Partial and Vague Knowledge for Similarity Measures

Timo Steffens
Institute of Cognitive Science
Kolpingstr. 7, 49074 Osnabrueck, Germany
tsteffen@uos.de

Abstract

This paper proposes to enhance similarity-based classification by virtual attributes from imperfect domain theories. We analyze how properties of the domain theory, such as partialness and vagueness, influence classification accuracy. Experiments in a simple domain suggest that partial knowledge is more useful than vague knowledge. However, for data sets from the UCI Machine Learning Repository, we show that vague domain knowledge that in isolation performs at chance level can substantially increase classification accuracy when being incorporated into similarity-based classification.

1 Introduction

One of the most prominent challenges in machine learning is to identify appropriate features for representing instances, since learning performance depends heavily on the features used. Particularly, the performance of similarity-based classification degrades with the number of irrelevant features [Griffiths and Bridge, 1996]. It is also known from work on constructive induction (CI) that adding features can improve classification accuracy [Aha, 1991]. While CI is a bottom-up approach, this paper proposes a top-down approach on identifying abstract features. The focus of CI was mainly on logical and rule-based processes, whereas this paper shows how additional features can extend similarity measures for similarity-based classification.

The main contribution of this paper is to show that additional features can be derived from domain theories that are imperfect. This will alleviate the knowledge acquisition bottleneck, as it reduces the requisites of obtaining expert knowledge. Although similarity-based classification is only used in domains where no perfect domain theories exist, usually there is imperfect domain knowledge and isolated chunks of knowledge [Aamodt, 1994; Bergmann *et al.*, 1994; Cain *et al.*, 1991; Porter *et al.*, 1990]. For example, in [Aamodt, 1994] open and weak domain theories were integrated into a case-based reasoning system. Similarly, matching knowledge was used to improve the performance of the well-known PROTOS system [Porter *et al.*, 1990]. Furthermore, it was shown that the combination of CBR and a domain theory outperforms both CBR and the theory itself [Cain *et al.*, 1991]. In contrast to those weak theories, strong domain theories were used to filter irrelevant features [Bergmann *et al.*, 1994].

We present a new approach that exploits imperfect domain knowledge in similarity-based classification by inferring additional abstract features. Furthermore, we analyze the impact of the knowledge's vagueness and partialness.

The next section specifies the representation of cases, the similarity measure, and domain theories. Section 3 gives an overview over how additional features can improve classification accuracy. Section 4 reports experiments with two domains from the UCI Machine Learning Repository [Blake and Merz, 1998]. Finally, the last section concludes and outlines future work.

2 Representation of the CBR modules

2.1 Cases and the similarity measure:

A case C is made up of a set of attributes $\{A_1, A_2, \ldots, A_n\}$. While the original attributes can be either discrete or numeric, the additional virtual attributes in this paper are binary.

Following the well-known local-global principle, we compute the similarity between two cases as the weighted average aggregation of the attributes' similarities:

$$sim(A, B) = \sum_{i=1}^{n} (\omega_i * s_i)$$

where $s_i = sim(A_i, B_i)$ are the local similarity values, and the ω_i are the corresponding weights.

2.2 Domain knowledge:

In this paper, we examine only domain knowledge that can be represented as a domain theory of the following form: A domain theory is a set of inference rules that relate concepts to each other. These rules specify which concepts exist in the domain and describe how abstract concepts can be inferred from more primitive ones [Bergmann *et al.*, 1994]. For example, consider the relation in the rule $A_1 \leftarrow A_2 \wedge A_3 < A_4$ which says that there are the binary attributes A_1, A_2 and the ordinal attributes A_3, A_4. Also, the rule states that A_1 is satisfied, if A_2 is true and A_3 is smaller than A_4. We assume that the case representation language is compatible with the

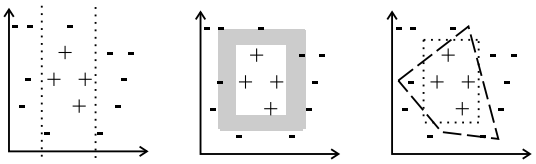

Figure 1: Properties of domain theories. The theories describe parts of the target concept, of which there are positive (+) and negative (-) instances. Left: Partial knowledge, only parts of the concept boundaries are known. Middle: Vague knowledge, concept boundaries are believed to be somewhere within the shaded areas. Right: Inconsistent knowledge, different rules make differing predictions.

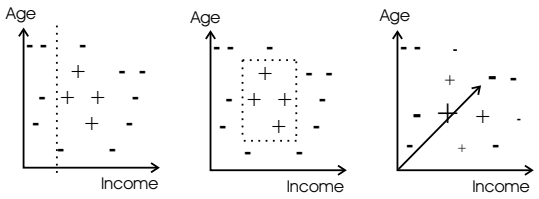

Figure 2: Types of virtual attributes. Left: A binary virtual attributes divides the instance space into instances satisfying or not satisfying it. Middle: A conjunction of binary attributes. Right: The most general type of virtual attributes is to add a dimension to the instance space.

language of the domain theory, either by sharing primitives or by using a bridging language. The formal definition of the domain theory is skipped here, since it is equivalent to Horn clauses, including logical connectors, equality, and comparison operators.

According to [Mooney and Ourston, 1991], the concepts in a domain theory can be divided into three types: observables are attributes that are directly represented in the cases. Classification goals are attributes that are to be inferred or approximated. All other attributes are called intermediates.

Intermediate attributes are the focus in this paper, because they are natural candidates for virtual attributes, that is, they can be added to the similarity measure in order to enhance the classification accuracy.

2.3 Properties of domain theories:

Domains in which CBR is applied usually lack a perfect domain theory. Hence, the domain theories (or parts thereof) that we work with have at least one of the following properties (cf. Figure 1):

- **Partialness:** This is the case if some parts of the domain are not modelled, for example a) if conditions are used but not defined, or b) the relation of intermediates or observables to the classification goal is not known, or c) the classification goal does not exist in the rulebase at all. Note that these situations correspond to gaps at the "top" or "bottom" of the domain theory [Mooney and Ourston, 1991].
- **Vagueness:** Values can only be given within a certain confidence interval. If a value is picked from such an interval, it is likely to be incorrect.
- **Inconsistency:** There are two or more rules (or even alternative theories) that make different classifications and it is not known which one is correct. CBR is often used to overcome this problem, because the cases provide knowledge on which classification is correct for certain cases.

In this paper, we focus on partial and vague theories.

3 Virtual attributes

Virtual attributes [Richter, 2003] are attributes that are not directly represented in the cases but can be inferred from the already existing attributes. They are useful if the monotonicity-principle is violated. If $sim(A, B) > sim(A, C)$ is necessary to reflect class membership, then there must at least be one pair of local similarities, so that $sim(A_i, B_i) > sim(A_i, C_i)$. If such a pair does not exist, the similarity measure must make use of interdependencies between attributes. For example, the similarity may not depend on two attributes A_1, A_2 themselves, but on their difference $A_1 - A_2$. Virtual attributes can express such interdependencies (e. g., $deposit(A) = income(A) - spending(A)$) and can also encapsulate non-linear relations.

We propose to use intermediate concepts of domain theories as virtual attributes. Virtual attributes can easily be added to the set of attributes of each instance.

Every virtual attribute forms an additional dimensions of the instance space (see Figure 2 (right)). This is most intuitive for numerical attributes. An example is the concept $expectedWealthTillRetirement(C) = (65 - age(C)) * income(C)$ Unfortunately, these dimensions can change assumptions about instance distributions and are most likely not orthogonal to the other dimensions, since they are inferrable from other attributes.

In this paper we focus on binary virtual attributes. Although formally they are additional dimensions, they can be visualized as separating lines within the original instance space (see Figure 2 (left)). They divide the instance space into two regions. For example, $taxFree(C) \leftarrow income(C) < 330$ may divide some instance space into salaries that are or are not subject to paying taxes in Germany. We will show that virtual attributes that describe target concept boundaries are especially useful

Intermediate attributes that are fully defined (i. e., that do not have gaps at the bottom of the domain theory) can be computed from the values of observables and other intermediates. In order to use an intermediate as a virtual attribute, it is added to the local similarities of the similarity measure, that is, $s_i = 1$, iff both instances satisfy the intermediate concept or both do not satisfy it, and $s_i = 0$ otherwise. In the following, virtual attributes are assumed to be discrete.

Let us look at how binary virtual attributes influence classification. Assume for sake of illustration that the instance space is formed by the attributes $temp$ and $press$ denoting the temperature and pressure of a manufacturing oven. Let us assume furthermore that the (to be approximated) target

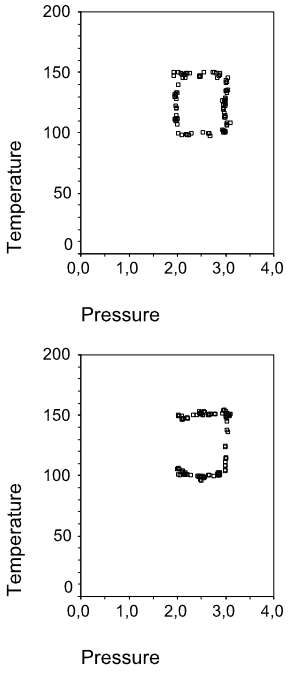

Figure 3: Distribution of errors for the target concept $hardened(C) \leftarrow temp(C) > 100 \wedge temp(C) < 150 \wedge press(C) > 2 \wedge press(C) < 3$ without virtual attributes (top) and with the virtual attribute $V(C) \leftarrow press(C) <= 2$ (bottom).

concept is $hardened(C) \leftarrow temp(C) > 100 \wedge temp(C) < 150 \wedge press(C) > 2 \wedge press(C) < 3$.

The error distribution of an unweighted kNN-classifier for the target concept is depicted in Figure 3 (top). Not surprisingly, the misclassifications occur at the boundaries of the target concept.

Now let us analyze the effect of different amounts and different qualities of domain knowledge on the classification. In order to control the independent variables like partialness and correctness of the domain knowledge, we created a simple test domain. There were two continuous attributes X and Y, uniformly distributed over the interval [0,100]. The target concept was $T(C) \leftarrow X(C) > 30 \wedge X(C) < 70 \wedge Y(C) > 30 \wedge Y(C) < 70$. We used a square centered in the instance space as target concept, because it is one of the few concepts for which the optimal weight setting for kNN-classification can be calculated analytically. The optimal weight setting for the target concept is to use equal weights [Ling and Wang, 1997]. Thus, the accuracy of 1-NN with equal weights is the optimal accuracy that can be achieved without adding additional attributes. There were 100 randomly generated cases in the case-base and 200 test cases were used. Each experiment was repeated 1000 times with random cases in the case-base and random test cases.

3.1 Partialness of the domain theory:

We operationalize the partialness of the domain knowledge as number of known target concept boundaries. The more boundaries are known, the less partial it is.

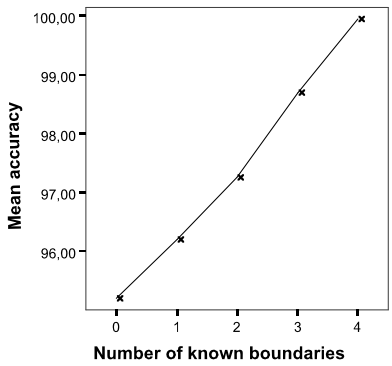

Figure 4: Percentage of correctly classified cases with different numbers of target concept boundaries described by virtual attributes.

Adding virtual attributes that correctly specify a boundary of the target concept makes the misclassifications at those boundaries disappear (see Figure 3 (bottom)). Thus, by adding virtual attributes that describe a boundary correctly, the classification accuracy is increased (see Figure 4).

Obviously, even partial knowledge (e. g. adding only one virtual attribute) can improve classification accuracy. However, in this experiment we assumed that the virtual attributes were correct. In the next experiment we analyzed the influence of the correctness of virtual attributes.

3.2 Correctness:

Vague knowledge can be informally described as knowing that an attribute should be more or less at a certain value. The higher the vagueness, the higher is the probability for high incorrectness. We operationalize correctness of a virtual attribute as its distance from the correct value. We created virtual attributes of the form $V(C) \leftarrow X(C) < c$, where c was varied from 0 to 100 at steps of 5. Remember that the correct X-value (which was used in the domain theory to generate the cases) was 30. The accuracy of classification when adding these virtual attributes is depicted in Figure 5.

The results are a bit disappointing. The accuracy drops rapidly if the virtual attribute is inaccurate. Fortunately, the accuracy with inaccurate virtual attributes is not much lower than using no virtual intermediates (the similarity measure with no virtual attribute is equivalent to setting $c = 0$ or $c = 100$). The second peak at $X = 70$ which is the other boundary on the X-attribute is due to the fact that similarity-based classification is direction-less: only the position of the concept boundary has to be known, the side on which positive and negative instances are located is encoded in the cases.

These experiments with a simple domain suggest that partial knowledge is more useful than vague knowledge. Adding partial knowledge is likely to increase the classification accuracy, whereas vague knowledge is only useful if there is good evidence that the knowledge is correct.

In the next section we will evaluate the influence of virtual

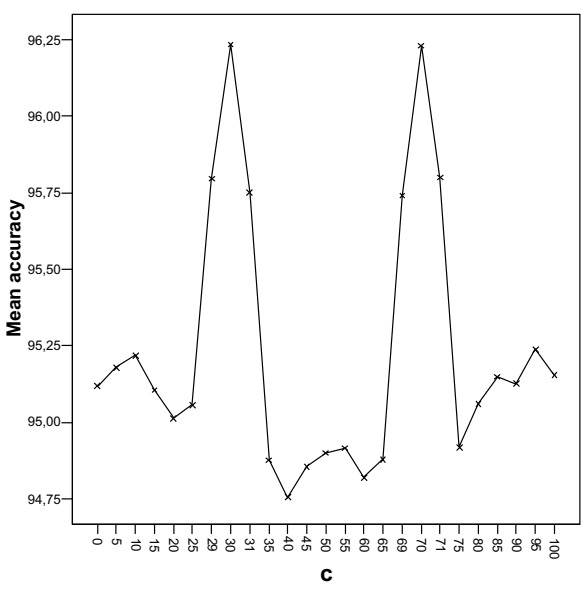

Figure 5: Accuracy of similarity measures using the virtual attribute $V(c) \leftarrow X(C) < c$, where c forms the horizontal axis. c-axis is stretched at the positions of the concept boundaries.

attributes in several domains from the UCI Machine Learning Repository [Blake and Merz, 1998].

4 Experiments

4.1 The domains:

The domain of the previous section allowed us to vary the correctness and partialness of the domain theory. However, since the domain was handcrafted and simple, we ran additional experiments with two data sets from the UCI Machine Learning Repository. Note that some data sets in the repository come along with perfect domain models, as the instances were created by those models. However, we used only data sets whose domain theories were imperfect.

- Japanese Credit Screening (JCS): This domain comes with a domain theory that was created by interviewing domain experts. Accordingly, the theory is imperfect and classifies only 81% of the cases correctly.
- Promoter gene sequences (PGS): This domain theory reflects the knowledge of domain experts in the field of promoter genes. It is highly inaccurate and performs at chance level when used in isolation [Towell *et al.*, 1990]. We included this domain to serve as a worst case scenario, since the domain knowledge is most inaccurate.

It is known that not all intermediate concepts will increase classification accuracy when used as virtual attributes [Steffens, 2004]. Hence, mechanisms to select or weight virtual attributes are necessary. In this paper we investigate whether weighting virtual attributes is more appropriate than selecting them. In the experiments we apply several existing weighting approaches which will be described in section 4.3.

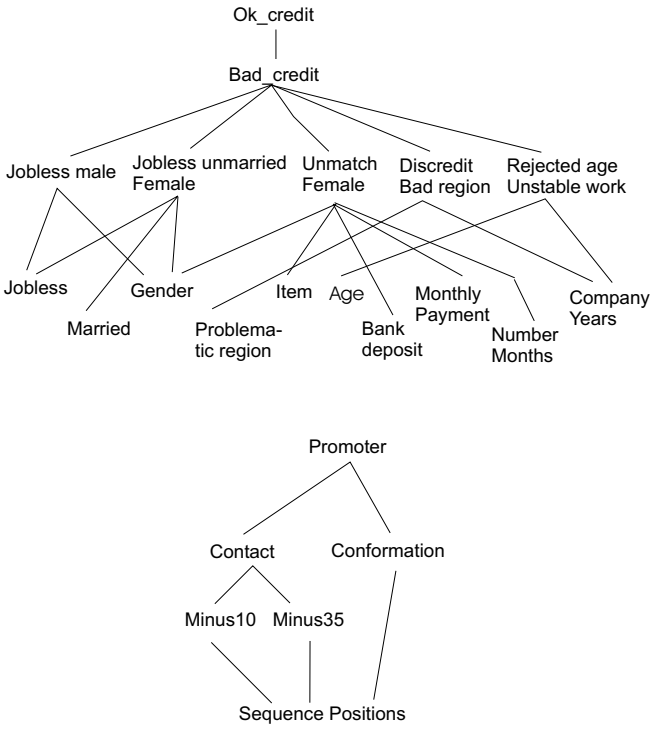

Figure 6: The domain theory of the JCS domain (top) and of the PGS domain (bottom).

4.2 The virtual attributes:

The domain theories of JCS and PGS have been created by domain experts for real world applications. Hence, they do not separate positive from negative instances in a perfect way. The accuracy of the JCS domain theory is 81%, the accuracy of the PGS domain theory is only 50%. The structure of both theories is depicted in Figure 6.

Most of the concepts process several observables. For example, *rejected_age_unstable_work* processes the observables *age* and *number_years*[1]:

```
rejected_age_unstable_work(S) :-
    age_test(S, N1),
    59 < N1,
    number_years_test(S, N2),
    N2 < 3.
```

Although the concepts are very imperfect (i. e., they miscategorize training cases), our experiments described in section 4.4 show that these concepts can improve classification accuracy when used as virtual attributes.

4.3 Weighting methods:

According to the classification of weighting methods as proposed in [Wettschereck *et al.*, 1997], we selected four methods with performance bias, and six with preset bias (i. e., statistical and information-theoretic methods).

[1]This attribute denotes the number of years that the applicant worked at the same company.

- Performance bias: Weighting methods with a performance bias classify instances in a hill-climbing fashion. They update weights based on the outcome of the classification process. The performance bias performs well if there are many irrelevant features [Wettschereck *et al.*, 1997]. Since the intermediate concepts of the domain theories can be assumed to be relevant, we expected performance bias methods to perform badly.
  1. EACH [Salzberg, 1991] increases the weight of matching features and decreases the weight of mismatching features by a hand-coded value.
  2. IB4 [Aha, 1992] is a parameter-free extension of EACH. It makes use of the concept distribution and is thus sensitive to skewed concept distributions. It assumes that the values of irrelevant features are uniformly distributed.
  3. RELIEF [Kira and Rendell, 1992] is a feature selection- rather than feature weighting-algorithm. It calculates weights based on the instance's most similar neighbors of each class and then filters attributes whose weights are below a hand-coded threshold. We used extensions for non-binary target classes and kNN with $k > 1$ as proposed in [Kononenko, 1994].
  4. ISAC [Bonzano *et al.*, 1997] increases weights of matching attributes and decreases weights of mismatching attributes by a value that is calculated from the ratio of the prior use of the instance. The more often the instance was retrieved for correct classifications, the higher the update value.
- Preset bias: The bias of the following methods is based on probabilistic or information-theoretic concepts. They process each training instance exactly once.
  1. CCF [Creecy *et al.*, 1992] binarizes attributes and weights them according to the classes' probability given a feature.
  2. PCF [Creecy *et al.*, 1992] is an extension of CCF which takes the distribution of the feature's values over classes into account. It calculates different weights for different classes.
  3. MI [Daelemans and van den Bosch, 1992] calculates the reduction of entropy in the class distribution by attributes and uses it as the attribute weight.
  4. CD [Nunez *et al.*, 2002] creates a correlation matrix of the discretized attributes and the classes. The weight of an attribute increases with the accuracy of the prediction from attribute value to class.
  5. VD [Nunez *et al.*, 2002] extends CD in that it considers both the best prediction for a class and the predictions of all attributes.
  6. CVD [Nunez *et al.*, 2002] combines CD and VD.

4.4 Results:

For brevity, we will refer to the similarity measure which uses only observables as the *non-extended* measure. The similarity measure which uses virtual attributes will be called *extended*. We used the leave-one-out evaluation method.

Table 1: Classification accuracies of the non-extended and the extended similarity measures. The columns report the accuracies for the unweighted classification and for several weighting methods.

Domain	unw.	EACH	RELIEF	IB4	ISAC
JCS (w/o)	74.19	74.19	78.23	74.19	72.58
JCS (w/)	74.19	72.58	79.03	72.58	79.03
PGS (w/o)	86.79	89.62	96.23	88.68	50.0
PGS (w/)	85.85	93.40	96.23	90.57	96.23

CCF	PCF	MI	CD	VD	CVD
72.58	72.58	74.19	74.19	72.58	71.77
73.39	75.0	75.0	77.42	75.0	75.0
85.85	87.74	68.87	88.68	77.36	83.02
91.51	86.79	98.11	88.68	97.17	87.74

For most of the weighting methods, the extended similarity measure performs better than the non-extended one. In table 1 we underline the accuracy of the extended similarity measure if it outperformed the non-extended similarity measure when using the same weighting method. In the PGS domain, seven of ten weighting methods perform better if the similarity measure is extended with virtual attributes. Even more so, in the JCS domain the accuracies of eight of ten weighting methods were improved by using virtual attributes.

In its optimal setting, with an accuracy of 98.11% our approach performs also better than the results from the literature reported for the PGS domain. The accuracy of KBANN in [Towell *et al.*, 1990] is 96.23%, which to our knowledge was the highest accuracy reported so far and also used the leave-one-out evaluation. We found no classification accuracy results for JCS in the literature[2].

Obviously, these improvements are not restricted to a certain class of weighting methods. Methods with performance bias (most notably ISAC), information-theoretic bias (i. e. MI), and with a statistical correlation bias (e. g. VD) benefit from processing virtual attributes.

Even in the PGS domain, the improvements are substantial. This is surprising, since the domain knowledge is the worst possible and classifies at chance level when used for rule-based classification. This is a promising result as it shows that adding intermediate concepts may increase accuracy even if the domain theory is very inaccurate. We hypothesize that this is due to the fact that even vague rules-of-thumb provide some structure in the instance space which will be exploited by the similarity measure.

5 Conclusion and future work

The main contribution of this paper is to show that imperfect domain knowledge can benefit similarity-based classification. This facilitates knowledge elicitation from domain experts as it removes the requirements of completeness and accurateness. Our experiments in a simple domain suggest that partial knowledge is more useful than vague knowledge. How-

[2]The domain often referred to as 'credit screening' with 690 instances is actually the credit card application domain.

ever, we showed in the domains from the Machine Learning Repository that even highly inaccurate domain knowledge can be exploited to drastically improve classification accuracy. Future work includes experiments in further domains and transforming intermediate attributes by feature generation [Fawcett and Utgoff, 1992].

References

[Aamodt, 1994] Agnar Aamodt. Explanation-driven case-based reasoning. In Stefan Wess, Klaus-Dieter Althoff, and Michael M. Richter, editors, *Topics in Case-Based Reasoning*, pages 274–288. Springer, 1994.

[Aha, 1991] David W. Aha. Incremental constructive induction: An instance-based approach. In Lawrence Birnbaum and Gregg Collins, editors, *Proceedings of the Eighth International Workshop on Machine Learning*, pages 117–121, Evanston, IL, 1991. Morgan Kaufmann.

[Aha, 1992] David W. Aha. Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies*, 36(2):267–287, 1992.

[Bergmann *et al.*, 1994] Ralph Bergmann, Gerhard Pews, and Wolfgang Wilke. Explanation-based similarity: A unifying approach for integrating domain knowledge into case-based reasoning. In Stefan Wess, Klaus-Dieter Althoff, and Michael M. Richter, editors, *Topics in Case-Based Reasoning*, pages 182–196. Springer, 1994.

[Blake and Merz, 1998] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998.

[Bonzano *et al.*, 1997] Andrea Bonzano, Padraig Cunningham, and Barry Smyth. Using introspective learning to improve retrieval in cbr: A case study in air traffic control. In David Leake and Enric Plaza, editors, *Proceedings of the second ICCBR conference*, pages 291–302, Berlin, 1997. Springer.

[Cain *et al.*, 1991] Timothy Cain, Michael J. Pazzani, and Glenn Silverstein. Using domain knowledge to influence similarity judgements. In *Proceedings of the Case-Based Reasoning Workshop*, pages 191–198, Washington D.C., U.S.A., 1991.

[Creecy *et al.*, 1992] Robert H. Creecy, Brij M. Masand, Stephen J. Smith, and David L. Waltz. Trading mips and memory for knowledge engineering. *Communications of the ACM*, 35(8):48–64, 1992.

[Daelemans and van den Bosch, 1992] Walter Daelemans and Antal van den Bosch. Generalization performance of backpropagation learning on a syllabification task. In *Proceedings of the Third Twente Workshop on Language Technology: Connectionism and Natural Language Processing*, pages 27–37, Enschede, The Netherlands, 1992. Unpublished.

[Fawcett and Utgoff, 1992] Tom Elliott Fawcett and Paul E. Utgoff. Automatic feature generation for problem solving systems. In Derek H. Sleeman and Peter Edwards, editors, *Proceedings of the 9th International Conference on Machine Learning*, pages 144–153. Morgan Kaufmann, 1992.

[Griffiths and Bridge, 1996] Anthony D. Griffiths and Derek G. Bridge. A yardstick for the evaluation of case-based classifiers. In Ian D. Watson, editor, *Proceedings of Second UK Workshop on Case-Based Reasoning*, 1996.

[Kira and Rendell, 1992] Kenji Kira and Larry A. Rendell. A practical approach to feature selection. In Derek H. Sleeman and Peter Edwards, editors, *Proceedings of the Ninth International Workshop on Machine Learning*, pages 249–256. Morgan Kaufmann Publishers Inc., 1992.

[Kononenko, 1994] Igor Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In F. Bergadano and L. de Raedt, editors, *Proceedings of the European Conference on Machine Learning*, pages 171–182, Berlin, 1994. Springer.

[Ling and Wang, 1997] Charles X. Ling and Hangdong Wang. Computing optimal attribute weight settings for nearest neighbour algorithms. *Artificial Intelligence Review*, 11:255–272, 1997.

[Mooney and Ourston, 1991] Raymond J. Mooney and Dirk Ourston. Constructive induction in theory refinement. In Lawrence Birnbaum and Gregg Collins, editors, *Proceedings of the Eighth International Machine Learning Workshop*, pages 178–182, San Mateo, CA, 1991. Morgan Kaufmann.

[Nunez *et al.*, 2002] H. Nunez, M. Sanchez-Marre, U. Cortes, J. Comas, I. Rodriguez-Roda, and M. Poch. Feature weighting techniques for prediction tasks in environmental processes. In *Proceedings of the 3rd Workshop on Binding Environmental Sciences and Artificial Intelligence (BESAI 2002)*, 2002.

[Porter *et al.*, 1990] Bruce W. Porter, Ray Bareiss, and Robert C. Holte. Concept learning and heuristic classification in weak-theory domains. *Artificial Intelligence*, 45(1-2):229–263, 1990.

[Richter, 2003] Michael M. Richter. Fallbasiertes Schliessen. *Informatik Spektrum*, 3(26):180–190, 2003.

[Salzberg, 1991] Steven Salzberg. A nearest hyperrectangle learning method. *Machine Learning*, 6(3):251–276, 1991.

[Steffens, 2004] Timo Steffens. Similarity-measures based on imperfect domain-theories. In Steffen Staab and Eva Onainda, editors, *Proceedings of STAIRS 2004*, pages 193–198. IOS Press, Frontiers in Artificial Intelligence and Applications, 2004.

[Towell *et al.*, 1990] Geofrey G. Towell, Jude W. Shavlik, and Michael O. Noordenier. Refinement of approximate domain theories by knowledge based neural network. In *Proceedings of the Eighth National Conference on AI*, volume 2, pages 861–866, 1990.

[Wettschereck *et al.*, 1997] Dietrich Wettschereck, David W. Aha, and Takao Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11:273–314, 1997.