Feature Subset Selection Using a Genetic Algorithm

Jihoon Yang and Vasant Honavar, Iowa State University

PRACTICAL PATTERN-CLASSIFICATION AND KNOWLEDGE-DISCOVERY PROBLEMS REQUIRE THE SELECTION OF A SUBSET OF ATTRIBUTES OR FEATURES TO REPRESENT THE PATTERNS TO BE CLASSIFIED. THE AUTHORS' APPROACH USES A GENETIC ALGORITHM TO SELECT SUCH SUBSETS, ACHIEVING MULTICRITERIA OPTIMIZATION IN TERMS OF GENERALIZATION ACCURACY AND COSTS ASSOCIATED WITH THE FEATURES.

IN PRACTICAL PATTERN-CLASSIFICATION tasks such as medical diagnosis, a classification function learned through an inductive learning algorithm assigns a given input pattern to one of a finite set of classes. Typically, the representation of each input pattern consists of a vector of attribute, feature, or measurement values. The choice of features to represent the patterns affects several aspects of pattern classification, including

- *Accuracy.* The features used to describe the patterns implicitly define a pattern language. If the language is not expressive enough, it fails to capture the information necessary for classification. Hence, regardless of the learning algorithm, the amount of information given by the features limits the accuracy of the classification function learned.
- *Required learning time.* The features describing the patterns implicitly determine the search space that the learning algorithm must explore. An abundance of irrelevant features can unnecessarily increase the size of the search space and hence the time needed for learning a sufficiently accurate classification function.
- *Necessary number of examples.* All other things being equal, the larger the number of features describing the patterns, the larger the number of examples needed to train a classification function to the desired accuracy.
- *Cost.* In medical diagnosis, for example, patterns consist of observable symptoms along with the results of diagnostic tests. These tests have various associated costs and risks; for instance, an invasive exploratory surgery can be much more expensive and risky than, say, a blood test.

In the automated design of pattern classifiers, these variables present us with the *feature subset selection problem.* This is the task of identifying and selecting a useful subset of pattern-representing features from a larger set of features. The features in the larger set have different associated measurement costs and risks, and some may be irrelevant or mutually redundant.

A significant, practical example of such a scenario is the task of selecting a subset of clinical tests—each with a different financial cost, diagnostic value, and associated risk—to be performed for medical diagnosis. Other instances of the feature subset selection problem arise in, for example, large-scale data-mining applications and power system control.

Several approaches to feature subset selection exist (see the "Related work" sidebar); ours employs a genetic algorithm. The experiments we describe in this article demonstrate the effectiveness of our approach in the automated design of neural networks for pattern classification and knowledge discovery.

1094-7167/98/$10.00 © 1998 IEEE

Related work

There have been several proposals of approaches to feature subset selection. (We discuss only a few of these in this article; our recent work[1] contains a more complete list of references.) Some of these approaches involve searching for an optimal subset of features based on particular criteria of interest.

Feature weighting is a variant of feature selection. It involves assigning a real-valued weight to each feature. The weight associated with a feature measures its relevance or significance in the classification task.[2] Feature subset selection is a special case of weighting with binary weights.

Several authors have examined the use of a *heuristic* search for feature subset selection; this often operates in conjunction with a branch-and-bound search.[3] Others have explored *randomized*[4] and randomized, population-based heuristic search techniques such as genetic algorithms[5,6] to select feature subsets for use with decision-tree or nearest-neighbor classifiers.

Feature subset selection algorithms fall into two categories based on whether or not they perform feature selection independently of the learning algorithm that constructs the classifier. If the technique performs feature selection independently of the learning algorithm, it follows a *filter* approach. Otherwise, it follows a *wrapper* approach.[3]

The filter approach is generally computationally more efficient. However, its major drawback is that an optimal selection of features may not be independent of the inductive and representational biases of the learning algorithm that constructs the classifier. The wrapper approach, on the other hand, incurs the computational overhead of evaluating candidate feature subsets by executing a selected learning algorithm on the data set using each feature subset under consideration.

Because exhaustive search over all possible combinations of features is not computationally feasible, most current approaches assume monotonicity of some measure of classification performance and then use branch-and-bound search. This ensures that adding features does not worsen performance. Techniques that make this monotonicity assumption in some form appear to work reasonably well with linear classifiers. However, they can exhibit poor performance with nonlinear classifiers such as neural networks.[7] Furthermore, many practical scenarios do not satisfy the monotonicity assumption. For example, irrelevant features (for example, social security numbers in medical records in a diagnosis task) can significantly worsen a decision tree classifier's generalization accuracy. Also, most of the proposed feature selection techniques (with the exception of those using genetic algorithms) are not designed to handle multiple selection criteria (classification accuracy, feature measurement cost, and so on).

The multicriteria approach that we explore in this article is wrapper-based and uses a genetic algorithm in conjunction with a relatively fast, interpattern distance-based, neural-network learning algorithm. However, this general approach works with any inductive learning algorithm.

References

1. J. Yang and V. Honavar, "Feature Subset Selection Using a Genetic Algorithm," *Feature Extraction, Construction and Selection—A Data Mining Perspective*, Liu and Motoda, eds., Kluwer Academic Publishers, Boston, forthcoming, 1998.
2. S. Cost and S. Salzberg, "A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features," *Machine Learning*, Vol. 10, No. 1, Jan. 1993, pp. 57–78.
3. G. John, R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," *Proc. 11th Int'l Conf. Machine Learning*, Morgan Kaufmann, San Francisco, 1994, pp. 121–129.
4. H. Liu and R. Setiono, "A Probabilistic Approach to Feature Selection—A Filter Solution," *Proc. 13th Int'l Conf. Machine Learning*, Morgan Kaufmann, 1996, pp. 319–327.
5. F. Brill, D. Brown, and W. Martin, "Fast Genetic Selection of Features for Neural Network Classifiers," *IEEE Trans. Neural Networks*, Vol. 3, No. 2, Mar. 1992, pp. 324–328.
6. M. Richeldi and P. Lanzi, "Performing Effective Feature Selection by Investigating the Deep Structure of the Data," *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, Calif., 1996, pp. 379–383.
7. B. Ripley, *Pattern Recognition and Neural Networks*, Cambridge Univ. Press, New York, 1996.

Why a genetic algorithm?

Feature subset selection in the context of practical problems such as diagnosis presents a multicriteria optimization problem. The criteria to be optimized include the classification's accuracy, cost, and risk. Evolutionary algorithms offer a particularly attractive approach to multicriteria optimization because they are effective in high-dimensional search spaces.[1]

Neural networks are densely interconnected networks of relatively simple computing elements—for example, threshold or sigmoid neurons. Neural networks' potential for parallelism and their fault and noise tolerance make them an attractive framework for the design of pattern classifiers for real-world, real-time, pattern-classification tasks.

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.

Assume C is a finite set of classes, n a finite number of discrete or real-valued attributes, $\Re$ the set of real numbers, and D a finite set of discrete values. Multilayer networks of nonlinear computing elements (such as threshold neurons) can realize any classification function $\phi: \Re^n \rightarrow C$ or $\phi: D^n \rightarrow C$. If the attributes are symbolic, they must first be mapped to numeric values using appropriate coding schemes. Evolutionary algorithms are generally quite effective for rapid global search of large search spaces in multimodal optimization problems. Neural networks are particularly effective for fine-tuning solutions once promising regions in the search space have been identified.[1] Against this background, genetic algorithms offer an attractive approach to feature subset selection for neural-network pattern classifiers.

However, if we use traditional neural-network training algorithms to train the pattern classifiers, the use of genetic algorithms for subset selection presents some practical problems:

- Traditional neural-network learning algorithms (such as back-propagation) 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 weight setting. For example, if the network has too few neurons, the learning algorithm will miss the desired classification function. If the network has far more neurons than necessary, it can result in overfitting of the training data, which leads to poor gen-

eralization. Either case would make it difficult to evaluate the usefulness of a feature subset describing the training patterns for 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 neural networks' training patterns.
- A typical run of a genetic algorithm involves many generations. In each generation, evaluation of an individual (a feature subset) involves training the neural network and computing its accuracy and cost. This can make the fitness evaluation rather expensive, because gradient-based algorithms are typically quite slow. The problem is exacerbated because we must use multiple neural networks to sample the space of ad hoc network architecture choices to get a reliable fitness estimate for each feature subset represented in the population.

Fortunately, constructive neural-network learning algorithms[2] eliminate the need for ad hoc and often inappropriate a priori choices of network architectures. In addition, such algorithms can potentially discover near-minimal networks whose size is commensurate with the complexity of the classification task implicitly specified by the training data. Several new, provably convergent, and relatively efficient constructive learning algorithms for multicategory real and discrete-valued pattern classification tasks have begun to appear in the literature.[3,4] Many of these have demonstrated very good performance in terms of reduced network size, learning time, and generalization in several experiments with both artificial and fairly large real-world data sets.

DistAl

The results we present in this article are from experiments using neural networks constructed by DistAl,[3] a simple and fast constructive neural-network learning algorithm for pattern classification. DistAl's key feature is to add hidden neurons one at a time, using a greedy strategy that ensures that each 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—that is, when the network correctly classifies the entire training set. At this point, 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.

DistAl is guaranteed to converge to 100% classification accuracy on any finite training set in time that is polynomial in the number of training patterns. Earlier experiments[3] show that DistAl, despite its simplicity, yields classifiers that compare quite favorably with those generated by learning algorithms that are more sophisticated and substantially more demanding computationally. This makes DistAl an attractive choice for experimenting with evolutionary approaches to feature subset selection for neural-network pattern classifiers. Figure 1 shows the key steps in our approach.

DistAl adds hidden neurons one at a time, using a greedy strategy that ensures that each hidden neuron correctly classifies a maximal subset of training patterns belonging to a single class.

Implementation

We ran our experiments using a standard genetic algorithm with a rank-based selection strategy. The probability of selection of the highest ranked individual is p (where $0.5 < p < 1.0$ is a user-specified parameter); that of the second highest ranked individual is $p(1-p)$; that of the third highest ranked individual is $p(1-p)^2$; and that of the last ranked individual is 1 – (sum of the probabilities of selection of all the other individuals).[1] Our results are based on ten random partitions for each classification task with the following parameter settings:

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

We based these parameter settings on the results of several preliminary runs. The probabilities of crossover, mutation, and selection of the highest ranked individual are close to the typical values used in standard genetic algorithms.[1]

Each individual in the population represents a candidate solution to the feature subset selection problem. Let m be the total number of features 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. (Given m such features, there exist 2^m possible feature subsets. Thus, for large values of m, an exhaustive search is not feasible). Each feature subset is represented by a binary vector of dimension m. If a bit is a 1, it means that the corresponding feature is selected. A value of 0 indicates that the corresponding feature is not selected.

We determine an individual's fitness 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 combines two criteria—the accuracy of the classification function realized by the neural network and the cost of performing the classification. We can estimate the classification function's accuracy by calculating the percentage of patterns in a test set that the neural network in question correctly classifies. Several measures of classification cost suggest themselves: the cost of measuring the value of a particular feature needed for classification (the cost of performing the necessary test in a medical diagnosis application), the risk involved, and so on. To keep things simple, we chose this two-criteria fitness function:

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

Here, $fitness(x)$ is the fitness of the feature subset represented by x; $accuracy(x)$ is the test accuracy of the neural-network classifier

trained by DistAl using the feature subset represented by x; $cost(x)$ is the sum of measurement costs of the feature subset represented by x; and $cost_{max}$ is an upper bound on the costs of candidate solutions. In this case, $cost_{max}$ is simply the sum of the costs associated with all of the features. This is clearly a somewhat ad hoc choice. However, it does discourage trivial solutions—such as a zero-cost solution with very low accuracy—from being selected over reasonable solutions that yield high accuracy at a moderate cost. It also ensures that $\forall x\ 0 \leq fitness\ (x) \leq (100 + cost_{max})$.

In practice, we must define suitable tradeoffs between the multiple objectives based on knowledge of the domain. In general, it is a nontrivial task to combine multiple optimization criteria into a single fitness function. The literature on utility theory examines a wide variety of approaches.[5]

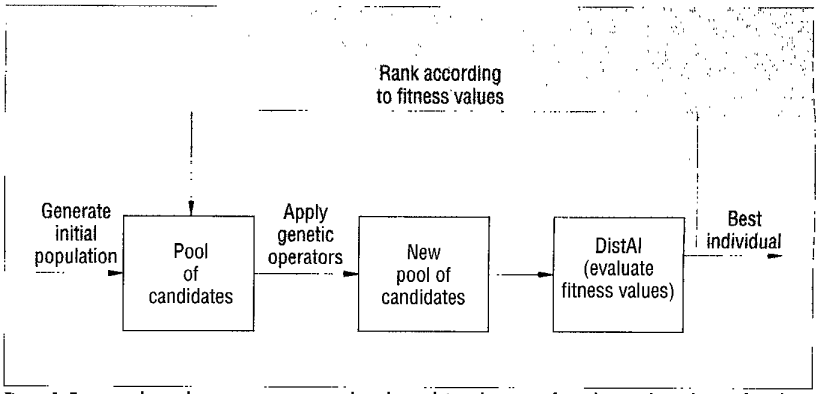

Figure 1. Feature subset selection using a genetic algorithm with DistAl. Starting from the initial population of candidates having different feature subsets, we generate new populations repeatedly from the previous ones by applying genetic operators (crossover and mutation) to the selected parents. DistAl evaluates the fitness values of offspring and ranks them according to their fitness values. The last generation of the process yields the best individual.

Experimental data sets

The experiments we report here used real-world data sets as well as a carefully constructed artificial data set (called 3-bit parity) to explore the feasibility of using genetic algorithms for feature subset selection for neural-network classifiers. We obtained the real-world data sets from the machine-learning data repository at the University of California, Irvine (*http://www.ics.uci.edu/AI/ML/MLDBRepository.html*).

3-bit parity data set. We constructed this data set to explore the genetic algorithm's effectiveness in selecting an appropriate subset of relevant features in the presence of redundant features. If successful, the genetic algorithm would minimize the cost and maximize the accuracy of the resulting neural-network pattern classifier.

To introduce redundancy to the training set, we replicated the original features once, thereby doubling the number of features. Then, we generated an additional set of irrelevant features and assigned them random Boolean values. We generated 100 7-bit random vectors and augmented them with the 6-bit vectors (corresponding to the original three bits plus an identical set of three bits). We assigned each feature in the resulting data set a random cost between 0 and 9.

Real-world data sets. Our objective with real-world data sets was to compare the neural networks built using feature subsets that the genetic algorithm selected with neural networks using the entire set of features available. Table 1 summarizes the data sets' characteristics.

Some medical data sets include measurement costs for the features, but most sets lack this information. Thus, our experiments focused on identifying a minimal subset of features to yield high-accuracy neural-network classifiers for all data sets. Where measurement costs were available, we compared the performance considering the cost in addition to the accuracy (see Equation 1) with that we obtained by considering the accuracy alone.

Experimental results

We partitioned each data set into a training and test set (with 90% of the data used for training and the remaining 10% for testing). We did this partition 10 times and used each partition in five independent runs of the gen-

Table 1. Data sets used in the experiments.

Data set	Description	Size (no. of patterns)	Input features (no.)	Feature type	Output classes (no.)
3P	3-bit parity problem	100	13	Numeric	2
Annealing	Annealing database	798	38	Numeric, nominal	5
Audiology	Audiology database	200	69	Nominal	24
Bridges	Pittsburgh bridges	105	11	Numeric, nominal	6
Cancer	Breast cancer	699	9	Numeric	2
CRX	Credit screening	690	15	Numeric, nominal	2
Flag	Flag database	194	28	Numeric, nominal	8
Glass	Glass identification	214	9	Numeric	6
Heart	Heart disease	270	13	Numeric, nominal	2
HeartCle	Heart disease (Cleveland)	303	13	Numeric, nominal	2
HeartHun	Heart disease (Hungarian)	294	13	Numeric, nominal	2
HeartLB	Heart disease (Long Beach)	200	13	Numeric, nominal	2
HeartSwi	Heart disease (Swiss)	123	13	Numeric, nominal	2
Hepatitis	Hepatitis domain	155	19	Numeric, nominal	2
Horse	Horse colic	300	22	Numeric, nominal	2
Ionosphere	Ionosphere structure	351	34	Numeric	2
Liver	Liver disorders	345	6	Numeric	2
Pima	Pima Indians diabetes	768	8	Numeric	2
Promoters	DNA sequences	106	57	Nominal	2
Sonar	Sonar classification	208	60	Numeric	2
Soybean	Large soybean	307	35	Nominal	19
Votes	House votes	435	16	Nominal	2
Vehicle	Vehicle silhouettes	846	18	Numeric	4
Vowel	Vowel recognition	528	10	Numeric	11
Wine	Wine recognition	178	13	Numeric	3
Zoo	Zoo database	101	16	Numeric, nominal	7

etic algorithm. Tables 2, 3, and 4 show averaged performance. The table entries correspond to means and standard deviations, shown in the form mean ± standard deviation. (See our recent work[6] for more thorough experiments.)

Improving generalization. To study the effect of feature subset selection on generalization, we ran experiments using classification accuracy as the fitness function. The results shown in Table 2 indicate that the networks constructed using a GA-selected subset of features compare quite favorably with networks that use all of the features. In particular, feature subset selection resulted in significant generalization improvement.

Table 3 compares the results of our approach with other GA-based approaches[7] and several non-GA-based approaches cited in our recent work.[6] (These non-GA approaches use a decision-tree algorithm.) We limited the comparisons to only those data sets for which at least one of the two studies[6,7] reported results that could be compared with the results of our experiments. (It is not generally feasible to do a completely fair and thorough comparison between different approaches without complete knowledge of the parameters and setup used in the experiments.) The results indicate that our approach provided higher generalization accuracy in almost all cases, although it occasionally used more features.

Minimizing cost and maximizing accuracy. For this experiment, we based subset selection on both generalization accuracy and the features' measurement cost. (See the fitness function in Equation 1.) We used the 3-bit parity problem, hepatitis, Cleveland heart disease, and the Pima Indians diabetes data sets (with random costs in the 3-bit parity problem.) Table 4 shows the results.

The fitness function that combined both accuracy and cost outperformed that based on accuracy alone in every respect: the number of features used, generalization accuracy, and the number of hidden neurons. This is not surprising, because the former tries to minimize cost while maximizing accuracy; this reduces the number of features. The latter emphasizes only the accuracy. Some of the runs resulted in feature subsets that did not necessarily have minimum cost. This suggests that we can improve the results with a more principled choice of a fitness function combining accuracy and cost.

GENETIC ALGORITHMS OFFER AN attractive approach to solving the feature subset selection problem in inductive learning of pattern classifiers in general and neural-network pattern classifiers in particular. This task finds applications in the cost-sensitive design of classifiers for tasks such as medical diagnosis and computer vision. Other applications of interest include automated data-mining and knowledge discovery from data sets with an abundance of irrelevant or redundant features. In such cases, identifying a relevant subset that adequately captures the regularities in the data can be particularly useful.

THE FITNESS FUNCTION THAT COMBINED BOTH ACCURACY AND COST OUTPERFORMED THAT BASED ON ACCURACY ALONE IN EVERY RESPECT: THE NUMBER OF FEATURES USED, GENERALIZATION ACCURACY, AND THE NUMBER OF HIDDEN NEURONS.

Some directions for further research in this field include

- the application of approaches based on genetic algorithms to feature subset selection for large-scale pattern classification tasks that arise in power systems control,[8] gene sequence recognition, and data-mining and knowledge discovery;
- extensive experimental and theoretical comparison of the performance of our approach with that of conventional methods for feature subset selection;
- more principled design of multiobjective fitness functions for feature subset selection using domain knowledge along with mathematically well-founded tools of multiattribute utility theory.[5]

Some of these research directions are currently being explored. ■

Acknowledgments

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

References

1. M. Mitchell, *An Introduction to Genetic Algorithms,* MIT Press, Cambridge, Mass., 1996.
2. V. Honavar and L. Uhr, "Generative Learning Structures and Processes for Connectionist Networks," *Information Sciences*, Vol. 70, 1993, pp. 75–108.
3. J. Yang, R. Parekh, and V. Honavar, *DistAl: An Inter-Pattern Distance-Based Constructive Learning Algorithm*, Tech. Report ISU-CS-TR 97-05, Iowa State Univ., Ames, Ia., 1997. Also to appear in *Proc. Int'l Joint Conf. Neural Networks,* IEEE, Piscataway, N.J., 1998.
4. R. Parekh, J. Yang, and V. Hanovar, *Constructive Neural Network Learning Algorithms for Multi-Category Real-Valued Pattern Classification*, Tech. Report ISU-CS-TR 97-06, Dept. Computer Science, Iowa State Univ., 1997.
5. R. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs,* John Wiley & Sons, New York, 1976.
6. J. Yang and V. Honavar, "Feature Subset Selection Using a Genetic Algorithm," *Feature Extraction, Construction and Selection—A Data Mining Perspective,* Liu and Motoda, eds., Kluwer Academic Publishers, Boston, forthcoming in 1998.
7. M. Richeldi and P. Lanzi, "Performing Effective Feature Selection by Investigating the Deep Structure of the Data," *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, Calif., 1996, pp. 379–383.
8. G. Zhou, J. McCalley, and V. Honavar, "Power System Security Margin Prediction Using Radial Basis Function Networks," *Proc. 29th Ann. North American Power Symp.*, 1997, pp. 192–198.

Table 2. Neural-network pattern classifiers constructed using the entire set of features compared with those constructed using the most accurate subsets selected by the genetic algorithm. The column labeled accuracy shows the generalization accuracy, and the column labeled hidden shows the number of hidden neurons generated in the neural networks.

Data set	All attributes			GA-selected subset		
	Features (no.)	Accuracy (%)	Hidden	Features (no.)	Accuracy (%)	Hidden
3P	13	79.0 ± 12.2	5.0 ± 2.0	6.6 ± 1.6	100 ± 0.0	9.2 ± 4.9
Annealing	38	96.6 ± 2.0	12.1 ± 2.4	21.0 ± 3.1	99.5 ± 0.9	11.1 ± 2.9
Audiology	69	66.0 ± 9.7	24.7 ± 4.8	36.4 ± 3.5	83.5 ± 8.2	27.4 ± 5.6
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
Flag	28	65.8 ± 9.5	9.1 ± 6.2	14.0 ± 2.6	78.1 ± 7.8	11.2 ± 6.5
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
Soybean	35	81.0 ± 5.6	20.2 ± 3.2	19.4 ± 2.7	92.8 ± 5.9	23.3 ± 4.3
Vehicle	18	65.4 ± 3.5	23.7 ± 5.0	9.1 ± 1.7	68.8 ± 4.3	36.2 ± 18.2
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

Table 3. Comparison between various approaches for feature subset selection. The non-GA column shows the best performance among several approaches not based on genetic algorithms;[6] the Richeldi column shows the performance reported by Richeldi and Lanzi;[7] and the DistAl column shows the performance of our approach.

Data set	Non-GA		Richeldi		DistAl	
	Features	Accuracy	Features	Accuracy	Features	Accuracy
Annealing	-	-	8	95.0 ± 2.3	21.0 ± 3.1	99.5 ± 0.9
Cancer	4	74.7	-	-	5.4 ± 1.4	99.3 ± 0.9
CRX	6	85.0	7	85.1 ± 6.1	8.0 ±2.1	91.5 ± 2.8
Glass	4	62.5	4	70.5 ± 7.8	5.5 ± 1.4	80.8 ± 5.0
Heart	3	79.2	5	80.8 ± 6.5	7.2 ± 1.6	93.9 ±3.8
Hepatitis	4	84.6	-	-	9.2 ± 2.3	97.1 ± 4.3
Horse	4	85.3	-	-	11.1 ± 2.3	92.6 ± 3.4
Pima	-	-	3	73.2 ± 3.8	3.8 ± 1.5	79.5 ± 3.1
Sonar	-	-	16	76.0 ± 9.0	30.7 ± 3.7	97.2 ± 2.9
Vehicle	-	-	7	69.6 ± 6.1	9.1 ± 1.7	68.8 ± 4.3
Votes	4	97.0	5	95.7 ± 3.5	8.9 ± 1.8	98.8 ± 1.2

Table 4. Performance comparison: neural-network pattern classifiers that use features selected based on accuracy alone compared to those that use features selected based on both accuracy and cost.

Data set	Accuracy only			Accuracy and cost			
	Features	Accuracy	Hidden	Features	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

Jihoon Yang is a graduate student working on a PhD in computer science at Iowa State University. His research interests include intelligent agents, data mining and knowledge discovery, machine learning, neural networks, pattern recognition, and evolutionary computing. Yang received a BS in computer science from Sogang University, Seoul, Korea, and an MS in computer science from Iowa State University. Yang is a member of the AAAI and the IEEE. Contact him at the AI Research Group, Dept. of Computer Science, Iowa State Univ., Ames, IA 50011; yang@cs.iastate.edu.

Vasant Honavar is an associate professor in Iowa State University's Artificial Intelligence Research Laboratory, which he founded, and is also on the faculty of the Interdepartment Graduate Program in Neuroscience. His research interests include AI, cognitive science, machine learning, neural networks, intelligent agents and multiagent systems, adaptive systems, and intelligent manufacturing systems. He received a BE in electrical engineering from Bangalore University, Bangalore, India, an MS in electrical and computer engineering from Drexel University, an MS in computer science from the University of Wisconsin, and a PhD in computer science and cognitive science, also from the University of Wisconsin. He edited *Advances in Evolutionary Synthesis of Neural Systems* (MIT Press, 1998). He is a member of the IEEE, ACM, AAAI, Cognitive Science Society, Society for Neuroscience, Neural Network Society, and Sigma Xi, and is an associate of the Behavior and Brain Sciences Society. Contact him at the AI Research Group, Dept. of Computer Science, Iowa State Univ., Ames, IA 50011; honavar@cs.iastate.edu.