1 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
U.S.A.
{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 is due to the fact that the performance of the classifier (usually induced by some learning algorithm) and the cost of classification are sensitive to the choice of the features used to construct the classifier. Exhaustive evaluation of possible feature subsets is usually infeasible in practice because of the large amount of computational effort required. Genetic algorithms, which belong to a class of randomized heuristic search techniques, offer an attractive approach to find near-optimal solutions to such optimization problems. This paper presents an approach to feature subset selection using a genetic algorithm. Some advantages of this approach include the ability to accommodate multiple criteria such as accuracy and cost of classification into the feature selection process and to find feature subsets that perform well for particular choices of the inductive learning algorithm used to construct the pattern classifier. Our experiments with several benchmark real–world pattern classification problems demonstrate the feasibility of this approach to feature subset selection in the automated design of neural networks for pattern classification and knowledge discovery.

1.1 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 features 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.

- The time needed for learning a sufficiently accurate classification function: For a given representation of the classification function, the features used to describe the patterns implicitly determine the search space that needs to be explored by the learning algorithm. 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.

- The number of examples needed for learning a sufficiently accurate classification function: All other things being equal, the larger the number of features 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 [Langley, 1995; Mitchell, 1997].

- 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.

- The comprehensibility of the knowledge acquired through learning: A primary task of an inductive learning algorithm is to extract *knowledge* (e.g., in the form of classification rules) from the training data. Presence of a large number of features, especially if they are irrelevant or misleading, can make the knowledge difficult to comprehend by humans. Conversely, if the learned rules are based on a small number of relevant features, they would much more concise and hence easier to understand, and use by humans.

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 features to be used to represent patterns from a larger set of often mutually redundant, possibly irrelevant, features 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 [Zhou et al., 1997], construction of user interest profiles for text classification [Yang et al., 1998a] and sensor subset selection in the design of autonomous robots [Balakrishnan and Honavar, 1996b].

The rest of the paper is organized as follows: Section 1.2 summarizes various approaches to the feature subset selection. Section 1.3 describes our approach that uses a genetic algorithm for neural network pattern classifiers. Section 1.4 explains the implementation details in our experiments. Section 1.5 presents the results of various experiments designed to evaluate the performance of our approach on some benchmark classification problems as well as a document classification task. Section 1.6 concludes with summary and discussion of some directions for future research.

1.2 RELATED WORK

A number of approaches to feature subset selection have been proposed in the literature. (See [Siedlecki and Sklansky, 1988; Doak, 1992; Langley, 1994; Dash and Liu, 1997] for surveys). These approaches involve searching for an optimal subset of features based on some criteria of interest. Feature subset selection problem can be viewed as a special case of the *feature weighting* problem. 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 [Cost and Salzberg, 1993; Punch et al., 1993; Wettschereck et al., 1995]. If we restrict the weights to be binary valued, the feature weighting problem reduces to the feature subset selection problem. The focus of this paper is on feature subset selection.

Let $\mu(S)$ be a performance measure that is used to evaluate a feature subset S with respect to the criteria of interest (e.g., cost and accuracy of the resulting classifier). Feature subset selection problem is essentially an optimization problem which involves searching the space of possible feature subsets to identify one that is optimal or near-optimal with respect to μ. Feature subset selection algorithms can broadly be classified into three categories according to the characteristics of the search strategy employed.

1.2.1 Feature Subset Selection Using Exhaustive Search

In this approach, the candidate feature subsets are evaluated with respect to the performance measure μ and an *optimal* feature subset is found using exhaustive search. The Focus algorithm [Almuallim and Dietterich, 1994] employs the breadth-first search algorithm to find the minimal combination of features sufficient to construct a hypothesis that is consistent with the training examples. The algorithm proposed by [Sheinvald et al., 1990] uses the *minimum description length* criterion [Rissanen, 1978] to select an optimal feature subset using exhaustive enumeration and evaluation of candidate feature subsets. Exhaustive search is computationally infeasible in practice, except in those rare instances where the total number of features is quite small.

1.2.2 Feature Subset Selection Using Heuristic Search

Since exhaustive search over all possible subsets of a feature set is not computationally feasible in practice, a number of authors have explored the use of *heuristics* for feature subset selection, often in conjunction with branch and bound search, a technique that is well–known in combinatorial optimization [Cormen et al., 1990] and artificial intelligence [Russell and Norvig, 1995]. *Forward selection* and *backward elimination* are the most common sequential branch and bound search algorithms used in feature subset selection [Narendra and Fukunaga, 1977; Devijver, 1982; Foroutan and Sklansky, 1987; Fukunaga, 1990]. Forward selection starts with an empty feature set and adds a feature at a time, at each stage choosing the addition that most increases μ. Backward elimination starts with the entire feature set and at each step drops the feature whose absence least decreases μ. Both forward and backward selection procedures are optimal at each stage, but are unable to anticipate complex *interactions* between features that might affect the performance of the classifier. A related approach, called the *exchange strategy* starts with an initial feature subset (perhaps found by forward selection or backward elimination) and then tries to exchange a feature in the selected subset with one of the features that is outside it. We can often find a feature subset that is guaranteed to be the best for a given size of the feature subset without considering all possible subsets using branch and bound search [Narendra and Fukunaga, 1977] if we assume that μ is monotone. That is, adding features is guaranteed to not decrease μ. It is worth pointing out that in many practical pattern classification 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 [Quinlan, 1993]. Furthermore, feature subset selection techniques that rely on the monotonicity of the performance criterion, although they appear to work reasonably well with linear classifiers, can exhibit poor performance with non-linear classifiers such as neural networks [Ripley, 1996].

The use of systematic search to find a feature subset that is consistent with training data by forward selection using a reliability measure is reported in [Schlimmer, 1993]. Five greedy hillclimbing procedures (with different sequential search methods) for obtaining good generalization with decision tree construction algorithms (ID3 and C4.5) [Quinlan, 1993] were proposed in [Caruana and Freitag, 1994]. In related work, [John et al., 1994] used both forward selection and backward elimination to minimize the cross validation error of decision tree classifiers [Quinlan, 1993]; [Kohavi, 1994; Kohavi and Frasca, 1994] used hillclimbing and best-first search for feature subset selection for decision tree classifiers. Koller et al. [Koller and Sahami, 1996; Koller and Sahami, 1997] used forward selection and backward elimination to select a feature that is subsumed by the remaining features (determined by the *Markov blanket*, the set of features that render the selected feature conditionally independent of the remaining features) for constructing Naive Bayesian [Duda and Hart, 1973; Mitchell, 1997] and decision tree classifiers [Quinlan, 1993]. The Preset algorithm [Modrzejewski, 1993] employs the *rough set theory* [Pawlak, 1991] to select a feature subset by rank ordering the features to generate a minimal decision tree. A class of techniques based for feature subset selection using the probability of error and correlation among features is reported in [Mucciardi and Gose, 1971].

1.2.3 Feature Subset Selection Using Randomized Search

Randomized algorithms [Motwani and Raghavan, 1996] make use of randomized or probabilistic (as opposed to deterministic) steps or sampling processes. Several researchers have explored the use of such algorithms for feature subset selection. The Relief algorithm [Kira and Rendell, 1992] assigns weights to features (based on their estimated effectiveness for classification) using the randomly sampled instances. Features whose weights exceed a a user-determined threshold are selected in designing the classifier. Several extensions of Relief have been introduced to handle noisy or missing features as well as multi-category classification [Kononenko, 1994]. A randomized hillclimbing search for feature subset selection for nearest neighbor classifiers [Cover and Hart, 1967; Diday, 1974; Dasarathy, 1991] was proposed in [Skalak, 1994]. The LVF and LVW algorithms [Liu and Setiono, 1996b; Liu and Setiono, 1996a] are randomized algorithms that generate several random feature subsets and pick the one that has the least number of *unfaithful* patterns in the space defined by the feature subset (LVF) or the one that has the lowest error using a decision tree classifier (LVW) giving preference to smaller feature subsets. (Two patterns are said *unfaithful* if they have the same feature values but different class labels). Several authors have explored the use of randomized population-based heuristic search techniques such as genetic algorithms (GA) for feature subset selection for decision tree and nearest neighbor classifiers [Siedlecki and Sklansky, 1989; Brill et al., 1992; Punch et al., 1993; Richeldi and Lanzi, 1996] or rule induction systems [Vafaie and De Jong, 1993]. A related approach used *lateral feedback* networks [Guo, 1992; Kothari and Agyepong, 1996] to evaluate feature subsets [Guo and Uhrig, 1992]. Feature subset selection techniques that employ genetic algorithms do not require the restrictive monotonicity assumption. They also readily lend themselves to the use of multiple selection criteria (e.g., classification accuracy, feature measurement cost, etc.). This makes them particularly attractive in the design of pattern classifiers in many practical scenarios.

1.2.4 Filter and Wrapper Approaches to Feature Subset Selection

Feature subset selection algorithms can also 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 performed independently 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 that is 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. This is feasible only if the learning algorithm used to train the classifier is relatively fast. Figure 1.1 summarizes the filter and wrapper approaches. The approach to feature subset selection proposed in this paper is an instance of the wrapper approach. It utilizes a genetic algorithm for feature subset selection. Feature subsets are evaluated by computing the generalization accuracy of (and optionally cost of features used in) the neural network classifier constructed using a computationally efficient neural network learning algorithm called DistAl [Yang et al., 1998b].

1.3 FEATURE SELECTION USING A GENETIC ALGORITHM 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 features used to describe the patterns. Genetic algorithms offer a particularly attractive approach for multi-criteria optimization.

Neural networks offer an attractive framework for the design of trainable pattern classifiers for real-world real-time pattern classification tasks on account of their potential for parallelism and

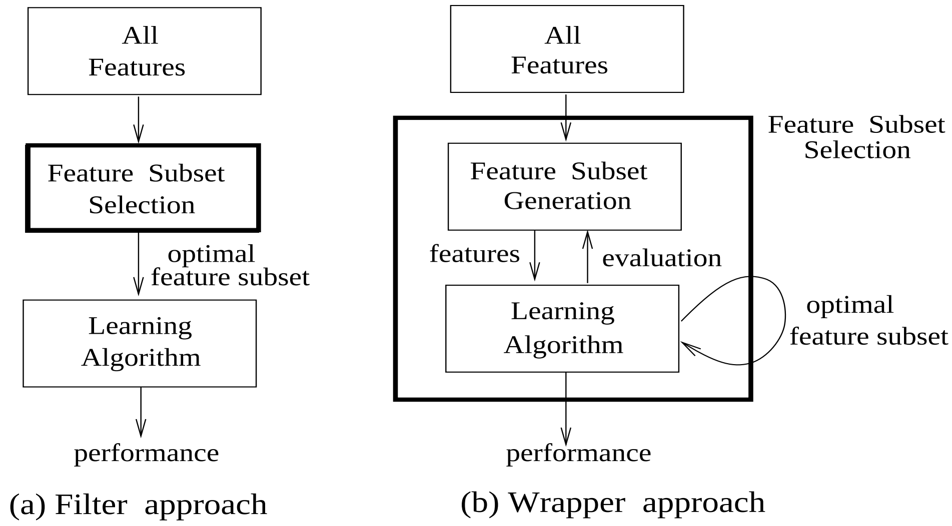

Figure 1.1 Two approaches to feature subset selection based on the incorporation of the learning algorithm. Features are selected independently of the learning algorithm in *filter* approach, while feature subsets are generated and evaluated by a learning algorithm in *wrapper* approach.

fault and noise tolerance, [Gallant, 1993; Honavar, 1994; Hassoun, 1995; Ripley, 1996; Mitchell, 1997; Honavar, 1998a; Honavar, 1998b].

While genetic algorithms are generally quite effective for rapid global search of large search spaces in difficult optimization problems, neural networks offer a particularly attractive approach to finetuning promising solutions once they have been identified. Thus, it is attractive to explore combinations of global and local search techniques in the solution of difficult design or optimization problems [Mitchell, 1996]. Against this background, the use of genetic algorithms for feature subset selection in the design of neural network pattern classifiers is clearly of interest.

This paper explores GADistAI, 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 called DistAI. However, the general approach can be used with any inductive learning algorithm. The interested reader is referred to [Honavar, 1994; Langley, 1995; Mitchell, 1997; Honavar, 1998a; Honavar, 1998b] for surveys of different approaches to inductive learning.

1.3.1 Genetic Algorithms

Evolutionary algorithms [Goldberg, 1989; Holland, 1992; Koza, 1992; Fogel, 1995; Michalewicz, 1996; Mitchell, 1996; Banzaf et al., 1997] include a class related randomized, population-based heuristic search techniques which include genetic algorithms [Goldberg, 1989; Holland, 1992; Mitchell, 1996], genetic programming [Koza, 1992; Banzaf et al., 1997], evolutionary programming [Fogel, 1995], and variety of related approaches [Michalewicz, 1996; Mitchell, 1996]. They are inspired by processes that are modeled after biological evolution. Central to such evolutionary systems is the idea of a population of potential solutions (individuals) that corresponds to members of a high-dimensional search space.

The individuals represent candidate solutions to the optimization problem being solved. A wide range of genetic representations (e.g., bit vectors, LISP programs, matrices, etc.) can be used to encode the individuals depending on the space of solutions that needs to be searched. In genetic algorithms [Goldberg, 1989; Michalewicz, 1996; Mitchell, 1996], the individuals are typically represented by n-bit binary vectors. The resulting search space corresponds to an n-dimensional boolean space. In the feature subset selection problem, each individual would represent a feature subset.

It is assumed that the quality of each candidate solution (or fitness of the individual in the population) can be evaluated using a fitness function. In the feature subset selection problem, the fitness function would evaluate the selected features with respect to some criteria of interest (e.g., cost of the resulting classifier, classification accuracy of the classifier, etc.). In this case, it is essentially the μ function defined earlier.

Evolutionary algorithms use some form of fitness-dependent probabilistic selection of individuals from the current population to produce individuals for the next generation. A variety of selection techniques have been explored in the literature. Some of the most common ones are *fitness-proportionate* selection, *rank-based* selection, and *tournament-based* selection [Goldberg, 1989; Michalewicz, 1996; Mitchell, 1996]. The selected individuals are subjected to the action of genetic operators to obtain new individuals that constitute the next generation. The genetic operators are usually designed to exploit the known properties of the genetic representation, the search space, and the optimization problem to be solved. Genetic operators enable the algorithm to *explore* the space of candidate solutions. See [Balakrishnan and Honavar, 1995] for a discussion of some desirable properties of genetic representations and operators.

Mutation and *crossover* are two of the most commonly used operators that are used with genetic algorithms that represent individuals as binary strings. Mutation operates on a single string and generally changes a bit at random. Thus, a string 11010 may, as a consequence of random mutation, get changed to 11110. Crossover, on the other hand, operates on two parent strings to produce two offspring. With a randomly chosen crossover position 4, the two strings 01101 and 11000 yield the offspring 01100 and 11001 as a result of crossover. Other genetic representations (e.g., matrices, `LISP` programs) require the use of appropriately designed genetic operators [Michalewicz, 1996; Mitchell, 1996; Banzaf et al., 1997].

The process of fitness-dependent selection and application of genetic operators to generate successive generations of individuals is repeated many times until a satisfactory solution is found (or the search fails). It can be shown that evolutionary algorithms of the sort outlined above simulate highly opportunistic and exploitative randomized search that explores high-dimensional search spaces rather effectively under certain conditions [Holland, 1992]. In practice, the performance of evolutionary algorithms depends on a number of factors including: the choice of genetic representation and operators, the fitness function, the details of the fitness-dependent selection procedure, and the various user-determined parameters such as population size, probability of application of different genetic operators, etc. The specific choices made in the experiments reported in this paper are summarized in Section 1.4.

1.3.2 Neural Networks

Neural networks are densely connected, massively parallel, shallowly serial networks of relatively simple computing elements or neurons [Gallant, 1993; Honavar, 1994; Hassoun, 1995; Ripley, 1996; Mitchell, 1997; Honavar, 1998a; Honavar, 1998b]. Each neuron computes a relatively simple function of its inputs and transmits outputs to other neurons to which it is connected via its output links. A variety of neuron functions are used in practice. Each neuron has associated with it a set of parameters which are modifiable through learning. The most commonly used parameters are the so-called *weights*.

The computational capabilities (and hence pattern classification abilities) of a neural network depend on its architecture (connectivity), functions computed by the individual neurons, and the setting of parameters or weights used. 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 non-numeric (e.g., nominal), they have to be first mapped to numeric values using appropriate coding scheme).

Since the function computed by a neural network is determined by its topology as well as the computations performed by individual neurons, designing a neural network for a particular pattern classification task reduces to determination of the network architecture (number of neurons, their connectivity, etc.), the types of neurons (e.g., linear, sigmoid, threshold, etc.), as well as the parameter or weight values. This is typically accomplished through a combination of design (using a-priori knowledge or guesswork) and inductive learning (which may be used to modify, among other things, the weights, network architecture, or both) [Gallant, 1993; Honavar and Uhr, 1993; Honavar, 1994; Parekh et al., 1997a; Honavar, 1998a].

1.3.3 Genetic Algorithm Wrapper approach to Feature Subset Selection for Neural Network Pattern Classifiers: Some Practical Considerations

Genetic algorithms offer an attractive technique for feature subset selection for neural network pattern classifiers for several reasons, some of which were mentioned above. However, we are faced with several difficulties in using this approach in practice.

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 a feature subset employed to represent the training patterns used to train the neural networks. This is due to the fact that the poor performance of the classifier might be due to the failure of the learning algorithm, and not the feature subset used.

Fortunately, constructive neural network learning algorithms [Gallant, 1993; Honavar and Uhr, 1993; Honavar, 1998a] 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 [Yang et al., 1996; Parekh et al., 1997a; Parekh et al., 1997b; Yang et al., 1998b; Honavar, 1998a]. 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 and Uhr, 1993; Parekh et al., 1997a; Yang et al., 1998b]. However, most of them, with the exception of DistAl [Yang et al., 1998b] use time–consuming iterative training algorithms for setting the weights of the neurons.

Using genetic algorithms for feature subset selection for the design of neural network pattern classifiers involves running a genetic algorithm for several generations. In each generation, evaluation of an individual (a feature subset) requires training the corresponding neural network and computing its accuracy and cost. This evaluation has to be performed for each of the individuals in the population. Thus, it is not feasible to use computationally expensive iterative weight update algorithms for training neural network classifiers for evaluating candidate feature subsets. Against this background, DistAl offers an attractive approach to training neural networks.

1.3.4 DistAl*: A Fast Algorithm for Constructing Neural Network Pattern Classifiers*

DistAl [Yang et al., 1998b] is a simple and relatively fast constructive neural network learning algorithm for pattern classification. The results presented in this paper are based experiments using neural networks constructed by DistAl. The key idea behind DistAl is to add *hyperspherical* 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 the pattern set becomes empty (that is, 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

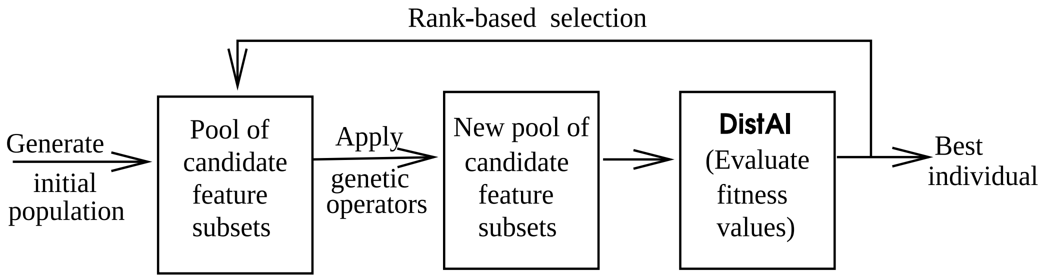

Figure 1.2 GADistAI: Feature subset selection using a genetic algorithm with DistAI. Starting from the initial population (of candidates having different feature subsets), new populations are generated repeatedly from the previous ones by applying genetic operators (i.e., crossover and mutation) to the selected parents, evaluating the fitness values of offsprings by DistAI and ranking them according to their fitness values. The best individual is obtained after the last generation.

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 [Yang et al., 1998b]. Experiments reported in [Yang et al., 1998b] show that DistAI, despite its simplicity, yields classifiers that compare quite favorably with those generated using more sophisticated (and substantially more computationally demanding) learning algorithms. This makes DistAI an attractive choice for experimenting with evolutionary approaches to feature subset selection for neural network pattern classifiers. Key steps in our approach are shown in Figure 1.2.

1.4 IMPLEMENTATION DETAILS

As explained earlier, the use of a genetic algorithm in any search or optimization problem requires:

- the choice of a representation for encoding candidate solutions to be manipulated by the genetic algorithm
- the definition of a fitness function that is used to evaluate the candidate solutions
- the definition of a selection-scheme (e.g., fitness-proportionate selection)
- the definition of suitable genetic operators that are used to transform candidate solutions (and thereby explore the search space)
- setting of user-controlled parameters (e.g., probability of applying a particular genetic operator, size of the population, etc.)

Our experiments were run using a genetic algorithm [Goldberg, 1989; Mitchell, 1996] using 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 $p(1-p)^2$,..., that of the last ranked individual is $1-$(sum of the probabilities of selection of all the other individuals). The rank-based selection strategy gives a non-zero probability of selection of each individual [Mitchell, 1996]. Our experiments used 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

The parameter settings were based on results of several preliminary runs. They are comparable to the typical values mentioned in the literature [Mitchell, 1996].

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. It is represented by a binary vector of dimension m (where m is the total number of features). 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. The fitness of an individual is determined by evaluating the neural network constructed by DistAI 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 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 feature 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 2-criteria fitness function defined as follows:

$$fitness(x) = accuracy(x) - \frac{cost(x)}{accuracy(x) + 1} + cost_{max} \tag{1.1}$$

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 DistAI using the feature subset represented by x, $cost(x)$ is the sum of measurement costs of 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 features. 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 and Raiffa, 1976].

1.5 EXPERIMENTS

1.5.1 Description of Datasets

The experiments reported here used a wide range of real-world datasets from the machine learning data repository at the University of California at Irvine [Murphy and Aha, 1994] 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 feature subset selection using DistAI is also applied to document classification problem for journal paper abstracts and news articles.

3-bit Parity Dataset. This dataset was constructed to explore the effectiveness of the genetic algorithm in selecting an appropriate subset of relevant features in the presence of redundant features so as 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 features are replicated once (to introduce redundancy) thereby doubling the number of features. Then an additional set of irrelevant features 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 feature in the resulting dataset is assigned a random cost between 0 and 9. The performance considering the random costs in addition to the accuracy (see equation (1.1)) was compared with that obtained by considering the accuracy alone.

Datasets from UCI Repository. In our experiments with real world datasets, our objective was to compare the neural networks built using feature subsets selected by the genetic algorithm

Table 1.1 Datasets used in the experiments. *Size* is the number of patterns in the dataset, *Features* is the number of input features, and *Class* is the number of output classes.

Dataset	*Size*	*Features*	*Feature Type*	*Class*
3-bit parity problem (**3P**)	100	13	numeric	2
annealing database (**Annealing**)	798	38	numeric, nominal	5
audiology database (**Audiology**)	200	69	nominal	24
pittsburgh bridges (**Bridges**)	105	11	numeric, nominal	6
breast cancer (**Cancer**)	699	9	numeric	2
credit screening (**CRX**)	690	15	numeric, nominal	2
flag database (**Flag**)	194	28	numeric, nominal	8
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
pima indians diabetes (**Pima**)	768	8	numeric	2
DNA sequences (**Promoters**)	106	57	nominal	2
sonar classifiction (**Sonar**)	208	60	numeric	2
large soybean (**Soybean**)	307	35	nominal	19
vehicle silhouettes (**Vehicle**)	846	18	numeric	4
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
paper abstracts 1 (**Abstract1**)	100	790	numeric	2
paper abstracts 2 (**Abstract2**)	100	790	numeric	2
news articles 1 (**Reuters1**)	939	1568	numeric	6
news articles 2 (**Reuters2**)	139	435	numeric	4
news articles 3 (**Reuters3**)	834	1440	numeric	8

with those that use the entire set of features available. Table 1.1 summarizes the characteristics of the datasets. Some medical datasets include measurement costs for the features, but most of the datasets lack this information. Therefore, our experiments with the datasets from UCI repository focused on identifying a minimal subset of features that yield high accuracy neural network classifiers. Where measurement costs were available, the performance considering the cost in addition to the accuracy was compared with that obtained by considering the accuracy alone.

Document Datasets. The paper abstracts were chosen from three different sources: IEEE Expert magazine, Journal of Artificial Intelligence Research and Neural Computation. The news articles were obtained from Reuters dataset. Each document is represented in the form of a vector of numeric weights for each of the words (terms) in the vocabulary. The weights correspond to the term frequency and inverse document frequency (TFIDF) [Salton and McGill, 1983; Yang et al., 1998a] values for the corresponding words. The training sets for paper abstracts were generated based on the classification of the corresponding documents into two classes (interesting and not interesting) by two different individuals, resulting in two different data sets (**Abstract1** and **Abstract2**). The classifications for news articles were given based on their topics (6, 4 and 8 classes) following [Koller and Sahami, 1997], resulting in three different datasets (**Reuters1**, **Reuters2** and

Reuters3), respectively. These datasets are also summarized in Table 1.1. Since these datasets do not have measurement costs for the features, our experiments with document datasets also focused on identifying a minimal subset of features that yield high accuracy neural network classifiers.

1.5.2 Experimental Results

Two different sets of experiments were run to explore the performance of GADistAI.

The first set of experiments were designed to explore the effect of feature subset selection on the performance of DistAI on a given choice of training and test sets. Each dataset was randomly partitioned into a training and test set (with 90% of the data used for training and the remaining 10% for testing). The genetic algorithm was used to select the best feature subset on the basis of this choice of training and test sets. The results were averaged over 5 independent runs of the genetic algorithm, for a given choice of training and test set. This process was repeated 10 times with 10 different choices of training and test set. The results of these experiments (which represent $5 \times 10 = 50$ runs of the genetic algorithm) are shown in Table 1.2 and 1.5. The entries in the tables give the means (and standard deviations) in the form *mean* ($\pm$ *standard deviation*).

The second set of experiments explored a somewhat different, but related question. Since feature subset selection in GADistAI is guided by the fitness function, it seems reasonable to expect that the quality of fitness estimates will have some impact on the performance of DistAI. Thus, it is interesting to explore the performance of GADistAI when the fitness estimates are obtained using several training and test sets. Thus, in this set of experiments, fitness estimates used by GADistAI were obtained by averaging the observed fitness values for 10 different partitions of the data into training and test sets. The reported results represent averages over 5 independent runs of the algorithm. The results are shown in Table 1.3, 1.4 and 1.6.

Improvement in Generalization using Feature Subset Selection. To study the effect of feature subset selection on generalization, experiments were run using classification accuracy as the fitness function. The results in Table 1.2 indicate that the networks constructed using GA-selected subset of features compare quite favorably with networks that use all of the features in all randomly partitioned datasets. In particular, feature subset selection resulted in substantial improvement in generalization on many of the datasets. (For example, 100% accuracy were yielded in **P3, Promoters**, and **Zoo** datasets). Also, the number of features selected is significantly smaller than the total number of features present in the original data representation in all of the datasets.

The results shown in Table 1.3 indicate that the networks constructed using GA-selected subset of features are comparable to the networks that use all of the features in most of the datasets with 10-fold cross-validation. Clearly, GADistAI outperformed plain DistAI (with all features) in the parity problem in the sense that it successfully selected important features giving 100% generalization. For the remaining datasets, the improvement is generalization ranged from modest in some cases to marginal in others. The best individual generated by GADistAI outperformed DistAI in almost all datasets. Again, the number of features selected is significantly smaller than the total number of features present in the original data representation in all of the datasets.

Table 1.4 compares the results of GADistAI with the results of other GA-based [Richeldi and Lanzi, 1996] and several non GA-based approaches that are available in the literature [Liu and Setiono, 1996a; Liu and Setiono, 1996b; Kohavi, 1994; Kohavi and Frasca, 1994; Koller and Sahami, 1996; Koller and Sahami, 1997]. A '-' indicates that the result is not reported in the corresponding reference. The results indicate that GADistAI gave higher generalization accuracy than the other techniques or comparable accuracy in almost all cases (except **Vehicle** dataset) although it occasionally selected more features. GADistAI produced feature subsets with larger number of features than the approach in [Koller and Sahami, 1996; Koller and Sahami, 1997] for **Reuters** datasets. This can be explained by that the former found the feature subsets using a genetic algorithm for datasets with relatively large number of features while the latter set up the number of features to select a-priori. It should be noted that it is not generally feasible to do a completely fair and thorough comparison between different approaches without the complete knowledge of the parameters and the set up used in the experiments.

Table 1.2 Comparison of neural network pattern classifiers constructed by DistAl using the entire set of features with the best network constructed by GADistAl using GA-selected subsets of features for randomly partitioned datasets. *Features* is the number of features used and *Accuracy* is the generalization accuracy obtained in the neural networks. The reported accuracy of DistAl is obtained by 10-fold cross-validation, and that of GADistAl represents averages over 50 runs of genetic algorithm (10 partitions of the dataset, 5 runs for each partition). See Section 1.5.2. for details.

	DistAl		GADistAl	
Dataset	*Features*	*Accuracy*	*Features*	*Accuracy*
3P	13	79.0±12.2	6.6 ± 1.6	100 ± 0.0
Annealing	38	96.6±2.0	21.0 ± 3.1	99.5 ± 0.9
Audiology	69	66.0±9.7	36.4 ± 3.5	83.5 ± 8.2
Bridges	11	63.0 ± 7.8	5.6 ± 1.5	81.6 ± 7.6
Cancer	9	97.8 ± 1.2	5.4 ± 1.4	99.3 ± 0.9
CRX	15	87.7 ± 3.3	8.0 ± 2.1	91.5 ± 2.8
Flag	28	65.8 ± 9.5	14.0 ±2.6	78.1 ±7.8
Glass	9	70.5 ± 8.5	5.5 ±1.4	80.8 ±5.0
Heart	13	86.7 ± 7.6	7.2 ±1.6	93.9 ±3.8
HeartCle	13	85.3 ± 2.7	7.3 ±1.7	92.9 ±3.6
HeartHun	13	85.9 ± 6.3	7.0 ±1.2	93.0 ±4.0
HeartLB	13	80.0 ± 7.4	7.1 ±1.7	91.0 ±5.7
HeartSwi	13	94.2 ± 3.8	6.6 ±1.7	98.3 ±3.3
Hepatitis	19	84.7 ± 9.5	9.2 ±2.3	97.1 ±4.3
Horse	22	86.0 ± 3.6	11.1 ±2.3	92.6 ±3.4
Ionosphere	34	94.3 ± 5.0	17.3 ±3.5	98.6 ±2.4
Pima	8	76.3 ± 5.1	3.8 ±1.5	79.5 ±3.1
Promoters	57	88.0 ± 7.5	28.8 ±3.3	100 ±0.0
Sonar	60	83.0 ± 7.8	30.7 ±3.7	97.2 ±2.9
Soybean	35	81.0 ± 5.6	19.4 ± 2.7	92.8 ±5.9
Vehicle	18	65.4 ± 3.5	9.1 ±1.7	68.8 ±4.3
Votes	16	96.1 ± 1.5	8.9 ±1.8	98.8 ±1.2
Vowel	10	69.8 ± 6.4	6.5 ±1.2	78.4 ±3.8
Wine	13	97.1 ± 4.0	6.7 ±1.6	99.4 ±2.1
Zoo	16	96.0 ± 4.9	9.3 ±1.6	100 ±0.0
Abstract1	790	89.0±9.4	393.7 ± 12.9	97.6 ± 4.7
Abstract2	790	84.0±12.0	393.8 ± 14.6	94.4 ± 7.3
Reuters1	1568	91.6±2.9	786.1 ± 19.1	94.9 ± 2.5
Reuters2	435	88.5±10.5	218.3 ± 9.7	97.5 ± 4.7
Reuters3	1440	96.4±1.6	715.4 ± 20.3	98.7 ± 1.0

Minimizing Cost and Maximizing Accuracy using Feature Subset Selection. The selection was based on both the generalization accuracy and the measurement cost of features. (See the fitness function in equation (1.1)). The 3-bit parity problem, Cleveland heart disease, hepatitis domain and pima indians diabetes datasets were used for the experiment (with the random costs in the 3-bit parity problem). The results are shown in Table 1.5 and 1.6 for randomly partitioned and 10-fold cross-validation datasets, respectively.

As we can see from Table 1.5, 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 cost. This is not surprising because the former tries to minimize cost (while maximizing the accuracy), which reduces the number of features, while the latter emphasizes only on the accuracy.

Table 1.6 also shows the fitness function that combined both accuracy and cost outperforms that based on accuracy alone in all datasets except **HeartCle**. The generalization accuracy was higher and the cost was also higher with the fitness function that is based on accuracy alone in **HeartCle**

Table 1.3 Comparison of neural network pattern classifiers constructed by DistAl using the entire set of features with the best network constructed by GADistAl using fitness estimates based on 10-fold cross-validation. GADistAl (best) represents the mean (and the standard deviation) of the accuracy of the best network produced by GADistAl using 10-fold crossvalidation among the 5 independent runs of the genetic algorithm. GADistAl (average) represents the mean and the standard deviation (computed over 5 independent runs of the genetic algorithm) of the accuracy of the best network produced by GADistAl. See Section 1.5.2 for details.

	DistAl		GADistAl (average)		GADistAl (best)	
Dataset	*Features*	*Accuracy*	*Features*	*Accuracy*	*Features*	*Accuracy*
3P	13	79.0±12.2	4.8 ± 0.7	100 ± 0.0	4	100 ± 0.0
Annealing	38	96.6±2.0	20.0 ± 1.4	98.8 ± 0.4	18	99.5 ± 1.2
Audiology	69	66.0±9.7	37.2 ± 1.8	72.6 ± 2.8	39	76.5 ± 13.8
Bridges	11	63.0 ± 7.8	4.9 ± 0.6	56.9 ± 7.6	5	67.0 ± 11.9
Cancer	9	97.8 ± 1.2	6.0 ± 1.1	98.0 ± 0.3	8	98.6 ± 0.9
CRX	15	87.7 ± 3.3	7.4 ± 2.6	87.7 ± 0.4	6	88.0 ± 2.8
Flag	28	65.8 ± 9.5	14.2 ± 2.8	63.9 ±6.1	18	70.0 ± 8.8
Glass	9	70.5 ± 8.5	4.4 ± 0.8	69.3 ±2.5	5	71.0 ± 9.4
Heart	13	86.7 ± 7.6	7.6 ± 0.8	85.5 ±0.7	7	85.9 ± 5.4
HeartCle	13	85.3 ± 2.7	8.4 ± 0.8	86.9 ±0.6	9	87.7 ± 4.0
HeartHun	13	85.9 ± 6.3	7.4 ± 1.4	85.4 ±1.3	8	87.2 ± 2.2
HeartLB	13	80.0 ± 7.4	7.6 ± 1.0	79.8 ±1.9	6	83.0 ± 6.0
HeartSwi	13	94.2 ± 3.8	7.4 ± 1.7	95.3 ±1.1	8	96.7 ± 4.1
Hepatitis	19	84.7 ± 9.5	10.2 ± 1.6	85.2 ±2.9	10	88.7 ± 9.5
Horse	22	86.0 ± 3.6	9.6 ± 2.7	83.2 ±1.6	5	85.0 ± 7.0
Ionosphere	34	94.3 ± 5.0	16.6 ± 3.0	94.5 ± 0.8	13	96.0 ± 4.3
Pima	8	76.3 ± 5.1	4.0 ± 1.7	73.1 ±3.1	2	76.8 ± 3.8
Promoters	57	88.0 ± 7.5	30.6 ± 2.1	89.8 ±1.7	31	92.0 ± 7.5
Sonar	60	83.0 ± 7.8	32.2 ± 2.2	84.0 ±1.6	28	85.5 ± 7.6
Soybean	35	81.0 ± 5.6	21.0 ± 1.4	83.1 ± 1.1	19	84.3 ± 7.2
Vehicle	18	65.4 ± 3.5	9.4 ± 2.1	50.1 ± 7.9	11	59.4 ± 4.7
Votes	16	96.1 ± 1.5	8.2 ± 1.5	97.0 ±0.7	7	97.9 ± 1.3
Vowel	10	69.8 ± 6.4	6.8 ± 1.2	70.2 ±1.6	6	71.5 ± 5.7
Wine	13	97.1 ± 4.0	8.2 ± 1.2	96.7 ±0.7	7	97.1 ± 3.9
Zoo	16	96.0 ± 4.9	8.8 ± 1.6	96.8 ±2.0	9	99.0 ± 3.0
Abstract1	790	89.0±9.4	402.2 ± 14.2	89.2 ± 1.0	387	91.0 ± 9.4
Abstract2	790	84.0±12.0	389.8 ± 5.2	84.0 ± 1.1	382	85.0 ± 10.2
Reuters1	1568	91.6±2.9	766.0 ± 12.0	90.2 ± 0.7	750	91.5 ± 0.7
Reuters2	435	88.5±10.5	222.4 ± 14.7	90.3 ± 0.8	195	91.5 ± 10.6
Reuters3	1440	96.4±1.6	721.0 ± 16.6	96.2 ± 0.7	712	96.9 ± 1.6

dataset. This explains how the fitness function (equation (1.1)) works in GADistAl and verifies the rationale behind it. Also, note that 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.

1.6 SUMMARY AND DISCUSSION

An approach to feature subset selection using a genetic algorithm for neural network pattern classifiers is proposed in this paper. A fast inter-pattern distance-based constructive neural network algorithm, DistAl, is employed to evaluate the fitness (in terms of the generalization accuracy) of candidate feature subsets in the genetic algorithm. 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 pattern classifiers in general, and neural network pattern classifiers in particular.

Table 1.4 Comparison between various approaches for feature subset selection. The first column (non-GA) shows the best performance among the several non GA-based approaches cited in Section 1.2 [Liu and Setiono, 1996a; Liu and Setiono, 1996b; Kohavi, 1994; Kohavi and Frasca, 1994; Koller and Sahami, 1996; Koller and Sahami, 1997], the second column (ADHOC) shows the performance reported in [Richeldi and Lanzi, 1996], and the last column (GADistAI) shows the performance of our approach.

	non-GA		ADHOC		GADistAI	
Dataset	*Features*	*Accuracy*	*Features*	*Accuracy*	*Features*	*Accuracy*
Annealing	-	-	8	95.0	18	99.5
Cancer	4	74.7	-	-	8	98.6
CRX	6	85.0	7	85.1	6	88.0
Glass	4	62.5	4	70.5	5	71.0
Heart	3	79.2	5	80.8	7	85.9
Hepatitis	4	84.6	-	-	10	88.7
Horse	4	85.3	-	-	5	85.0
Pima	-	-	3	73.2	2	76.8
Sonar	-	-	16	76.0	28	85.5
Vehicle	-	-	7	69.6	11	59.4
Votes	4	97.0	5	95.7	7	97.9
Reuters1	40	94.1	-	-	750	91.5
Reuters2	40	90.0	-	-	195	91.5
Reuters3	80	98.6	-	-	712	96.9

Table 1.5 Comparison of performance of neural network pattern classifiers constructed by GADistAI that use features selected based on accuracy alone vs. features selected using both accuracy and cost for randomly partitioned datasets.

	Accuracy only			Accuracy & Cost		
Dataset	*Features*	*Accuracy*	*Cost*	*Features*	*Accuracy*	*Cost*
3P	6.6	100	46.1	4.3	100	26.7
HeartCle	7.3	92.9	335.7	6.1	93.0	261.5
Hepatitis	9.2	97.1	22.8	8.3	97.3	19.0
Pima	3.8	79.5	28.5	3.1	79.5	22.8

Table 1.6 Comparison of performance of neural network pattern classifiers constructed by GADistAI that use features selected based on accuracy alone vs. features selected using both accuracy and cost for datasets arranged by 10-fold cross-validation.

	Accuracy only			Accuracy & Cost		
Dataset	*Features*	*Accuracy*	*Cost*	*Features*	*Accuracy*	*Cost*
3P	4.8	100	35.6	3.8	100	25.4
HeartCle	8.4	86.9	390.5	7.2	85.7	317.8
Hepatitis	10.2	85.2	23.4	10.0	85.3	23.2
Pima	4.0	73.1	29.3	4.2	76.1	20.8

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. It also offers a natural approach to feature subset selection by taking into account, the distribution of available data. This is due to the fact that feature selection is driven by estimated fitness values, which if based on multiple partitions of the dataset into training and test data, provide a robust measure of performance of the feature subset. This is not generally the case with many of the greedy stepwise algorithms that select features based on a single partition of the data into training and test sets. Consequently, the feature subsets selected by such algorithms are likely to perform rather poorly on other random partitions of the data into training and test sets.

The approach to feature subset selection is able to naturally incorporate multiple criteria (e.g., accuracy, cost) into the feature selection process. This finds applications in cost-sensitive design of classifiers for tasks such as medical diagnosis, computer vision, among others. Another interesting application is automated data mining and knowledge discovery from datasets 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, particularly in scientific knowledge discovery tasks. Techniques similar to the one discussed in this paper have been successfully used recently to select feature subsets for pattern classification tasks that arise in power system security assessment [Zhou et al., 1997], sensor subsets in the design of behavior and control structures for autonomous mobile robots [Balakrishnan and Honavar, 1996a; Balakrishnan and Honavar, 1996b; Balakrishnan and Honavar, 1996c].

Additional experiments with GADistAI in scientific knowledge discovery tasks in bioinformatics (e.g., discovery of protein structure–function relationships, carcinogenicity prediction, gene sequence identification) are currently in progress. Some directions for future research include: Extension of feature subset selection by incorporating *feature construction* and *genetic programming* [Koza, 1992]; Extensive experimental (and wherever 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 [Keeney and Raiffa, 1976].

Acknowledgments

This research was partially supported by National Science Foundation Grant IRI-9409580 and John Deere Foundation Grant to Vasant Honavar. The authors wish to thank Mehran Sahami for providing **Reuters** document datasets. The authors are grateful to Dr. Pazzani of the Department of Information and Computer Science at the University of California at Irvine for managing the repository of machine learning datasets and making it available to us. An earlier version of this paper appears in IEEE Expert.

References

Almuallim, H. and Dietterich, T. (1994). Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1-2):279–305.

Balakrishnan, K. and Honavar, V. (1995). Properties of genetic representations of neural architectures. In *Proceedings of WCNN'95 July 17-21 Washington D.C.*, volume 1, pages 807–813.

Balakrishnan, K. and Honavar, V. (1996a). Analysis of neurocontrollers designed by simulated evolution. In *Proceedings of the International Conference on Neural Networks*, Washington, D.C.

Balakrishnan, K. and Honavar, V. (1996b). On sensor evolution in robotics. In Koza, Goldberg, Fogel, and Riolo, editors, *Proceedings of the 1996 Genetic Programming Conference – GP-96*, pages 455–460. MIT Press, Cambridge, MA.

Balakrishnan, K. and Honavar, V. (1996c). Some experiments in the evolutionary synthesis of robotic neurocontrollers. In *Proceedings of the World Congress on Neural Networks (WCNN'96)*, pages 1035–1040, San Diego, CA.

Banzaf, W., Nordin, P., Keller, R., and Francone, F. (1997). *Genetic Programming - An Introduction*. Morgan Kaufmann, Palo Alto, CA.

Brill, F., Brown, D., and Martin, W. (1992). Fast genetic selection of features for neural network classifiers. *IEEE Transactions on Neural Networks*, 3(2):324–328.

Caruana, R. and Freitag, D. (1994). Greedy attribute selection. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 28–36, New Brunswick, NJ. Morgan Kaufmann.

Cormen, T., Leiserson, C., and Rivest, R. (1990). *Introduction to Algorithms*. MIT Press, Cambridge, MA.

Cost, S. and Salzberg, S. (1993). A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10(1):57–78.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27.

Dasarathy, B. (1991). *Nearest Neighbor (NN) Norms: NN Pattern Classification Techiniques*. IEEE Computer Society Press, Los Alamitos, CA.

Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, 1(3).

Devijver, P. (1982). *Pattern Recognition: A Statistical Approach*. Prentice Hall.

Diday, E. (1974). Recent progress in distance and similarity measures in pattern recognition. In *Proceedings of the Second International Joint Conference on Pattern Recognition*, pages 534–539.

Doak, J. (1992). An evaluation of feature selection methods and their application to computer security. Technical Report CSE-92-18, Department of Computer Science, University of California, Davis, CA.

Duda, R. and Hart, P. (1973). *Pattern Classification and Scene Analysis*. Wiley, New York.

Fogel, D. (1995). *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ.

Foroutan, I. and Sklansky, J. (1987). Feature selection for automatic classification of non-gaussian data. *IEEE Transactions on Systems, Man and Cybernetics*, 17:187–198.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press, New York.

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

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

Guo, Z. (1992). *Nuclear Power Plant Fault Diagnostics and Thermal Performance Studies Using Neural Networks and Genetic Algorithms*. PhD thesis, University of Tennessee, Knoxville, TN.

Guo, Z. and Uhrig, R. (1992). Using genetic algorithms to select inputs for neural networks. In *Proceedings of COGANN'92*, pages 223–234.

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

Holland, J. (1992). *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA.

Honavar, V. (1994). Toward learning systems that integrate multiple strategies and representations. In Honavar, V. and Uhr, L., editors, *Artificial Intelligence and Neural Networks: Steps Toward Principled Integration*, pages 615–644. Academic Press: New York.

Honavar, V. (1998a). Machine learning: Principles and applications. In Webster, J., editor, *Encyclopedia of Electrical and Electronics Engineering*. Wiley, New York. To appear.

Honavar, V. (1998b). Structural learning. In Webster, J., editor, *Encyclopedia of Electrical and Electronics Engineering*. Wiley, New York. To appear.

Honavar, V. and Uhr, L. (1993). Generative learning structures for generalized connectionist networks. *Information Sciences*, 70(1-2):75–108.

John, G., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 121–129, New Brunswick, NJ. Morgan Kaufmann.

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

Kira, K. and Rendell, L. (1992). A practical approach to feature selection. In *Proceedings of the Ninth International Conference on Machine Learning*, pages 249–256. Morgan Kaufmann.

Kohavi, R. (1994). Feature subset selection as search with probabilistic estimates. In *AAAI Fall Symposium on Relevance*.

Kohavi, R. and Frasca, B. (1994). Useful feature subsets and rough set reducts. In *Third International Workshop on Rough Sets and Soft Computing*.

Koller, D. and Sahami, M. (1996). Toward optimal feature selection. In *Machine Learning: Proceedings of the Thirteenth International Conference*. Morgan Kaufmann.

Koller, D. and Sahami, M. (1997). Hierarchically classifying documents using very few words. In *Proceedings of the International Conference on Machine Learning*, pages 170–178.

Kononenko, I. (1994). Estimating attributes: Analysis and extension of relief. In *Proceedings of European Conference on Mahcine Learning*, pages 171–182.

Kothari, R. and Agyepong, K. (1996). On lateral connections in feed-forward neural networks. In *Proceedings of the International Conference on Neural Networks*, pages 13–18.

Koza, J. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA.

Langley, P. (1994). Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*, pages 1–5, New Orleans, LA. AAAI Press.

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

Liu, H. and 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. and Setiono, R. (1996b). A probabilistic approach to feature selection - a filter solution. In *Proceedings of the Thirteenth International Conference on Machine Learning.* Morgan Kaufmann.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, New York, third edition.

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

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

Modrzejewski, M. (1993). Feature selection using rough sets theory. In *Proceedings of the European Conference on Machine Learning*, pages 213–226. Springer.

Motwani, R. and Raghavan, P. (1996). Randomized algorithms. *ACM Computing Surveys*, 28(1):33–37.

Mucciardi, A. and Gose, E. (1971). A comparison of seven techniques for choosing subsets of pattern recognition. *IEEE Transactions on Computers*, 20:1023–1031.

Murphy, P. and Aha, D. (1994). Repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, CA.

Narendra, P. and Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, 26:917–922.

Parekh, R., Yang, J., and Honavar, V. (1997a). Constructive neural network learning algorithms for multi-category real-valued pattern classification. Technical Report ISU-CS-TR97-06, Department of Computer Science, Iowa State University.

Parekh, R., Yang, J., and Honavar, V. (1997b). MUpstart - a constructive neural network learning algorithm for multi-category pattern classification. In *Proceedings of the IEEE/INNS International Conference on Neural Networks, ICNN'97*, pages 1924–1929.

Pawlak, Z. (1991). *Rough Sets, Theoretical Aspects of Reasoning about Data.* Kluwer Academic.

Punch, W., Goodman, E., Pei, M., Chia-Shun, L., Hovland, P., and Enbody, R. (1993). Further research on feature selection and classification using genetic algorithms. In *Proceedings of the International Conference on Genetic Algorithms*, pages 557–564. Springer.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA.

Richeldi, M. and Lanzi, P. (1996). Performing effective feature selection by investigating the deep structure of the data. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 379–383. AAAI Press.

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

Rissanen, J. (1978). Modelling by shortest data description. *Automatica*, 14:465–471.

Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach.* Prentice Hall, Englewood Cliffs, NJ.

Salton, G. and McGill, M. (1983). *Introduction to Modern Information Retrieval.* McGraw Hill, New York.

Schlimmer, J. (1993). Efficiently inducing determinations: A complete and systematic search algorithm that uses optmal pruning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 284–290, Amherst, MA. Morgan Kaufmann.

Sheinvald, J., Dom, B., and Niblack, W. (1990). A modelling approach to feature selection. In *Proceedings of the Tenth International Conference on Pattern Recognition*, pages 535–539.

Siedlecki, W. and Sklansky, J. (1988). On automatic feature selection. *International Journal of Pattern Recognition*, 2:197–220.

Siedlecki, W. and Sklansky, J. (1989). A note on genetic algorithms for large-scale feature selection. *IEEE Transactions on Computers*, 10:335–347.

Skalak, D. (1994). Prototype and feature selection by sampling and random mutation hill-climbing algorithms. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 293–301, New Brunswick, NJ. Morgan Kaufmann.

Vafaie, H. and De Jong, K. (1993). Robust feature selection algorithms. In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, pages 356–363.

Wettschereck, D., Aha, D., and Mohri, T. (1995). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. Technical Report AIC95-012, Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence, Washington, D.C.

Yang, J., Pai, P., Honavar, V., and Miller, L. (1998a). Mobile intelligent agents for document classification and retrieval: A machine learning approach. In *14th European Meeting on Cybernetics and Systems Research. Symposium on Agent Theory to Agent Implementation*, Vienna, Austria.

Yang, J., Parekh, R., and Honavar, V. (1996). MTiling - a constructive neural network learning algorithm for multi-category pattern classification. In *Proceedings of the World Congress on Neural Networks'96*, pages 182–187, San Diego.

Yang, J., Parekh, R., and Honavar, V. (1998b). DistAl: An inter-pattern distance-based constructive learning algorithm. In *Proceedings of the International Joint Conference on Neural Networks*, Anchorage, Alaska. To appear.

Zhou, G., McCalley, J., and Honavar, V. (1997). Power system security margin prediction using radial basis function networks. In *Proceedings of the 29th Annual North American Power Symposium*, Laramie, Wyoming.

Jihoon Yang is a graduate student of Computer Science at Iowa State University. His current research interests include intelligent agents, data mining and knowledge discovery, machine learning, neural networks, pattern recognition, and evolutionary computing. He holds a B.S. in Computer Science from Sogang University (Seoul, Korea), and an M.S. in Computer Science from Iowa State University, and is currently working towards his Ph.D. in Computer Science at Iowa State University. Jihoon Yang a member of AAAI and IEEE.

Vasant Honavar is Associate Professor of Computer Science and Neuroscience at Iowa State University. His current research interests include artificial intelligence, intelligent agents, machine learning, data mining and knowledge discovery, neural and evolutionary computing, and bioinformatics. Dr. Honavar holds a B.E. in Electronics Engg. from Bangalore University (India), an M.S. in Electrical and Computer Engg. from Drexel University, and M.S. and Ph.D. degrees in Computer Science from the University of Wisconsin-Madison. Dr. Honavar is a member of AAAI, ACM, and IEEE.