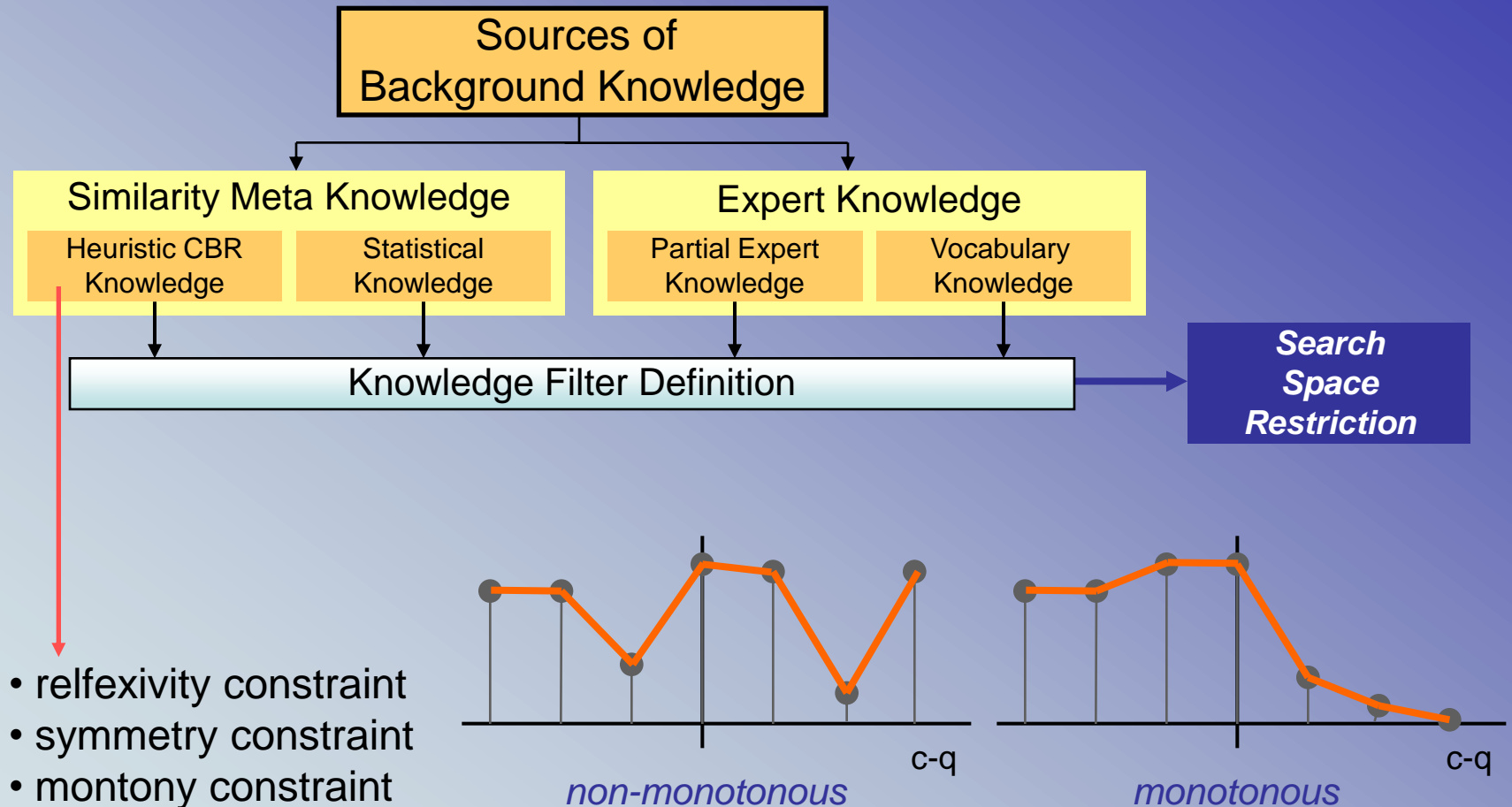
- Realisation




- Modification of Created Hypotheses




# Sources of Knowledge

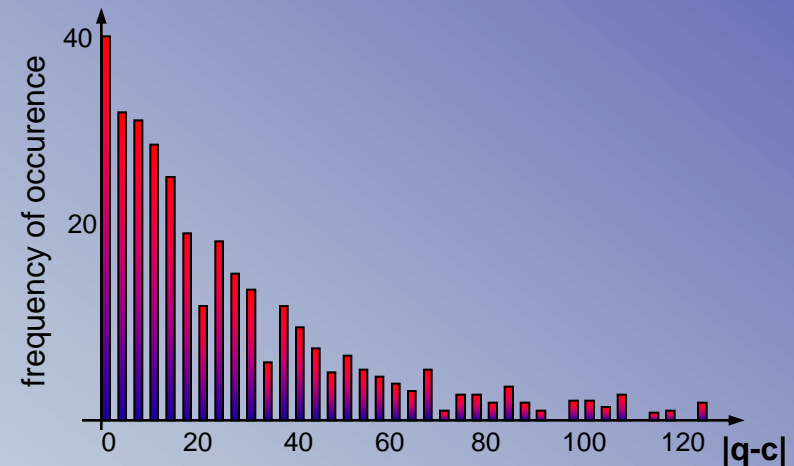


# Mining Knowledge from the Case Base

- Local Measure Definition: **high vs. low importance regions**
  - consulted frequently 
  - high impact on measure's performance
  - utmost correct definition necessary

 **Focus of Learning Algorithm**
- Statistical Case Base Analysis:  
Which combination of query and case value occurs how often if each case is used as query once?
- Assumptions
  - substantial case base
  - representative for queries occurring in practice

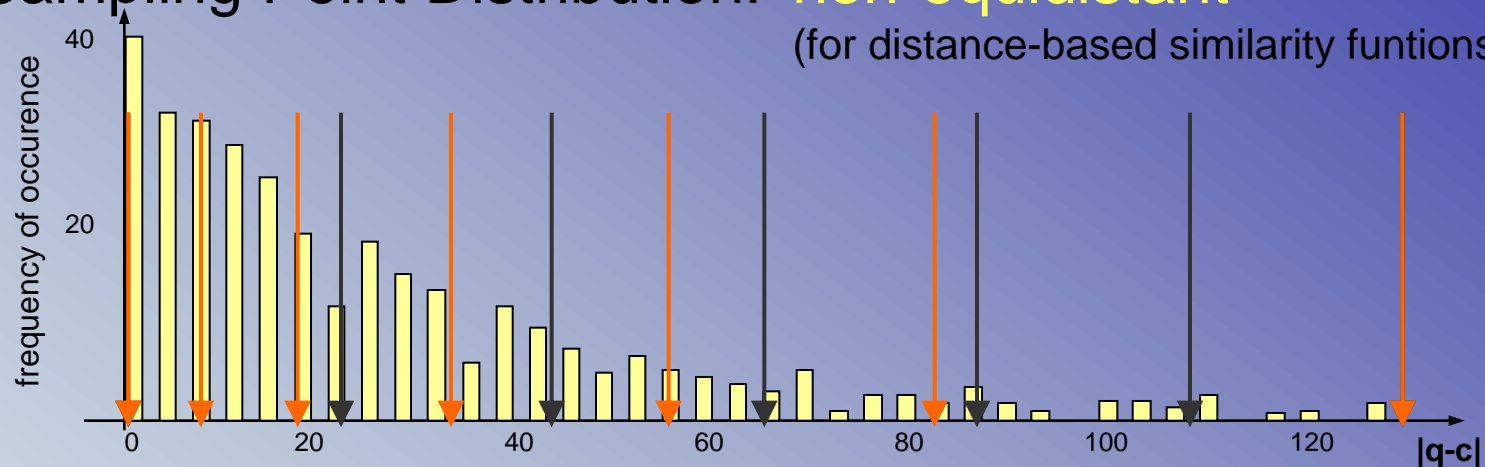
	a	b	c	d	e
a	1.0	0.1	0.2	0.1	0.3
b	0.1	1.0	0.4	0.8	0.2
c	0.2	0.4	1.0	0.6	0.7
d	0.1	0.8	0.6	1.0	0.3
e	0.3	0.2	0.7	0.3	1.0



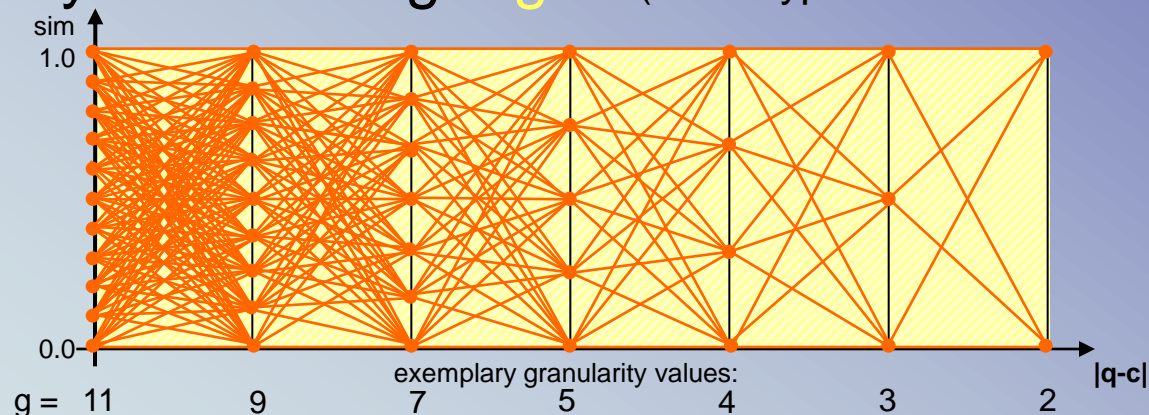
# Employment of the Mined Knowledge

- Sampling Point Distribution: **non-equidistant**

(for distance-based similarity functions only)



- Granularity: introducing a **grid** (for all types of local measures)



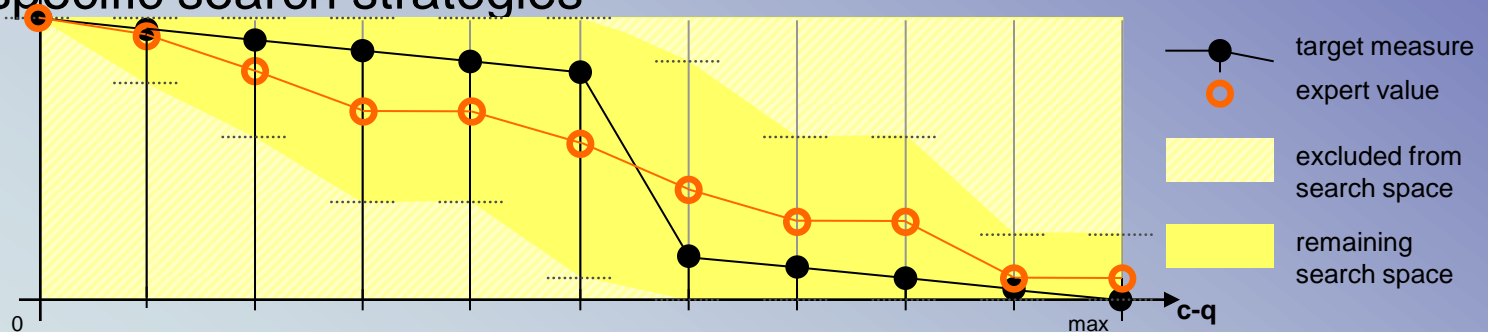
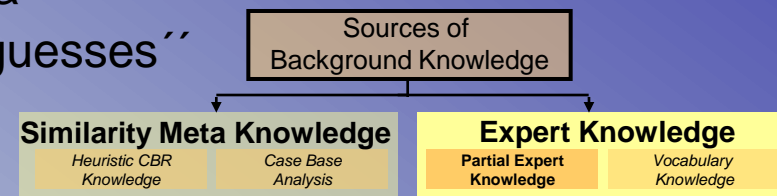
# Partial Expert Knowledge

- Motivation

- shortening the gap between fully automatic (learning) and fully manual (knowledge engineer) definition of similarity measures
- benefits:
  - reduced knowledge acquisition effort
  - exclusion of overfit-minima
  - avoidance of “educated guesses”

- Approaches

- attribute and weight preferences
- expert-estimated values with confidence levels
- specific search strategies

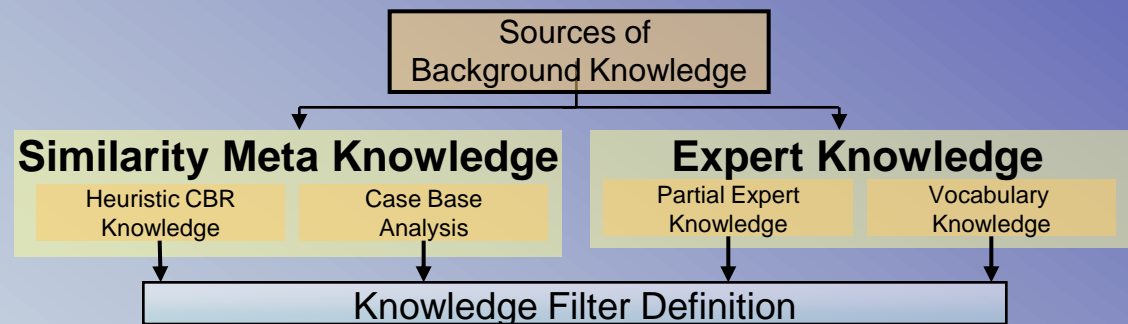


# Experimental Evaluation (I)

- Learning Experiments in various Classification and Regression Domains
- Comparison: Accuracies achieved with
  1. default similarity measures (knowledge-poor)
  2. learnt similarity measures
  3. similarity measures learnt with help of knowledge filters

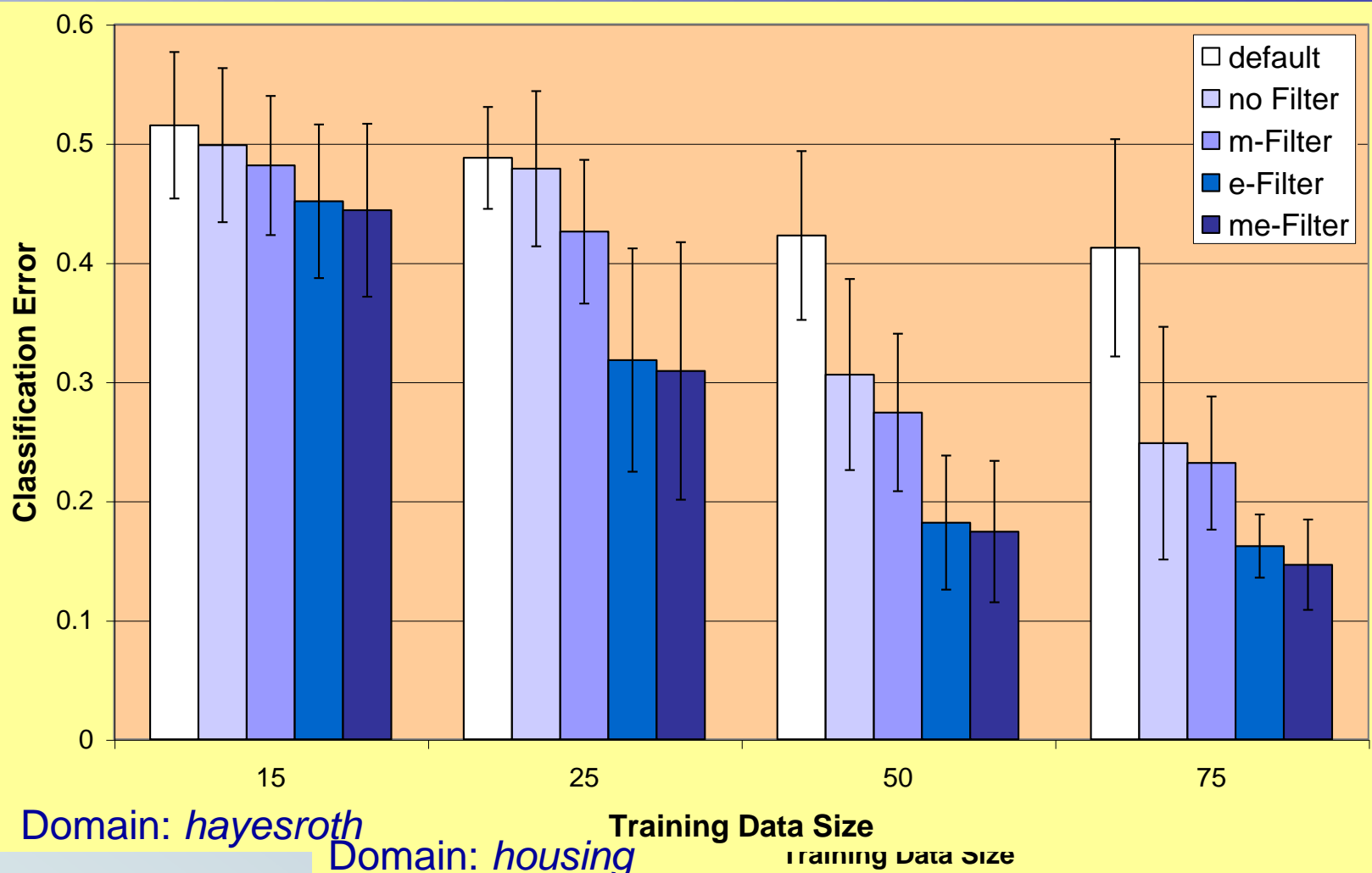
- Filter Definition

- m-Filters
- e-Filters
- me-Filters



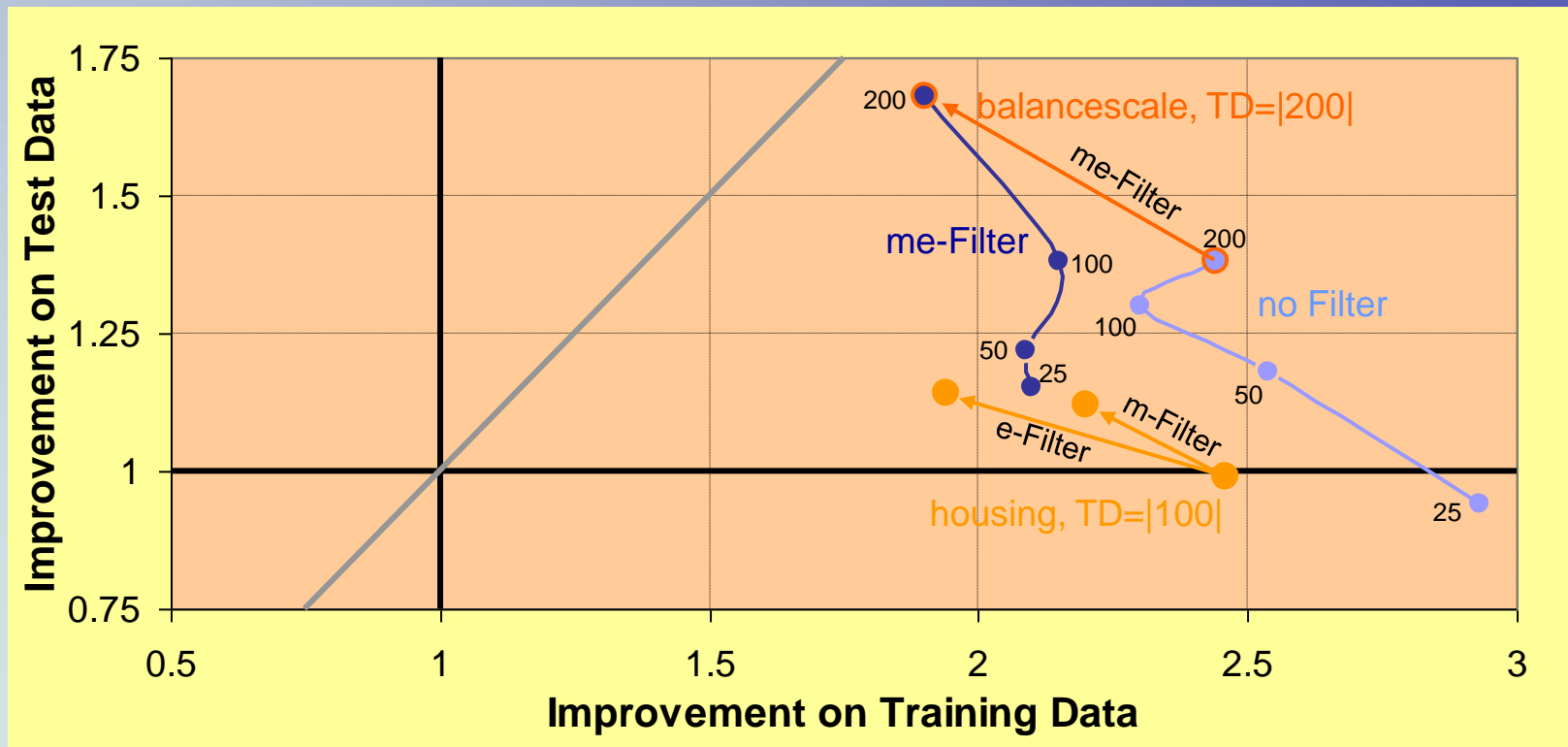
- Dependency on different Training Data Sizes
- Occurrence and Reduction of Overfitting

# Experimental Evaluation (II)



# Experimental Evaluation (III)

- Overfitting Analysis
  - x-values: quality of learnt vs. default measure on training data
  - y-value: quality of learnt vs. default measure on test data





# Conclusions

- Utilisation of Additional Background Knowledge
  - similarity meta knowledge and expert knowledge
  - search space restriction via knowledge-based optimisation filters
- Benefits
  - reduction of susceptibility to overfitting
  - more directed search, avoiding irrelevant parts of the search space
  - hybrid similarity measure definition: partially defined manually, partially learnt
- Experimental Examinations
  - clear outperforming of default similarity measures
  - clear improvement via knowledge filters