Towards a Better Understanding of Memory-Based Reasoning Systems*

John Rachlin Simon Kasif
Steven Salzberg
Dept. of Computer Science
Johns Hopkins University
Baltimore, MD 21218
lastname@cs.jhu.edu

David W. Aha
Navy Center for Applied Research in AI
Naval Research Laboratory
Washington DC 20375
aha@aic.nrl.navy.mil

Extended Abstract

As scientific databases continue to grow, the analysis of scientific data becomes an increasingly important application area for machine learning research. Much of the research on scientific data is directed towards predicting properties of a physical process. Such processes are often described in terms of a function defined over the attributes of the domain or a stochastic model such as a Markov chain. The first question one should ask when studying such problems is: what is the best machine learning technique to use for this problem domain? A general solution to this problem remains elusive. It has been argued (e.g., by the case-based reasoning community) that instead of producing a description of the problem domain in terms of logical rules, functional descriptions, or a complex statistical model, it is possible to store a collection of memories (cases) and perform prediction by interpolating from them. In this practical context, the second question to ask is: are memory-based reasoning (MBR) methods as effective for a given problem domain as more complex approaches (e.g., probabilistic networks). This paper addresses this question both experimentally and formally.

We were originally motivated to study the relative strengths and weaknesses of probabilistic vs. memory based reasoning approaches because of the recent application of these methods to the analysis of biological data such as DNA and RNA sequences, proteins, and genetic mapping data. Consider, for example, the problem of predicting the secondary structure of a protein from its primary amino acid sequence. These amino acids have many important physical and chemical properties (e.g., hydrophobicity, polarity, and various geometric features) that influence their folding behavior. Thus, the solution to this fundamentally important problem from molecular biology might be expressed as a function of a large number of (currently unknown) variables. Alternatively, it may be modelled as a stochastic process involving the forces between the atoms that comprise the protein. Although both approaches are areas of active research, a precise description of the physical process and all the relevant attributes is not known. Instead, most current methods approach this problem using a symbolic representation for the 20 amino acids as the primary input. Partially as a result of this representational limitation, most current approaches for predicting protein structure have met with only moderate success, typically achieving about 64% accuracy. More intricate methods obtain slightly better results, but the problem is still largely unsolved. Two relatively successful techniques for this and similar problems in computational biological have been Bayesian and MBR methods (Zhang, Mesirov, & Waltz 1992; Cost & Salzberg 1993; Yi & Lander 1993; Delcher *et al.* 1993). In addition, there are other prominent domains where MBR and probabilistic methods are currently competing for best performance (e.g., speech recognition and natural language parsing).

We therefore decided to perform a detailed study of the relative strengths of Bayesian and MBR approaches for a range of different prediction tasks involving symbolic attributes. As a baseline for the comparison, we also demonstrate the performance of a nearest neighbor algorithm using the simple overlap metric. This study is particularly intriguing because of several important similarities that exist between MBR methods (with value-difference metrics) and Bayesian classifiers:

- both methods make the same independence assumptions,
- both require the same computational time and

*The full version of this paper will appear in: *Machine Learning: Proceedings of the Eleventh International Conference*, Morgan Kaufmann Publishers, San Francisco, CA. (1994)

space during training, although the output space complexity and classification time is higher for the MBR algorithm,

- these are the primary methods employed for the study of the fundamental biology problems cited above, and
- the Bayesian classifier abstracts case-specific information into a set of conditional and unconditional probabilities, while MBR stores the cases themselves. Thus, our comparisons highlight the utility of using case-specific information.

Our analysis substantially enhances our understanding of the relative strengths and weaknesses of these methods, and suggests that MBR has a number of important advantages.

Stanfill and Waltz (Stanfill & Waltz 1986; Stanfill 1987) introduced the Value Difference Metric (VDM) to define similarity when using symbolic-valued features and empirically demonstrated its benefits. They argued that the VDM is more appropriate than the simple *overlap* distance measure, which defines distance as the number of mismatching features between two instances. More recently, Cost and Salzberg (Cost & Salzberg 1993) demonstrated that PEBLS, which uses a modification of the VDM distance measure called MVDM, can outperform the overlap measure and several other learning algorithms on a number of practical problems. However, these studies did not perform a detailed analysis of MVDM's learning ability, which is a primary contribution of this paper.

The MVDM used by PEBLS defines the distance between two examples as the sum of the value differences across all features. The value difference between two values v_1 and v_2 of a given feature is defined as:

$$\delta(v_1, v_2) = \sum_{i=1}^{k} \left| \frac{v_{1_i}}{V_1} - \frac{v_{2_i}}{V_2} \right| \qquad (1)$$

where k is the number of classes, v_{1_i} is the number of times v_1 occurred for instances of class C_i and V_1 is the number of times v_1 occurred for all classes. Thus the ratio v_{1_i}/V_1 is the probability of class C_i given that the feature has value v_1, $P(C_i|v_1)$.

The simple Bayesian classifier represents each concept with a single probabilistic expression, defined by class base rates and feature probability distributions. For each class C_i and feature value v_j, we estimate $P(v_j|C_i)$ from the training data. A new point is classified into C_i if $P(C_i)\prod_j P(v_j|C_i)$ is maximal.

Computationally, PEBLS computes the same statistics as the Bayesian classifier. However, while Bayes summarizes these statistics in simple rules, PEBLS uses them as part of an MBR classifier.

In summary, we present in the full version of this paper a methodology aimed at trying to evaluate an MBR system based on particular performance criteria. This approach has two important components: 1) it examines learning problems of varying complexity in a controlled attempt to better understand the *specific* conditions under which a reasoning system will perform well, and 2) it applies standard techniques to the same problems as a baseline for comparison. By this approach we have added significantly to our understanding of these types of systems.

Acknowledgements

This paper is based on work supported by the National Science foundation under Grants No. IRI-9223591 and IRI-9220960.

References

Cost, S., and Salzberg, S. 1993. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning* 10(1):57–78.

Delcher, A. L.; Kasif, S.; Goldberg, H. R.; and Hsu, B. 1993. Probabilistic prediction of protein secondary structure using causal networks. In *Proceedings of the 11th National Conference on Artificial Intelligence*, 316–321.

Rachlin, J.; Kasif, S.; Salzberg, S.; and Aha, D. W. 1994. Towards a better understanding of memory-based reasoning systems. In Cohen, W. W., and Hirsh, H., eds., *Machine Learning:Proceedings of the Eleventh International Conference.* San Francisco, CA: Morgan Kaufmann.

Stanfill, C., and Waltz, D. 1986. Toward memory-based reasoning. *Communications of the ACM* 29(12):1213–1228.

Stanfill, C. 1987. Memory-based reasoning applied to English pronunciation. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, 577–581.

Yi, T.-M., and Lander, E. 1993. Protein secondary structure prediction using nearest-neighbor methods. *Journal of Molecular Biology* 232:1117–1129.

Zhang, X.; Mesirov, J.; and Waltz, D. 1992. A hybrid system for protein secondary structure prediction. *Journal of Molecular Biology* 225:1049–1063.

A Case Study on Case-Based and Symbolic Learning* (Extended Abstract)

Stefan Wess and Christoph Globig
University of Kaiserslautern, P.O. Box 3049
D-67653 Kaiserslautern, Germany
{wess, globig}@informatik.uni-kl.de

Abstract

Contrary to symbolic learning approaches, which represent a learned concept *explicitly*, case-based approaches describe concepts *implicitly* by a pair (CB, sim), i.e. by a measure of similarity sim and a set CB of cases. This poses the question if there are any differences concerning the learning power of the two approaches. In this article we will study the relationship between the case base, the measure of similarity, and the target concept of the learning process. To do so, we transform a simple symbolic learning algorithm (the version space algorithm) into an equivalent case-based variant. The achieved results strengthen the hypothesis of the equivalence of the learning power of symbolic and case-based methods and show the interdependency between the measure used by a case-based algorithm and the target concept.

Introduction

In this article (which is a short version of the work presented in (Wess & Globig 1994)) we want to compare the learning power of two important learning paradigms – the *symbolic* and the *case-based* approach (Aha 1991). As a first step in this direction, (Jantke 1992) has already analyzed the common points of inductive inference and case-based learning. Under the term *symbolic learning*[1] we subsume approaches, e.g. (Michalski, Carbonell, & Mitchell 1983), that code the knowledge provided by the presentation of the cases into a *symbolic representation of the concept* only, e.g. by formulas, rules, or decision trees. The learning task we want to study is the classification of objects. The aim of a classification task is to map the objects x of a universe U to concepts $C \subseteq U$, i.e to subsets of the universe. In the most simple scenario we have to decide the membership problem of a certain concept C.

For this *special scenario* we will show that a case-based approach has the same learning power as a symbolic approach. We will therefore present a simple symbolic learning algorithm (the Version Space (Mitchell 1982)) and transform this algorithm into a equivalent case-based variant. Based on this example we will show that for case-based approaches there exists a strong tradeoff between the set of learnable concepts and the minimal number of cases in the case base. We will conclude that for our scenario the used *bias* must have a comparable strength in both approaches.

Basic Algorithm for Case-Based Classification

The fundamental problem the two approaches have to solve during the learning phase is the same. At every moment the learner knows the correct classification of a finite subset of the universe only. The knowledge that the algorithm is able to use is incomplete and, therefore, the computed hypothesis needs not to be correct. In the application phase a case-based system tries to classify a new case with respect to a set of stored cases, the case base CB. For simplicity, we consider cases as tuples $(x, class(x))$ where x is a description of the case and $class(x)$ is the classification. Given a new case $(y, ?)$ with unknown classification, the system searches in the case base CB for the nearest neighbor $(x, class(x))$ (or the most similar case) according to a given measure of similarity[2] $sim : U \times U \rightarrow [0, 1]$. Then it states the

*The presented work was partly supported by the *Deutsche Forschungsgemeinschaft*, SFB 314: "Artificial Intelligence and Knowledge-Based Systems" and the project IND-CBL.

[1] Case-based systems may also use symbolic knowledge. The use of the term "symbolic learning" in this work may therefore be confusing to the reader. But, since the term "symbolic learning" is also used to contrast a special class of learning approaches to systems which use neural networks, we think that the use of the term "symbolic learning" as characterization of these approaches is appropriate.

[2] The dual notion is that of a *distance measure* $d : U \times U \rightarrow \mathcal{R}^+$. In the sequel we will use the term *measure* if we do not want to distinguish between similarity and distance measures. Both types of measures have the same power (Richter & Wess 1991) and we will use them with respect to the context of the examples.

classification $class(x)$ of the nearest neighbor as the classification of the new case $(y, ?)$, i.e $(y, class(x))$. For the basic case-based algorithm cf. (Aha 1991; Aha, Kibler, & Albert 1991).

From the viewpoint of machine learning, case-based learning may be seen as a *concept formation task.* This raises the question how the learned concepts are represented in case-based approaches. Contrary to symbolic learning systems, which represent a learned concept *explicitly,* e.g. by formulas, rules, or decision trees, case-based systems describe a concept C *implicitly* (Holte 1990) by a pair (CB, sim). The relationship between the case base and the measure used for classification may be characterized by the equation:

$$\boxed{\textbf{Concept} = \textbf{Case Base} + \textbf{Similarity Measure}}$$

This equation indicates in analogy to arithmetic that it is possible to represent a given concept C in multiple ways, i.e. there exist many pairs $C = (CB_1, sim_1)$, $(CB_2, sim_2), \dots, (CB_k, sim_k)$ for the same concept C. Furthermore, the equation gives a hint how a case-based learner can improve its classification ability. There are three possibilities to improve a case-based system. The system can (1) store new cases in the case base, (2) change the measure of similarity or (3) change both, the case base and the similarity measure. During the learning phase a case-based system gets a sequence of cases $X_1, X_2, \dots$ with $X_i = (x_i, class(x_i))$ and builds up a sequence of pairs $(CB_1, sim_1), (CB_2, sim_2), \dots, (CB_k, sim_k)$ with $CB_i \subseteq \{X_1, X_2, \dots, X_i\}$. The aim is to get in the limit a pair (CB_n, sim_n) that needs no further change, i.e. $\exists n \; \forall m \geq n \; (CB_n, sim_n) = (CB_m, sim_m)$, because it is a correct classifier for the target concept C.

Case-based systems apply techniques of nearest-neighbor classification in symbolic domains. The basic idea is to use the knowledge of the known cases directly to solve new problems. By directly we mean, that the case-based system does not try to extract explicit knowledge during the learning phase and apply this abstract knowledge during the application phase.

A Case-Based Variant of a Symbolic Learner

To demonstrate the fundamental equivalence of the learning power of symbolic and case-based learners, we transform a well-known symbolic learner – the Version Space (VS) from (Mitchell 1982) – in an equivalent case-based variant. The Version Space algorithm is a simple and well-known symbolic learning algorithm. Because of its simplicity it is easy to show a lot of properties, which hold for many other learning algorithms, where it would be difficult to prove them.

The Symbolic Version Space

The universe U of cases consists of finite vectors over finite value sets W_i ($U = W_1 \times \cdots \times W_n$). We want to decide the membership problem of a certain concept C. The concepts to learn fix the value of certain attributes[3]. We can describe these concepts C as vectors $(C_1, \dots, C_n)$, with $C_i = *$ or $C_i = a_{ij} \in W_i$. A case $((a_1, \dots, a_n), class(a))$ fulfills the concept C, if for all $1 \leq i \leq n$ holds: $C_i = *$ or $C_i = a_i$, i.e. $C_i = *$ is fulfilled by every $x \in W_i$. We further demand that $C_i \neq *$ for at least one i.

A concept C is called consistent with a set of cases, if all positive cases of the set fulfill the concept and none of the negative does. The symbolic version space solves the learning problem by updating two sets S and G of concepts. S contains the most specific concept that is consistent with the known cases and G includes the most general concepts consistent with the known cases. The task of the symbolic algorithm is to change the sets S and G in order to preserve their properties. For the algorithm cf. (Mitchell 1982). It is important that at every moment all cases subsumed by S are known to be positive, and all cases that are not subsumed by any concept of G are known to be negative. This observation leads to a partial decision function $VS : U \to \{0, 1\}$ that can be used to classify new cases:

$$VS(x) = \begin{cases} 1 & \text{if } \forall C \in S[C(x) = 1] \\ 0 & \text{if } \forall C \in G[C(x) = 0] \\ ? & \text{otherwise} \end{cases}$$

As long as $S \neq G$ VS will not classify all cases of the universe. If a case is covered by S but not by G it is not clear whether it belongs to the concept C or not. So VS will not return an answer for those cases (this is the semantics of the "?" in the decision function).

A Case-Based Variant of the Version Space

If we analyze the version space algorithm, it is obvious that the main learning task is to distinguish between relevant and irrelevant attributes. We will use this observation to construct a case-based variant VS-CBR of the algorithm of the previous section. An attribute value is called *relevant,* if it is part of the target concept $C = (a_1, \dots, a_n)$. For every attribute i, we define a function f_i that maps $x \in W_i$ to $\{0, 1\}$ with the following definition:

$$f_i(x) = \begin{cases} 1 & \text{if } C_i = x \\ 0 & \text{otherwise} \end{cases}$$

[3] i.e. these concepts represent the conjunctions of atomic formulas $x_i = a_i$, e.g. $shape = circle \wedge size = big$.

The functions f_i will be combined to $f: U \rightarrow \{0,1\}^n$ $f((a_1,\ldots,a_n)) = (f_1(a_1),\ldots,f_n(a_n))$. The distance between two cases a and b is then defined using the city-block metric as follows:

$$d_f(a,b) := |f_1(a_1) - f_1(b_1)| + \cdots + |f_n(a_n) - f_n(b_n)|$$

It is obvious that every change of the functions $f_1, f_2, \ldots, f_n$ causes a change of the underlying measure d_f. The intended function f_i is learnable by the algorithm in Fig. 1. The algorithm expects the first case to be positive.

```
Algorithm to Learn $f$ for VS-CBR
1. Initialize $f_i(x_i) = 0$ for all $i, x_i \in W_i$
2. Let the first positive case be $((a_1,\ldots,a_n),+)$. Let $f_i(a_i) = 1$ and $CB = \{(a,+)\}$
3. Get a new case $((b_1,\ldots,b_n), class(b))$.
4. If $class(b)$ is negative, store $b$ in the case base $CB$, i.e. $CB := CB \cup \{(b,-)\}$
5. If $class(b)$ is positive and $f_i(b_i) = 0$, then let $f_i(x_i) = 0$ for all $x_i \in W_i$ ($f_i$ maps now every value to zero).
6. If there exist two cases $(a, class(a)), (b, class(b)) \in CB$ with $d_f(a,b) = 0$ and $class(a) \neq class(b)$ then ERROR: The target concept $C$ is not member of the version space.
7. If the concept $C$ is determined then STOP: The concept is learned. The classifier $(CB, d_f)$ consists of the case base $CB$ and the measure $d_f$
8. Go to step 3.
```

Figure 1: Algorithm to learn f for VS-CBR

If the concept is learned, the function f and the case base CB are used for classification. Given a new case $(c,?)$, the set

$$F := \{x \in CB \mid \forall\, y \ \in CB \ d_f(x,c) \leq d_f(y,c)\}$$

is computed. The classification $class(x)$ of the most similar case (x, $class(x)$) is then used for the classification of the new case $(c,?)$. If F contains more than one case and these cases have different classifications then $class(c)$ is determined by a fixed strategy to solve this conflict. Different strategies are possible and each strategy will induce a own decision function for VS-CBR. For example, one conflict solving strategy may state the minimal classification according to a given ordering of the concepts. To solve the membership problem, we assume that a case $(c,?)$ is classified as negative if it has the same minimal distance from a positive and a negative case, i.e. $d((a,+),(c,?)) = d((b,-),(c,?))$ is minimal. To achieve this behavior of the classifier the ordering of the concepts must be *negative* < *positive.*

Analysis

Now let us analyze VS-CBR's way of classification in more detail. Positive and negative cases are used differently in VS-CBR during the learning phase:

- Positive cases are used to change f, i.e. to adapt the distance measure d_f. They will not be stored in the case base (with the exception of the very first positive case).
- Negative cases are stored in the case base CB but do not change the distance measure d.

The information that is used by VS to change S and G is used by VS-CBR to change the case base or the measure of similarity. It is easy to show that all cases which are classified by the symbolic VS will also be classified correctly by the case-based one. The difference is that the case-based variant VS-CBR computes a classification for every case of the universe (because the distance measure is total) while the symbolic VS classifies only if it knows that the proposed classification must be correct. Otherwise (i.e. the case fulfills a concept from G but not the concept in S) it will not produce any classification at all. If we add a test, whether the classification of the nearest neighbor is correct to VS-CBR, we can force VS-CBR to produce only certain classifications, too. But this test would more or less be a variant of the original VS algorithm.

Relationships between CB, sim, and C

We have shown that it is possible to reformulate the Version Space algorithm in a case-based manner so that the case-based variant behaves as the symbolic algorithm. It is important to understand the implications of a measure of similarity to the set of representable concepts.

On one hand, case-based systems (CB, sim) use the cases in the case base CB to fill up the equivalence classes induced by the measure sim. On the other hand, they use the cases to lower the number of equivalence classes by changing the measure sim. Thereby, the target concept C may be identified by fewer cases. But, a lower number of equivalence classes means that the modified measure sim' can distinguish between fewer concepts. Having this in mind, we can be compare case-based systems with respect to two dimensions: *minimality* and *universality*. The first dimension relates to the implicit knowledge that is coded into the used measure *sim*. Because we are not able to measure this implicit knowledge directly, we have to look at the size of the case base instead. More knowledge coded in the used measure *sim* will result in a smaller (minimal) size of the case base CB within the classifier (CB, sim).

Definition 1 *The similarity measure* sim_1 *of a case-based system* (CB_1, sim_1) *is called* better informed *than a measure* sim_2 *of a system* (CB_2, sim_2) *iff both systems are classifiers for the same concept* C, $|CB_1| < |CB_2|$ *holds, and there is no* $CB_i' \subset CB_i$ *so that* (CB_i', sim_i) *is a classifier for the concept* C.

The second dimension relates to the set of learnable concepts. We must distinguish between the representability and the learnability of a concept. A concept C is called representable by a measure *sim*, if there ***exists*** a finite case base CB such that (CB, sim) is a classifier for C. A concept C is called learnable by a measure *sim*, if there exists a ***strategy to build*** a finite case base CB such that in the limit (CB, sim) is a classifier for the concept.

Definition 2 *A similarity measure* sim_1 *is called* more universal *than a similarity measure* sim_2 *iff the set of concepts that are learnable by* sim_2 *is a proper subset of the set of concepts that are learnable by* sim_1.

Using an universal similarity measure conflicts the minimality of the case base. Reducing the size of the case base, which means to code more knowledge into the measure, usually results in a less universal similarity measure. We can distinguish two extreme situations:

All knowledge is coded into the case base: The similarity is maximal if and only if the compared cases are identical, i.e. $sim(x, y) = 1 \iff x = y$, 0 otherwise. The measure is universal because it is able to learn every binary concept C_i in the given universe U. But to do so, it needs the whole universe as a case base, i.e $CB := U$. Thus, the resulting system $(U, =)$ is universal but not minimal.

All knowledge is coded into the measure: The similarity is maximal if and only if the classification of the compared cases $C(x)$ is identical, i.e. the measure of similarity *sim* knows the definition of the concept C to learn. Nearly the whole knowledge about the concept is then coded into the measure. The case base contains almost one positive c^+ and one negative case c^- and is used only to choose between some trivial variations. The measure $sim(x, y) := 1 \iff C(x) = C(y)$ (0 otherwise) can only distinguish between four concepts (C, $\neg C$, *True* – i.e. all cases are positive, *False* – i.e. all cases are negative). Thus, the resulting system $(\{c^+, c^-\}, C(x) = C(y))$ is minimal but not universal.

In a case-based learner, two processes – reducing the size of the set of learnable concepts (hypothesis space) and increasing the size of the case base – should be performed. The measure $sim(x, y) \iff C(x) = C(y)$ indicates a simple way to reformulate any symbolic algorithm in a case-based manner, i.e. use the actual symbolic hypothesis to construct such a measure and store one positive and one negative case in the case base.

Discussion

The symbolic as well as the case-based approach compute a classification when a new case is presented. If only the input and the output of the algorithms are known, we will not be able to distinguish between the symbolic and the case-based approach. The symbolic algorithm builds up its hypothesis by revealing the *common characteristics* of the cases in a predefined *hypothesis language*. The hypothesis describes the *relation between a case and the concept*. One component of a case-based learner is a measure, that states the similarity or the distance between cases. The measure defines a *preference relation* between two cases and is therefore independent from the existence of a concept. A main difference between case-based and symbolic classification algorithms is the representation of the learned concept. A case-based classifier (CB, sim) consists of a case base CB and a measure of similarity *sim*. It is possible to represent the same concept C in multiple ways, i.e. by different tuples (CB_i, sim_i). But, neither the case base CB nor the measure *sim* is sufficient to build a classifier for C. The knowledge about the concept C is spread to both. Thus, the hypothesis produced by a case-based algorithm represents the concept only *implicitly*, while symbolic procedures build up an *explicit* representation of the learned concept.

If the problems and the power of case-based and symbolic approaches are similar as we have seen for our simple scenario, the question arises whether the two approaches can be interchanged in all situations. We assume that we want to get a classifier only and not an explicit description of the concept. In the second case, a case-based system cannot be the appropriate choice. Within this perspective, the symbolic and the case-based approach seem to be interchangeable in the described context. The symbolic approach corresponds to a kind of *compilation process* whereas the case-based approach can be seen as a kind of *interpretation* during run time. Which approach should be used in a concrete situation is a question of an adequate *representation* *of the previous knowledge.* If previous knowledge contains a *concept of neighborhood* that leads to appropriate hypotheses, a case-based approach is a good choice. In this scenario we are able to code the neighborhood principle into the measure used. The case-based approach will then produce good hypotheses before the concept is learned, i.e. when not all equivalence classes of the measure are filled.

We have analyzed the relationship between the measure of similarity, the case base, and the target concept in the described scenario of classification tasks (cf. (Globig & Wess 1994)). The learning algorithm *needs strong assumptions* about the target concept in order to solve its task with an acceptable number of cases. Assumptions exclude certain concepts from the hypothesis space. Symbolic learners use these assumptions to restrict the language to represent their hypotheses. A case-based learner have to code this assumptions into the measure of similarity. These restrictions of the hypothesis space are called *bias.* (Rendell 1986) divides the abstraction done by a learning system in two parts: *the bias* (to describe the amount of assumptions), and the *power of the learner.* We have characterized case-based systems by the *number of learnable concepts* and the *number of cases* they need to identify a target concept. Case-based algorithms use the cases of the case base to fill equivalence classes induced by the measure used. On the other hand, they use the knowledge from the cases to lower the number of equivalence classes by changing the measure. Thereby, the target concept may be identified by fewer cases. The used measure defines the set of the learnable concepts and the cases in the case base select a concept from this set.

The *bias* relates to the restriction of the set of learnable concepts induced by the measure of similarity and is therefore comparable to the *degree of universality.* The *minimal size* of the case base reflects the information the learner needs to come to a correct hypothesis, i.e. the power of the learner (Rendell 1986). Using an universal similarity measure conflicts the minimality of the case base. Reducing the size of the case base, which means to code more knowledge into the measure, usually results in a less universal similarity measure. We have stressed that the measure (respectively the way to modify the measure) is the *bias of case-based reasoning.* Because case-based systems are based on a bias that cannot be deduced from the cases, we reject the thesis (Cost & Salzberg 1993) that case-based classification is more appropriate in situations with a low amount of previous knowledge.

We conclude that for classification tasks there is no fundamental advantage in the learning power of case-based systems as maintained by (Cost & Salzberg 1993). Since the number of cases an algorithm need to learn a concept is directly related to the size of the hypothesis space, the used bias must have a comparable strength in both approaches. While symbolic approaches use this extra evidential knowledge to restrict the language to represent their hypotheses, the case-based algorithms need it to get appropriate measures of similarity.

Acknowledgement: We would like to thank M.M. Richter, K.-D. Althoff, H.-D. Burkhard, and K.P. Jantke for many helpful discussions, and the anonymous reviewers for their comments.

References

Aha, D. W.; Kibler, D.; and Albert, M. K. 1991. Instance-Based Learning Algorithms. *Machine Learning* 6:37–66.

Aha, D. W. 1991. Case-Based Learning Algorithms. In Bareiss, R., ed., *Proceedings CBR Workshop 1991,* 147–158. Morgan Kaufmann Publishers.

Cost, S., and Salzberg, S. 1993. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning* 10(1):56–78.

Globig, C., and Wess, S. 1994. Symbolic Learning and Nearest-Neighbor Classification. In Bock, H.-H.; Lenski, W.; and Richter, M., eds., *Information Systems and Data-Analysis,* 17–27. Springer Verlag.

Holte, R. S. 1990. Commentary on: PROTOS an exemplar-based learning apprentice. In Kodtratoff and Michalski (1990). 128–139.

Jantke, K. P. 1992. Case-Based Learning in Inductive Inference. In *Proceedings of the 5th ACM Workshop on Computational Learning Theory (COLT-92),* 218–223. ACM Press.

Kodratoff, Y., and Michalski, R., eds. 1990. *Maschine Learning: An Artificial Inteligence Approach,* volume III. Morgan Kaufmann.

Michalski, R.; Carbonell, J. G.; and Mitchell, T., eds. 1983. *Machine Learning: An Artificial Intelligence Approach,* volume 1. Palo Alto, California: Tioga.

Mitchell, T. 1982. Generalization as search. *Artificial Intelligence* 18(2):203–226.

Rendell, L. 1986. A general framework for induction and a study of selective induction. *Machine Learning* (1):177–226.

Richter, M. M., and Wess, S. 1991. Similarity, Uncertainty and Case-Based Reasoning in PATDEX. In Boyer, R. S., ed., *Automated Reasoning,* Essays in Honor of Woody Bledsoe. Kluwer Academic Publishing. 249–265.

Wess, S., and Globig, C. 1994. Case-Based and Symbolic Learning - A Case Study. In Wess, S.; Althoff, K.-D.; and Richter, M. M., eds., *Topics in Case-Based Reasoning - selected papers from the First European Workshop on Case-Based Reasoning, Kaiserslautern, 1993.* Springer Verlag.