The Feature Selection Problem : Traditional Methods and a New Algorithm

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 61820, U.S.A.
rendell@cs.uiuc.edu

Abstract

For real-world concept learning problems, feature selection is important to speed up learning and to improve concept quality. We review and analyze past approaches to feature selection and note their strengths and weaknesses. We then introduce and theoretically examine a new algorithm **Relief** which selects relevant features using a statistical method. Relief does not depend on heuristics, is accurate even if features interact, and is noise-tolerant. It requires only linear time in the number of given features and the number of training instances, regardless of the target concept complexity. The algorithm also has certain limitations such as non-optimal feature set size. Ways to overcome the limitations are suggested. We also report the test results of comparison between Relief and other feature selection algorithms. The empirical results support the theoretical analysis, suggesting a practical approach to feature selection for real-world problems.

1 Introduction

The representation of raw data often uses many features, only some of which are relevant to the target concept. Since relevant features are often unknown in real-world problems, we must introduce many candidate features. Unfortunately redundant 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.

For many real-world problems, which possibly involve much feature interaction, we need a reliable and practically efficient method to eliminate irrelevant features. Approaches and their problems are discussed in Section 2. Intended to circumvent some of the problems, a new algorithm is described in Section 3. Its detailed theoretical analysis and brief empirical evaluation are given in Sections 4 and 5. Section 6 addresses current limitations and future work. Section 7 concludes.

2 Past Approaches and Their Problems

We assume two-class classification problems. An instance is represented by a vector composed of p feature values. $\mathbb{S}$ denotes a set of training instances with size n. $\mathbb{F}$ 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 the feature f_j of X.

Typical approaches need a function $J(\mathbb{E}, \mathbb{S})$ which evaluates the subset $\mathbb{E}$ of $\mathbb{F}$ using the data $\mathbb{S}$. The subset $\mathbb{E}_1$ is better than $\mathbb{E}_2$ if $J(\mathbb{E}_1, \mathbb{S}) > J(\mathbb{E}_2, \mathbb{S})$. If J is to examine all the training instances, the complexity of J, $\Theta(J)$, must be at least $\Theta(n)$.

2.1 Concept Learners Themselves

Concept learners, such as ID3 [Quinlan 1983] or PLS1 [Rendell, Cho & Seshu 1989], select relevant features by themselves, using measures such as information gain for J. Hence, one might think that feature selection is not a problem at all. But hard concepts having feature interaction 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 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 relevant. Since real-world problems may involve feature interaction, it is not always enough to apply concept learners only.

2.2 Exhaustive Search

One way to select a necessary and sufficient subset is to try exhaustive search over all subsets of $\mathbb{F}$ and find the subset that maximizes the value of J. This exhaustive

search is optimal - it gives the smallest subset maximizing J. But since the number of subsets of $\mathcal{F}$ is 2^p, the complexity of the algorithm is $O(2^p)\cdot O(J)$. This approach is appropriate only if p is small and J is computationally inexpensive.

Almuallim and Dietterich [1991] introduced FOCUS, an exhaustive search algorithm [Figure 1]. They showed that FOCUS can 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. They defined the complexity for a concept c to be the number of bits needed to encode c using their bit-vector representation, and showed that FOCUS will terminate in time $O((2p)^{\log(s\ -\ p)}n)$ where s is the complexity of the target concept.

But the complexity can be as large as $\Theta(2^p)$, for example when all the features are relevant. Since the complexity of the target concept is generally not known *a priori*, we have to expect as much as $\Theta(2^p pn)$ time for the worst case when all the subsets of $\mathcal{F}$ are to be examined. Moreover, with noisy data, FOCUS would select a larger subset of $\mathcal{F}$, since the optimal subset would not give clear class separation.

```
FOCUS(S, F)
  For i = 0, 1, 2, ···
    For all E ⊆ F of size i
      If there exist no two instances in S that agree on
      all features in E but do not agree on the class
      then return E and exit
```

Figure 1 FOCUS

2.3 Heuristic Search Algorithms

Devijver and Kittler [1982] review heuristic feature selection methods for reducing the search space. Their definition of the feature selection problem, "select the best d features from $\mathcal{F}$, given an integer $d \le p$" requires the size d to be given explicitly and differs from ours in the sense. This is problematic in real-world domains, because the appropriate size of the target feature subset is generally unknown. The value d may be decided by computational feasibility, but then the selected d features may result in poor concept description even if the number of relevant features exceeds d only by 1.

Figure 2 shows Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) algorithms. These algorithms use a strong heuristic, "the best feature to add (remove) in every stage of the loop is the feature to be selected (discarded)." These algorithm are much more efficient. SFS's complexity is $\Theta(\frac{p!}{(p-d)!})\cdot O(J)$. SBS's complexity is $\Theta(\frac{p!}{d!})\cdot O(J)$.

But the heuristic also causes a problem. These algorithms perform poorly with feature interaction. Interacting features (e.g. in protein folding, parity (Section 5)) may not maximize J individually, even though they maximize it together.

2.4 Feature weight based approaches

Research in AI tends not to view feature selection as a distinct problem but rather handles it as an implicit part of induction. The following approaches handle feature selection implicitly.

STAGGER [Schlimmer 1987, Schlimmer & Granger 1986] selects source features for constructing a new feature, judging from the feature weights based on their 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.

```
SFS(S, F)
  E = ∅
  For i = 1 to d
    Find a feature f_max ∈ F − E, where
      J(E ∪ {f_max}, S) = max_{f ∈ F − E} J(E ∪ {f}, S)
  E = E ∪ {f_max}
  Return E
```

(a) SFS

```
SBS(S, F)
  E = F
  For i = 1 to (p − d)
    Find a feature f_max ∈ E, where
      J(E − {f_max}, S) = max_{f ∈ E} J(E − {f}, S)
  E = E − {f_max}
  Return E
```

(b) SBS

Figure 2 Heuristic Search Algorithms

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. CABOT updates the weights by asking the domain expert to identify the best case. This dependency on the expert makes the system less autonomous, which is problematic for feature selection.

3 Relief Algorithm

Relief is a feature weight based 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 ($0 \le \tau \le 1$). 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,

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

When x_k and y_k are numerical,

$diff(x_k, y_k)=(x_k - y_k)/nu_k$

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

```
Relief(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 ≥ t)
            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 3 Relief Algorithm

Relief (Figure 3) 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 to determine 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.

The following theoretical analysis shows that Relief is different from other feature weight based algorithms in that it can handle feature interaction, or that it is more autonomous.

4 Theoretical Analysis

Relief has two critical components: the averaged weight vector **Relevance** and the threshold τ. **Relevance** is the averaged vector of the value $-(x_i - \textbf{near-hit}_i)^2 + (x_i - \textbf{near-miss}_i)^2$ for each feature f_i over m sample triplets. Each element of **Relevance** corresponding to a feature shows how relevant the feature is to the target concept. τ is a relevance threshold for determining whether the feature should be selected.

The complexity of Relief is $\Theta(pmn)$. Since m is an arbitrarily chosen constant, the complexity is $\Theta(pn)$. Thus the algorithm can select statistically relevant features in linear time in the number of features and the number of training instances.

Relief is valid only when (1) the relevance level is large for relevant features and small for irrelevant features, and (2) τ retains relevant features and discards irrelevant features. We will show why (1) and (2) hold in the following sections.

4.1 Relevance level

Let Δ be a vector of random variables $\{\delta_i\}$ such that

$$\delta_i = -(x_i - \textbf{near-hit}_i)^2 + (x_i - \textbf{near-miss}_i)^2$$

Figure 3 shows that the **update-weight** function accumulates the value of δ_i for each feature over m samples. **Relevance** gives the averaged value of δ_i for each feature f_i.

If f_i is a relevant feature, x_i and $\textbf{near-hit}_i$ are expected to be very close in the neighborhood of **X**. In contrast, the values of at least one of the relevant features of **X** and **Near-miss** are expected to be different. Therefore, $\textbf{near-hit}_i$ is expected to be close to x_i more often than $\textbf{near-miss}_i$ to x_i, and $\textbf{relevance}_i = E(\delta_i) \gg 0$.

[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.

If instead f_i is an irrelevant feature, the values of random variables x_i, **near-hit**$_i$ and **near-miss**$_i$ do not depend on one another. Therefore, (x_i – **near-hit**$_i$) and (x_i – **near-miss**$_i$) are independent. Since **near-hit**$_i$ and **near-miss**$_i$ are expected to obey the same distribution,[2]

$$E((x_i - \textbf{near-hit}_i)^2) = E((x_i - \textbf{near-miss}_i)^2)$$

$$E(\delta_i) = -E((x_i - \textbf{near-hit}_i)^2) + E((x_i - \textbf{near-miss}_i)^2) = 0$$

$$\textbf{relevance}_i = E(\delta_i) = 0$$

Therefore, statistically, the relevance level of a relevant feature is expected to be larger than zero and that of an irrelevant one is expected to be zero (or negative)[2].

4.2 Threshold τ

Figure 3 shows that those features whose relevance levels are greater than or equal to τ are selected and the rest are discarded. Hence the problem to pick a proper value of τ. Relief can be considered to statistically estimate the relevance level δ_i for each feature f_i, using interval estimation. First we assume all the features are irrelevant ($E(\delta_i) = 0$). τ gives the acceptance and critical regions of the hypothesis.

$$\text{acceptance-region} = \{ \xi_i \mid |\xi_i - E(\delta_i)| \leq \tau \} = \{ \xi_i \mid |\xi_i| \leq \tau \}$$

$$\text{critical-region} = \{ \xi_i \mid |\xi_i - E(\delta_i)| > \tau \} = \{ \xi_i \mid |\xi_i| > \tau \}$$

If the relevance level of a feature is in the acceptance region of the hypothesis, it is considered to be irrelevant. If the relevance level of a feature is in the critical region of the hypothesis, it is considered to be relevant.

One way to determine τ is to use Cebysev's inequality,

$$P(|\rho - E(\rho)| \leq h\sigma(\rho)) > 1 - 1/h^2$$

for any distribution of ρ

where $\sigma(\rho)$ is the standard deviation of ρ and $h > 0$.

Since x_i, **near-hit**$_i$, and **near-miss**$_i$ are normalized, $-1 \leq \delta_i \leq 1$. Since f_i is assumed to be irrelevant, $E(\delta_i) = 0$ and therefore $\sigma(\delta_i) \leq 1$. Since the relevance level relevance$_i$ is the average of δ_i over m sample instances, $E(\text{relevance}_i) = 0$ and $\sigma(\text{relevance}_i) = \sigma(\delta_i)/\sqrt{m} \leq 1/\sqrt{m}$. Therefore,

$$P(|\text{relevance}_i| \leq h/\sqrt{m}) \geq P(|\text{relevance}_i| \leq h\sigma(\text{relevance}_i)) > 1 - 1/h^2$$

According to the inequality, if we want the probability of a Type I error (rejecting the hypothesis when it is true) to be less than α, $1/h^2 \leq \alpha$ is sufficient. Therefore $h = 1/\sqrt{\alpha}$ is good enough. It follows that $\tau = h/\sqrt{m} = 1/\sqrt{\alpha m}$ is good enough to make the probability of a Type I error to be less than α.

Note that Cebysev's inequality does not assume any specific distribution of δ_i. h can usually be much smaller than $1/\sqrt{\alpha}$. Also, $\sigma(\delta_i)$ can be much smaller than 1 (e.g. for the discrete distribution of (0 : 1/2, 1 : 1/2), σ = 0.707. For the continuous uniform distribution over [0, 1], σ =0.0666). Since we only want $\tau = h\sigma$, τ can be much smaller than $1/\sqrt{\alpha m}$.

While the above formula determines τ by α (the value to decide how strict we want to be) and m (the sample size), experiments show that the relevance levels display clear contrast between relevant and irrelevant features [Kira & Rendell '92]. τ can also be determined by inspection.

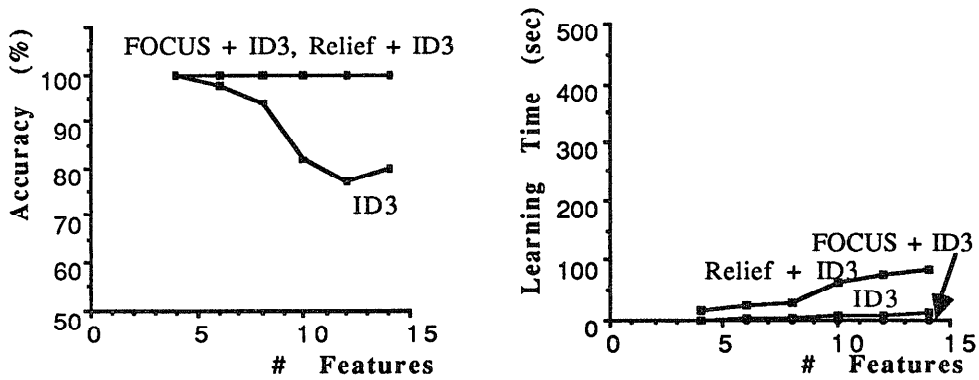
5 Empirical Evaluation

In section 2, we discussed three types of past approaches. One is concept learners alone, another is exhaustive search, the third is heuristic search. In this section, we compare Relief with these approaches. ID3 represents concept-learner-alone approach and also 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. Exhaustive search is represented by FOCUS [Almuallim & Dietterich 1991].

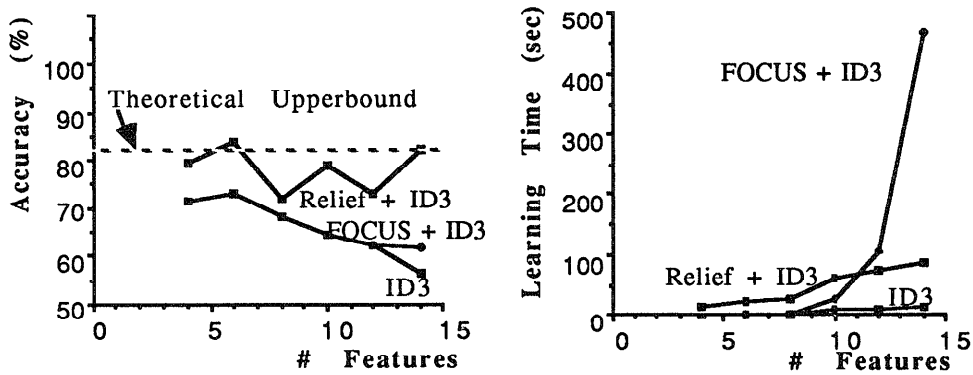
Figure 4 shows the results of comparing (1) ID3 alone, (2) FOCUS + ID3, and (3) Relief (m = 40, τ = 0.1) + ID3 in terms of predictive accuracy and learning time in a parity domain. The target concept is $f_1 \oplus f_2 = 1$. The horizontal axis shows the size of the given feature set $\mathcal{F}$ in which only two are relevant features. The results are the averages of 10 runs.

The predictive accuracy of ID3 alone was inferior to both FOCUS + ID3 and Relief + ID3. This shows the importance of feature selection algorithms. With noise-free data, both FOCUS + ID3 and Relief + ID3 learned the correct concept. FOCUS + ID3 is more effective than Relief + ID3, because FOCUS can select the two relevant features more quickly than Relief. With noisy data, however, the predictive accuracy of ID3 with Relief is higher than with FOCUS. In fact, 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. Thus Relief is a useful algorithm even when feature interaction is prevalent and the data is noisy. These results show that Relief is significantly faster than exhaustive search and more accurate than heuristic search.

[2] Strictly speaking, the distributions differ slightly. Since Relief does not allow X to be identical with Near-hit, some of their irrelevant feature values are expected to be different. On the other hand, since X and Near-miss are not identical (if there is no noise), all of their irrelevant feature values can be the same at the same time. This asymmetry tends to make $E(\delta_i)$ negative for irrelevant features.



(a) Tests with noise-free data



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

Figure 4 Test Results in Parity Domain

6 Current Limitation and Future Work

Relief requires retention of data in incremental uses. However it can be easily 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 them even though only a fraction are necessary for concept description.

Relief is applicable only to the two-class classification problem. However the algorithm 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 [Rendell & Seshu 1990] as **Near-hit** (Figure 3). It is crucial for Relief to pick real near-hit instances. One way is to give enough near-hit instances for all instances. 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 provide enough near-hit instances to detect relevance of those new features to the concept. These limitations also suggest research directions.

7 Conclusion

Relief is a simple algorithm which relies entirely on a statistical method. The algorithm employs few heuristics, and is less often fooled. It is efficient - its computational complexity is polynomial ($\Theta(pn)$). Relief is also noise-tolerant and is unaffected by feature interaction. This is especially important for hard real-world domains such as protein folding.

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 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., A Practical Approach to Feature Selection, Machine Learning : Proceedings of the Ninth International Conference (ML92), 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.