Genetic Algorithm for feature selection: application to prediction of mortality during hypotensive episodes in patients with sepsis and severe sepsis

Louis Mayaud, Lionel Tarassenko and Gari D. Clifford

October 11, 2012

Abstract

This document describes the algorithm used to select features for prediction of mortality during the first hypotensive episode requiring medical intervention in a population of patients with sepsis and severe sepsis. Results are compared with state-of-the-art feature selection techniques and shown equivalent performance. The potential of the Genetic Algorithm as an optimization technique is briefly discussed. The code, on which this research is based, is freely available on [Google Code].

1 Introduction

A Genetic Algorithm (GA) is a heuristic algorithm, used as an optimization technique, which mimics the mechanisms of DNA duplication and natural selection. It is initialized with random ‘individuals’ (also called genomes constituting a binary vector which defines which subset of variables will be used). The performance of each individual is estimated with a fitness function (which measures how well an individual, or given subset of variables, discriminates and calibrates in a prediction task). The initial (random) choice of individuals is likely to be sub-optimal, and so an iterative process of ‘natural selection’ is used to converge to a stable population of suitable individuals. At each iteration, a percentage of the best individuals are bred to generate an offspring (the new generation), on which selection and breeding is subsequently applied. The very best individuals are cloned (to ensure the best individual in the next generation are at least as good as the previous), and the poorest performing individuals are removed from the ‘breeding cycle’. Eventually, this evolutionary process selects the most adapted subset of variables with

respect to the given fitness function \[1 \ 3 \ 9\]. This technique has been successfully applied to variable selection \[14\] and in particular on biomedical and clinical datasets \[15 \ 4\].

2 Definitions

Each patient in the database is called an observation. For each patient various measurements are recorded (laboratory results, vital signs, age, \ldots) and will further referred as features or variables. The value of the \(j^{th}\) variable observed in the \(i^{th}\) patient will be noted \(x_{ij}\) where \(i = 1..N\) and \(j = 1..P\), with \(N\) and \(P\) the number of patients and variables, respectively.

2.1 The data

The dataset presented in this work is composed of \(N = 2,113\) patients, for whom \(P = 189\) variables were collected or computed. The design set was randomly split over \(N_{\text{design}} = 1500\) and \(N_{\text{test}} = 613\) patients.

2.2 Cross-validation

The training-validation set, or design set, is defined as: \(X_{\text{design}} = X_{\text{test}} = x_i\), where \(i = 1..N_{\text{design}}\). The test set is defined as \(X_{\text{test}} = x_i\), where \(N_{\text{design}} < i < N + 1\). This data will not be used during the process of feature selection described here, nor will it be during model fitting. This allows us to interpret the performance on the test set as a good estimate of how the algorithm will perform on any future data drawn from a similar patient population.

Supervised classification techniques are designed to find relation between a group of variables and another (the former being usually refered as the data and the latter as the outcome). A good use of these techniques will usually try to avoid overfitting and provide good generalization. To do so, during the model design, data is usually split into training and validation sets, a process often refered as cross-validation \[5\]. As one would expect, performance will vary depending on which training set is drawn from the data, as well as on which validation set it is applied to. Bootstrapping is one of the techniques that takes into account this variability \[5\]: \(B\) random subsets of \(N_{\text{train}}\) samples, \(X_{\text{train}}^{(b)}\), are independently drawn from the data for training and results are evaluated on the \(N_{\text{test}}\) remaining samples, \(X_{\text{test}}^{(b)}\), providing \(B\) different values of the performance metric.
2.3 Genetic algorithm

A Genetic Algorithm mimics mechanisms of DNA replication under external constraints as observed in nature. Its terminology is therefore largely inspired from the field of biology:

- an *individual* or *genome*, noted $w_j$, is a binary vector of length the number of variables in the initial dataset $P$, and indicates whether a variable is selected (1), or not (0);
- a *population* or *generation*, noted $W$, is composed of $Q$ individuals and is represented by a binary matrix of dimension $Q \times P$. Each line in the population (i.e. a genome) represents a different subset of variables. Each column in the population designate a feature and denotes how often a variable was selected by the genomes in the current generation.

The evaluation of each subset of variable (genome) will be estimated during a bootstrapping procedure: out of the $N_{design}$ observations, $B = 100$ independent training sets of size $N_{training} = 1000$ are drawn (with replacement). The *outcome*, noted $y$, is a binary vector set to one for non-survivors and zero for patients who were discharged from hospital alive.

2.4 Logistic regression

The logistic function with zero mean $\mu = 0$ and unit variance $\sigma = 1$ is expressed as follows:

$$
\pi(g) = \frac{1}{1 + e^{-g}}
$$

(1)

Given a binary outcome $y$, the logistic regression try to maximize the following *likelihood*, $l(\beta|X)$, of the $\beta_j$ parameters:

$$
l(\beta|X) = \arg \max_{\beta_j} \sum_{i \in \text{training}} (1 - y)(1 - g) \times y g
$$

(2)

where $g(\beta, X) = \pi \left( \beta_0 + \sum_{j=1}^{P} \beta_j x_{ij} \right)$. It is however often convenient to find the *log* of equation 2 and therefore instead maximize the *log-likelihood*:

$$
\arg \max_{\beta_j} \sum_{i \in \text{training}} [(1 - y)(1 - g) + yg]
$$

(3)
3 Description of the algorithm

3.1 Initialization

The first population $W^{(0)}$ is a random population of $Q = 100$ genomes, each drawn from a Bernoulli distribution with the mean arbitrarily set to $p = N_{Non-survivors}/(10 * 0.7 * N_{design})$. This initialization ensures that, on average, genomes do not select more than 10 variables per positive event (patient dies in the hospital) found in the training set. Hence, each row is a random sequence of $P = 189$ zeros and ones, where 1 in a column indicates that the corresponding variable is included in the subset of features designed by the line (i.e., the genome).

3.2 Iteration

![Figure 1: Description of iteration $k$ of the Genetic Algorithm. Each row (genome/individual) in generation $k$ is evaluated as follows: (1) in the original design data, the features indicated by the binary genome are extracted; (2) during a bootstrapping procedure the performance of the subset of variables is estimated on different validation sets; finally (3) the new generation $k+1$ is created as detailed in figure 2.](image)

Each generation (starting from the one described above) is evaluated as detailed in figure 1.

For each genome:

1. The subset of features selected by each genome is extracted from the dataset reducing the dimensionality from 189 to $p_{kj}$, the number of feature selected by the $j^{th}$ genome in the
2. each genome is evaluated during a bootstrapping procedure, with respect to the fitness function detailed in section 3.2.1;

3. finally, the population is ordered with the fitness function and bred as explained in section 3.2.2;

### 3.2.1 Fitness function

For each training set $b$ in the bootstrap procedure, the performance of the subset of variables is estimated as follows:

1. logistic regression $\beta^{(b)}$ parameters are fitted with equation 3;

2. probability of death is then estimated with $\hat{y}^{(b)} = \pi(\beta^{(b)}, X_{\text{train}}^{(b)});

3. finally, the performance of the genome on this validation data $X_{\text{validation}}^{(b)}$ is taken as the log-likelihood described in equation 3.

The final score given to the $j^{th}$ subset of variables is computed from all the bootstrap values:

$$l(\beta|X^{(j)}) = \frac{1}{B} \sum_{b=1}^{B} \left( \sum_{i \in \text{validation}^{(b)}} (1 - y^{(b)}(1 - g(\beta, x_{i}^{(j)}))) + y^{(b)} g(\beta, x_{i}^{(j)}) \right)$$ (4)

### 3.2.2 Breeding

The breeding process is detailed in figure 2. It shows that a children population is composed of:

- 10% of the best genomes from the parent population (3b);
- 90% of mutated offspring (3a), which were created during a three-step process, that mimics DNA replication:
  - 3a-i genomes falling in the best 45% of the parent population are randomly paired up;
  - 3a-ii selected pairs of parents are crossed over at random locations, creating two children;
  - 3a-iii finally, children undergo random mutation of 20% of their genomes.

The parameters in this section were initially set to state-of-the-art values [14] and subsequently tuned. However, the algorithm was not found to be sensitive to them.
Figure 2: Genomes in $k^{th}$ population are sorted by descending log-likelihood on validation set. The first 10% genomes are directly passed down to the next generation (Elitism, 3b). The first 45% are bred: (3a-i) genomes are randomly paired up; (3a-ii) pairs of parents are crossed-over at random positions to create 2 offspring; finally (3a-iii), a random 20% of children’s genome are mutated (bits are flipped). Eventually, the $(k + 1)^{th}$ generation is composed of 10% best genomes from previous population and 90% of mutated offspring.

### 3.2.3 Stopping criterion

The maximum number of generations was set to $K = 200$ and, in order to prevent the selected variables from overfitting to the splits of the data chosen in the bootstrap, an early stopping criterion was defined (from the $10^{th}$ generation) as:

$$l_{j+10} - l_j < 0.001$$

### 4 Selecting variables

The GA was run $G = 500$ times in parallel. The best genome of each last population was extracted. The importance of each variable was defined as the percentage of inclusions over all 500 best genomes. The model dimensionality was selected as the median of the number of features included by these 500 best genomes.
5 Comparing Genetic Algorithm

5.1 The Least Absolute Shrinkage And Selection Operator (LASSO) technique

The Least Absolute Shrinkage And Selection Operator (LASSO) estimate was introduced by Tibshirani [10] and defined by

$$\hat{\beta} = \arg \min \sum_{i=1}^{N} \left( y_i - \sum_j \beta_j x_{ij} \right)^2$$

subject to $\sum_j |\beta_j| < t$, where $t$ is a tuning parameter. This formulation is equivalent to a $L_1$-norm regularization:

$$\hat{\beta} = \arg \min \sum_i (y_i - \hat{y}_i)^2 + \lambda \sum_j |\beta_j|$$

The main benefit of the LASSO over the ridge regression [2] (or $L_2$-norm regularization) is that it discards irrelevant features ($\beta_i = 0$) when $\lambda$ is small enough. It is a quadratic programming problem for which standard numerical analysis algorithms can be used to search for minima. Because of these advantages, LASSO has gained popularity over the past decade and is a standard technique for feature selection which has been successfully applied, among other fields, to predict mortality [11].

5.2 The Evidence Procedure

Bayesian statistics provide an excellent framework for model comparison, which was detailed in length by MacKay [6] and in [8, 7]. Comparison is achieved by looking at the posterior probability of a model’s assumptions, $\mathcal{H}$, given the data:

$$P(\mathcal{H}|D) \propto P(D|\mathcal{H})P(\mathcal{H})$$

The Evidence for the data $P(D|\mathcal{H})$ in equation 8 can be marginalized as follow:

$$P(D|\mathcal{H}) = \int P(D|\beta, \mathcal{H})P(\beta|\mathcal{H})d\beta$$

where $\beta$ are the parameters of our model. Assuming that the posterior $P(\beta|D, \mathcal{H}) \propto P(D|\beta, \mathcal{H})P(\beta|\mathcal{H})$
has a strong peak at the most probable parameters $\beta_{MP}$, we can estimate the evidence by the height of the integrands’ peak times its width $\sigma_{\beta|D}$ (see figures in MacKay et al. [8]):

$$P(D|\mathcal{H}) = P(D|\beta_{MP}, \mathcal{H}) \times P(\beta_{MP}|\mathcal{H}) \times \sigma_{\beta|D}$$  \hspace{1cm} (10)

The first term of equation 10 is the best fit likelihood that can be estimated from equation 4. The second term only depends on the priors given to the parameters (i.e. our initial hypothesis on the model). The width of the parameters given the data (the posterior $\sigma_{\beta|D}$), can be estimated from the Hessian $A$ (or the inverse of the covariance matrix) as $\sigma_{\beta|D} = \det^{-1/2}(A/2\pi)$.

Equation 10 will favor simple models with good generalization against complex models: this naturally embodies the concept of Occam’s razor. One main advantage of this technique is that the whole design data can be dedicated to select the variable selection without the need of bootstrapping procedures.

We implemented the following fitness function on the GA, where $l_j$ is the fitness function defined in equation 4:

$$fitness_j = l_j + \log \left( \prod_{l=1}^{p_k} P(\beta_l|\mathcal{H}) \times \frac{1}{\sqrt{\det(A/2\pi)}} \right)$$  \hspace{1cm} (11)

5.3 Results

Figure 3 shows the evolution of the Area Under the Curve (AUC) for different feature selection techniques.

5.4 Discussion

The performance of Genetic Algorithm partly depends on the chosen fitness function. If different techniques optimize the exact same criteria, it is reasonable to believe that they will provide equivalent solution. Therefore, with a fitness function based on Mutual Information it is possible that GA will not outperform Joint Mutual Information (JMI) based techniques [3]. Similarly, with a fitness function based on the likelihood of parameters (LLH) from a logistic regression, it should provide equivalent result than that of LASSO. Indeed, we compared GA (with LLH fitness function) to LASSO and did not find any statistically significant difference in discriminative power (Area Under the Receiver Operating Curve). Figure 3 shows that the performance of GA
Figure 3: Area Under the Receiver Operating curve (AUROC) on the test set (n=613) plotted against models dimensionality. Features are added one by one in order of importance given by each feature selection technique: Stepwisefit, LASSO, GA and GA Evidence.

and LASSO are equivalent with nearly overlapping curves. This result show that there is no reason to prefer one technique to the other on this dataset.

Thanks to their generic nature however, GAs allow optimization of different types of fitness function. Two cases are particularly interesting:

1. where no analytic solution can be found (or at the cost of drastic assumptions);

2. where gradient descent or sequential elimination techniques are likely to fail given: 1) the large size of the search space and 2) the presence of local minima.

For instance, the fitness function based on the Evidence procedure provides a promising way to select the optimal model dimensionality with the right subset of features without the need of computationally intense cross-validation procedure: with only half the number of variables included in models from LASSO and GA-LLH, this model show equivalent performance. This illustrates the potential advantage of GA over traditional techniques. Fitness functions derived from support vector machines [12] is another example of current research by the authors, but such classifiers lead to far a more computationally intensive optimization.
6 Conclusion

We provide in this document an extensive description of the way GA was designed for this work. The algorithm provided performance equivalent to that of state-of-the-art feature selection techniques. However, we propose that the GA has extended capabilities due to its generic nature. The code used for this work is freely available online on Google Code and is provided with a graphical user interface.

References


