Improving Genetic Based Feature Selection by Reducing Data Dimensionality

Extended Abstract

M. Richeldi
CSELT
Centro Studi E Laboratori Telecomunicazioni S.p.A.
Via G. Reiss Romoli 274 - Torino, Italy

P. L. Lanzi[†]
Dipartimento di Elettronica Informatica e Automatica
Politecnico di Milano
Piazza Leonardo da Vinci 5 - Milano, Italy

Corresponding author:
Marco Richeldi
Tel.: +39 11 228 6999 Fax: +39 11 228 6862
E-mail: Marco.Richeldi@cselt.stet.it

1. Introduction

Feature selection plays a central role in the data analysis process since irrelevant features often degrade the performance of algorithms devoted to data characterization, extraction of rules from data, and construction of predictive models, both in speed and in predictive accuracy. Machine learning algorithms decrease their generalization capability when dealing with excess features [4]. Irrelevant and redundant features interfere with useful ones, so that most supervised learning algorithms fail to properly identify those features that are necessary to describe the target concept. Effective feature selection, by enabling generalization algorithms to focus on the best subset of useful features, substantially increases the likelyhood of obtaining simpler, more understandable and predictive models of the data.

Feature selection algorithms in the literature can be categorized into two classes, according to the type of information extracted from the training data and the type of the induction algorithm [2].

Feature selection may be accomplished independently of the performance of the learning algorithm used in the knowledge extraction stage. Optimal feature selection is achieved by maximizing or minimizing a criterion function. Such approach may be referred to as the *absolute* or *filter feature selection model*. Conversely, the effectiveness of the *performance-dependent* or *feedback feature selection model* is directly related to the performance of the concept discovery algorithm, usually in terms of its predictive accuracy.

[†] E-mail: Lanzi@elet.polimi.it. This research was conducted at CSELT. During this work, P. Lanzi was supported by a research grant from CSELT.

This paper presents an effective solution to the feature selection issue, that is based upon the genetic algorithm paradigm.

Genetic algorithms (GAs) are adaptive search techniques, based on the analogy with biology, in which a group of possible solutions evolves via natural selection. In recent years, some researchers addressed the feature selection problem using genetic algorithms [18, 19, 20, 21]. In these works, genetic algorithms are used to explore the space of all possible subsets of feature and obtain the set of features that minimize the error rate of a given learning algorithm. This approach reveals effective but suffers from a major shortcoming. The search space grows exponentially with the number of features. Accordingly, the number of generations needed to ensure convergence to a good solution grows proportionally to the number of features. As a consequence, finding a predictive feature subset may take quite a long time when the genetic search is enforced over the whole set of features.

We propose an alternative approach to GA-based feature selection, in which a preprocessing step, named *Data Reduction* (DR), is applied to inspect the underlying structure of the data and initialize the genetic search algorithm.

The data reduction step is aimed at reducing the dimensionality of the data and is independent of the knowledge discovery algorithm. An iterative, statistical method is applied to explore direct (first-order) and indirect (higher-order) dependencies between data and to partition the set of observed features into a small number of clusters (factors) such that those features in a given cluster can be regarded as measures of the same underlying construct. A genetic algorithm has been designed that explores the factors obtained in the data reduction step and determines the most informative subset of features. The fitness of the genetic algorithms is the performance of the induction algorithm employed in the knowledge extraction step.

Experimental results support the claim that effective data reduction yields an excellent starting point for the genetic search. The genetic search process on the reduced set of features is much faster than the one over the whole set of features, and extracts a feature subset at least equally predictive.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 presents our approach to genetic-based feature selection. Experimental results are summarized in Section 4.

2. Related Work

A rich literature on the feature selection issue is available. Due to lack of space, we only mention some of the most interesting research works and refer the interested reader to [6] and [11] for further key references.

[6] reviews and provide comparative evaluation of several feature selection methods which are suitable to be used with lazy learning algorithms. Lazy learning algorithms, such as k-nn, employ a distance function to generate predictions from stored instances. Conversely, [11] references many studies originated in the statistical community. Most of these works focus on subset selection using linear regression. Branch and bound methods are presented by [10, 11] while feature selection using genetic algorithms are introduced in [18, 19, 20, 21].

[10] presents a comparison between a greedy-like search and a genetic algorithm-based method, that highlights a strong relation between the nature of the data and the behavior of both systems. [9] compares a classic “greedy” hill-climbing algorithm and a genetic algorithm-based method. Results are presented that support the claim that genetic algorithms may be used to

improve the robustness of feature selection without sacrificing too much computational efficiency. [8] embeds a feature construction step in a standard genetic architecture. The goal of the new architecture is finding, through selection and/or construction, an adequate set of features to be used by a tree induction system. which creates new features combining existing features using arithmetic operators. [11] introduces a genetic algorithm that uses a simple nearest-neighbor classifier to evaluate feature sets. A counterpropagation network is trained on the resulting feature subset to fit a predictive model to the data. In order to speed up feature evaluation, a sampling method is introduced in which only a portion of the training set is used on any given evaluation. This methods finds subsets that are as good for counterpropagation than those chosen by an evaluation that uses the entire training set.

3. The Proposed Architecture

Genetic-based feature selection algorithms that have emerged in the literature fit the feedback model, and are usually implemented as follows.

Group of features (the *individuals*) are represented as binary strings. Each binary digit (*gene*) stands for the presence or the absence of a given feature. Standard genetic operators, i.e. crossover and mutation [12], are applied without any modification. The error rate of the induction algorithm used in the concept discovery stage is selected as the GA evaluation function(*fitness*) to be minimized. That is, fitness associated to a feature subset **x** is the estimated predictive accuracy of the induction algorithm that would learn the data characterized by the **x** features only.

This approach has been proven effective at finding good performing subsets of features in some empirical studies. However, the efficiency and, consequently, the effectiveness of genetic algorithms which implement the feedback model declines markedly as the number of features grows. The search of a good subset of features can be regarded as a search in a state space in which each state represents a feature subset. States are associated with the objective function used to evaluate the feature subset. Since the search space size grows exponentially along with the number of features, effective feature selection algorithms will be those which constrain the search as much as possible without discarding rewarding states. Search in the space originated by all the original features is reasonable when the number of features is limited (less than 30/40 features). Conversely, combinatorial explosion does affect the search performance severely.

A possible manner to handle the problem is by identifying the best starting point for the search. That is, by focusing the search on that subset of the observed features which is more likely to contain all the relevant features for the learning task[1]. This approach has not been investigated yet, although [1] suggests that search might start with the feature subset used by a learning algorithm, such as a decision tree.

In our opinion, a feature subset cannot be truly informative and, consequently, good performing on unseen cases, unless it contains at least one feature which contribute to define every dimension underlying the data. On the one hand, if some structure exists in the data, a good predictive model of the data can be obtained solely if any important concept is accounted for by the model. If there exist *n* important concepts that contribute to the target concept, and a feedback model identifies a feature subset with less than *n* features as the best performing, it is very likely

[1] The size of the starting subset ought to be significantly less than the size of the original set of features to gain some computational advantage.

that the subset overfit the data. On the other hand, if data has no structure, every feature can be regarded as reflecting a single concept and the search would start from the entire set of features.

In this paper, we propose a genetic-based approach to feature selection that rests upon the above considerations. In our solution, depicted in Fig. 1, a *data reduction* step precedes the genetic search of relevant features.

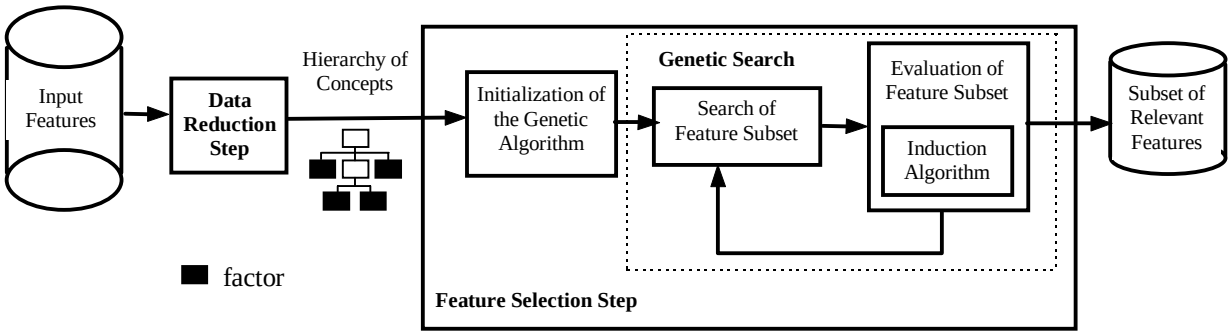

Figure 1. Architecture of the proposed approach to genetic-based feature selection

Briefly, data reduction is a statistical technique used to obtain an insight into the data structure. It yields a clustering of the data features, where each cluster holds features that are mutually related or, in general, that describe the same concept.

Clusters generated in the data reduction step can be used as the starting point for the genetic search of informative features. Indeed, a genetic algorithm may be configured that selects a limited number of features from every cluster. As a consequence, genetic search efficiency is enhanced as the algorithm works through a far smaller space and is driven by the knowledge about the deep structure of the data. Since feature subsets are selected that reflect all the problem dimensions, search effectiveness improves as well.

Effective reduction of data dimensionality is the success factor in our approach. We reviewed statistical approaches to data reduction and found them somewhat inadequate to be used in the feature selection framework. Section 3.1 presents a novel data reduction algorithm that overcomes some of the problems of statistical methods. Section 3.2 shows how genetic search can leverage the knowledge derived from the data reduction step.

3.1. The Data Reduction Step

A number of multivariate statistical techniques can be used to accomplish data reduction, that is, to identify a small number of factors that represent relationships among sets of many interrelated features. Some of them are well known such as factor analysis and cluster analysis.

Factor analysis assumes that statistical correlations among features results from their sharing a set of common factors. The mathematical model for factor analysis appears to be similar to a multiple regression equation. Each feature is expressed as a linear combination of factors that are not known in advance but are determined by factor analysis. Cluster analysis can be used to find homogeneous groups of features rather than of objects. When cluster analysis applies to features, the absolute value of the correlation between features is usually employed to measure their similarity and partition them into a set of groups.

Cluster analysis and factor analysis are well established procedures that are effective in many domains. But the set of mathematical assumptions on which they rely diminishes their applicability in a number of machine learning applications. This is mainly due to the following reasons: these methods (i) fit a linear model of the data, (ii) are suitable to handle numeric features only, (iii) are often fooled by spurious and/or masked correlation.

We designed a novel statistical technique that overcomes some of the problems which degrade the performance of standard approaches to data reduction. This method, described in full detail in [7], is briefly summarized below.

As for cluster and factor analysis, our technique is concerned with reducing the dimensionality of the data and is independent of the knowledge discovery algorithm. An iterative process is applied to explore direct (first order-) and indirect (higher order-) dependencies between the data and to partition the set of observed features into a small number of clusters (factors) such that those features in a given cluster are thought to be measures of the same underlying construct.

Investigating higher-order dependencies is beneficial under two point of views. On the one hand, the method is less likely to be fooled by spurious and masked correlations between the features. When correlations are either imposed artificially or masked by other features, or the data is so noisy that all the correlations assume similar values, the cluster analysis model will fail to fit the data. In this case, indirect relationships between features need to be investigated, since direct associations, which are measured by correlation, do not convey enough information to discover the deep structure of the data. On the other hand, higher-order correlations simplifies the detection of non-linear relationships which underlie the data.

Our method identifies groups of features that are equivalent measures of some factor. It can be regarded, therefore, as a clustering technique. However, the mechanism underlying the formation of clusters is very different from the one employed by cluster analysis of features. The method searches for direct and indirect association among features using the concept of feature *profile*. The profile of a feature F denotes which features F is related to and which ones F is not related to. For example, let A, B, C, D, E, and F be six features that characterize a given data set. Also, let 0.2, 0.1, -0.8, 0.3, and 0.9, be estimates of the direct relationships between F and A, B, C, D, and E, respectively[2]. F's profile is defined as the vector <0.2, 0.1, -0.8, 0.3, 0.9, 1.0>.

Higher-order correlations between features are computed by applying a variation of the Pearson's product moment statistics to feature profiles. Nth-order correlations result in a matrix called the *Nth-order dependency matrix*. By examining the Nth-order dependency matrix, one can determine the strength of relationship between features, and group those features that appear to be related. The recursive process halts when the profile similarity of features in each cluster goes over a predefined threshold or a given number of iterations have been done.

Features may be partitioned into four different categories of clusters. They are called *Positive_Concept*, *Negative_Concept*, *Unrelated_Features*, and *Undefined*, respectively, and reflect the different typology and strength of dependencies that may exist between a set of features. Features that share very similar profiles, i.e., that appear to contribute to the same dimension of the phenomenon, are grouped into a cluster of type Positive_Concept. Features related by a negative association to other features are assigned to a Negative_Concept-type cluster. Features which appear not to influence or to be influenced significantly by the rest of the

[2]The relatioship among real-valued features could be estimated by applying a Pearson's product moment correlation coefficient, for example.

features form the Unrelated_Features cluster. The Undefined cluster contains all the remaining features which can not be assigned to one of the other three types of clusters.

As a result, the process returns a hierarchy of clusters which reflects the hierarchy of concepts that characterize the observed phenomenon. Each level provides a description of the concepts underlying the data from a different level of abstraction. A test of homogeneity of content is then applied to every level of the hierarchy to determine a good factorization of features.

3.2. The Feature Selection Step: The Genetic Search

A genetic-based feature selection algorithm was designed to take advantage of the knowledge extracted by the data reduction method described in the previous section. This section illustrates the architecture of the genetic algorithm, that is, representation of individuals, genetic operators and fitness function.

Representation of individuals

In the framework of genetic-based feature selection, individuals represent feature subsets and populations of individuals represent samples of the feature space to be searched. Our approach to genetic search for relevant features differs from the ones in the literature [18, 19, 20, 21] in that the genetic algorithm is forced to select at most one feature from each factor resulting from the data reduction step. As a consequence, the best performing, least-sized feature subset which covers all the data dimensions are discovered by the genetic algorithm.

The data reduction technique described in the previous section classifies data features into four kinds of factors. These factors may be further dichotomized into two groups, according to our confidence in the capability of the data reduction method to identify true data dimensions. *Positive_Concept* and *Negative_Concept* factors are designated *Neat_Concept* factors, as they represent well-defined, neat concepts in the data. On the converse, *Unrelated* and *Undefined* factors are designated *Blurred_Concept* factors, as they contain features that do not contribute to a precise, unique concept underlying the data. Blurred_Concepts hold features that share weak or no relationship at all. The data reduction method is not able to determine whether or not these features convey interesting information and define a concept in the data.

The genetic algorithm handles features that belong to factors of different types as follows. It can select:

- At most one feature from every Neat_Concept factor. This because features in a Neat_Concept may be regarded as equivalent representations of the same concept.
- Any number of features from Blurred_Concepts, as each feature in a Blurred_Concept may represent an interesting concept by itself.

An representation for individuals shown in Figure 2 has been designed to implement the above feature selection policy.

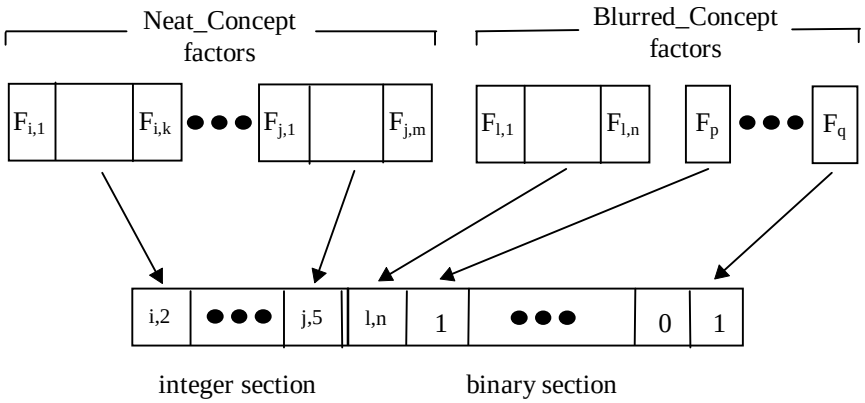

Figure 2. Representation of individuals

The representation has an integer section and a binary section. Every individual is composed by an integer gene for each factor of the Neat_Concept type, and by a binary gene for each feature that belong to a Blurred_Concept factor. The value of an *allele* in an integer gene is either set to the index of the feature that has been selected among the features in the factor associated to the gene, or to zero if no feature is selected. The value of an allele in a binary gene is set according to the presence or absence of the feature associated to the gene.

Mutation Operator

The mutation operator has a different behavior according to whether the selected gene corresponds to a Neat_Concept or to a Blurred_Concept factor. In the former case (integer gene), the zero value or the index of another feature in the factor is selected. In the latter case (binary gene), mutation changes the binary value as in standard mutation operator [13].

Crossover Operator

A standard two point crossover operator is applied to integer genes [13].

Fitness Function

We selected Quinlan's C4.5 [5] as the learning algorithm used to implement the fitness function that the genetic search has to minimize. C4.5 is a well known inducer of classification trees. C4.5 was given the feature subset represented by the individual. The fitness value was computed as the error rate of the classification tree on the test set. The error rate (generalization performance) of C4.5 was estimated by averaging the result of a ten-fold cross-validation over three runs. The pruning confidence level was set to the default 25%.

4. Experiments and Results

We conducted a set of experiments to compare the following three approaches to genetic-based feature selection:

1. The standard genetic implementation proposed in literature [19, 20, 21]. This architecture is characterized as follows: features are represented as binary strings; each binary *gene* stands for the presence or the absence of a given feature; standard crossover and mutation are applied without any modification; fitness to be minimized is the error rate of the induction algorithm used in the concept discovery stage.
2. The implementation of our approach, that has been presented in section 3.
3. An implementation of our approach in which standard factor analysis was used as data reduction technique.

Architectures 1 and 2 were compared to assess the advantages of our approach in terms of running time and predictive performance of the selected feature subset. Architectures 2 and 3 were compared to evaluate the effectiveness of our method of data reduction with respect to standard multivariate statistical data reduction techniques.

The genetic algorithms were implemented using GENESIS [14]. As mentioned above, the generalization performance of C4.5 was chosen as the evaluation function of the genetic algorithm. C4.5's error rate was estimated by averaging the result of a ten-fold cross-validation over three runs.

Factor analysis was run using different factor extraction and rotation methods, then reporting the best performance result in case more alternative factor sets were discovered.

Three real-world datasets were used in the experiments: the Sonar Data Set, the Segment Data Set [6] and the Texture Data Set used in [9]. For each dataset experimentation was conducted as follows.

First, the standard genetic implementation was applied to the data to select the best performing feature subset. Second, factor analysis was applied to the data set. Factors generated by factor analysis cannot be categorized into different types. Thus, genetic search was designed to select at most one feature from each subset of the partition. Finally, the method proposed in Section 3 was applied to the data set.

Table 1 presents comparison results. The column labeled by C4.5 shows the number of features in the original data set and the performance of C4.5 on the original data. Next columns show the predictive accuracy (E.R.- error rate) achieved by the best performing subset selected by each of the three genetic architectures described above. The number of resulting features (n.f.) and the number of fitness evaluations (n.e.) are reported as well.

Dataset	C4.5		Standard GA			Data Reduction: Factor Analysis			Data Reduction: proposed approach		
	E.R	n.f.	E.R.	n.f.	n.e.	E.R.	n.f	n.e.	E.R.	n.f.	n.e.
Segment	4.5	19	4.7	7	1586	6.1	5	319	5.1	12	578
Sonar	27.9	60	23.4	24	1055	27.9	10	1050	23.6	12	656
Texture	24.8	8	23.1	3	114	27.3	2	48	21.7	4	49

Table 1. Results of experiments. E.R. - Error Rate. n.f. - Number of features. n.e. - Number of fitness evaluations to obtain the given solution.

The following observation can be made by analyzing comparison results.

Both the standard genetic search and the proposed approach do select very good features. Indeed, C4.5's generalization performance improves in two out of three experiments (Sonar and

Texture data sets) when run on the reduced set of features. This result supports the claim that effective feature selection benefits the inductive process.

The proposed approach shows the same performance of the standard genetic search, but it needs half the number of fitness evaluations to achieve convergence. As a result, the proposed approach shows a three-fold running time speed-up. This speed-up factor does not diminish when the running time for reducing data dimensionality is taken into account. Indeed, the data reduction method proposed in Section 3 takes some minutes to complete, while genetic search over the Sonar and Segment data sets may take some hours.

The introduction of a data reduction step always improves the running time performance of the genetic search. Indeed, better efficiency is achieved by using both factor analysis and the method proposed in section 3. However, the proposed data reduction method can discover far better informative features than factor analysis. This result supports the claim that effective data reduction is essential to make our approach successful.

For the experiment conducted on the Texture Data Set and for each of the three genetic architectures, Fig. 3 reports the resulting average error rate (fitness value) for every generation until convergence is achieved.

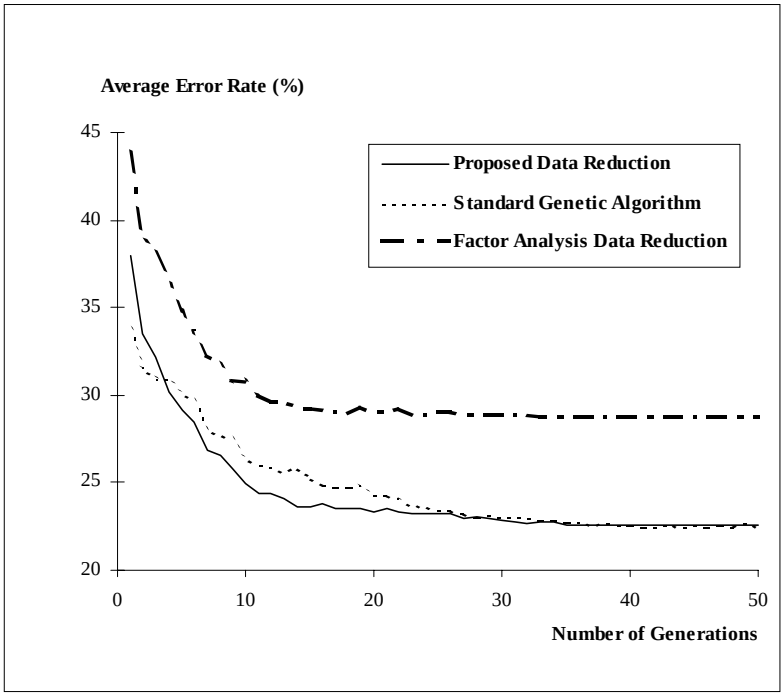

Figura 1

Figure 3. Average error rates for the three search approaches

It is worth noting that when data reduction precedes the genetic search, the number of fitness evaluation that are required to ensure convergence diminishes markedly. However, poor data reduction results in far less predictive features. Indeed, the proposed data reduction technique

substantially outperforms factor analysis and is more efficient than the standard genetic-based feature selection.

5. References

[1] R. Kohavi “Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology,” *First International Conference on Knowledge Discovery and Data Mining*, Morgan Kaufmann, 1995.

[2] D. Wettschereck, T. Mohri, D. W. Aha. “A review and Comparative Evaluation of Feature Weighting Methods for Lazy Learning Algorithms,” *AI Review Journal*, 1995.

[3] G. H. John and R. Kohavi and K. Pfleger., "Irrelevant Features and the Subset Selection Problem," *Proc. of the 11th Int. Conf. on Machine Learning*, 1994.

[4] K. Kira and L. Rendell. "The feature selection problem: Traditional methods and a new algorithm," *Proc. of the Tenth National Conf. on AI*, 129-134, MIT Press, 1992.

[5] J. R. Quinlan. "C4.5 Programs for Machine Learning," Morgan Kaufmann, 1993.

[6] P. M. Murphy and D. W. Aha. UCI repository of machine learning databases. 1994.

[7] M. Richeldi and M. Rossotto. "Combining Statistical Techniques and Search Heuristic to Perform Effective Feature Selection," to appear in "Machine Learning and Statistics: the Interface", Eds. C. Taylor and R. Nakhaeizadeh, Springer Verlag, 1995.

[8] H. Vafaie, K. De Jong “Genetic Algorithms as a Tool for Restrcturing Feature Space Representation” *Proceedings of the International Conference on Tools with Artificial Intelligence* 1995, Herndom, VA.

[9] H. Vafaie, K. De Jong “Robust Feature Selection Algorithms” *International Conference on Tools with AI,* Boston, Massachusetts, 1993.

[10] H. Vafaie, Ibrahim F. Imam “Feature Selection Methods: Genetic Algorithm vs. Greedy-like Search”, *Proceedings of the International Conference on Fuzzy and Intelligent Control Systems, 1994.*

[11] F.Z. Brill III “Genetic Algorithms for Feature Selection” Master Thesis. [10] H. Vafaie, Ibrahim F. Imam “Feature Selection Methods: Genetic Algorithm vs. Greedy-like Search”, *Proceedings of the International Conference on Fuzzy and Intelligent Control Systems, 1994.*

[12] Z. Michalewicz “Genetic Algorithms+Data Structures = Evolution Programs”,Second Edition, Springer Verlag 1994

[13] J.H. Holland “Adaptation in Natural and Artificial Systems”, *University of Michigan Press, Ann Arbor, MI., 1975.*

[14] John J. Grefenstette Technical Report CS-83-11, Computer Science Dept., Vanderbilt Univ., 1984