Appears in *Working Notes, Workshop on Data Engineering for Inductive Learning,*
Fourteenth International Joint Conference on Artificial Intelligence (1995).

Rapidly Estimating the Quality of Input Representations for Neural Networks[1]

Kevin J. Cherkauer **Jude W. Shavlik**

Department of Computer Sciences, University of Wisconsin-Madison
1210 West Dayton Street, Madison WI 53706, U.S.A.
Phone: 1-608-262-6613, Fax: 1-608-262-9777, E-mail: {cherkaue,shavlik}@cs.wisc.edu

Keywords: *Inductive Learning, Input Representation Selection/Quality Estimation, Empirical Evaluation*

Abstract

The choice of an input representation for machine learning can have a profound impact on the accuracy of the learned model in classifying novel instances. A reliable method of rapidly estimating the value of a representation, independent of the learner, would be a powerful tool in the search for better representations. We introduce a fast representation-quality measure that is more accurate than Rendell and Ragavan's *blurring* metric in rank ordering input representations for neural networks on two difficult, real-world datasets. This work constitutes a step forward both in representation quality measures and in our understanding of the characteristics that engender good representations.

1 Introduction

A major area of machine learning research is *inductive learning from examples,* where a system uses a set of classified *training examples* to induce a model, or "concept," that will accurately predict the classes of future examples. A main component of this approach is selecting an input representation for the examples. The most common technique describes each example by a vector of feature-value pairs. However, the particular features used are crucial to the learner's success. Given a large pool of candidate features, there are combinatorially many subsets that could be chosen, motivating the need for automated selection methods [4, 5, 6]. To consider a large number of representations, we need a fast measure of representation quality. This is most important for computationally expensive learning systems such as backpropagation-trained artificial neural networks (ANNs) [19], where a full cross-validation learning experiment may use weeks of computer time. In such cases it is impossible to search many representations without an inexpensive quality estimator.

In this paper we introduce three such estimators, which we call *transparency* measures. We evaluate the measures empirically using two difficult, real-world datasets, demonstrating that they produce reliable estimates of ANN input representation quality. In addition, these estimates are better than those produced by Rendell and Ragavan's *blurring* measure [17].

[1] **Acknowledgments:** This work was supported by ONR Grant N00014-93-1-0998, NSF grant CDA-9024618 (CM-5 use), and a NASA Graduate Student Researcher Program Fellowship held by KJC.

2 Why Transparency?

Other work in representation evaluation concentrates mainly on the ability to separate examples of different classes (which we call *coverage*) and on representation size (*economy*) [1, 11, 12, 13, 14, 21], the implicit assumption being that these are all one needs to evaluate representation quality. A typical heuristic is, "If two representations have equal coverage, prefer the more economical (smaller) one." We believe this intuition is incomplete. There is more to representation quality than coverage and economy, as our experiments show. We call the missing factor *transparency.*

Transparency describes the ease of learning an accurate concept under a given representation. If a representation tends to foster easily learnable, accurate concepts, it is transparent, while if most of the expressible accurate models are difficult to learn, it is opaque—even if it has good coverage and economy—and learners using it will likely perform poorly.

We must decide first what we mean by models that are "easy to learn," and second what it is about *representations* that influences this. We believe the ease of learning a model is determined by its complexity. Models with high-order feature interactions (e.g. lengthy conjunctions) or many disjunctions are difficult to learn. There are f-choose-d possible order-d feature interactions in a set of f features, so as interactions increase linearly, the search space containing such models grows combinatorially. Disjunctions partition the training data, so there are fewer examples available to learn each disjunct [9], leading to less accurate models. The problem of quickly estimating how well a *representation* fosters simple, accurate models is the focus of the remainder of this paper.

3 Transparency Measures

This paper's main question is, "How do we estimate a representation's tendency to produce simple, accurate models?" This is not the same as asking, "What is the simplest accurate model constructible under a given representation?" Answering the latter question requires an intractable search and may tell us little about how well we can expect to do in practice. What interests us is the *average* complexity of the accurate models possible under a representation. This is the representation's *transparency*, which estimates the complexity of the accurate models we expect typical learners to find. Transparency is related to minimum description length (MDL) [18] in that both have biases toward low complexity. MDL is a heuristic for estimating *model* quality ("smaller encodings are better"). Transparency is a heuristic for estimating *representation* quality based on the average complexity of the models a typical learner will build under a representation.

Some authors concerned with representation quality do not address transparency at all, and design feature-selection systems that may even minimize it (e.g. [1]). Others construct new features with the implicit goal of raising transparency, but do not attempt a formal definition of it (e.g. [11, 12, 13, 14, 21]). These methods likely do increase *transparency,* but the authors explain that this is done to (indirectly) reach an explicit goal of *economy.* The idea that transparency itself has value (indeed, more value than economy) seldom appears. In the following sections we discuss several different transparency estimators.

3.1 Blurring as a Transparency Measure

Rendell and Ragavan [17] do address the issue of transparency explicitly and present a method for quantifying it in a metric they call *blurring.* They claim that the less a representation requires the use of feature interactions to produce accurate models, the more transparent it is. However, *blurring* does not examine feature interactions *per se.* Instead it measures the average value, or information content, of a representation's individual features. This may often correlate with the level of feature interaction needed, but it does not directly measure it. *Blurring* is equivalent to the (negation of the) average information gain [15] of a representation's features with respect to a training set. (We show this in our Appendix.)

3.2 Model-Based Transparency Measures

We now introduce a new class of "model-based" transparency measures that sample actual models of the target concept to make their estimates. In our experiments, the sampled models are decision trees, chosen because they are simple, inexpensive, and commonly used. The experiments demonstrate that tree-based measures make good predictions for ANN performance, suggesting that the measures capture fundamental properties of the representations.

We evaluate three model-based measures below, two of which calculate the average complexity of n random decision trees that perfectly classify the training set ($n = 100$ below). Random tree-building is top-down; features are chosen with uniform probability from those which further partition the training examples (ignoring example class). Tree building terminates either when each leaf contains examples of only one class or when there are no more features that further partition the examples.

Our first measure estimates the expected disjunctive complexity of accurate models as the average number of leaves in n random decision trees:[2]

$$avg(leaves) = \frac{1}{n}\sum_{t=1}^{n} leaves(t)$$

where $leaves(t)$ is the number of leaves in tree t. High values indicate high model complexity (low transparency), since each leaf corresponds to a disjunct.

Min(leaves) is similar, except it uses the minimum number of leaves over the n trees:

$$min(leaves) = \min_{t=1,n}(leaves(t))$$

This is more guided than *avg(leaves)*, basing its estimate on the "best" model seen.

ID3leaves counts the number of leaves in the tree grown by Quinlan's [15] ID3 algorithm:

$$ID3leaves = leaves(ID3)$$

This is the most guided transparency metric we examine, since it bases its estimate on the most intelligently chosen model.

[2] Measures based on conjunctive complexity (i.e. leaf depth, or "number of decisions") performed more poorly and so are not included here.

4 Experiments

We evaluate the above transparency measures using ANNs as the learning systems for two problems: prediction of gene reading frames in DNA segments (six classes), and handwritten digit recognition (ten classes). In other work we have found that ANNs make more accurate class predictions than decision trees for these datasets, motivating our interest in developing inexpensive representation quality measures for ANNs, as the high ANN training cost limits the number of representations we can try directly.

The DNA data is from Craven and Shavlik [7] and consists of 20,000 examples from 30 "sufficiently independent" *E. coli* genes. Each example is a 61-base window from one gene, and the learner predicts which of six possible reading frames encodes the gene. The data is in four 5,000-example sets, with all examples from each gene kept in a single set, allowing four-fold cross-validation (CV) with no gene overlap between training and testing sets.

The digit data is the NIST "F13" set, available by ftp from `sequoyah.ncsl.nist.gov`. There are 3,471 images of the digits 0–9 from 49 different writers. We put 10% of the examples of each class randomly into each of ten files and used ten-fold CV for our experiments.

4.1 Experiment 1—Raw Representations

This experiment compares the transparency measures on ten input representations of different qualities and sizes, created by an algorithm we call RS ("Representation Selector"). RS first constructs a large pool of candidate Boolean features according to general user-defined feature types. This pool contained 5,460 features for DNA and 251,679 for NIST. For each CV fold, RS sorts the features by information gain on the entire training set. Then it scans the list, selecting each feature that is pairwise independent of all other features already selected. Independence is determined by calculating the X^2 statistic of a 2×2 contingency table for the training examples, where each example is one count in the table. The table rows and columns are indexed by the two possible values of each feature.

Table 1 shows such a contingency table and the corresponding X^2 statistic, which is approximately χ^2 with one degree of freedom. We test the hypothesis that the features are independent, rejecting it only if the X^2 value is ≥ 400, which implies feature *dependence* with greater than 0.999999 confidence. Such an easy acceptance criterion is needed because as the representation grows, new features must past tens or hundreds of tests to be included. Feature selection stops once the selected features can fully separate the training examples into their respective classes.

The above procedure produces a single reasonable input representation. (Similar methods

Table 1: **Left:** a contingency table tallying training examples according to the values of two features. **Right:** the corresponding X^2 statistic.

		Feature 1	
		0	1
Feature 2	0	a	b
	1	c	d

$$X^2 = \frac{(ad - bc)^2 \times (a + b + c + d)}{(a + c)(b + d)(a + b)(c + d)}$$

are used as feature-selection algorithms by others [2, 3].) To obtain different representations, we ran RS again with different initial feature pools. The new pools were derived from the original pool by deleting features whose training-set information gains were greater than or equal to various thresholds. We created nine additional representations for each problem by using the thresholds $\{0.30, 0.20, 0.15, 0.10, 0.05, 0.04, 0.03, 0.02, 0.01\}$. The representations generally degrade as more high-scoring features are deleted, giving us a range of representation qualities to test our measures on.

We evaluated the different measures by comparing the rank ordering they assigned the representations to that given by ANN test-set accuracies. We are not primarily interested in the ANN accuracies for these experiments; we only use them to create a ground-truth ranking of representation quality. The experiments test the ability of our transparency measures to approximate this ranking.

Our ground-truth rankings are from feed-forward ANNs trained with backpropagation (learning rate = 0.01, momentum = 0.9, 15 epochs for DNA, 50 epochs for NIST). We used a tuning set comprising 10% of the training set to pick the stopping epoch. For each representation, we tried six different numbers of hidden units in one layer: 0, 10, 20, 40, 80, and 160. The best CV accuracy among these six trials was used to rank the input representation relative to the others. Table 2 (left) shows the ANN accuracies and resultant rankings. With one exception, better initial feature pools yielded higher ANN test-set accuracies.

To evaluate the transparency measures, we computed the rank correlation coefficients, r_S, between the rankings they assigned and the ground-truth ranking for each dataset:

$$r_S = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}$$

where d_i is the difference between the integer ranks assigned to representation i by ground-

Table 2: **Left:** ground-truth ranking of the ten representations for each dataset in Exper. 1. *Rk:* rank. *Rep:* allowable information gains of features in the pool from which each representation was chosen. *% Acc:* test-set accuracy of the best ANN CV run for each representation; these determine the ranking. **Right:** ground-truth ranking compared to rankings assigned by transparency measures (listed by name) in Exper. 1. r_S: rank correlation coefficient.

	DNA		**NIST**	
Rk	Rep	% Acc	Rep	% Acc
1	All	82.2	All	92.4
2	≤ 0.30	79.7	≤ 0.30	91.9
3	≤ 0.20	76.5	≤ 0.20	91.0
4	≤ 0.10	75.2	≤ 0.15	89.6
5	≤ 0.15	74.5	≤ 0.10	87.6
6	≤ 0.05	70.2	≤ 0.05	77.1
7	≤ 0.04	69.8	≤ 0.04	74.4
8	≤ 0.03	66.4	≤ 0.03	67.4
9	≤ 0.02	58.8	≤ 0.02	62.9
10	≤ 0.01	49.3	≤ 0.01	50.0

DNA		**NIST**	
Measure	r_S	Measure	r_S
ID3leaves	0.988	ID3leaves	1.000
Min(leaves)	0.939	Min(leaves)	
Avg(leaves)	0.782	Avg(leaves)	
Blurring		Blurring	

truth and the transparency measure. Table 2 (right) shows these. For DNA, the best rankings are given by the most guided measures, *ID3leaves* and *min(leaves). Blurring* tied for third place with *avg(leaves).* For NIST, all measures produced perfect rankings, adding credence to their usefulness for diverse real-world problems. There is high confidence (greater than 0.98 for DNA and 0.99 for NIST) that a true correlation exists between each measure's ranking and ground-truth.

4.2 Experiment 2—Same-Size Representations

Some of the difference in representation quality observed in Exper. 1 may be due to differing representation sizes, not to transparency differences. Therefore, in Exper. 2 we equalized the representation sizes. Exper. 2 is the same as Exper. 1 except that feature selection was no longer terminated when all training examples were separated into their classes. Instead, RS continued to add features until a fixed number were selected, chosen to be greater than the number needed to correctly classify the training examples (200 features for DNA, 250 for NIST). Each Exper. 2 representation R_i^2 ($i = 1, \dots, 10$) is thus a proper superset of the corresponding Exper. 1 representation R_i^1, and all representations for a given dataset in Exper. 2 are identical in size and allow perfect classification of the training examples. Additionally, all the extra features added to each Exper. 2 representation had training-set information gains as bad or worse than the worst feature in the corresponding Exper. 1 representation.

Table 3 (left) shows the ground-truth rankings and ANN accuracies for Exper. 2. Again, better feature pools on average tended to produce better representations. Comparing to Table 2, note that in every case expanding the representations increased the ANN accuracies, even though the added features were poorer than the original ones. This shows that smaller consistent feature sets are not necessarily better as is often claimed [1, 11, 12, 13, 14, 21].

Table 3 (right) compares the transparency measures in Exper. 2. Again, all measures performed perfectly on the NIST data. On the DNA data, the three model-based mea-

Table 3: **Left:** ground-truth ranking of the ten representations for each dataset in Exper. 2. **Right:** ground-truth rankings compared to rankings assigned by transparency measures in Exper. 2.

	DNA		NIST	
Rk	Rep	% Acc	Rep	% Acc
1	All	84.4	All	94.0
2	≤ 0.30	82.9	≤ 0.30	93.7
3	≤ 0.15	82.8	≤ 0.20	93.0
4	≤ 0.10	82.3	≤ 0.15	92.7
5	≤ 0.20	82.2	≤ 0.10	92.0
6	≤ 0.04	80.2	≤ 0.05	88.5
7	≤ 0.05	79.4	≤ 0.04	85.3
8	≤ 0.03	77.7	≤ 0.03	80.8
9	≤ 0.02	71.5	≤ 0.02	73.7
10	≤ 0.01	55.3	≤ 0.01	54.4

DNA		NIST	
Measure	r_S	Measure	r_S
Min(leaves)	0.988	Min(leaves)	1.000
Avg(leaves)	0.964	Avg(leaves)	
ID3leaves	0.952	ID3leaves	
Blurring	0.806	Blurring	

sures did well, with rank correlations greater than 0.95. *Blurring's* predictions are poorer ($r_S = 0.806$). Exper. 2 strengthens our empirical evidence that transparency measures based on typical model complexity can be more accurate than simpler measures like *blurring.* Moreover, having equalized *coverage* and *economy* across representations in Exper. 2, we believe *transparency* accounts for most of the observed differences in ANN test-set accuracy.

4.3 Experiment 3—Redundant Features

ANNs can often use redundant features to advantage (e.g. see [20]), an ability generally not attributed to decision trees. Therefore, a transparency measure based on decision trees may be unable to accurately rank a pair of representations R^1 and R^2 to be used by an ANN, where R^2 is just R^1 plus some redundant features. Exper. 3 shows that a tree-based transparency measure can in fact accurately predict which representation is better for ANNs.

In Exper. 3 we compared the ten representations from Exper. 1 with the ten corresponding representations from Exper. 2. Recall that each Exper. 2 representation R_i^2 ($i = 1, \ldots, 10$) was created by adding extra features to an Exper. 1 representation R_i^1 (so $R_i^2 \supset R_i^1$). The additional features in Exper. 2 were not needed to separate the training data and also had lower training-set information gains than the original features of Exper. 1 they were added to. Thus, each pair of Exper. 1–Exper. 2 representations fits the situation described in the previous paragraph.

Across these ten representation pairs, in every case ANN test-set accuracy was higher for the Exper. 2 (larger, more redundant) representation (see Tables 2 and 3). Exper. 3 tests the ability of the various transparency measures to predict this. For each pair of corresponding Exper. 1 and Exper. 2 representations, we counted the number of times each transparency measure correctly scored Exper. 2's representation higher. Table 4 shows the results.

The only measure that correctly scored *any* representation pairs was *ID3leaves,* which is based on the ID3 decision tree. It scored all 20 pairs correctly, while the other measures scored all pairs incorrectly. ID3 must benefit from redundant features, probably because having more splitting options allows better greedy choices to be made.

The measures based on random trees, on the other hand, suffer in the face of redundant features. The added features in Exper. 2 are poorer than those shared with Exper. 1 representations. This lowers the average predictiveness of the features in the Exper. 2 representations. The quality of random feature choices for the tree nodes degrades with the average quality of the available features, so the redundant representations yield more complex random trees.

Blurring examines only the information gains of individual features. The representations of Exper. 2 must necessarily score worse under *blurring* because the added features were all

Table 4: Number of correct rankings (out of 10) by each transparency measure of the pairs of corresponding Exper. 1 and Exper. 2 representations.

Measure	ID3leaves	Min(leaves)	Avg(leaves)	Blurring
DNA	10	0	0	0
NIST	10	0	0	0

of lower information gains than the original Exper. 1 features.

The probability of *ID3leaves* ordering all 20 pairs of representations correctly by chance is $\left(\frac{1}{2}\right)^{20}$, which is less than one in a million. We conclude that *ID3leaves,* though tree-based, is nonetheless robust in the face of redundant features. The other measures clearly are not.

5 Discussion

Our experiments show that what we call transparency is an important factor in the quality of input representations for ANN learning. We also found that model-based transparency measures can be better at predicting representation quality than the less sophisticated *blurring* metric, even though we created the different representations by thresholding feature information gains, which should fit *blurring's* biases well.

Blurring is, however, the fastest measure discussed in this paper. It computes the information gain of each feature on the training set once and consumed on the order of a few CPU seconds per representation evaluated on a SPARC 10/30 workstation. *ID3leaves* builds an ID3 decision tree, doing approximately *blurring's* amount of work for each tree node and using a few CPU minutes per representation. However, the ANN CV runs required several CPU days for the larger representations, so *ID3leaves* still saves three orders of magnitude. *Blurring* and *ID3leaves* present a tradeoff between ranking accuracy and time consumed.

The other transparency measures were slower than *blurring* or *ID3leaves* (on the order of hours) when we sampled 100 random trees, and therefore may not be fast enough for many purposes unless the number of samples can be greatly reduced. Rerunning the experiments with a sample size of 10, which requires about the same amount of time per measure as *ID3leaves,* degraded the DNA results only slightly and the NIST results not at all.

Even though all the new transparency measures we introduced are based on *decision-tree* models, the rankings they gave were accurate for *neural networks.* This suggests that they measure something fundamental about representation quality for a perhaps broad class of learning systems rather than just artifacts of the particular models measured. Further work is needed to verify this supposition.

John *et al.* [10] propose that representations should be scored on the basis of models constructed by the actual learning algorithm to be used for production runs, as is done, for example, by Caruana and Freitag [4] with ID3 and C4.5 [16]. They believe that learner-independent quality measures cannot be accurate enough. However, this approach is too costly to be applied effectively to the ANNs we use here. We showed in this paper that inexpensive, learner-independent measures can indeed perform well.

6 Future Work

We would like to develop a single formula in *coverage*, *transparency,* and *economy* (CTE) that efficiently predicts representation quality. The experiments reported here isolated transparency, but in general one must take all three factors into account. An open question is how learner-dependent such a formula will be. We suspect that many learning techniques build similar *types* of models, and we hope that one formula may therefore apply to a broad range of learners. We intend to test our ideas on other standard learning algorithms and additional real-world datasets.

Low-cost quality measures make it possible to search a larger number of potential representations. A CTE formula could thus form the basis of broadly applicable representation-selection schemes (e.g. genetic search over representations using CTE values in the fitness function). Fast representation evaluation is necessary to search many representations for expensive learners.

Another issue is extending our transparency measures to continuously valued features. Fortunately, effective methods of discretizing continuous features for decision trees have already been developed [8, 16] and can be inserted directly into our measures.

7 Conclusions

We discussed transparency as an important factor in representation quality and developed inexpensive, effective ways to measure it. Empirical tests on two real-world datasets demonstrated these measures' accuracy at ranking representations for ANN learning at much lower cost than training the ANNs themselves. This work is a step forward in understanding representation quality and solving the problem of its rapid estimation.

Appendix

We show here that Rendell and Ragavan's ("R&R") *blurring* metric [17] is equivalent to the average information gain of a representation's features. R&R define the entropy H of a binary concept y with class values 0 and 1 as

$$H(y) = -[p(y)\log_2 p(y) + p(\overline{y})\log_2 p(\overline{y})]$$

where $p(y)$ and $p(\overline{y})$ are the prior probabilities that $y = 1$ and $y = 0$, respectively. This is a standard entropy measure, which Quinlan [15] calls $I(p, n)$, p and n being R&R's $p(y)$ and $p(\overline{y})$, respectively. R&R also define the entropy of y conditioned on feature x_i as

$$H(y|x_i) = -\textstyle\sum_v p(x_i = v)[p(y|x_i = v)\log_2 p(y|x_i = v) + p(\overline{y}|x_i = v)\log_2 p(\overline{y}|x_i = v)]$$

over all values v of x_i. Quinlan calls this $E(A)$, where his A is R&R's x_i. Finally, R&R define *blurring* as the average value of $H(y|x_i)$ over the n features in the representation

$$\Delta = \tfrac{1}{n}\textstyle\sum_{i=1}^{n} H(y|x_i) \qquad (1)$$

But note that Quinlan's definition of information gain is $gain(A) = I(p, n) - E(A)$ which in R&R's notation would be $gain(x_i) = H(y) - H(y|x_i)$ so average information gain is just

$$\overline{gain} = \tfrac{1}{n}\textstyle\sum_{i=1}^{n}[H(y) - H(y|x_i)] = H(y) - \tfrac{1}{n}\textstyle\sum_{i=1}^{n} H(y|x_i) \qquad (2)$$

Substituting Δ (Equation 1) into Equation 2 gives $\Delta = H(y) - \overline{gain}$. *Blurring* ($\Delta$) is thus equivalent to a constant minus the average information gain of the representation's features.

References

[1] H. Almuallim and T.G. Dietterich. Learning boolean concepts in the presence of many irrelevant features. *Artif. Intell.*, 69(1–2):279–305, 1994.

[2] P.W. Baim. A method for attribute selection in inductive learning systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):888–896, 1988.

[3] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994.

[4] R. Caruana and D. Freitag. Greedy attribute selection. In *Proc. 11th Intl. Conf. on Machine Learning*, pages 28–36, New Brunswick, NJ, 1994. Morgan Kaufmann.

[5] K.J. Cherkauer and J.W. Shavlik. Protein structure prediction: Selecting salient features from large candidate pools. In *Proc. 1st Intl. Conf. on Intelligent Systems for Mol. Bio.*, pages 74–82, Bethesda, MD, 1993. AAAI Press.

[6] K.J. Cherkauer and J.W. Shavlik. Selecting salient features for machine learning from large candidate pools through parallel decision-tree construction. In H. Kitano and J.A. Hendler, editors, *Massively Parallel Artificial Intelligence*, pages 102–136. AAAI Press/MIT Press, Menlo Park, CA/Cambridge, MA, 1994.

[7] M.W. Craven and J.W. Shavlik. Learning to predict reading frames in *E. coli* DNA sequences. In *Proc. 26th Hawaii Intl. Conf. on System Sci.*, pages 773–782, Wailea, HI, 1993. IEEE Computer Society Press.

[8] U.M. Fayyad and K.B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. 13th Intl. Joint Conf. on Artif. Intell.*, pages 1022–1027, Chambéry, Savoie, France, 1993. Morgan Kaufmann.

[9] R. Holte, L. Acker, and B. Porter. Concept learning and the problem of small disjuncts. In *Proc. 11th Intl. Joint Conf. on Artif. Intell.*, Detroit, MI, 1989. Morgan Kaufmann.

[10] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proc. 11th Intl. Conf. on Machine Learning*, pages 121–129, New Brunswick, NJ, 1994. Morgan Kaufmann.

[11] K. Kira and L.A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proc. 10th Nat. Conf. on Artif. Intell.*, pages 129–134, San Jose, CA, 1992. AAAI Press/MIT Press.

[12] C.J. Matheus. *Feature Construction: An Analytic Framework and an Application to Decision Trees.* PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 1990.

[13] G. Pagallo and D. Haussler. Boolean feature discovery in empirical learning. *Machine Learning*, 5(1):71–99, 1990.

[14] S. Piramuthu and H. Ragavan. Improving connectionist learning with symbolic feature construction. *Connection Sci.*, 4(1):33–43, 1992.

[15] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[16] J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA, 1993.

[17] L. Rendell and H. Ragavan. Improving the design of induction methods by analyzing algorithm functionality and data-based concept complexity. In *Proc. 13th Intl. Joint Conf. on Artif. Intell.*, pages 952–958, Chambéry, Savoie, France, 1993. Morgan Kaufmann.

[18] J. Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific, New Jersey, 1989.

[19] D.E. Rumelhart, G.E. Hinton, and R. Williams. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*, volume 1, pages 318–363. MIT Press, Cambridge, MA, 1986.

[20] R.S. Sutton and S.D. Whitehead. Online learning with random representations. In *Proc. 10th Intl. Conf. on Machine Learning*, pages 314–321, Amherst, MA, 1993. Morgan Kaufmann.

[21] J. Wnek and R.S. Michalski. Hypothesis-driven constructive induction in AQ17-HCI: A method and experiments. *Machine Learning*, 14(2):139–168, 1994.