Feature Selection via Genetic Optimization

Sancho Salcedo-Sanz, Mario Prado-Cumplido, Fernando Pérez-Cruz, and Carlos Bousoño-Calzón *

Dpto. Teoría de la señal y Comunicaciones, Universidad Carlos III de Madrid, 28911 Leganés, Madrid, Spain.
{sancho, mprado, fernando, cbousono}@tsc.uc3m.es

Abstract. In this paper we present a novel Genetic Algorithm (GA) for feature selection in machine learning problems. We introduce a novel genetic operator which fixes the number of selected features. This operator, we will refer to it as *m-features* operator, reduces the size of the search space and improves the GA performance and convergence. Simulations on synthetic and real problems have shown very good performance of the *m-features* operator, improving the performance of other existing approaches over the feature selection problem.

1 Introduction

Feature selection is an important task in supervised classification problems due to irrelevant features, used as part of a classification procedure, can increase the cost and running time of the system, and make poorer its generalization performance [1].

There are two different approaches to the Feature Selection Problem (FSP). The first method tries to identifying an appropiate set of features, independently of its classification performance, which preserve most of the information provided by the original data. This approach is known as filter method for feature selection [1]. The second approach directly selects a subset of m features out of the n available ones in such a way that the performance of the classifier is improved or, at least, is not degraded. This method, known as wrapper method, is more powerful than filter methods, but it is also computationally more demanding [2], [3]. The search of the best feature subset can be performed by means of any search algorithm like hill-climbing, greedy or genetic algorithms, etc.

In this paper we present a novel Genetic Algorithm (GA) as wrapper method. We have introduced a novel operator which is able to fix the number of selected features, reducing this way the size of the search space and improving the GA performance on the FSP. We show that our approach has very good performance, achieving the best solution in problems with synthetic data and improving the results of the KDD Cup 2001 winner [4].

The article is structured as follows: Section 2 describes the FSP, Section 3 briefly reviews GAs and its application to feature selection, and propose the

* This work has been partially supported by a CICYT grant number: TIC-1999-0216.

m-features operator. In Section 4, we show, by means of computer experiments, the validity of the proposed approach for both synthetic and real data. Finally, Section 5 ends the paper with some concluding remarks.

2 Feature Selection Problem

The Feature Selection Problem (FSP) for a learning problem from samples can be addressed in the following way: Given a set of labeled data points $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{\pm 1\}$, choose a subset of m features ($m < n$), that achieves the lowest classification error [2].

Following [2], we will define the FSP as finding the optimum n-column vector $\boldsymbol{\sigma}$, where $\sigma_i \in \{1, 0\}$, that defines the subset of selected features, which is found as:

$$\boldsymbol{\sigma}^o = arg \min_{\boldsymbol{\sigma}, \alpha} \left(\int V(y, f(\mathbf{x} * \sigma, \alpha)) dP(\mathbf{x}, y) \right) \tag{1}$$

where $V(\cdot, \cdot)$ is a loss functional, $P(\mathbf{x}, y)$ is the unknown probability function the data was sampled from and we have defined $\mathbf{x} * \sigma = (x_1\sigma_1, \ldots, x_n\sigma_n)$. The function $y = f(\mathbf{x}, \alpha)$ is the classification engine that is evaluated for each subset selection, $\boldsymbol{\sigma}$, and for each set of its hyper-parameters, α. This functional can be solved using the ERM principle [5].

3 Genetic Algorithms for Feature Selection

Genetic Algorithms (GAs) have been previously used in feature selection in both filter methods [6] and wrapper methods [7]. However, up until now, its use in wrapper methods has been restricted to small size problems, due to the high computational cost of the resulting algorithm. In this paper we propose a novel GA for the FSP, which has a better performance and a faster convergence than other GA approaches to the problem. In the next section a brief summary of the standard GA is given, and then we present the improvements introduced in the GA for the FSP.

3.1 Background

Genetic Algorithms (GAs) are a class of robust problem solving techniques based on a population of solutions which evolves through sucessive generations by means of the application of the genetic operators: selection, crossover and mutation [8].

Selection is the process by which individuals in the population are randomly sampled with probabilities proportional to their fitness values. An elitist strategy, consisting in passing the highest fitness string to the next generation, is applied in order to preserve the best solution encontered so far in the evolution. The selected set, of the same size of the initial population, is subjected to the crossover operation. Firstly, the binary strings are coupled at random. Secondly, for each pair of strings, an integer position along the string is selected uniformly at random. Two new strings are composed by swapping all bits between the selected position and the end of the string. This operation is applied to the couples with probability P_c less than one. By means of the mutation operation, every bit in every string of the population may be changed from 1 to 0, or vice versa, with a very small probability, P_m.

3.2 A Genetic Algorithm for the FSP

The population of the GA for the FSP is formed by k binary vectors $\boldsymbol{\sigma}$, which evolves by the iterative procedure of the genetic operators described above.

In order to adapt the GA to the FSP, we introduce a novel genetic operator which fixes the number of features selected by the GA to m features, i.e fixes the number of 1s in the individuals to a given number m. We have named it *m-features* operator. The *m-features* operator works in the following way: after the crossover and mutation operators, the binary string will present p 1s, in general different from the desired number of features m. If $p < m$ the *m-features* operator adds $(m-p)$ 1s randomly; if $p > m$, the *m-features* operator randomly selects $(p-m)$ 1s and removes them from the binary string.

The *m-features* operator forces the GA to search in a reduced search space. In fact, the standard GA searchs for vector $\boldsymbol{\sigma}^{\circ}$ in a space of size 2^n, whereas introducing the *m-features* operator, the size of the search space is reduced to $\binom{n}{m} \ll 2^n$. The application of the *m-features* operator will also obtain as a consequence a much faster convergence of the genetic algorithm.

The GA can be used with the *m-features* operator (we call it *best m-features* mode) or without it (we say then that the GA is running in *standard* mode). The GA running in *standard* mode will search for the best set of features in the whole search space, whereas the GA running in *best m-features* mode searchs for the best set of m features in the search space. Note that the *m-features* operator is an adaptation of the GA to the FSP, due to the number of 1s in the individuals is the number of selected features.

The fitness function associated the each individual is the classification error obtained classifying l training points $(\mathbf{x} * \sigma, y)$. Due to GAs maximizes the fitness function, and the objetive function in the FSP is minimizing the error probability, a modified fitness function is introduced:

$$F = 100(1 - err) \tag{2}$$

where err is the probability of error given by the classifier.

4 Experiments

In order to show the performance of our algorithm, we solve two classification problems: the first one is taken from [2], and is formed by two artificial datasets, where we can test the ability of our algorithm to select a small number of features in presence of many irrelevant and redundant ones. The second is a real application in molecular bioactivity for drug design, also known as *Thrombin Binding Problem* used for the first time in the KDD (Knowledge Discovery and Data Mining) Cup in 2001 [4].

4.1 Synthetic Data

Linear Problem Six dimensions out of 202 were relevant. The labels are equiprobable. The first three features $\{x_1, x_2, x_3\}$ were drawn form $x_i = yN(i, 1)$ and the second three features $\{x_4, x_5, x_6\}$ were drawn as $x_i = N(0, 1)$ with probability of 0.7, otherwise the first three were drawn as $x_i = N(0, 1)$ and the second three as $x_i = yN(i - 3, 1)$. The remaining features were drawn for both class from $x_i = N(0, 20)$. In this problem four of the first six features are redundant and the rest are irrelevant, being 2 the optimal number of features.
Nonlinear Problem Two dimensions out of 52 were relevant. The labels are equiprobable. The data are drawn from the following: if $y = -1$ then $\{x_1, x_2\}$ are drawn from $N(\mu_1, \Sigma)$ or $N(\mu_2, \Sigma)$ with equal probability, $\mu_1 = \{-\frac{3}{4}, -3\}$ and $\mu_2 = \{\frac{3}{4}, 3\}$ and $\Sigma = I$. If $y = 1$ then $\{x_1, x_2\}$ are drawn again from two normal distributions with equal probability, with $\mu_1 = \{3, -3\}$ and $\mu_2 = \{-3, 3\}$ and the same Σ as before. The rest of the features are noise $x_i = N(0, 20)$.

We use the GA in *best 2-features* mode in both problems. Standard values for P_c and P_m have been used; the population size has been fixed to $k = 50$ binary strings, $P_c = 0.6$ and $P_m = 0.01$. The classifier used is a SVM machine with a linear kernel for the linear problem and a gaussian kernel for the nonlinear problem. Figure 1 shows the evolution of the GA in both problems for a training set of 50 samples, taking the average test error on 500 samples over 30 independent training sets. The GA with the *m-features* operator needs, in average, 24 and 19 generations for achieving the optimal solution in the linear and nonlinear problems respectively (see Figure 1).

4.2 Molecular bioactivity for drug desing

The first step in the discovery of a new drug is to identify and isolate the receptor to which it should bind, followed by testing many molecules for their ability to bind to the target site. Thus, it is needed determining what separates the active (binding) compounds from the inactive (non-binding) ones. In this case the problem consists in predicting if a given drug binds to a target site on thrombin, a key receptor in blood. This task can be seen in a machine learning context as a FSP. Further information of the problem can be found in [4].

The data used in this paper are taken from the ones used in the KDD Cup [4]. Each example (observation) has a length of 139351 binary features, but, following the winner criteria explained in [4], we selected 100 features using mutual information. In the training set there are 1909 examples and only 42 of them bind. Hence the data are highly unbalanced (42 positive examples is only 2.2% of the data). The test set contains 634 additional compounds, which were generated based on the assay results recorded for the training set.

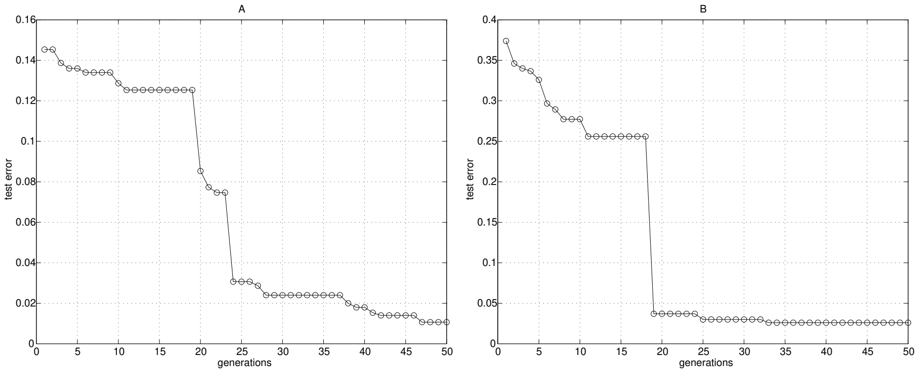

Fig. 1. GA (*best 2-features* mode) performance on A: linear problem, B: nonlinear problem. The x-axis is the GA generations, and the y-axis the test error as a fraction of test points.

We apply a GA using the *m-features* operator for feature selection using as inductive classifier a very simple approach, the OR operator. The use of this operator is fully justified because there are very few 1s in each feature and we are looking for features with 1s highly correlated with the positive outputs. In addition the OR classifier provides a very fast way of computing the percentage of error in classification for this problem, without training or adjusting any hyper-parameter. Following the KDD rules, the error in classification is evaluated according to a *weighted accuracy criterion*, due to the unbalanced nature of the number of positive and negative examples:

$$err_{wac} = \left[\frac{1}{2}\left(\frac{\#\{\hat{y} : y = 1 \wedge \hat{y} = 1\}}{\#\{y : y = 1\}}\right) + \frac{1}{2}\left(\frac{\#\{\hat{y} : y = -1 \wedge \hat{y} = -1\}}{\#\{y : y = -1\}}\right)\right]. \quad (3)$$

Table 1 shows the results of the best solution achieved by our GA running in *best 7-features*, *10-features*, *13-features* and *16-features* modes, *standard* mode and the results obtained for the winner of the KDD Cup 2001 (Bayes Network) [4].

Note that our algorithm achives better results that the best existing algorithm. The best solution was obtained with the GA working in *best 13-features* mode, with a $err_{wac} = 74.5\%$, an improvement of 6% over the best solution given in the KDD Cup[1]. The best solution obtained with the GA working in *standard* mode selected 18 features, with a $err_{wac} = 69.6\%$, which is still better than the best solution in the KDD Cup. However, in general, the GA running in *standard* mode performs worse than running in *m-feature* mode.

[1] Only the 7% of all competitors at the KDD achieved err_{wac} higher than 60.0%.

Table 1. Best err_{wac} obtained by GA running in different modes and the best err_{wac} in the KDD Cup 2001.

Algorithm	err_{wac}
GA (*7-features* mode)	70.8%
GA (*10-features* mode)	71.1%
GA (*13-features* mode)	**74.5**%
GA (*16-features* mode)	73.5%
GA (*Standard* mode)	69.6%
Winner of KDD Cup 2001 [4]	68.4%

5 Conclusions

In this paper a novel Genetic Algorithm for Feature Selection in classification problems has been presented. We have adapted the standard GA to the FSP, introducing a novel operator (*m-features* operator) which allows selecting the best m features of the input data for a given problem. This operator also reduces the size of the search space, improving the performance and the convergence of the GA in the FSP.

Experiments in synthetic problems and in a real application, the Thrombin Binding Problem, have shown very good performance of our algorithm, achieving the correct solution in synthetic problems in a short time and better results than any other existing approach for the Thrombin Binding Problem.

References

1. Blum, A. and Langley, P.: Selection of Relevant Features and Examples in Machine Learning. Artificial Intellingence, 97, (1997) 245–271.
2. Weston, H., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., Vapnik, V.: In: Sara A Solla, Todd K Leen, and Klaus-Robert Muller, (eds): Feature Selection for SVMs. Advances in NIPS 12, MIT Press, (2000) 526–532.
3. Kohavi, R., John, G. H.: Wrappers for Features Subset Selection. Int. J. Digit. Libr, 1, (1997) 108–121
4. Annual KDD Cup: http://www.cs.wisc.edu/~dpage/kddcup2001/.
5. Vapnik, V. N.: Statistical learning theory. John Wiley & sons, New York, 1998.
6. Chen, H. Y., Chen, T. C., Min, D. I., Fischer G. W. and Wu, Y. M: Prediction of Tacrolimus Blood Levels by Using the Neural Network with Genetic Algorithm in Liver Transplantation Patients. Therapeutic Drug Monitoring 21, (1999) 50–56.
7. Vafaie, H and De Jong, K. A.: Genetic Algorithms as a Tool for Features Selection in Machine Learning. Proc. of the 4th int. conference on tools with artificial systems, IEEE computer society press, Arlintong, VA, (1992) 200–204.
8. Goldberg, D. E: Genetic Algorithm in Search Optimization and Machine Learning. Addison-Wesley Publishing Company, Inc (1989).