A Practical Approach to Feature Selection

Kenji Kira
Computer & Information Systems Laboratory
Mitsubishi Electric Corporation
5-1-1 Ofuna, Kamakura
Kanagawa 247, Japan
kira@sy.isl.melco.co.jp

Larry A. Rendell
Beckman Institute and Department of Computer Science
University of Illinois at Urbana-Champaign
405 N. Mathews Avenue
Urbana, IL 61801, U.S.A.
rendell@cs.uiuc.edu

Abstract

In real-world concept learning problems, the representation of data often uses many features, only a few of which may be related to the target concept. In this situation, feature selection is important both to speed up learning and to improve concept quality. A new feature selection algorithm **Relief** uses a statistical method and avoids heuristic search. Relief requires linear time in the number of given features and the number of training instances regardless of the target concept to be learned. Although the algorithm does not necessarily find the smallest subset of features, the size tends to be small because only statistically relevant features are selected. This paper focuses on empirical test results in two artificial domains; the LED Display domain and the Parity domain with and without noise. Comparison with other feature selection algorithms shows Relief's advantages in terms of learning time and the accuracy of the learned concept, suggesting Relief's practicality.

1 INTRODUCTION

Since relevant features for many real-world concept learning problems are often unknown, we must introduce many candidate features. Unfortunately irrelevant features degrade the performance of concept learners both in speed (due to high dimensionality) and predictive accuracy (due to irrelevant information). The situation is particularly serious in constructive induction, as many candidate features are generated in order to enhance the power of the representation language. *Feature selection* is the problem of choosing a small subset of features that ideally is necessary and sufficient to describe the target concept.

Concept learners, such as ID3 [Quinlan 1983] or PLS1 [Rendell, Cho & Seshu 1989], select relevant features by themselves, using measures such as information gain. Hence one might think that feature selection is not a problem at all. But hard concepts having feature interactions are problematic for induction algorithms [Devijver & Kittler 1982, Pagallo 1989, Rendell & Seshu 1990]. For example, if the target concept is $f_1 \oplus f_2 = 1$ and the distribution of the feature values is uniform over {0, 1}, the probability of an instance's being positive (negative) is 50% when $f_1 = 1$ ($f_2 = 1$). There is little information gain in selecting either of f_1 or f_2 though they are the relevant features. Since real-world problems may involve feature interaction, it is not always enough to apply concept learners only. Problems such as protein folding and weather prediction are hard in the sense.

Many feature selection algorithms have been proposed. One of them is an exhaustive search algorithm over all subsets of the given feature set. However exhaustive search is intractable. Devijver and Kittler [1982] review heuristic methods for reducing the search space. But they are suboptimal - it is always possible for the methods to miss relevant but undetectable features.

For real-world problems involving much feature interaction, we need a reliable and practically efficient method to eliminate irrelevant features. The inefficiency of past approaches is caused by trying combinations of features, explicitly searching for the smallest sufficient subset of the given feature set. A different approach is to collect all the statistically-relevant features. Our algorithm is described in Section 2, and its empirical evaluation is given in Sections 3. Section 4 addresses limitations and Section 5 compares related work. Section 6 concludes and suggests future work.

2 Relief ALGORITHM

We assume two-class classification problems. An instance is represented by a vector composed of p feature values. $\mathcal{S}$ denotes a set of training instances with size n. $\mathcal{F}$ is the given feature set $\{f_1, f_2, \ldots, f_p\}$. An instance X is denoted by a p-dimensional vector $(x_1, x_2, \ldots, x_p)$, where x_j denotes the value of feature f_j of X.

Relief is a feature selection algorithm inspired by instance-based learning [Aha, Kibler & Albert 1991, Callan, Fawcett & Rissland 1991]. Given training data $\mathcal{S}$, sample size m, and a threshold of relevancy τ, Relief detects those features which are statistically relevant to the target concept. τ encodes a relevance threshold ($0 \leq t \leq 1$). We assume the scale of every feature is either nominal (including boolean) or numerical (integer or real). Differences of feature values between two instances X and Y are defined by the following function diff.

When x_k and y_k are nominal,

$$\text{diff}(x_k, y_k) = \begin{cases} 0 & \text{<if } x_k \text{ and } y_k \text{ are the same>} \\ 1 & \text{<if } x_k \text{ and } y_k \text{ are different>} \end{cases}$$

When x_k and y_k are numerical,

$\text{diff}(x_k, y_k) = (x_k - y_k)/nu_k$

where nu_k is a normalization unit to normalize the values of diff into the interval [0, 1]

Relief (Figure 1) picks a sample composed of m triplets of an instance X, its **Near-hit** instance[1] and **Near-miss** instance. Relief uses the p-dimensional Euclid distance for selecting **Near-hit** and **Near-miss**. Relief calls a routine to update the feature weight vector W for every sample triplet and determines the average feature weight vector **Relevance** (of all the features to the target concept). Finally, Relief selects those features whose average weight ('**relevance level**') is above the given threshold τ.

Relief is valid only when (1) the relevance level is large for relevant features and small for irrelevant features, and (2) τ can be chosen to retain relevant features and discard irrelevant features. Theoretical analysis [Kira & Rendell 1992] shows that (1) the relevance level is positive when the feature is relevant, and close to zero or negative when it is irrelevant, and that (2) a statistical method of interval estimation, can be used to determine the value of τ.

[1]We call an instance a near-hit of X if it belongs to the close neighborhood of X and also to the same category as X. We call an instance a near-miss when it belongs to the properly close neighborhood of X but not to the same category as X.

The complexity of Relief is $\Theta(pmn)$ because the distance between X and each of the n instances is calculated, taking $\Theta(p)$ time, to determine its **Near-miss** and **Near-hit** inside a loop iterating m times. m is a constant affecting the accuracy of relevance levels. Since m is chosen independently of p and n, the complexity is $\Theta(pn)$. Thus the algorithm can select statistically relevant features in linear time in terms of the number of features and the number of training instances.

```
Relief(S, m, τ)
    Separate S into S+ = {positive instances} and
        S- = {negative instances}
    W = (0, 0, . . . , 0)
    For i = 1 to m
        Pick at random an instance X ∈ S
        Pick at random one of the positive instances
            closest to X, Z+ ∈ S+
        Pick at random one of the negative instances
            closest to X, Z- ∈ S-
        if (X is a positive instance)
            then  Near-hit = Z+; Near-miss = Z-
            else  Near-hit = Z-; Near-miss = Z+
        update-weight(W, X, Near-hit, Near-miss)
    Relevance = (1/m)W
    For i = 1 to p
        if (relevance_i ≥ τ)
            then  f_i is a relevant feature
            else  f_i is an irrelevant feature

update-weight(W, X, Near-hit, Near-miss)
    For i = 1 to p
        W_i = W_i - diff(x_i, near-hit_i)^2 + diff(x_i, near-miss_i)^2
```

Figure 1 Relief Algorithm

3 EMPIRICAL EVALUATION

Two artificial domains, LED (Section 3.1) and Parity (Section 3.2) were used for evaluating Relief. We also compare Relief and other feature selection algorithms both in learning time and predictive accuracy of the learned concepts (Section 3.3).

3.1 LED DISPLAY DOMAIN

Our data in the LED Display Domain [Breiman et al. 1984, Aha, Kibler & Albert 1991], which were generated by the program provided in UC Irvine repository, consist of seven meaningful features, corresponding to the seven segments, and seventeen irrelevant features. Each feature has the value of 0 or 1. We also introduced noise. For the noisy case, 10% of

the feature values were negated. The number of training instances was 200.

The target concepts of this domain are the description of digits (0, 1, ... , 9). Table 1 shows which feature values are 1 for each concept.

Table 1 LED Display Domain : Target Concepts

digit	f1	f2	f3	f4	f5	f6	f7
1	0	0	1	0	0	1	0
2	1	0	1	1	1	0	1
3	1	0	1	1	0	1	1
4	0	1	1	1	0	1	0
5	1	1	0	1	0	1	1
6	1	1	0	1	1	1	1
7	1	0	1	0	0	1	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	0
0	1	1	1	0	1	1	1

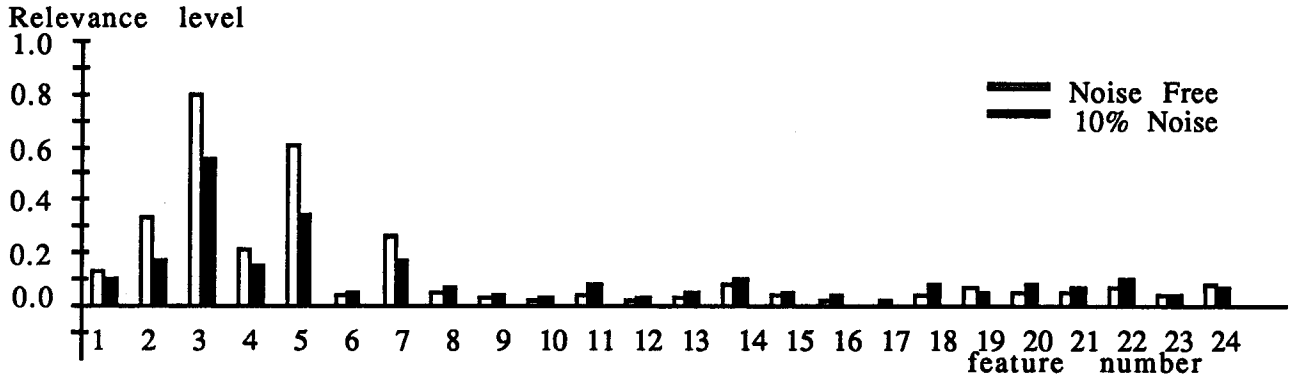


Figure 2 Relevance levels of the concept '6'

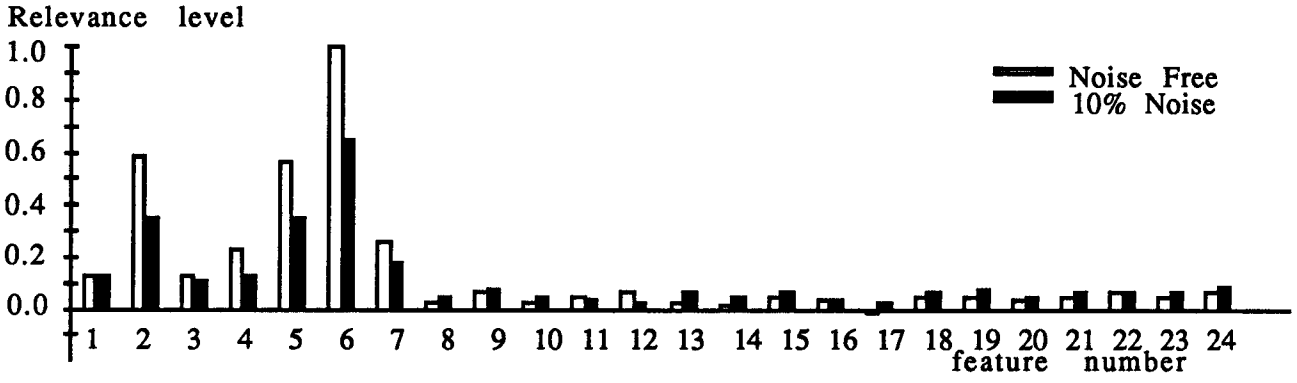


Figure 3 Relevance levels of the concept '2'

Figure 2 and 3 show the relevance levels of all the features for the concept '6' and the concept '2' respectively. In learning the concept for a digit (e.g. '6'), all the instances corresponding to the other nine digits are considered to be negative examples of the target concept. Each relevance level shown by a bar in the graphs is an average over 50 runs (5 runs for 10 different data sets).

Figure 2 shows that features f_3 and f_5 are highly relevant to the target concept '6'. All of the noise-free runs, and 45 out of the 50 noisy runs, picked f_3 and f_5 as the best two features. Table 1 shows that $\{f_3, f_5\}$ is the minimal feature set to describe the concept '6'.

Figure 3 shows that features f_2, f_5 and f_6 are highly relevant for the target concept '2'. In this case, $\{f_6\}$ is the minimal feature set, although Relief did not find the minimal set. Most runs (all the noise-free runs and 48 out of the 50 noisy runs) picked f_6 as the best feature. The size of the feature set selected by Relief is small in the sense that no irrelevant features were selected.

Both Figure 2 and 3 show that the same set of features have high relevance levels no matter whether the data is noise-free or noisy. In both cases (and others not reported here), given a threshold determined by inspection or theoretical considerations [Kira & Rendell 1992], Relief selects feature sets that are small and sufficient to describe the target concepts.

3.2 PARITY DOMAIN

Parity(h, k, r) denotes a parity domain consisting of h relevant features and k irrelevant features. For each relevant feature, r% of the instances have a negated (noisy) feature value. The target concept is $f_1 \oplus f_2 \oplus \ldots \oplus f_h = 1$. A parity concept is a hard concept having many *peaks* or disjuncts [Rendell & Seshu 1990] in the instance space. In other words, the relevant features interact with one another.

Table 2 shows the result of applying Relief to Parity(3, 7, 0) and Parity(3, 7, 5). For both of them, f_1, f_2 and f_3 are the only relevant features. Each row of the table shows the average and the standard deviation of the relevance level for each feature over 20 different data sets (each set consisting of 200 training instances) for each domain.

The last row shows the number of times (out of 20 runs) when Relief was *fooled* — an irrelevant feature has a higher relevance level than at least one of the relevant features.

Table 2 Parity(3, 7, 0) and Parity(3, 7, 5)
Rel. = Relevance level, Dev. = Standard deviation

	200 inst, No noise		200 inst, 5% noise	
	Rel.	Dev.	Rel.	Dev.
f1	0.2970	0.0589	0.1605	0.0471
f2	0.3075	0.0542	0.1815	0.0519
f3	0.3165	0.0511	0.1540	0.0589
f4	- 0.0740	0.0465	- 0.0335	0.0540
f5	- 0.0755	0.0663	- 0.0425	0.0559
f6	- 0.0860	0.0407	- 0.0335	0.0673
f7	- 0.0870	0.0500	- 0.0290	0.0607
f8	- 0.0935	0.0587	- 0.0395	0.0548
f9	- 0.0880	0.0671	- 0.0400	0.0616
f10	- 0.1085	0.0515	- 0.0810	0.0454
fooled	0/20		2/20	

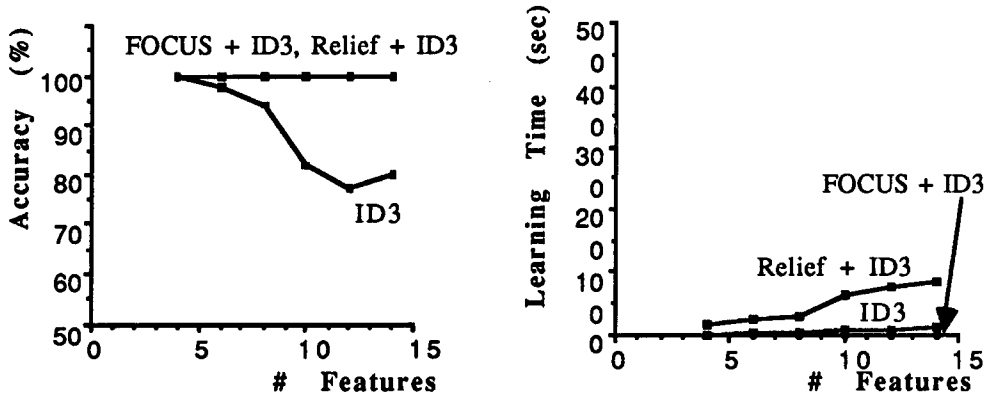
Table 2 shows a clear contrast between the relevance levels of f_1, f_2 and f_3 and those of the irrelevant features. The same set of features is highlighted both for noise-free and for noisy data, though less clearly for the latter.

When the data are noisy, Relief may be fooled. Note that 5% feature noise for the three relevant features corresponds to 4.75% classification noise (one-bit errors and three-bit errors lead to classification error). Near-miss instances picked by Relief may in fact be near-hit instances and vice versa.

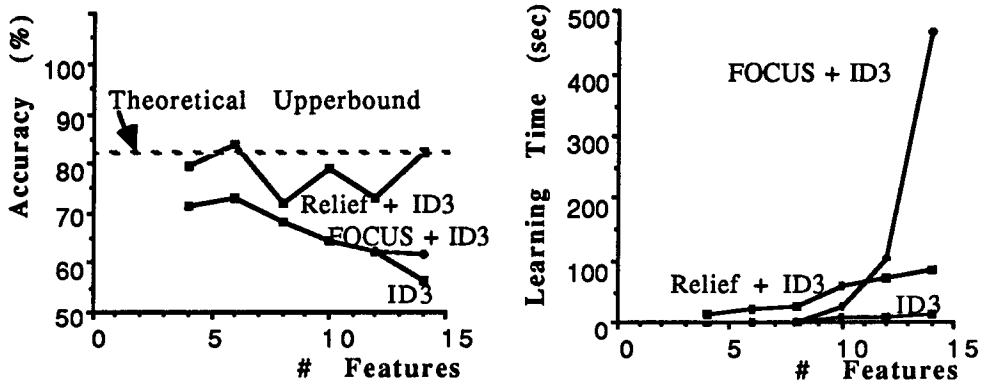
3.3 COMPARISON WITH OTHER ALGORITHMS

In the introduction, we discussed three types of feature selection algorithms. One is exhaustive search, another is heuristic search, and the third is Relief - a new statistical method. In this section, we compare these three types of algorithms. Exhaustive search is represented by FOCUS [Almuallim & Dietterich 1991], which is a recent algorithm that stops as soon as the smallest sufficient feature set is found. ID3 represents heuristic search - a kind of sequential forward search [Devijver & Kittler 1982], since it incrementally selects the best feature with the most information gain while building a decision tree. Figure 4 shows the results of comparing (1) ID3 alone, (2) FOCUS combined with ID3, and (3) Relief ($m = 40$, $\tau = 0.1$) combined with ID3, each in terms of predictive accuracy and learning time in a parity domain.

The noise-free test was done in Parity(2, α, 0) and the noisy test in Parity(2, α, 10), where α varied from 2 to 12. The horizontal axis shows the size of the given feature set $\mathcal{F}$. The results are the averages of 10 runs.



(a) Tests with noise-free data



(b) Tests with noisy data (10% feature noise)

Figure 4 Test Results in Parity Domain

The predictive accuracy of ID3 alone was inferior to FOCUS + ID3 and Relief + ID3 in both noise-free and noisy situations. This shows the importance of feature selection algorithms.

With noisy data, the predictive accuracy of Relief + ID3 is higher than FOCUS + ID3 because FOCUS selected more irrelevant features than Relief. The predictive accuracy of Relief + ID3 looks unstable. This is because Relief failed to select one of the relevant feature once in a while, which greatly affects the average accuracy. As Figure 4 shows, however, Relief + ID3 typically learns the correct concept. The learning time of FOCUS + ID3 increases exponentially as the size of $\mathcal{F}$ increases, while that of Relief + ID3 increases only linearly. Relief can select relevant features in linear time regardless of the complexity of the target concept. The algorithm also works for noisy data.

These results show that Relief is superior to exhaustive search in terms of learning time and superior to heuristic search in terms of predictive accuracy.

4 CURRENT LIMITATION AND FUTURE WORK

Relief requires retention of data in incremental uses. However it can be modified for incremental update of relevance levels. Relief does not help with redundant features. If most of the given features are relevant to the concept, it would select most of the given features even though only a small number of them are necessary for concept description.

Relief is applicable only to the two-class classification problem. However Relief can easily be extended for solving multiple-class classification problems by considering them as a set of two-class classification problems. Relief can also be extended for solving continuous value prediction problems.

Insufficient training instances *fools* Relief. Sparse distribution of training instances increases the probability of picking instances in different peaks or disjuncts as **Near-hit** (Figure 1). For example, if the same number of instances are given, Relief gets less accurate for a parity concept $f_1 \oplus f_2 \oplus \ldots \oplus f_h = 1$ as h increases. This is because the density of peaks increases as h increases.

It is crucial for Relief to pick real near-hit instances. There are two ways. One is to give enough near-hit instances for all instances (especially for those instances on the boundary of the target concept). Another is to apply feature construction [Matheus & Rendell 1989, Rendell & Seshu 1990, Yang, Blix & Rendell 1991]. By generating good new features, the number of peaks of the target concept is reduced. Accordingly the same training instances may then provide enough near-hit instances to detect relevance of those new features to the concept.

5 RELATED WORK

Traditional heuristic search algorithms given in [Devijver & Kittler 1982] heuristically select a subset of features of a given size d. But since we don't know the size of the subset to be selected, it is always possible to be fooled by a concept which requires more than d features to describe. Relief is designed to pick all the relevant features and it is not necessary to give the size of the subset to be selected. Instead, Relief requires a certain threshold to discard irrelevant features statistically.

The FOCUS algorithm [Almuallim & Dietterich 1991] is able to detect the necessary and sufficient features in quasi-polynomial time, provided (1) the complexity of the target concept is limited and (2) there is no noise. But since the complexity of the target concept is unknown and data are often noisy in real-world problems, FOCUS can be impractical.

Table 3 summarizes the comparison of Relief and other feature selection algorithms.

Aha [1989] and Aha & McNulty [1989] propose a method to learn relative attribute weights, corresponding to relevance levels. Their method suggests an IBL approach to feature selection. Their analysis of the weight vector implicitly assumed a uniform distribution for irrelevant feature values, which makes weights of irrelevant features statistically zero. In general, however, we do not know the distribution of irrelevant feature values. Relief does not assume anything for the distribution of irrelevant feature values.

Callan, Fawcett and Rissland [1991] also introduce an interesting feature weight update algorithm in their case-based system CABOT, which showed significant improvement over pure case-based reasoning in the OTHELLO domain. When CABOT retrieves a wrong case, it updates the weights by asking the domain expert to identify the best correct case. Since Relief does not require the domain expert to choose the best instance, it is more autonomous. Relief's feature relevance is supported by theoretical analysis [Kira & Rendell 1992].

Table 3 Comparison of Feature Selection Algorithms

	Learning Speed	Quality of Selection
ID3 alone	fast	• fooled by feature interaction
Heuristic Search	relatively fast	• fooled by feature interaction • fooled when no. of relevant features exceeds the limit
FOCUS	fast, when the target concept is simple and the data are noise-free can be very slow, when complex or noisy	• selects the optimal set when the data are noise-free • tends to select many irrelevant features when the data are noisy
Relief	relatively fast	• selects only statistically relevant features • not fooled by feature interaction • noise-tolerant

Schlimmer [1987] explores an extended method to construct new features for STAGGER [Schlimmer & Granger 1986] and [Aha 1991] applies the method for IBL. The method selects seeds (source features) for generating a new feature judging from weights assigned to each source feature based on its relevance to the concept. However, since the relevance is determined one feature at a time, the method does not work for domains where features interact with one another (e.g. parity domains, protein folding). Relief assesses feature sets and can handle complex domains.

Porter, Bareiss and Holte [1990] introduce feature importance, which corresponds to relative attribute weights in [Aha & McNulty 1989]. In the Protos system of Porter, Bareiss and Holte, feature importance is determined from the explanation of classification given by experts. Relief does not require any explanation of classification, and is more autonomous.

6 CONCLUSION

Relief is a simple algorithm which is partly inspired by learning relative feature weights in IBL. The algorithm relies entirely on statistical analysis and employs few heuristics, and is less often fooled. The algorithm is efficient. Its computational complexity is polynomial (Θ(pmn), or Θ(pn) if the sample size m is constant). The efficiency of our algorithm is brought about by (1) not searching the space of subsets explicitly and (2) giving up the minimality of the subset returned. Relief is also fairly noise-tolerant and is unaffected by feature interaction. This is especially important for hard real-world domains such as protein folding and weather prediction.

Though our approach is suboptimal in the sense that the subset acquired is not always the smallest, this limitation may not be critical for two reasons. One is that the smallest set can be achieved by subsequent exhaustive search over the subsets of all the features selected by Relief. The other mitigating factor is that the concept learners such as ID3 [Quinlan 1983] and PLS1 [Rendell, Cho & Seshu 1989] themselves can select necessary features to describe the target concept if the given features are all relevant.

More experiments and thorough theoretical analysis are warranted. The experiments should include combining our algorithm and various kinds of concept learners such as similarity-based learners, and connectionist learners. Relief can also be applied to IBL to learn relative weights of features for the similarity metrics and integrated with constructive induction.

Acknowledgements

The authors thank David Aha for discussion on IBL algorithms and Bruce Porter for discussion on feature importance. Thanks also to the members of the Inductive Learning Group at UIUC for comments and suggestions.

References

[Aha 1989] Aha, D. W. Incremental Instance-Based Learning of Independent and Graded Concept Descriptions, Proceedings of the Sixth International Workshop on Machine Learning.

[Aha 1991] Aha, D. W. Incremental Constructive Induction: An Instance-Based Approach, Proceedings of the Eighth International Workshop on Machine Learning.

[Aha, Kibler & Albert 1991] Aha, D. W., Kibler, D. & Albert, M. K. Instance-Based Learning Algorithms. Machine Learning, 6, 37-66.

[Aha & McNulty 1989] Aha, D. W. & McNulty, D. M. Learning Relative Attribute Weights for Instance-Based Concept Descriptions, Proceedings of the Eleventh Annual Conference of the Cognitive Science Society.

[Almuallim & Dietterich 1991] Almuallim, H. & Dietterich, T. G., Learning With Many Irrelevant Features, Proceedings of the Ninth National Conference on Artificial Intelligence, 1991, 547-552.

[Bareiss 1989] Bareiss, R., Exemplar-Based Knowledge Acquisition : A Unified Approach to Concept Representation, Classification, and Learning, Academic Press

[Breiman et al. 1984] Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J., Classification and Regression Trees, Wadsworth, 1984.

[Callan, Fawcett & Rissland 1991] Callan, J. P., Fawcett, T. E. & Rissland, E. L., CABOT : An Adaptive Approach to Case-Based Search, Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, 1991, 803-808.

[Devijver & Kittler 1982] Devijver, P. A. & Kittler, J., Pattern Recognition : A Statistical Approach, Prentice Hall.

[Kira & Rendell 1992] Kira, K. & Rendell, L. A., The Feature Selection Problem : Traditional Methods and a New Algorithm, Proceedings of the Tenth National Conference on Artificial Intelligence, 1992.

[Matheus & Rendell 1989] Matheus, C. & Rendell, L. A., Constructive Induction on Decision Trees. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 1989, 645-650.

[Pagallo 1989] Pagallo, G., Learning DNF by Decision Trees, Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 1989, 639-644.

[Porter, Bareiss & Holte 1990] Porter, B. W., Bareiss, R. & Holte, R. C. Concept Learning and Heuristic Classification in Weak-Theory Domains, Artificial Intelligence, 45, 229-263.

[Quinlan 1983] Quinlan, J. R. Learning Efficient Classification Procedures and Their Application to Chess End Games. Machine Learning : An Artificial Intelligence Approach, 1983, 463-482.

[Rendell, Cho & Seshu 1989] Rendell, L. A., Cho, H. H. & Seshu, R. Improving the Design of Similarity-Based Rule-Learning Systems. International Journal of Expert Systems, 2, 97-133.

[Rendell & Seshu 1990] Rendell, L. A. & Seshu, R. Learning Hard Concepts through Constructive Induction: Framework and Rationale. Computational Intelligence, Nov., 1990.

[Schlimmer 1987] Schlimmer, J. C., Learning and Representation Change, Proceedings of the Fifth National Conference on Artificial Intelligence

[Schlimmer & Granger 1986] Schlimmer, J. C. & Granger, R. H. Jr., Incremental Learning from Noisy Data, Machine Learning 1, 317-354

[Yang, Blix & Rendell 1991] Yang, D-S., Blix, G. & Rendell, L. A. The Replication Problem: A Constructive Induction Approach, Proceedings of European Working Session on Learning, march, 1991.