Exploiting Background Knowledge when Learning Similarity Measures

Thomas Gabel[1] and Armin Stahl[2]

[1] University of Karlsruhe, Institute AIFB
tga@aifb.uni-karlsruhe.de

[2] German Research Center for Artificial Intelligence DFKI GmbH
Research Group Image Understanding and Pattern Recognition (IUPR)
stahl@informatik.uni-kl.de

Abstract. The definition of similarity measures—one core component of each CBR application—leads to a serious knowledge acquisition problem if domain and application specific requirements have to be considered. To reduce the knowledge acquisition effort, different machine learning techniques have been developed in the past. In this paper, enhancements of our framework for learning knowledge-intensive similarity measures are presented. The described techniques aim on a restriction of the search space to be considered by the learning algorithm by exploiting available background knowledge.

1 Introduction

Similarity measures are certainly one of the most import aspects of Case-Based Reasoning. Since they are required for the first step of the CBR cycle [1], namely the retrieval of useful cases [3], the quality of the employed similarity measure strongly influences the entire problem solving process. For example, if suboptimal cases are retrieved, the subsequent adaptation process will be more expensive, might result in suboptimal solutions, or might even fail completely. In the most commercial CBR systems applied nowadays (typically classification, diagnosis or help-desk systems), the situation is even harder. Here, usually no case adaptation functionality is provided at all. Hence, the quality of the final output of the CBR system completely depends on the quality of the case data provided and the systems' ability to retrieve the best cases available.

One reason for the great success of CBR is certainly the reduced knowledge acquisition effort. However, one also has to pay a price for this advantage of CBR: correctness and high quality of the output cannot be guaranteed in general. Particularly when employing quite simple distance metrics (e.g. the Hamming or the Euclidean distance), retrieval results are often unsatisfactory. While these *knowledge-poor similarity measures (kpSM)* can be defined with little effort, the drawback of them is, that they are "blind" regarding the application and domain specific requirements. Hence, in many applications more sophisticated and domain specific heuristics to select accurate cases have been used. However,

when employing such *knowledge-intensive similarity measures (kiSM)*, one is confronted again with a serious knowledge acquisition problem. Domain experts able to provide the mandatory domain knowledge and experienced knowledge engineers able to model this knowledge by using formal similarity measures are required. If such experts are available at all, this results in significantly increased development costs for the CBR application.

In order to avoid this drawback of kiSM, we have proposed to support their definition by applying machine learning techniques [9–12]. The basic idea of this approach is to acquire feedback about the quality of retrieval results from which learning algorithms can deduce proper similarity measures automatically. Although experimental evaluations have shown that our framework allows to facilitate the definition of high quality measures significantly, under certain circumstances some open problems might prevent its application in daily practice.

This paper presents some techniques leading to a broader and improved applicability of our framework. The major idea is to incorporate easily available background knowledge into the learning process in order to avoid typical problems of machine learning, such as overfitting.

Section 2 provides some basics of our learning framework and points out the objectives of the extensions presented in this paper. However, since these extensions described in Section 3 and 4 directly put on our previous work, for more details about our learning approach the reader is referred to [12]. Section 5 presents the results of an experimental evaluation demonstrating the power of the described techniques. Finally, Section 6 concludes with some remarks on related work and further research issues.

2 Learning Similarity Measures

In principle, learning similarity measures from some kind of training data is not a novel issue. A lot of work in this direction has been done, for example, in the area of nearest-neighbour classification [13]. Here, one tries to adjust feature weights—which can also be seen as a simple form of a kiSM—by examining pre-classified training data. However, our learning framework [9–12] particularly addressed two novel issues:

Applicability Beyond Classification Domains. In the last years CBR has become very popular in quite different areas, such as e-Commerce or Knowledge Management. Here, traditional learning approaches originally developed for classification tasks are usually not applicable due to the absence of pre-classified data required for learning. One of our objectives was the development of a learning framework suited for a broader range of CBR applications. This framework is based on a particular kind of training data, called *utility feedback*, where the cases' utility has to be estimated by some *similarity teacher*. This similarity teacher might be, for example, a human domain expert, the users of the system or some software agent [9].

Learning of Local Similarity Measures. Besides the definition of attribute weights, successful CBR tools applied nowadays also allow the specification

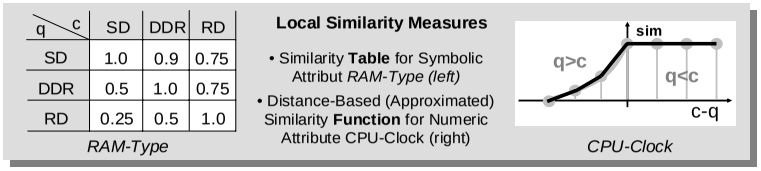

Fig. 1. Modelling Local Similarity Measures

of so-called *local similarity measures*. These measures are responsible for the similarity calculation regarding one single aspect of the case representation[1], i.e. the local similarity between the query and the case value of a particular attribute. Depending on the data type of the attribute, quite different representation formalisms are used for modelling local measures (compare Figure 1). These measures can be used to encode a lot of knowledge about the underlying domain making their definition difficult and time-consuming. One novelty of our learning framework is that it embraces also learning of local similarity measures.

2.1 Usage of Evolutionary Algorithms

When considering attribute weights only, the learning process can efficiently be realised with gradient descent algorithms [13, 9]. However, when extending the learning functionality on local similarity measures, the more complex and heterogeneous representations demand alternative, more flexible learning algorithms.

We have employed evolutionary algorithms which are well-known for their capability to perform well in such environments. The details of our approach – including the representation of local similarity measures as individuals, specialised genetic operators used, an appropriate control algorithm as well as experimental results – are described in [12, 11]. The experiments have shown that the combination of utility feedback and evolutionary algorithms represents a powerful approach for supporting the modelling of kiSM in different application scenarios. However, some open questions and interesting research issues motivate an enhancement of our framework to be presented in this paper.

2.2 Motivation for an Enhancement

Extending learning on local similarity measures leads to a serious problem: a huge search or hypothesis space, respectively, to be treated by the learning algorithm. In particular the representations (see Figure 1) that we have chosen for representing typical local similarity measures [12], namely similarity tables (for unordered data types) and difference-based similarity functions (for ordered data types), consist of numerous parameters enabling the learning algorithm to generate very specific measures. The following reasons motivate an enhancement of the learning algorithms in order to ensure a more goal-directed learning process:

[1] We assume the commonly used attribute-value based representation.

- Especially when obtaining little training data, huge search spaces increase the risk that the learner creates models (here, similarity measures) that fit "too" good to the training data, while showing poor performance and little ability in generalising on some independent test data set. This behaviour is known as "overfitting" in many machine learning approaches.
- Due to the chosen representation, the huge search space is populated with plenty of (local) similarity measures whose usage in practice is highly improbable. Hence, the evolutionary algorithm wastes time searching regions of the search space which do not correspond to "realistic" measures. Since evolutionary algorithms are well-known to be very demanding regarding computational resources, a speeding up of the learning process will increase the applicability of our framework in commercial practice.
- Completely manual definition and fully automated learning of knowledge-intensive similarity measures are two extremes of a wide spectrum of modelling possibilities [10]. Both approaches are coupled with certain advantages and disadvantages. For example, learning might result in suboptimal measures (e.g. caused by overfitting) while a manual definition might lead to unacceptable development costs. In order to benefit of the advantages of both approaches, it seems promising to apply a hybrid approach. This means, one should consider easily available knowledge by defining it directly, while learning unknown or uncertain knowledge by applying our framework.

The three mentioned issues are the foundation of the extensions presented in the following. Here, the basic idea is to exploit easily available background knowledge in order to restrict the hypothesis space to be considered by the evolutionary algorithm. Therefore, we identified two main sources of background knowledge that can be exploited easily and utilised to improve the process of learning similarity measures.

3 Determination of Knowledge Sources

The knowledge sources we employ to enhance the learning can be divided into two groups: Firstly, the used representation of local similarity measures allows us to define several forms of *meta knowledge* representing general demands on the appearance of learnt measures. Secondly, the aid of a knowledge engineer and the incorporation of his/her *expert knowledge* into the learning process may be rewarding (see Section 3.2).

3.1 Similarity Meta Knowledge

Experience in the practical usage of CBR systems has shown that most of the similarity measures that are applied feature certain characteristics. We refer to that kind of experience as similarity meta knowledge.

3.1.1 Heuristic Constraints

There are awkward definitions for similarity measures (e.g. non-reflexive) which actually contradict the CBR paradigm, according to which similar problems have similar solutions, and which thus are very improbable to be used in practice. Nevertheless, those peculiar measures are part of the search space and, accordingly, have the chance of getting involved in the learning process. Defining a set of basic heuristic constraints, that should be fulfilled by local similarity measures, we intend to exclude unusual, highly improbable metrics from the search and thus to restrict the search space.

Reflexivity Constraint Most similarity measures employed in practice are supposed to be reflexive, as any entity can be characterised as being maximally similar to itself. Hence, if there is no justified reason to make use of non-reflexive similarity measures, the constraint for reflexivity of a local similarity measure sim_A for attribute A seems to be advantageous:

$$c_{refl} \;=\; \ll sim_A(x,x) = 1 \;\; \forall x \in D_A \gg$$

Symmetry Constraint The idea of introducing a symmetry constraint is inviting, since its application would approximately halve the search space. In the case of a symbolic attribute A_s with $|D_{A_s}| = n$, for example, the number of alterable entries in the respective similarity measure (individual) sinks from n^2 to $\frac{n^2+n}{2}$. The symmetry constraint is denoted as follows:

$$c_{symm} \;=\; \ll sim_A(x,y) = sim_A(y,x) \;\; \forall x,y \in D_A \gg$$

Of course, in many application scenarios, asymmetric local similarity measures are indispensable (e.g. in product recommendation systems).

Monotony Constraint According to common understanding, larger distances between the query's and case's value of an attribute make them more dissimilar, while smaller ones let them appear rather similar. Based on that foundation, we now define a constraint for monotony (only applicable to distance-based similarity functions, i.e. for ordered data types):

$$c_{mon} \;=\; \ll sim_A(x,y_2) - sim_A(x,y_1) \leq 0 \;\; \forall x,y_1,y_2 \in D_A$$
$$\text{with } 0 \leq y_1 - x \leq y_2 - x \text{ or } x - y_2 \leq x - y_1 \leq 0 \gg$$

3.1.2 Mining Knowledge from the Case Base

Considering a single local similarity measure sim, one can say that the similarity knowledge that is included in sim is distributed over several elements: in the case of similarity tables over its n^2 entries and in the case of similarity functions over a few parameters or sampling points describing that function.

Without any doubt, some of the measure's elements can be characterised as carrying "more valuable knowledge" insofar as they are consulted more frequently. That means they are more frequently used to determine a query's and a case's similarity regarding the respective attribute. Therefore, we ought to aim that those parts of the measure are outmost correct, as an erroneous similarity definition in these regions would have a higher negative impact on the CBR system's overall performance. An example for such a region is, in the case of a distance-based similarity function the area around the y-axis, i.e. all $sim_A(x,y)$

with $|x-y|$ near zero. For a learning algorithm this implies that it should focus more on such "high-importance regions", searching more thoroughly there, while it should be permitted to spend less effort in other "low-importance regions".

But, which regions are of high importance? We intend to answer that question in the following by means of a statistical case base analysis. In so doing, we want to find out (specifically for single attributes and local similarity measures, respectively) which entries within a similarity table and which regions of a similarity function's domain are more frequently consulted and which can thus be considered to be of higher importance. We need to stress in prior, that all considerations we are doing here, presume a sufficiently substantial case base which in particular is also representative (in its attribute-value distributions for all attributes) for queries occurring typically in practice.

Illustrating our idea of obtaining knowledge from analysing case data, we present a little example regarding a local similarity measure for a symbolic attribute $A_{optDrive}$, which describes optical disc drives and which might be used in an application scenario where personal computers are described.

We assume a case base CB of 160 cases whose attribute values for $A_{optDrive}$ (with $D_{A_{optDrive}} = \{CD\text{-}R, CD\text{-}RW, DVD\text{-}R, Combo, DVD\text{-}RW\}$) are set according to Figure 2. That distribution is called attribute-specific value frequency (cf. Definition 1) and we proceed on the assumption that it is representative and reflects the frequency with which certain values from $D_{A_{optDrive}}$ occur in practice.

Definition 1 (Attribute-Specific Value Frequency). *Let A be an attribute with domain D_A and $CB = \{c_1, \ldots, c_m\}$ a case base. The* ***attribute-specific value frequency*** *h_A is defined as*

$$\begin{aligned} h_A : D_A &\to \mathbb{N} \\ x &\mapsto \textstyle\sum_{i=1}^{m} \delta(c_i.A, x) \end{aligned}$$

where $\delta(x,y) = 1$, if $x = y$, and $\delta(x,y) = 0$, else.

To find out which combinations of certain case values and query values occur how often (and thus to find out which elements of the respective similarity table are supposed to be consulted how frequently), we pursue a leave-one-out test strategy as formalised in Definition 2: Each $c \in CB$ is used once as a query, while being excluded from the case base at that point of time, and it is summed up how many table look-ups (separately for each table element) have to be performed when carrying out an entire retrieval with c as query.

Although our example was designed for a symbolic attribute we want to emphasise that this analysis can be applied to numeric attributes as well. Then, however, the determined consultation frequencies refer to specific case-query distances, i.e. parts along the similarity function's x-axis instead of single elements within a similarity table.

Definition 2 (Local Similarity Measure Consultation Frequency). *Let $CB = \{c_1, \ldots, c_m\}$ be a case base and let A be an attribute with domain $D_A = [A_{min}, A_{max}]$, if A is numeric, and $D_A = \{d_1, \ldots, d_n\}$, if A is symbolic, respectively. Moreover, let sim_A be a local similarity measure for A an let h_A be the attribute-specific value frequency. Then, the* ***consultation frequency*** *of sim_A and D_A, respectively, is the auto correlation of h_A, which is defined as*

– *if A is numeric:* $c_n : D_A \to \mathbb{N}$

$$d \quad \mapsto \begin{cases} \sum_{i=-\infty}^{\infty} h_A(i) \cdot h_A(i-d) & \text{if } d \neq 0 \\ \sum_{i=-\infty}^{\infty} h_A(i) \cdot (h_A(i)-1) & \text{else} \end{cases}$$

– *if A is symbolic:* $c_s : D_A \times D_A \to \mathbb{N}$

$$(x,y) \quad \mapsto \begin{cases} h_A(x) \cdot h_A(y) & \text{if } x \neq y \\ h_A(x) \cdot (h_A(x)-1) & \text{else} \end{cases}$$

In that definition the mentioned leave-one-out strategy is taken into consideration by the "else" cases: If a certain case c_j is used as query during (leave-one-out) retrieval and in so doing is excluded from CB, the number of cases c_i, for which it holds $c_i.A = c_j.A$, must be decreased by one. Figure 2 summarises the results for the given example and highlights high vs. low importance regions of $sim_{optDrive}$'s similarity table with different shades of gray.

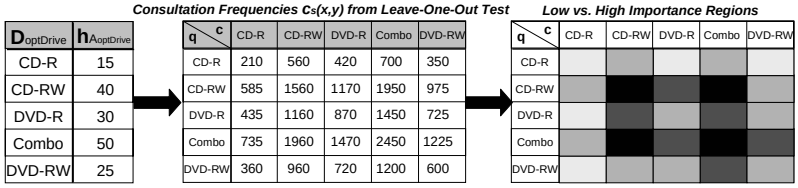

Fig. 2. Example for Low and High Importance Regions in a Similarity Table

3.1.3 Exploitation of the Knowledge

The crucial issue, after having defined heuristic constraints and/or conducted a case base analysis, concerns the employment of the knowledge obtained. In the former case a straightforward application of the constraints is possible: Local measures not conforming to those heuristics are disregarded and excluded from the search. For knowledge about high/low importance regions we have defined two approaches by means of which the knowledge about consultation frequencies c_n or c_s can be incorporated into the optimisation process.

a) Granularity Restriction

Each entry v in a similarity table as well as each sampling point's similarity value v can be chosen from $[0;1] \subset \mathbb{R}$. Accordingly, a self-evident strategy to efficiently restrict the search space is to introduce a grid for the respective similarity value v forcing it to be an element of $\{0, \frac{1}{g}, \ldots, \frac{g-1}{g}, 1\}$, where $g \in \mathbb{N}^+$ determines the allowed degree of granularity.

The results from a statistical case base analysis may be perfectly used to determine appropriate granularities. On the one hand, the consultation frequency $c_s(x,y)$ of a specific entry in a similarity table may be very high. So, this entry is supposed to be consulted very frequently and therefore should be adjusted as accurate as possible. Moreover, this reveals that the case base contains a lot of information about the combination "x as query value and y as case value". Then,

one may conclude that learning of $sim_A(x, y)$ is less vulnerable to overfitting and that its value may be chosen on the basis of a higher granularity.

On the other hand, $c_s(x, y)$ may be rather low. Then, a fine-grained definition for $sim_A(x, y)$ is, generally speaking, not necessary or, at least, not very important (regarding the CBR system's overall performance). Furthermore, the low value of $c_s(x, y)$ suggests that CB features little information about that case-query combination for attribute A only. Hence, learning $sim_A(x, y)$ suffers from a high risk of overfitting, so that the restriction of $sim_A(x, y)$ to a comparatively low number of possible values seems promising.

The following definition introduces our approach to restrict the search space via granularity levels.

Definition 3 (Granularity Values from Consultation Frequencies). *Let A be an attribute with domain D_A, CB a case base of m cases and sim_A the local similarity measure under consideration*

- *where $sim_A(x, y) \in [0, 1]$ with $x, y \in D_A$, if A is symbolic*
- *with sampling points $s_k \in [A_{min} - A_{max}, A_{max} - A_{min}]$ ($k \in \{1, \ldots, s\}$) and belonging similarity values $sim_{s_k} = sim(x, y)$ with $x - y = s_k$, if A is numeric.*

Then, it holds for the ***granularity value*** *g:*

- *if A is symbolic:* $g = 1 + \left\lceil \frac{c_s(x,y)}{m^\gamma} \right\rceil$
- *if A is numeric:* $g = 1 + \left\lceil \frac{\int_{s_k-\delta}^{s_k+\delta} w(x,s_k) \cdot c_n(x) dx}{m^\gamma} \right\rceil$ *with* $\delta = \frac{2 \cdot (A_{max} - A_{min})}{s-1}$ *and* $w(x, y) = 1 - \frac{|x-y|}{\delta}$

where γ is a parameter to scale the entire granularity assessment.

In the case of dealing with a symbolic attribute, the consultation frequencies can be employed directly to determine appropriate granularity values. As A's domain is continuous, if it is a numeric attribute, the computation of integrals is necessary. However, because the amount of available case data is finite, that calculation scales back to forming according finite sums instead of integrals. In practical tests we found that setting $\gamma = 1$ produces convincing results. While this granularity approach may be employed for all kinds of local measures, the following strategy works for numeric attributes only.

b) Modified Sampling Point Distribution

In [12] we have presumed that the sampling points, that are used to represent a distance-based similarity function sim_A, are distributed uniformly over $I_A = [A_{min} - A_{max}, A_{max} - A_{min}]$ of sim_A. As mentioned above some regions of I_A may be of less importance than other ones—namely those regions with a rather low consultation frequency c_n. Consequently, in those regions we actually need not to interpolate sim_A as elaborate as in other, high-importance regions. So, less sampling points ought to be placed in the former, while a higher sampling rate and thus a better approximation of sim_A seems suitable for the latter regions. In short, an equidistant distribution of sampling points does not correspond to nuances in consultation frequency. In [7] we present an algorithm that distributes the sampling points with respect to computed consultation frequencies.

3.2 Expert Knowledge

Without learning functionality, a similarity measure's competence relies exclusively on the expert modelling it. Contrary, if the knowledge engineer lacks sufficient domain knowledge, the only way to obtain adequate retrieval knowledge is via machine learning (provided that training data is available). Thus, as proclaimed in Section 2.2 a combination of these two converse ways to define similarity measures permits several advantages:

- The knowledge acquisition effort is decreased. The expert first issues his/her (partial) knowledge about the respective measure, and the remaining parts of the similarity measure are learnt automatically.
- Since the expert is not urged to specify the similarity measure completely, he/she is not forced to make educated "guesses" about elements of the measure he/she actually does not know much about.
- Due to the additional partial knowledge given by the expert, the learner might less likely tend to overfit its learning results to the training data.

In the following we examine three strategies to incorporate a knowledge engineer's partial knowledge into the learning process.

3.2.1 Attribute and Weight Preferences

Feature weights have a crucial influence on the entire similarity calculation [9]: A wrong choice of weights can distort the entire similarity assessment, even if a lot of effort has been put into tuning local measures. Experts usually do not care or cannot say, whether A_1 is two or three times as important as A_2. We argue that it is much easier for an expert to formulate a number of preference relations by which he/she can determine a partial order of weights, ordered with respect to the relevance of the corresponding attributes. The task of assigning concrete numerical values to the weights can then be left to a learning algorithm.

An expert may, for example, utter that he/she considers A_1 to be less important than A_3, A_2 to be more important than A_3, while having no idea about A_4's importance. The number of allowed weight values for $w_1, \ldots, w_4$—and thus the number of corresponding individuals that may be created in the scope of an evolutionary algorithm—is reduced when those relational constraints are taken into consideration. Of course, the degree of restriction depends on the number of preference relations the expert is able to provide.

3.2.2 Expert Estimations

The situation for local similarity measures, is even worse, because here a large number of concrete parameter or similarity values has to be devised for each attribute. As a consequence, many of the values that have to be determined during a manual definition of a similarity measure are left to the expert's intuitiveness and thus mostly represent estimations of the correct value only. Nevertheless, even estimated values may help to support and bias the learning process:

Bounds Approach An obvious possibility enabling an expert to directly cut off parts of the search space is by allowing him/her to define lower and upper bounds for specific similarity values. For instance, he/she may decide that for a symbolic attribute's values d_1 and d_2 it holds: $sim(d_1, d_2) \in [0.6; 0.8]$. Consequently, the learning algorithm does not have to take the whole interval $[0; 1]$ into consideration for $sim(d_1, d_2)$, but only that subinterval.

Confidence-Based Approach In response to the increased specification effort of the bounds approach (e.g. $2n^2$ bound values for a similarity table) we also introduce a more human-centred, confidence-based approach. Here, the expert may (eventually only partially) define the similarity measure in the usual manner, but is allowed to add an assertion about his/her level of confidence $c \in C = \{uncertain, low, average, high, certain\}$ regarding his/her specification. For example, he/she may state $sim(d_1, d_2) = 0.7$ and $c_{sim(d_1,d_2)} = high$ to express that it holds $sim(d_1, d_2) \approx 0.7$.

The specificational effort is less for the second approach as confidence levels may be defined for an entire measure as a whole or a part of it (e.g. for a number of rows in a table). The semantic of levels from C may be interpreted differently by the learner, e.g. depending on the application domain or on the knowledge engineer's experience. This means, the search space restriction induced by confidence levels is not as strict and inflexible as the one induced by bound specifications.

3.2.3 Exploitation of Structured Data Types

The knowledge engineer is fully responsible for the definition of the CBR system's vocabulary knowledge as well. That task comprises not only the definition of an appropriate case representation, but also the determination of attributes and their data types. Knowledge about the vocabulary, in particular about structured data types, can be employed a-priori to restrict the search space. With the term "structured data types" we here refer to symbolic data types, namely taxonomic and ordered symbolic data types, whose values $D = \{d_1, \ldots, d_n\}$ can be arranged in a tree structure or total order, respectively. Of course, that taxonomic/total ordering effects the belonging local similarity measures and the way the similarity between a query and a case value is computed [2].

Our approach to exploit that kind of vocabulary knowledge is primarily based upon employing a more compact representation of local measures as an evolutionary algorithm's individuals (more compact compared to a similarity table with n^2 entries). For instance, for taxonomic symbolic attributes we defined similarity tree individuals (consisting maximally of $2n-1$ entries), devised appropriate specialised genetic operators for those individuals and thus enabled the EA to directly operate on tree structures. The actual search space restriction is reached not only by that more compact representation of local measures, but also via implicit constraints (e.g. restrictions of the similarity values to be associated with single tree nodes) that hold for parts of taxonomic/ordered symbolic individuals. A more detailed description of that approach can be found in [7].

4 Incorporation of Background Knowledge

As mentioned in the previous section it is one of our aims to restrict the search space by exerting a bias on the respective learning algorithm so that it prefers certain regions of that space to other ones or completely avoids searching some subspaces of the entire search space.

4.1 Knowledge-Based Optimisation Filters

In order to realise the restriction of the search space we introduce the concept of *knowledge-based optimisation filters (kbOF)* restricting the search space. With that term we refer to entities that, on the one hand, hold the gathered knowledge concerning the learning of similarity measures. On the other hand, they are meant to play an active role to explicitly direct the search. As similarity measures are composed of several elements (e.g. a local measure for each attribute of the chosen case representation), we should avoid using a single filter for the entire measure. Instead, the definition of a special kbOF for each attribute (i.e. each local measure) as well as one additional filter for the feature weights is necessary. Hence, for a case representation consisting of n attributes we need $n + 1$ filters.

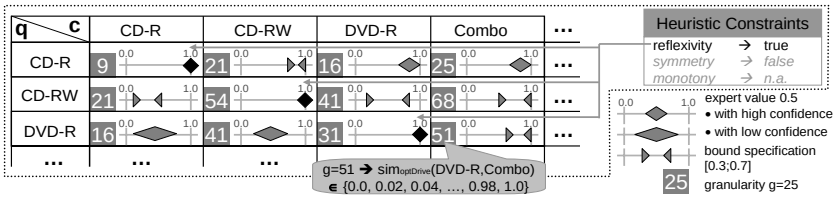

Fig. 3. Example of a Knowledge-Based Optimisation Filter for $A_{optDrive}$

In particular, each kbOF for a local similarity measure (we disregard feature weights here) may include the following pieces of knowledge:

- set of heuristic constraints (e.g. constraint for symmetry)
- set of granularity values g
- list of non-equidistantly distributed sampling points for numeric attributes
- set of bound specifications (subintervals of $[0, 1]$)
- set of expert-estimated similarity values with corresponding confidence levels
- characteristic information on how to exploit structured data types, only for (taxonomic and ordered) symbolic attributes

In Figure 3 we give a visualisation of a possible kbOF for the example of attribute $A_{optDrive}$ introduced in Section 3.1.2.

4.2 Intervening the Learning Process

An important question is how a kbOF interferes with the search process in order to exert a bias toward certain types of similarity measures. Our approach

considers the filter's background knowledge already during the creation of new hypotheses, i.e. during the creation of new candidate similarity measures. Here, the kbOFs' task is to supervise and control the generation of new candidates in such a way that no (or as few as possible) contradictions to its prior knowledge occur, i.e. hard constraints must always and soft ones should mostly be met.

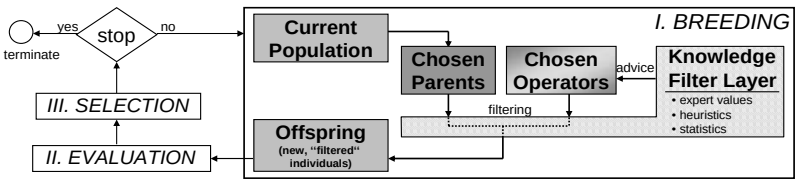

Fig. 4. Intervention of Knowledge-Based Optimisation Filters

In the context of an EA this bias is supposed to be done during the generation of new individuals in the breeding stage of the evolutionary loop. At this, the kbOFs exert their influence at the level of chromosomes: Based on the information they carry, these filters may make some values for the genes of a new individual more probable or forbid other values, for instance. They are also allowed to adapt the behaviour of mutation and crossover operators by giving advice to them in form of particular parameter values. In Figure 4 we outline the simplified control algorithm with its three main phases and show how knowledge filters can intervene the creation of offspring in general. The refined breeding phase of the EA is extended by the layer of kbOFs, in which the actual intentions (concerning offspring creation) of the evolutionary algorithm are "filtered" so that they are in accordance to the filter's knowledge. Hence, a kbOF uses its heuristic, expert and statistical knowledge to manipulate the creation of descendants.

5 Evaluation

The main focus of our experiments is laid upon a comparison of filterless and kbOF-enhanced learning of similarity measures. We have chosen a set of classification and solution prediction[2] application domains from the UCI Machine Learning Repository[3] and used our learning framework to learn the similarity measures in such a way that the CBR system's prediction accuracy is maximised. The CBR system's classification decision and solution prediction are based on a specialised k-nearest neighbour approach [7].

5.1 Experimental Settings

The number of presented techniques to improve the learning of similarity measures is numerous and, accordingly, the number of definable kbOF is extremely

[2] Here, the accurate numeric value of a so-called solution attribute has to be predicted.
[3] `http://www.ics.uci.edu/~mlearn/MLRepository.html`

high as well. Hence, to structure out experiments we have identified a number of *classes* of kbOF we compare against one another:

m-Filters contain similarity meta knowledge only (cf. Section 3.1). This means, they may make use of the reflexivity, symmetry or monotony constraint, for example, or of an arbitrary combination of them. Moreover, they are allowed to employ the knowledge mined from the case base, e.g. by introducing a grid all similarity values have to fit in. It is important to emphasise, that m-Filters are of special interest since the knowledge acquisition effort to define them is only marginal.

e-Filters are enhanced via specific expert knowledge. Hence, those knowledge-based optimisation filters can include bound specifications, expert estimations, confidence levels etc. Although e-Filters are also permitted to exploit the advantages of structured (taxonomic and ordered symbolic) attributes, none of our experiments covers these opportunities.

me-Filters represent the combination of the former ones, incorporating both kinds of additional knowledge to guide the optimisation process.

Apart from comparing the effect of utilising several filter types we also distinguished between different amounts of training examples used for learning. In each experiment (consisting of 10 repetitions for each domain) we have split the set of available cases into a training and test data set, CB_{train} and CB_{test}. Then, we started the learning of similarity measures for incrementally increasing subsets (15, 25, 50, ... cases) of CB_{train} and calculated average classification and solution prediction (i.e. numeric difference between correct and predicted value of the solution attribute) accuracies, respectively, on CB_{test}.

5.2 Results

All in all, the employment of knowledge-based optimisation filters led to improved learning results. The magnitudes of achieved improvements, however, differed enormously over the various application domains. In most cases, the me-Filter produced the highest learning improvements, which is plausible as those kbOFs had been enriched by the maximal amount of available background knowledge. Furthermore, it became obvious that the incorporation of expert knowledge in general generates higher gains than the usage of similarity meta knowledge only. This is not too surprising insofar as expert knowledge can be described as more exhaustive and substantial than similarity meta knowledge. That kind of outperforming, however, must be paid with the higher knowledge acquisition effort that has to be invested when employing expert knowledge.

In Figure 5 we summarise the achieved error reductions (exemplarily for two of the application domains we have chosen) for increasing training data sizes (x-axis). The baseline similarity measure represents, on the one hand, a knowledge-poor (default) similarity measure, into whose construction no further knowledge engineering effort has been put, i.e. the similarity assessment is here based on an uninformed syntactic match. On the other hand, we illustrate the accuracies that resulted from a similarity measure that was obtained from filterless learning.

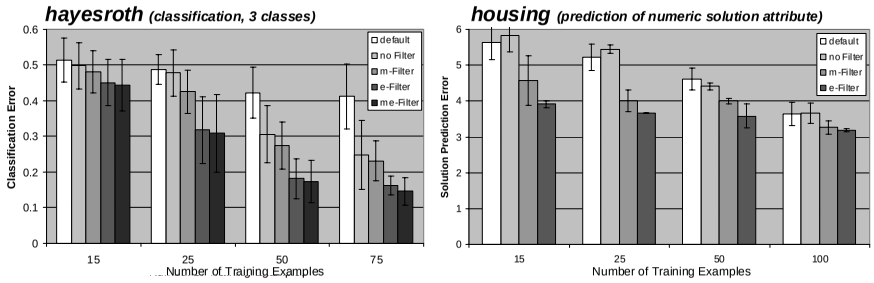

Fig. 5. Achieved Results for two Example Domains

The charts' subsequent data rows sketch the increased accuracies that could be gained due to the incorporation of explicit background knowledge through the mentioned three types of kbOFs.

So, for example, in the *hayesroth* domain (left chart) the additional knowledge yields the creation of high-quality similarity measures an thus of a k-nearest neighbour CBR classifier which produces a $7\% - 17\%$ smaller classification error. In the *housing* scenario (prediction of real-estate prices) the resulting CBR system's prediction accuracy is increased clearly (relative improvements between 15% and 35%), especially for small training data sizes which means that the system's susceptibility to overfitting is reduced. Similar results could be achieved for the remaining application domains as well [7].

Although the additional learning improvements, that could be obtained due to the utilisation of kbOFs compared to filterless learning, varied with respect to application domain and training data sizes used, in Table 1 we summarise the average "kbOF gains" in percent ($100\% \cdot \frac{error_{filtered}}{error_{unfiltered}}$).

$\lvert S_{train}\rvert$	15	25	50	100	200
no kbOF	100.0%	100.0%	100.0%	100.0%	100.0%
m-Filter	93.6%	86.9%	90.1%	92.8%	93.6%
e-Filter	83.2%	78.4%	80.9%	87.6%	94.5%
me-Filter	83.2%	73.4%	80.2%	85.8%	91.9%

Table 1. Relative Reductions of Classification/Solution Predicition Errors due to kbOFs (averaged over 6 application domains)

6 Related Work and Conclusion

Besides the classic work to feature weight learning (for an overview see [13]), in the last years also learning of user preferences has come into focus of research [5, 6]. Concerning the exploitation of re-ranking feedback some work can be found in [6, 14]. While these approaches usually apply gradient descent algorithms, Jarmulak et al. presented an approach based on a genetic algorithm [8]. However, all these learning approaches are restricted to learning weights only and do not incorporate local similarity measures.

In this paper we have presented further improvements of our learning framework [9–12] which particularly addresses two novel issues: learning of local similarity measures and a broad applicability beyond classification tasks. The core idea of these improvements is a restriction of the search space to be considered during learning by exploiting different sources of background knowledge. While we could show the power of our learning techniques in several test domains, an application in commercial practice is still outstanding. Moreover, an interesting issue for further research may be an extension of our framework on more sophisticated similarity measures, e.g. like required for object-oriented case representations [4].

References

1. A. Aamodt and E. Plaza. Case-based reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(1):39–59, 1994.
2. R. Bergmann. On the Use of Taxonomies for Representing Case Features and Local Similarity Measures. In *Proceedings of the 6th German Workshop on Case-Based Reasoning*, 1998.
3. R. Bergmann, M. Michael Richter, S. Schmitt, A. Stahl, and I. Vollrath. Utility-Oriented Matching: A New Research Direction for Case-Based Reasoning. In *Professionelles Wissensmanagement: Erfahrungen und Visionen. Proceedings of the 1st Conference on Professional Knowledge Management*. Shaker, 2001.
4. R. Bergmann and A. Stahl. Similarity Measures for Object-Oriented Case Representations. In *Proceedings of the 4th European Workshop on Case-Based Reasoning*. Springer, 1998.
5. K. Branting. Acquiring Customer Preferences from Return-Set Selections. In *Proceedings of the 4th International Conference on Case-Based Reasoning*. Springer, 2001.
6. L. Coyle and P. Cunningham. Exploiting Re-ranking Information in a Case-Based Personal Travel Assistent. In *Workshop on Mixed-Initiative Case-Based Reasoning at the 5th International Conference on Case-Based Reasoning*. Springer, 2003.
7. T. Gabel. Learning Similarity Measures: Strategies to Enhance the Optimisation Process. Master thesis, Kaiserslautern University of Technology, 2003.
8. J. Jarmulak, S. Craw, and R. Rowe. Genetic Algorithms to Optimise CBR Retrieval. In *Proceedings of the 5th European Workshop on Case-Based Reasoning*. Springer, 2000.
9. A. Stahl. Learning Feature Weights from Case Order Feedback. In *Proceedings of the 4th International Conference on Case-Based Reasoning*. Springer, 2001.
10. A. Stahl. Defining Similarity Measures: Top-Down vs. Bottom-Up. In *Proceedings of the 6th European Conference on Case-Based Reasoning*. Springer, 2002.
11. A. Stahl. *Learning of Knowledge-Intensive Similarity Measures in Case-Based Reasoning*. Ph.D. thesis, Technical University of Kaiserslautern, 2003.
12. A. Stahl and T. Gabel. Using Evolution Programs to Learn Local Similarity Measures. In *Proceedings of the 5th International Conference on Case-Based Reasoning*. Springer, 2003.
13. D. Wettschereck and David W. Aha. Weighting Features. In *Proceeding of the 1st International Conference on Case-Based Reasoning*. Springer, 1995.
14. Z. Zhang and Q. Yang. Dynamic Refinement of Feature Weights Using Quantitative Introspective Learning. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 1999.