# Fast Feature Selection with Genetic Algorithms: A Filter Approach

Pier Luca Lanzi
Dipartimento di Elettronica e Informazione
Politecnico di Milano
via Ponzio 34, I-20133 Milano Italia
lanzi@elet.polimi.it

*Abstract*—**The goal of the feature selection process is, given a dataset described by $n$ attributes (features), to find the minimum number $m$ of relevant attributes which describe the data as well as the original set of attributes do. Genetic algorithms have been already used to implement feature selection algorithms. Previous algorithms presented in the literature used the predictive accuracy of a specific learning algorithm as the fitness function to maximize over the space of possible feature subsets. Such an approach to feature selection requires a large amount of CPU time to reach a good solution on large datasets.**

**This paper presents a genetic algorithm for feature selection which improves previous results presented in the literature for genetic-based feature selection and: (i) is independent from a specific learning algorithm; (ii) requires less CPU time to reach a relevant subset of features. Reported experiments shows that proposed algorithm is at least ten times faster than standard genetic algorithm for feature selection without no loss of predictive accuracy when a learning algorithm is applied to reduced data.**

## I. Introduction

The goal of the feature selection process is, given a dataset that describes a target *concept* using $n$ attributes, to find the minimum number $m$ of relevant attributes which describe the concept as well as the original set of attributes do. As an example consider a dataset containing information of customers that applied for a credit to a bank. The *concept*, i.e. the class attribute, is represented by the risk level, *low* or *high*, assigned to each customer by a credit manager of the bank. Attributes represent the customer current credit situation, the past credit history and other general information. Thus the data, corresponding to each customer, can be regarded as examples of how the risk level should be assigned to a customer. Learning algorithms, such as supervised classifiers, can use these examples to learn how the risk rate of a customer, not previously classified by the manager, should be assigned. Feature selection, in this context, is employed to find the minimal set of attributes which can be used to define, represent, the risk level of a customer.

Feature selection plays a central role in the data analysis process since irrelevant features often degrade the performance of algorithms devoted to data characterization, rule extraction and construction of predictive models, both in speed and in predictive accuracy. 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[10]. Effective feature selection, by enabling generalization algorithms to focus on the best subset of useful features, substantially increases the likelihood of obtaining simpler, more understandable and predictive models of the data.

Feature selection algorithms presented in the literature can be classified in two classes according to the type of information extracted from the training data and the type of the induction algorithm [29]. Feature selection can be accomplished independently from the performance of a specific learning algorithm. Optimal feature selection is achieved by maximizing or minimizing a criterion function. Such an approach is referred to as the *filter* feature selection model. Conversely, the effectiveness of the performance-dependent or *wrapper*, feedback, feature selection model is directly related to the performance of the learning algorithm, usually in terms of its predictive accuracy. This paper presents an effective solution to the feature selection issue, based upon the genetic algorithm paradigm, that fits the filter model.

Genetic algorithms (GAs) are adaptive search techniques, based on the analogy with biology, in which a set of possible solutions evolves via natural selection. In recent years, some researchers addressed the feature selection problem using genetic algorithms in a wrapper model approach[26], [3]. In these works, genetic algorithms were used to explore the space of all possible subsets of feature so to obtain a set of features which maximizes the predictive accuracy of a specific learning algorithm. Using this approach the time required for reaching a subset of relevant features strongly depends on the complexity of the learning algorithm used for fitness evaluation. In fact each fitness evaluation requires the application of a learning algorithm to the data reduced to the subset of attributes specified by each individual. Experiments reported in [20] evidences that, using such an approach, genetic feature selection can require some hours of CPU time to obtain a good feature subset on large datasets.

This paper introduces an alternative approach to fitness evaluation for genetic feature selection, based on the filter model which results in a significant reduction in terms of computational time needed to reach a subset of relevant features. A feature subset is now evaluated using the *inconsistency rate*, as defined in [22], which measures to what extent the dimensionally reduced data can be accepted. An high rate means that the features selected do not describe the

0-7803-3949-5/97/$10.00 ©1997 IEEE

data as well as the original set of features does. Conversely a small rate stands for an acceptable reduction on the data. A null rate, for example, indicates that the considered subset is, according to the ratio, as descriptive as the original set of attributes. Inconsistency rate is independent from any learning algorithm, and can be computed rapidly. Consequently genetic feature selection rapidly reaches a good subset of features and thus can be applied to large datasets.

Experimental results support the claim that inconsistency rate is a good heuristic to evaluate the fitness of a subset of features. Resulting genetic search process is an order of magnitude faster than previous implementations proposed and extracts feature subsets at least equally predictive.

The rest of the paper is organized as follows. Section II presents our approach to genetic-based feature selection while experimental results are summarized in section III. Section IV presents an overview of related works presented in the literature.

## II. The Genetic Algorithm

Genetic-based feature selection algorithms that have emerged in the literature are usually implemented as follows. Individuals represent subsets of features by means of binary strings. Each binary digit (gene) stands for the presence (1) or the absence (0) of a given feature. Standard genetic operators, crossover and mutation [16], are applied without any modification. The predictive accuracy of a given learning algorithm is used to measure fitness of individual. That is, fitness associated to a feature subset $x$ is the estimated predictive accuracy of the induction algorithm that would learn the data reduced to the $x$ features only. The genetic algorithm proposed for feature selection purposes maintain the above representation and the standard genetic operators. Instead the fitness of individuals is computed using the inconsistency rate.

The inconsistency rate specifies to what extent the reduced data still represent the original dataset[22] and can be considered a measure of how much inconsistent the data become when only a subset of attributes is considered. Consider now Figure 1 where two items of a dataset with four attributes $\{att_1 \ldots att_4\}$ and one class, *class*, with values $\{c_0, c_1\}$ is shown.

| Items | $att_1$ | $att_2$ | $att_3$ | $att_4$ | *class* |
|---|---|---|---|---|---|
| ⋮ | | | | | ⋮ |
| $item_i$ | 0 | 1 | 2 | 4 | $c_0$ |
| ⋮ | | | | | ⋮ |
| $item_j$ | 0 | 1 | 3 | 4 | $c_1$ |
| ⋮ | | | | | ⋮ |

Fig. 1. Two items, $item_i$ and $item_j$ in a dataset with four attributes, $\{att_1 \ldots att_4\}$ and one class attribute (*class*).

The two items have different class values but differs only in attribute $att_3$ so that if the feature subset $\{att_1, att_2, att_4\}$ is considered we get an inconsistency in the data. In fact looking at Figure 2 where only the subset $\{att_1, att_2, att_4\}$ is considered the two items are equal with respect to attribute values but differs for class attribute values. Thus, from the point of view of a learning algorithm, there is an example that has been classified with two different label: this is inconsistent.

| Items | $att_1$ | $att_2$ | $att_4$ | *class* |
|---|---|---|---|---|
| ⋮ | | | | ⋮ |
| $item_i$ | 0 | 1 | 4 | $c_0$ |
| ⋮ | | | | ⋮ |
| $item_j$ | 0 | 1 | 4 | $c_1$ |
| ⋮ | | | | ⋮ |

Fig. 2. The two items, $item_i$ and $item_j$ in the previous dataset when only the subset of features $\{att_1, att_2, att_4\}$ is considered.

Inconsistency is introduced in the data when the number of attributes is reduced; the rate measures how much inconsistency is introduced when only a certain feature subset is considered. The rate is computed as follows:

1. two items of the given dataset are considered inconsistent if they match except for they class labels with respect to the subset of features considered;
2. for all matching instances the inconsistency count is the number $n$ of instances minus the largest number of instances of the most frequent class label; for example if there are two class label $c_1$ and $c_2$ with respectively $n_1$ and $n_2$ instances $(n_1 + n_2 = n)$ then the inconsistency count is equal to $(n - max(n_1, n_2))$.
3. the inconsistency rate is computed as the quotient of the sum of all the inconsistency counts divided by the total number of instances.

Inconsistency rate can be computed more rapidly than the performance of any learning algorithm. The rate, in fact, does not require any memory allocation as, for example, tree induction classifiers require. Moreover to calculate the performance of a learning algorithm on a certain dataset usually cross-validation is employed to avoid over-fitting [7]. Instead inconsistency rate is a simple statistics and does not require cross-validation.

Nevertheless the rate is only an approximated measure of the information loss when a subset of features is considered thus may be non-informative in some cases. Our experiments on real world data evidence that the criterion tends to be non informative when datasets contain continuous attributes with many values. This phenomenon is avoided if data are discretized before the genetic algorithm is applied. In such a case the choice of the discretization algorithm to employ is a main issue.

Experiments, reported in the next section, show that the algorithm is at least ten times faster than standard implementation of genetic based feature selection. Moreover inconsistency rate successfully selects subset of feature at least as informative as the original set of attributes.

## III. Experimental Results

Proposed algorithm was applied to real world datasets to: (i) test the effectiveness of inconsistency rate as a criterion to evaluate fitness of feature subsets and (ii) to evaluate the performance of the algorithm in terms of CPU time needed to reach a good solution. Experiments were conducted as follows.

First a group of datasets was selected from the UCI repository[21]. Discretization was applied to continuous attributes using the algorithm proposed in [18]. The genetic algorithm, was then applied to each dataset using the inconsistency criterion as fitness function to minimize. Best subsets of features selected by the genetic algorithm and the original set of features were compared running the $C4.5$ tree induction algorithm [25] on both feature sets. To avoid over-fitting five iterations of a two-fold cross-validation were applied and the classification accuracy was measured on the test sets. Predictive accuracies on the original data and on the reduced data were averaged and finally compared using a paired two tail T-test[7]. The algorithm has been developed using the GNU C++ compiler v2.7.0 and is based upon the GENESIS genetic algorithm[9]. Parameters for the GA were set using the default values given in GENESIS.

Table I shows the predictive accuracy (*p.a.*) of $C4.5$ on all the features (*Raw Data*) and on the reduced data obtained with the proposed algorithm (*Red. Data*). The table also reports the number of feature (*n.f.*) in the original data and in the reduced data.

| Dataset | Raw Data | | Red. Data | | p-value |
|---|---|---|---|---|---|
| | p.a. | n.f. | p.a. | n.f. | |
| Australian | 84.0 | 14 | 85.2 | 7 | 0.03 |
| CRX | 85.0 | 15 | 85.4 | 7 | 0.65 |
| Diabetes | 71.2 | 8 | 72.3 | 4 | 0.19 |
| German | 69.4 | 20 | 72.3 | 11 | 0.05 |
| Glass | 64.8 | 9 | 65.3 | 5 | 0.82 |
| Heart | 74.1 | 13 | 73.3 | 6 | 0.59 |
| Segment | 94.8 | 19 | 94.7 | 8 | 0.69 |
| Vehicle | 69.5 | 18 | 68.3 | 10 | 0.20 |
| Vote | 94.0 | 16 | 94.3 | 8 | 0.31 |

TABLE I

Performance of C4.5 induction algorithm on raw data (Raw Data) and reduced data (Red. Data). p.a. indicates the predictive accuracy of C4.5. n.f. indicates the number of features. p-values were calculated using a paired two tailed T test.

Comparison of the predictive accuracy evidences that the subsets of features extracted by the genetic algorithm using the inconsistency rate are at least as descriptive as original set of features but contains, on the average, only half the starting features. Specifically for the *Australian* and *German* datasets reduction of the feature set has significantly improved the predictive accuracy of the induction algorithm. A further comparison of the results in Table I with the results presented in [19] for a more complex algorithm evidence that the proposed approximated approach reach feature subsets that:

- contain almost the same number of features for each dataset as the previously proposed algorithm;
- have the same predictive accuracy as [19] except for the *glass* and *heart* datasets on which previous algorithm performs better: 70.5% for the *glass* dataset and 80.8% for the *heart* dataset.

Experiments have evidenced three main facts. The inconsistency rate can find subset of features that are at least as predictive as the original set of features. Second, the inconsistency rate is an approximated measure and thus in certain cases can perform worse than other algorithms that are based on almost exact, but more complex, measures. Last but most important, inconsistency rate significantly speeds up feature selection process with no decrease in predictive accuracy with respect to original data. A not optimized version of the algorithm requires at least ten times less the CPU time required by previous genetic implementations [20]. For example standard genetic feature selection applied to the *Segment* dataset requires some hours of CPU time while proposed algorithm terminates within fifteen minutes.

## IV. Related Works

This section presents a review of the most interesting research works presented in the literature on the general feature selection issue and for the specific genetic-based implementations.

[29] reviews and provide comparative evaluation of several feature selection methods which are suitable to be used with lazy learning algorithms. Conversely, [12] 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 [8], [14]. Feedback models can be categorized into two groups: those that repeatedly pass through the training set and those that process each training case exactly once. The first group includes methods which are based on genetic algorithms [28], hill-climbing search [12] and best-first search [11]. [11] introduces compound operators in the best-first search process to change the topology of the search space and make use of information available from the evaluation of feature subsets. [17] presents a schemata search which speeds up the search by using backward and forward hill climbing techniques. [5] employs greedy hill-climbing procedures which generalize well with $C4.5$. A caching scheme is introduced that makes attribute hill-climbing more efficient computationally. The second group of feature selection models include algorithms that have been designed to support the instance-based learning paradigm [2] such as those presented in [24], [2], [1].

Interesting algorithms which implement the filter model are those presented in [6], [10]. [6] exhaustively explores all feature subsets, selecting the minimal feature subset which is sufficient to determine the class label and it was originally defined for noise-free boolean domains. The method presented in [10] is a randomized algorithm which assigns

a relevance score to each feature according to its relevance to the target concept.

As the inconsistency rate defined in [22], other authors have recently proposed some approximate measures to evaluate a feature subset without applying a precise learning algorithms[23], [13]. [23] focus on feature selection for classification problems. Feature subset are evaluated using *contingency table analysis*. Effectiveness of the proposed criterion is tested only on artificial datasets using $C4.5$. [13] introduces an optimal feature selection criterion and gives an efficient algorithm to compute an approximation of the optimal criterion.

Genetic algorithms have been already used for feature selection using different learning algorithms to evaluate the fitness of subsets of attributes. In [26], [27] GAs are compared to other greedy search algorithms for feature selection. 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. [28] embeds a feature construction step in a standard genetic architecture using genetic programming paradigm[15]. The goal of the new architecture is finding, through selection and/or construction, an adequate set of features to be used by the $C4.5$ tree induction system. [3] introduces a genetic algorithm that uses a simple nearest- neighbor classifier to evaluate feature sets. A counter propagation 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, for counter propagation, as good as those chosen by an evaluation that uses the entire training set. In [20] a preprocessing step on data is introduced to reduce the search space for standard genetic-based feature selection algorithm.

## V. Conclusions

This paper presents an efficient solution to feature selection problem based on genetic algorithm paradigm that fits the feedback model. The proposed algorithm uses an inconsistency rate to evaluate the fitness of individuals in the population independently from a learning algorithm and rapidly. Experimental results shows that inconsistency rate speeds up the feature selection process without any decrease of predictive accuracy in the reduced data when a learning algorithm is applied to the selected subsets of features.

## Acknowledgments

I would like to thank Marco Richeldi who introduced me to data mining.

## References

[1] Aha D. W.: A study of instance-based learning algorithms for supervised learning tasks: Mathematical, empirical and psychological evaluations. Tech. Report 90-42. Irvine. CA

[2] Aha D. W.: Tolerating Noisy, Irrelevant, and Novel Attributes in Instance-Based Learning Algorithms International Journal of Man-Machine Studies. 36, 267–287. (1992)

[3] Brill III F. Z., Genetic Algorithms for Feature Selection. Master Thesis.

[4] Dietterich T. G.: Statistical Tests for Comparing Supervised Classification Learning Algorithms. Tech. Report. Department of Computer Science. Oregon State University. (1996)

[5] Caruana R., Freitag D.: Greedy Attribute Selection. Proc. of the 11th International Conference on Machine Learning. (1994)

[6] Almuallin H., Dietterich T. G.: Learning with many irrelevant features. Proc. of the 9th International Conference on Machine Learning. (1994)

[7] Dietterich T. G.: Statistical Tests for Comparing Supervised Classification Learning Algorithms. Tech. Report. Department of Computer Science. Oregon State University. (1996)

[8] Narendra L., Fukunaga K.: A branch and bound algorithm for feature subset selection. IEEE Transaction on Computers. C-26(9). (1977) Springer Verlag. (1988)

[9] Grefenstette J. J.: Technical Report CS-83-11 Computer Science Dept., Vanderbilt Univ.,

[10] Kira K., Rendell L.: The feature selection problem: Traditional methods and a new algorithm. Proc. of the Tenth Conference on National Conference on Artificial Intelligence. MIT Press. (1992)

[11] Kohavi R.: Feature Subset Selection Using the Wrapper Method: Over-fitting and Dynamic Search Space Topology. Proc. of the First Conference on Knowledge Discovery and Data Mining. Morgan Kaufmann. (1995)

[12] John G. H., Kohavi R. and Pfleger K.: Irrelevant Features and the Subset Selection Problem. Proc. of the 11th International Conference on Machine Learning. (1994)

[13] Koller D., Sahami M.: Toward Optimal Feature Selection Proc. of the 13th International Conference on Machine Learning. Bari, Italy. (1996)

[14] Kanal L., Kumar V. Search in Artificial Intelligence. Springer Verlag. (1988)

[15] Koza J. R.: Genetic Programming Cambridge (MA). MIT Press. (1992)

[16] Michalewicz Z.: Genetic Algorithms+Data Structures = Evolutionary Programs. Second Edition. Springer Verlag. (1994)

[17] Moore W. A., Lee M. S.: Efficient algorithms for minimizing cross validation error. Proc. of the 11th International Conference on Machine Learning. (1994)

[18] Richeldi M., Rossotto M.: Supervised Quantization of Continuous Predictor Variables. eminars on New Techniques and Technology for Statistics. Bonn 20-22 November. (1995)

[19] Richeldi M., Lanzi P. L.: Performing Effective Feature Selection by Investigating the Deep Structure of the Data Proc. of the Second Conference on Knowledge Discovery and Data Mining. Portland (OR). (1996)

[20] Richeldi M., Lanzi P. L.: Improving Genetic Based Feature Selection by Reducing Data Dimensionality Proc. of the ICML Workshop on Evolutionary Computation. Bari (1996)

[21] Murphy P. M., Aha D. W.: UCI Repository of Machine Learning Databases.

[22] Liu H., Setiono R.: A Probabilistic Approach to Feature Selection: A Filter Solution Proc. of the 13th International Conference on Machine Learning. Bari, Italy. (1996)

[23] Lu H., Sung S. Y. and Lu Y.: On Preprocessing Data for Effective Classification. Workshop on Research Issue on Data Mining and Knowledge Discovery in Databases. (1996)

[24] A nearest hyper-rectangle learning method. Machine Learning (1) 81-106. (1991)

[25] Quinlan J. R.: C4.5 Programs for Machine Learning. Morgan Kaufmann

[26] Vafaie H., De Jong K. Robust Feature Selection Algorithms. Proc. of the International Conference on Tools with Artificial Intelligence. Boston (MA). (1993)

[27] Vafaie H., Imam F. I.: Feature Selection Methods: Genetic Algorithms vs. Greedy-like Search. Proc. of the International Conference on Fuzzy and Intelligent Control Systems. (1994)

[28] Vafaie H., De Jong K. Genetic Algorithms as a Tool for Restructuring Feature Space Representation. Proc. of the International Conference on Tools with Artificial Intelligence. Herndom (VA). (1995)

[29] Wettschereck D., Mohri T. and Aha D. W.: A review and Comparative Evaluation of Feature Weighting Methods for Lazy Learning Algorithms" AI Review Journal. (1995)