5-3-1997

Feature Subset Selection Using a Genetic Algorithm

Jihoon Yang
Iowa State University

Vasant Honavar
Iowa State University

Feature Subset Selection Using A Genetic Algorithm

TR #97-02a

Jihoon Yang and Vasant Honavar

May 3, 1997

ACM Computing Classification System Categories (1991):
I.2.6 [*Artificial Intelligence*] Learning — connectionism and neural nets;
I.5.1 [*Pattern Recognition*] Models — neural nets;
I.5.2 [*Pattern Recognition*] Design Methodology — feature evaluation and selection.

Keywords:
feature subset selection, genetic algorithms, constructive neural net learning algorithms.

Artificial Intelligence Research Group
Department of Computer Science
226 Atanasoff Hall
Iowa Sate University
Ames, Iowa 50011-1040, USA

Feature Subset Selection Using A Genetic Algorithm[*]

Jihoon Yang and Vasant Honavar [†]
Artificial Intelligence Research Group
Department of Computer Science
226 Atanasoff Hall
Iowa State University
Ames, IA 50011
{yang|honavar}@cs.iastate.edu

Abstract

Practical pattern classification and knowledge discovery problems require selection of a subset of attributes or features (from a much larger set) to represent the patterns to be classified. This paper presents an approach to the multi-criteria optimization problem of feature subset selection using a genetic algorithm. Our experiments demonstrate the feasibility of this approach for feature subset selection in the automated design of neural networks for pattern classification and knowledge discovery.

1 Introduction

Many practical pattern classification tasks (e.g., medical diagnosis) require learning of an appropriate classification function that assigns a given input pattern (typically represented using a vector of attribute or feature values) to one of a finite set of classes. The choice of features, attributes, or measurements used to represent patterns that are presented to a classifier affect (among other things):

- The accuracy of the classification function that can be learned using an inductive learning algorithm (e.g., a decision tree induction algorithm or a neural network learning algorithm): The attributes used to describe the patterns implicitly define a pattern language. If the language is not expressive enough, it would fail to capture the information that is necessary for classification and hence regardless of the learning algorithm used, the accuracy of the classification function learned would be limited by this lack of information.

[*] This report is a revised version of TR97-02.

[†] This research was partially supported by the National Science Foundation (through grants IRI-9409580 and IRI-9643299) and the John Deere Foundation.

- The time needed for learning a sufficiently accurate classification function: For a given representation of the classification function, the attributes used to describe the patterns implicitly determine the search space that needs to be explored by the learning algorithm. An abundance of irrelevant attributes can unnecessarily increase the size of the search space, and hence the time needed for learning a sufficiently accurate classification function.

- The number of examples needed for learning a sufficiently accurate classification function: All other things being equal, the larger the number of attributes used to describe the patterns in a domain of interest, the larger is the number of examples needed to learn a classification function to a desired accuracy.

- The cost of performing classification using the learned classification function: In many practical applications e.g., medical diagnosis, patterns are described using observable symptoms as well as results of diagnostic tests. Different diagnostic tests might have different costs as well as risks associated with them. For instance, an invasive exploratory surgery can be much more expensive and risky than say, a blood test.

This presents us with a *feature subset selection problem* in automated design of pattern classifiers. The feature subset selection problem refers the task of identifying and selecting a useful subset of attributes to be used to represent patterns from a larger set of often mutually redundant, possibly irrelevant, attributes with different associated measurement costs and/or risks. An example of such a scenario which is of significant practical interest is the task of selecting a subset of clinical tests (each with different financial cost, diagnostic value, and associated risk) to be performed as part of a medical diagnosis task. Other examples of feature subset selection problem include large scale data mining applications, power system control, and so on.

2 Related Work

A number of approaches to feature subset selection have been proposed in the literature (of which only a few references are shown here). Some of these involve searching for an optimal subset of features based on some criteria of interest. [Almuallim & Dietterich, 1994] employs an exhaustive (breadth-first) search to find the minimal combination of features sufficient to construct a hypothesis consistent with the examples. Since exhaustive search over all possible combinations of features is not feasible, most current approaches to feature subset selection assume monotonicity of some measure of classification performance so that adding features is guaranteed to not worsen performance and use branch and bound search [Narendra & Fukunaga, 1977; Foroutan & Sklansky, 1987]. While they appear to work well with conventional statistical classifiers, their performance can be quite poor with non-linear classifiers such as neural networks [Ripley, 1996]. In many practical scenarios the monotonicity assumption is not satisfied.

Feature weighting is a variant of feature selection. It involves assigning a real-valued weight to each attribute. The weight associated with an attribute measures its relevance or importance in the classification task [Wettschereck *et al.*, 1995; Cost & Salzberg, 1993; Punch *et al.*, 1993]. Feature subset selection is a special case of weighting with binary weights.

Several authors have examined the use of *heuristic* search for feature subset selection (often in conjunction with branch and bound search) [Kira & Rendell, 1992; Modrzejewski, 1993; Liu & Setiono, 1995; John *et al.*, 1994; Kohavi, 1994; Kohavi & Frasca, 1994; Koller & Sahami, 1996]. Others have explored *randomized* [Liu & Setiono, 1996b; Liu & Setiono, 1996a] and randomized population-based heuristic search techniques such as genetic algorithms (GA) [Siedlecki & Sklansky, 1989; Punch *et al.*, 1993; Vafaie & De Jong, 1993; Brill *et al.*, 1992; Richeldi & Lanzi, 1996] to select feature subsets for use with decision tree or nearest neighbor classifiers.

Feature subset selection algorithms can be classified into two categories based on whether or not feature selection is done independently of the learning algorithm used to construct the classifier. If feature selection is done independent of the learning algorithm, the technique is said to follow a *filter* approach. Otherwise, it is said to follow a *wrapper* approach [John *et al.*, 1994]. While the filter approach is generally computationally more efficient than the wrapper approach, its major drawback is that an optimal selection of features may not be independent of the inductive and representational biases of the learning algorithm to be used to construct the classifier. The wrapper approach, on the other hand involves the computational overhead of evaluating candidate feature subsets by executing a selected learning algorithm on the dataset represented using each feature subset under consideration.

Feature subset selection techniques that make use of the monotonicity assumption in some form, although they appear to work reasonably well with linear classifiers, can exhibit poor performance with non-linear classifiers such as neural networks [Ripley, 1996]. Furthermore, in many practical scenarios the monotonicity assumption is not satisfied. For example, addition of irrelevant features (e.g., social security numbers in medical records in a diagnosis task) can significantly worsen the generalization accuracy of a decision tree classifier. Also, many of the feature selection techniques proposed in the literature (with the exception of GA) are not designed to handle with multiple selection criteria (e.g., classification accuracy, feature measurement cost, etc.).

3 Genetic Selection for Neural Network Pattern Classifiers

Feature subset selection in the context of many practical problems (e.g., diagnosis) presents an instance of a multi-criteria optimization problem. The multiple criteria to be optimized include the accuracy of classification, cost and risk associated with classification which in turn depends on the selection of attributes used to describe the patterns. Evolutionary algorithms offer a particularly attractive approach to multi-criteria optimization problems. This paper explores a wrapper-based multi-criteria approach to feature subset selection using a genetic algorithm in conjunction with a relatively fast inter-pattern distance-based neural network learning algorithm. However, the general approach can be used with any inductive learning algorithm. The interested reader is referred to [Honavar, 1994; Langley, 1995; Mitchell, 1997] for surveys of different approaches to inductive learning.

Neural networks – densely interconnected networks of relatively simple computing elements (e.g., threshold or sigmoid neurons) – offer an attractive framework for the design of pattern classifiers for real-world real-time pattern classification tasks on account of their potential for parallelism and fault and noise tolerance [Ripley, 1996; Hassoun, 1995; Gallant, 1993]. The classification function realized by a neural network is determined by the functions computed by the neurons, the connectivity of the network, and the parameters (weights) associated with the connections. It is well-known that multi-layer networks of non-linear computing elements (e.g., threshold neurons) can realize any classification function $\phi : \Re^n \rightarrow C$ or $\phi : D^n \rightarrow C$ where C is a finite set of classes and n is a finite number of discrete or real valued attributes, $\Re$ is the set of real numbers, and D is a finite set of discrete values. If the attributes are symbolic, they have to be first mapped to numeric values using appropriate coding schemes. While evolutionary algorithms are generally quite effective for rapid global search of large search spaces in multi-modal optimization problems, neural networks offer a particularly attractive approach to finetuning solutions once promising regions in the search space have been identified [Mitchell, 1996]. Against this background, genetic algorithms offer an attractive approach to feature subset selection for neural network pattern classifiers.

However, the use of genetic algorithms for feature subset selection for neural network pattern classifiers trained using traditional neural network training algorithms presents some practical problems:

- Traditional neural network learning algorithms (e.g., backpropagation) perform an error gradient guided search for a suitable setting of weights in the weight space determined by a user-specified network architecture. This ad hoc choice of network architecture often inappropriately constrains the search for an appropriate setting of weights. For example, if the network has fewer neurons than necessary, the learning algorithm will fail to find the desired classification function. If the network has far more neurons than necessary, it can result in overfitting of the training data leading to poor generalization. In either case, it would make it difficult to evaluate the usefulness of a feature subset employed to describe (or represent) the training patterns used to train the neural network.

- Gradient based learning algorithms although mathematically well-founded for unimodal search spaces, can get caught in local minima of the error function. This can complicate the evaluation of the usefulness of a feature subset employed to describe the training patterns used to train the neural networks.

- A typical run of a genetic algorithm involves many generations. In each generation, evaluation of an individual (a feature subset) involves training neural networks and computing their accuracy and cost. This can make the fitness evaluation rather expensive since gradient based algorithms are typically quite slow. The problem is further exacerbated by the fact that multiple neural networks have to be used to sample the space of ad-hoc choices of network architecture in order to get a reliable fitness estimate for each feature subset represented in the population.

Fortunately, constructive neural network learning algorithms [Honavar & Uhr, 1993; Gallant, 1993] eliminate the need for ad-hoc, and often inappropriate a-priori choices of network architectures; and can potentially discover near-minimal networks whose size is commensurate with the complexity of the classification task that is implicitly specified by the training data. Several new, provably convergent, and relatively efficient constructive learning algorithms for multi-category real as well as discrete valued pattern classification tasks have begun to appear in the literature [Parekh *et al.*, 1996; Yang *et al.*, 1997]. Many of these algorithms have demonstrated very good performance in terms of reduced network size, learning time, and generalization in a number of experiments with both artificial and fairly large real-world datasets [Honavar & Uhr, 1993; Parekh *et al.*, 1996; Yang *et al.*, 1997].

DistAI [Yang *et al.*, 1997] is a simple and fast constructive neural network learning algorithm for pattern classification. The results presented in this paper are based experiments using neural networks constructed by DistAI. The key idea behind DistAI is to add hidden neurons one at a time based on a greedy strategy which ensures that the hidden neuron correctly classifies a maximal subset of training patterns belonging to a single class. Correctly classified examples can then be eliminated from further consideration. The process terminates when this process results in an empty training set (when the network correctly classifies the entire training set). When this happens, the training set becomes linearly separable in the transformed space defined by the hidden neurons. In fact, it is possible to set the weights on the hidden to output neuron connections without going through an iterative process. It is straightforward to show that DistAI is guaranteed to converge to 100% classification accuracy on any finite training set in time that is polynomial in the number of training patterns. Experiments reported in [Yang *et al.*, 1997] show that DistAI, despite its simplicity, yields classifiers that compare quite favorably with those generated using more sophisticated (and substantially more computationally demanding) constructive learning algorithms. This makes DistAI an attractive choice for experimenting with evolutionary approaches to feature subset selection for neural network pattern classifiers.

4 Implementation Details

Our experiments were run using a standard genetic algorithm [Goldberg, 1989; Mitchell, 1996] with rank-based selection strategy. The reported results are based on 10-fold cross-validation for each classification task with the following parameter settings:

- Population size: 50
- Number of generation: 20
- Probability of crossover: 0.6
- Probability of mutation: 0.001
- Probability of selection of the highest ranked individual: 0.6

Each individual in the population represents a candidate solution to the feature subset selection problem. Let m be the total number of attributes available to choose from to represent the patterns to be classified. In a medical diagnosis task, these would be observable symptoms and a set of possible diagnostic tests that can be performed on the patient. (Note that given m such attributes, there exist 2^m possible feature subsets. Thus, for large values of m, exhaustive search is not feasible). It is represented by a binary vector of dimension m

(where m is the total number of attributes). If a bit is a 1, it means that the corresponding attribute is selected. A value of 0 indicates that the corresponding attribute is not selected. The fitness of an individual is determined by evaluating the neural network constructed by DistAl using a training set whose patterns are represented using only the selected subset of features. If an individual has n bits turned on, the corresponding neural network has n input nodes.

The fitness function has to combine two different criteria – the accuracy of the classification function realized by the neural network and the cost of performing classification. The accuracy of the classification function can be estimated by calculating the percentage of patterns in a test set (determined by 10-fold cross validation) that are correctly classified by the neural network in question. A number of different measures of the cost of classification suggest themselves: cost of measuring the value of a particular attribute needed for classification (or the cost of performing the necessary test in a medical diagnosis application), the risk involved, etc. To keep things simple, we chose a relatively simple form of a 2-criteria fitness function defined as follows:

$$fitness(x) = accuracy(x) - \frac{cost(x)}{accuracy(x)+1} + cost_{max} \tag{1}$$

where $fitness(x)$ is the fitness of the feature subset represented by x, $accuracy(x)$ is the test accuracy of the neural network classifier trained using DistAl using the feature subset represented by x, and $cost_{max}$ is an upper bound on the costs of candidate solutions. In this case, it is simply the sum of the costs associated with all of the attributes. This is clearly a somewhat ad hoc choice. However, it does discourage trivial solutions (e.g., a zero cost solution with a very low accuracy) from being selected over reasonable solutions which yield high accuracy at a moderate cost. It also ensures that $\forall x \; 0 \leq fitness(x) \leq (100 + cost_{max})$. In practice, defining suitable tradeoffs between the multiple objectives has to be based on knowledge of the domain. In general, it is a non-trivial task to combine multiple optimization criteria into a single fitness function. A wide variety of approaches have been examined in the utility theory literature [Keeney & Raiffa, 1976].

5 Experiments

5.1 Description of Datasets

The experiments reported here used real-world datasets as well as a carefully constructed artificial dataset (3-bit parity) to explore the feasibility of using genetic algorithms for feature subset selection for neural network classifiers. The real-world datasets were obtained from the machine learning data repository at the University of California at Irvine. [1]

5.1.1 3-bit Parity Dataset

This dataset was constructed to explore the effectiveness of the genetic algorithm in selecting an appropriate subset of relevant attributes in the presence of redundant attributes so as

[1] [http://www.ics.uci.edu/AI/ML/MLDBRepository.html]

Table 1: Datasets used in experiments.

Dataset	Size	Dimension	Attribute Type	Class
3-bit parity problem (**3P**)	100	13	numeric	2
pittsburgh bridges (**Bridges**)	105	11	numeric, nominal	6
breast cancer (**Cancer**)	699	9	numeric	2
credit screening (**CRX**)	690	15	numeric, nominal	2
glass identification (**Glass**)	214	9	numeric	6
heart disease (**Heart**)	270	13	numeric, nominal	2
heart disease [Cleveland](**HeartCle**)	303	13	numeric, nominal	2
heart disease [Hungarian](**HeartHun**)	294	13	numeric, nominal	2
heart disease [Long Beach](**HeartLB**)	200	13	numeric, nominal	2
heart disease [Swiss](**HeartSwi**)	123	13	numeric, nominal	2
hepatitis domain (**Hepatitis**)	155	19	numeric, nominal	2
horse colic (**Horse**)	300	22	numeric, nominal	2
ionosphere structure (**Ionosphere**)	351	34	numeric	2
liver disorders (**Liver**)	345	6	numeric	2
pima indians diabetes (**Pima**)	768	8	numeric	2
DNA sequences (**Promoters**)	106	57	nominal	2
sonar classifiction (**Sonar**)	208	60	numeric	2
house votes (**Votes**)	435	16	nominal	2
vowel recognition (**Vowel**)	528	10	numeric	11
wine recognition (**Wine**)	178	13	numeric	3
zoo database (**Zoo**)	101	16	numeric, nominal	7

to minimize the cost and maximize the accuracy of the resulting neural network pattern classifier. The modified training set is constructed as follows: The original attributes are replicated once (to introduce redundancy) thereby doubling the number of attributes. Then an additional set of irrelevant attributes are generated and are assigned random boolean values. 100 7-bit random vectors were generated and augmented with the 6-bit vectors (corresponding to the original 3 bits plus an identical set of 3 bits). Each attribute in the resulting dataset is assigned a random cost between 0 and 9.

5.1.2 Real-world Datasets

In our experiments with real world datasets, our objective was to compare the neural networks built using feature subsets selected by the genetic algorithm with those that use the entire set of attributes available. Some medical datasets have information about the costs of measuring the attributes [Turney, 1995], but most of the datasets do not have the information. Thus the focus was on identifying a minimal subset of attributes that yield high accuracy neural network classifiers for all datasets. Moreover, for the datasets with cost information, the performance considering the cost in addition to the accuracy (by equa-

Table 2: Comparison of neural network pattern classifiers constructed using the entire set of attributes against those constructed using the best (in accuracy) GA-selected subset.

	All Attributes			GA-selected Subset		
Dataset	*Dimension*	*Accuracy*	*Hidden*	*Dimension*	*Accuracy*	*Hidden*
3P	13	79.0±12.2	5.0 ± 2.0	6.6 ± 1.6	100 ± 0.0	9.2 ± 4.9
Bridges	11	63.0 ± 7.8	5.2 ± 3.3	5.6 ± 1.5	81.6 ± 7.6	17.6 ± 12.4
Cancer	9	97.8 ± 1.2	2.9 ± 1.2	5.4 ± 1.4	99.3 ± 0.9	5.7 ± 2.9
CRX	15	87.7 ± 3.3	7.7 ± 6.9	8.0 ± 2.1	91.5 ± 2.8	12.5 ± 7.6
Glass	9	70.5 ± 8.5	9.8 ± 6.9	5.5 ±1.4	80.8 ±5.0	14.5 ± 6.6
Heart	13	86.7 ± 7.6	5.7 ± 4.4	7.2 ±1.6	93.9 ±3.8	7.5 ± 3.9
HeartCle	13	85.3 ± 2.7	3.4 ± 1.1	7.3 ±1.7	92.9 ±3.6	7.6 ± 4.2
HeartHun	13	85.9 ± 6.3	5.0 ± 2.9	7.0 ±1.2	93.0 ±4.0	7.1 ± 3.7
HeartSwi	13	94.2 ± 3.8	2.2 ± 0.6	6.6 ±1.7	98.3 ±3.3	3.7 ± 1.5
HeartVa	13	80.0 ± 7.4	5.1 ± 2.6	7.1 ±1.7	91.0 ±5.7	8.5 ± 3.0
Hepatitis	19	84.7 ± 9.5	6.2 ± 4.0	9.2 ±2.3	97.1 ±4.3	8.1 ± 2.8
Horse	22	86.0 ± 3.6	5.3 ± 4.5	11.1 ±2.3	92.6 ±3.4	9.5 ± 4.1
Ionosphere	34	94.3 ± 5.0	5.5 ± 1.6	17.3 ±3.5	98.6 ±2.4	7.5 ± 2.4
Liver	6	72.9 ± 5.1	21.5 ± 27.3	4.1 ±0.7	77.8 ±4.0	25.9 ± 24.3
Pima	8	76.3 ± 5.1	8.1 ± 4.9	3.8 ±1.5	79.5 ±3.1	20.8 ± 21.2
Promoters	57	88.0 ± 7.5	2.2 ± 0.4	28.8 ±3.3	100 ±0.0	2.7 ± 1.0
Sonar	60	83.0 ± 7.8	6.4 ± 2.7	30.7 ±3.7	97.2 ±2.9	7.2 ± 3.0
Votes	16	96.1 ± 1.5	3.2 ± 1.5	8.9 ±1.8	98.8 ±1.2	4.0 ± 1.8
Vowel	10	69.8 ± 6.4	38.0 ± 8.3	6.5 ±1.2	78.4 ±3.8	41.5 ± 7.7
Wine	13	97.1 ± 4.0	5.5 ± 1.7	6.7 ±1.6	99.4 ±2.1	5.9 ± 2.1
Zoo	16	96.0 ± 4.9	6.1 ± 1.1	9.3 ±1.6	100 ±0.0	6.2 ± 1.1

tion (1)) was compared with that considering the accuracy only. Table 1 summarizes the characteristics of the datasets.

5.2 Experimental Results

The results with 10 different training/test sets based on 10-fold cross-validation were averaged and shown in the following tables.

5.2.1 Fitness Function with Accuracy Only

Selections in the genetic algorithm were based only on the accuracy to compare its performance with that made with the entire set of attributes. The results (in terms of the number of attributes chosen (*Dimension*), the generalization accuracy (*Accuracy*) and the network size (*Hidden*)) are shown in Table 2.

The results shown in Table 2 indicate that the networks constructed using GA-selected subset of attributes compare quite favorably with networks that use all of the attributes. The generalization accuracy always increased significantly with comparable network size (but

Table 3: Comparison of the GA-based neural network pattern classifiers of using accuracy vs. accuracy and measurement costs of attributes.

	Accuracy only			Accuracy & Cost			
Dataset	*Dimension*	*Accuracy*	*Hidden*	*Dimension*	*Accuracy*	*Cost*	*Hidden*
3P	6.6 ± 1.6	100 ± 0.0	9.2 ± 4.9	4.3 ± 1.2	100 ± 0.0	26.7 ± 7.6	7.3 ± 4.2
Hepatitis	9.2 ± 2.3	97.1 ± 4.3	8.1 ± 2.8	8.3 ± 2.4	97.3 ± 3.5	19.0 ± 8.1	7.4 ± 2.8
HeartCle	7.3 ± 1.7	92.9 ± 3.6	7.6 ± 4.2	6.1 ± 1.6	93.0 ± 3.4	261.5 ± 94.4	7.2 ± 5.1
Pima	3.8 ± 1.5	79.5 ± 3.1	20.8 ± 21.2	3.1 ± 1.0	79.5 ± 3.0	22.8 ± 9.7	16.0 ± 11.1

substantially fewer connections). The results are also comparable to those in [Richeldi & Lanzi, 1996] (though it is not generally feasible to do a fair comparison between different approaches without the complete knowledge on the parameters used).

5.2.2 Fitness Function with both Accuracy and Cost

Selection is based on both the generalization accuracy and the measurement cost of attributes. (See the fitness function in equation (1)). **3P**, **Hepatitis**, **HeartCle** and **Pima** datasets were used for the experiment (with the random costs in **3P**). The results are shown in Table 3.

As we can see from table 3, the combined fitness function of accuracy and cost outperformed that of accuracy only in every aspect: the dimension, generalization accuracy and network size. This is not surprising because the former tries to minimize cost (while maximizing the accuracy), which cuts down the dimension, while the latter emphasizes only on the accuracy. Some of the runs resulted in feature subsets which did not necessarily have minimum cost. This suggests the possibility of improving the results by the use of a more principled choice of a fitness function that combines accuracy and cost.

6 Summary and Discussion

The results presented in this paper indicate that genetic algorithms offer an attractive approach to solving the feature subset selection problem (under a different cost and performance constraints) in inductive learning of neural network pattern classifiers. This finds applications in cost-sensitive design of classifiers for tasks such as medical diagnosis, computer vision, among others. Other applications of interest include automated data mining and knowledge discovery from datasets with an abundance of irrelevant or redundant attributes. In such cases, identifying a relevant subset that adequately captures the regularities in the data can be particularly useful. The GA-based approach to feature subset selection does not rely on monotonicity assumptions that are used in traditional approaches to feature selection which often limits their applicability to real-world classification and knowledge acquisition tasks.

Some directions for further research include: Application of GA-based approaches to feature subset selection to large-scale pattern classification tasks that arise in power systems control, gene sequence recognition, and data mining and knowledge discovery; Extensive experimental (and whenever feasible, theoretical) comparison of the performance of the proposed approach with that of conventional methods for feature subset selection; More principled design of multi-objective fitness functions for feature subset selection using domain knowledge as well as mathematically well-founded tools of multi-attribute utility theory. Some of these topics are the focus of our ongoing research.

References

Almuallim, H., & Dietterich, T. (1994). Learning Boolean Concepts in the Presence of Many Irrelevant Features. *Artificial Intelligence*, **69**(1-2), 279–305.

Brill, F., Brown, D., & Martin, W. (1992). Fast Genetic Selection of Features for Neural Network Classifiers. *IEEE Transactions on Neural Networks*, **3**(2), 324–328.

Cost, S., & Salzberg, S. (1993). A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. *Machine Learning*, **10**(1), 57–78.

Foroutan, I., & Sklansky, J. (1987). Feature Selection for Automatic Classification of non-Gaussian Data. *IEEE Transactions on Systems, Man and Cybernetics*, **17**, 187–198.

Gallant, S. (1993). *Neural Network Learning and Expert Systems.* Cambridge, MA: MIT Press.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* New York: Addison-Wesley.

Hassoun, M. (1995). *Fundamentals of Artificial Neural Networks.* Boston, MA: MIT Press.

Honavar, V. (1994). Toward Learning Systems That Integrate Multiple Strategies and Representations. *Pages 615–644 of:* Honavar, V., & Uhr, L. (eds), *Artificial Intelligence and Neural Networks: Steps Toward Principled Integration.* Academic Press: New York.

Honavar, V., & Uhr, L. (1993). Generative Learning Structures and Processes for Connectionist Networks. *Information Sciences*, **70**, 75–108.

John, G., Kohavi, R., & Pfleger, K. (1994). Irrelevant Features and the Subset Selection Problem. *Pages 121–129 of: Machine Learning: Proceedings of the Eleventh International Conference.* Morgan Kaufmann.

Keeney, R., & Raiffa, H. (1976). *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* New York: Wiley.

Kira, K., & Rendell, L. (1992). A Practical Approach to Feature Selection. *Pages 249–256 of: Proceedings of the Ninth International Conference on Machine Learning.* Morgan Kaufmann.

Kohavi, R. (1994). Feature Subset Selection as Search with Probabilistic Estimates. *In: AAAI Fall Symposium on Relevance.*

Kohavi, R., & Frasca, B. (1994). Useful Feature Subsets and Rough Set Reducts. *In: Third International Workshop on Rough Sets and Soft Computing.*

Koller, D., & Sahami, M. (1996). Toward Optimal Feature Selection. *In: Machine Learning: Proceedings of the Thirteenth International Conference.* Morgan Kaufmann.

Langley, P. (1995). *Elements of Machine Learning.* Palo Alto, CA: Morgan Kaufmann.

Liu, H., & Setiono, R. (1995). Chi2: Feature Selection and Discretization of Numeric Attributes. *In: Proceedings of the Seventh IEEE International Conference on Tools with Artificial Intelligence.*

Liu, H., & Setiono, R. (1996a). Feature Selection and Classification - A Probabilistic Wrapper Approach. *In: Proceedings of the Ninth International Conference on Industrial and Engineering Applications of AI and ES.*

Liu, H., & Setiono, R. (1996b). A Probabilistic Approach to Feature Selection - A Filter Solution. *In: Proceedings of the Thirteenth International Conference on Machine Learning.* Morgan Kaufmann.

Mitchell, M. (1996). *An Introduction to Genetic algorithms.* Cambridge, MA: MIT Press.

Mitchell, T. (1997). *Machine Learning.* New York: McGraw Hill.

Modrzejewski, M. (1993). Feature Selection Using Rough Sets Theory. *Pages 213–226 of: Proceedings of the European Conference on Machine Learning.* Springer.

Narendra, P., & Fukunaga, K. (1977). A Branch and Bound Algorithm for Feature Subset Selection. *IEEE Transactions on Computers*, **26**, 917–922.

Parekh, R., Yang, J., & Honavar, V. (1996). *Constructive Neural Network Learning Algorithms for Multi-Category Real-Valued Pattern Classification.* Tech. rept. TR 96-14. Department of Computer Science, Iowa State University.

Punch, W., Goodman, E., Pei, M., Chia-Shun, L., Hovland, P., & Enbody, R. (1993). Further Research on Feature Selection and Classification Using Genetic Algorithms. *Pages 557–564 of: Proceedings of the International Conference on Genetic Algorithms.* Springer.

Richeldi, M., & Lanzi, P. (1996). Performing Effective Feature Selection by Investigating the Deep Structure of the Data. *Pages 379–383 of: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining.* AAAI Press.

Ripley, B. (1996). *Pattern Recognition and Neural Networks.* New York: Cambridge University Press.

Siedlecki, W., & Sklansky, J. (1989). A Note on Genetic Algorithms for Large-scale Feature Selection. *IEEE Transactions on Computers*, **10**, 335–347.

Turney, P. (1995). Cost-Sensitive Classification: Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm. *Journal of Artificial Intelligence Research*, **2**, 369–409.

Vafaie, H., & De Jong, K. (1993). Robust Feature Selection Algorithms. *Pages 356–363 of: Proceedings of the IEEE International Conference on Tools with Artificial Intelligence.*

Wettschereck, D., Aha, D., & Mohri, T. (1995). *A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms.* Tech. rept. AIC95-012. Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence, Washington, D.C.

Yang, J., Parekh, R., & Honavar, V. (1997). DistAI*: An Inter-pattern Distance-based Constructive Learning Algorithm.* Tech. rept. ISU-CS-TR 97-05. Iowa State University.