Les Cahiers du GERAD ISSN: 0711–2440

Bagged parallel genetic algorithms for objective model selection

M. Larocque,
J.-F. Plante, M. Adès

G–2018–70

September 2018

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2018
– Bibliothèque et Archives Canada, 2018

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2018
– Library and Archives Canada, 2018

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Bagged parallel genetic algorithms for objective model selection

Maxime Larocque [a]

Jean-François Plante [a]

Michel Adès [b]

[a] *GERAD & Department of Decision Sciences, HEC Montréal, Montréal (Québec), Canada, H3T 2A7*

[b] *Department of Mathematics, Université du Québec à Montréal, Montréal (Québec), Canada, H3T 2A7*

denis.larocque@hec.ca
jfplante@hec.ca
ades.michel@uqam.ca

September 2018
Les Cahiers du GERAD
G–2018–70

Abstract: Genetic algorithms are used for feature selection through a fitness function that drives the evolution of populations. With parallel universes, an importance score may be produced for each feature to determine subjectively from a plot which to retain. The authors derive the distribution of those importance scores under the null hypothesis that none of the features has predictive power and they determine an objective threshold for feature selection. The authors discuss the parameters for which the theoretical results hold. They illustrate their method on real data and run simulation studies to describe its performance.

Keywords: Feature selection, genetic algorithms, linear models, machine learning, predictions

Acknowledgments: For partial support of this work through research grants, thanks are due to the *Natural Sciences and Engineering Research Council of Canada*.

1 Introduction

In predictive models, feature selection can boost performance and help avoiding overfitting. Classical approaches such as stepwise regression, and all subset selection based on penalized statistics such as AIC and BIC are routinely taught (see e.g. (Draper and Smith, 2014)). Penalized regression methods such as the lasso select features and fit a model simultaneously (see e.g. (Tibshirani, 1996)). Genetic algorithms have also been proposed, including a parallel version by Zhu and Chipman (2006) who was revisited by Zhang et al. (2015) and Wang and Zhang (2015).

Biomimetics consists in imitating nature to solve complex human problems. For instance, genetic algorithms are inspired from natural selection. For feature selection, models are seen as individuals whose DNA is a vector of binary markers that indicate whether or not each feature is used in the model. Successive generations of models are generated by selecting among the fittest parents of the current population, and creating offsprings by selecting randomly the genes from both parents, a step called crossover. To improve the richness of the population, models may undergo mutation (where a bit is flipped). While the general steps of a genetic algorithm are clear, the actual recipe for each is chosen from numerous options.

Genetic algorithms often involve a single population that evolves until the fittest individual is found. A generalization proposed by Zhu and Chipman (2006) considers a fixed number of parallel populations who evolve independently of one another. After a number of steps that could be predetermined, the frequency of a feature in the last generation of all populations is used as a measure of its importance. A bubble plot allows to visually identify where the importance drops to separate important features from those who should be dismissed.

To illustrate the bubble plot, let us consider the Pollution dataset of McDonald and Schwing (1973) where a measure of mortality is predicted with linear regression from 15 features on 60 data points. The left panel of Figure 1 shows a bubble plot, a tool proposed by Zhu and Chipman (2006). The associated order plot is shown on the right panel. The order plot is identical to the bubble plot, except that the features are ranked in order of decreasing importance. The figure suggests the selection of features x_1, x_9 and x_{14}. Although Zhu and Chipman (2006) analyzed the Pollution dataset, Figure 1 was generated with a different genetic algorithm hence the different figure. While the bubble plot is interesting and intuitive, it requires a visual inspection, which makes the method subjective and limits its automation as a human intervention is required to complete the selection.

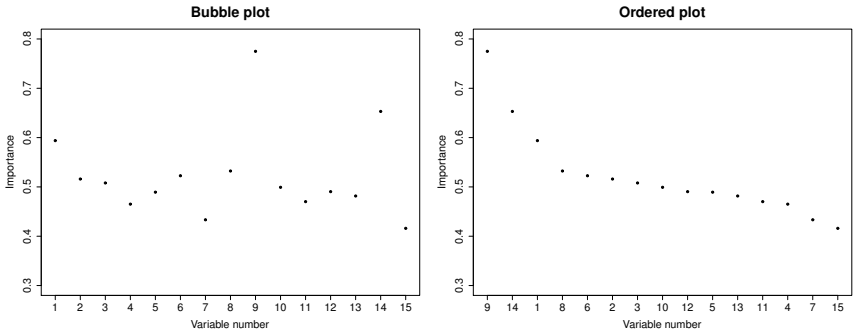

Figure 1: Example of a bubble plot and its associated ordered plot as suggested by Zhu and Chipman (2006). The importance of each feature is measured by its prevalence among the individuals of the last populations generated by parallel genetic algorithms.

To make parallel genetic algorithms more objective, we derive the distribution of the importance score of every feature under the null hypothesis that none of the features are good predictors of the response. To increase the richness of the parallel populations and their ability to detect truly important features, we also add a step of bagging, providing every parallel universe with a bootstrap sample of the data. The null distribution of the importance is used to determine an objective threshold at the global level α. Drawing the bubble plot then becomes optional since feature selection can be automatized by comparing the importance of each feature against the threshold.

Section 2 of the paper introduces the basics of genetic algorithms for model selection as well as some notation. Our method which we call Baggeg Genetic Algorithm (BGA) is described in Section 3 along with the mathematical results that support it. Case studies and Monte Carlo simulations are analyzed in Section 4. Section 5 discusses the choice of parameters in a bagged genetic algorithm. Section 6 offers a conclusion.

2 Genetic algorithms for model selection

Genetic algorithms are meta-heuristic optimization techniques mimicking a darwinian vision of evolution in order to solve complex, numerically intractable problems (see e.g. Shonkwiler and Mendivil (2009)). The general functioning of genetic algorithms consists of the successive application of three genetic operators – selection, crossover and mutation – to an initial population of candidate solutions or individuals. For model selection, each individual is encoded using a binary string which indicates whether or not each feature is present in the model. At every generation, the likelihood of an individual generating an offspring depends on its fitness, and the process is iterated a number of times.

We first describe the genetic operator in the more classic setting of a single-thread genetic algorithm, where the evolution of only one population is generated. We then extend the notation to the parallel universes proposed by Zhu and Chipman (2006).

2.1 Single-thread genetic algorithm (SGA)

In a single-thread algorithm, individual $i \in \{1, \ldots, I\}$ of generation $g \in \{1, \ldots, G\}$ is represented by the binary vector $B_{ig} = [b_{1ig}, \ldots, b_{Dig}]$ where b_{dig} is one when feature $d \in \{1, \ldots, D\}$ is active for this individual and zero otherwise. A typical application for model selection is to consider linear regression where each explanatory variable is included or not in the model as coded by b_{dig}.

2.1.1 Initial population

The initial population of individuals is generated randomly. The b_{di0} are hence drawn from independent Bernoullis with probability of success π_0, the activation rate.

2.1.2 Fitness

In a general genetic algorithm, the fitness function $f(B_{ig})$ measures the ability of an individual to "solve the problem". In a model selection setting, f should measure the predictive ability of the corresponding model. An ideal measure of fitness should not be biased by the number of features present in a model. Zhu and Chipman (2006) used a leave-one-out cross-validated residual sum of squares, but we prefer to use a validation subset to evaluate a properly scaled residuals sum of squares (RSS). Since better individuals should have a larger fitness, we use a negative power of our rescaled RSS.

2.1.3 Selection

At generation g, I individuals are available to become parents. To create an offspring, two individuals are selected randomly, with replacement (parthenogenesis is allowed). The probability of selection of an individual is proportional to its fitness. That is, for g fixed, the probability of selection of individual i' is $f(B_{i'g})/\{\sum_{i=1}^{I} f(B_{ig})\}$.

2.1.4 Crossover

Crossover determines how the genes of the two selected parents are combined to generate one offspring. Zhu and Chipman (2006) used one point crossover where an integer is chosen randomly between 1 and $D-1$. The genes of the father are used up to that integer, and the genes of the mother are used for the rest. One disadvantage of this approach is that the arbitrary order in which the variables are coded has an influence on the models that are possible to generate. We have a preference for uniform crossover where each gene of the

offspring is taken randomly from its mother or father. Let $\mathbf{C}_{ig}$ be a diagonal matrix where the diagonal entries are independent Bernoullis with probability of success 1/2. Then if individuals i_1 and i_2 of generation g are selected as the parents of individual i in generation $g+1$, the crossover will first generate the embryo

$$B^*_{ig} = \mathbf{C}_{ig} B_{i_1 g} + (\mathbf{I} - \mathbf{C}_{ig}) B_{i_2 g}$$

where $\mathbf{I}$ is the $D \times D$ identity matrix. The embryo B^* will become an individual in generation $g+1$ once the mutation step is complete.

2.1.5 Mutation

Mutation helps maintaining genetic diversity by making each gene of the embryo subject to a flip. Let $\mathbf{M}_{ig}$ be a diagonal matrix whose diagonal is filled with independent Bernoullis with parameter θ_g, the probability of a mutation, which could vary as generations evolve or be held fix. A decreasing θ_g, for instance, is akin to simulated annealing (see e.g. (Kirkpatrick et al., 1983)). Individual i of generation $g+1$ may then be obtained as

$$B_{i(g+1)} = (\mathbf{I} - \mathbf{M}_{ig}) B^*_{ig} + \mathbf{M}_{ig}(\mathbf{1} - B^*_{ig}).$$

2.1.6 Evolution

The process described is repeated recursively. While it is possible to iterate until a convergence criterion is met, we suppose a fixed number of generations, G. For a single-thread algorithm, the fittest individual of the last generation, $\arg\max f(B_{iG})$, is usually outputed as the solution.

As a meta-heuristic algorithm, there exists a very large number of variants for the genetic operators to which additional parameters may also be added. Our description of the operators is focused towards the method that we develop, but readers that are interested in learning more about the numerous uses of genetic algorithms and many variants of the genetic operators may consult, e.g. Mitchell (1998), Cantú-Paz (2000), Haupt and Haupt (2004) or Poli et al. (2008).

2.2 Parallel genetic algorithms (PGA)

Zhu and Chipman (2006) suggest to create U universes in which populations evolve in parallel. New indices need to be added to the notation introduced previsouly to account for the universe. Namely,

Genes: $B_{igu} = [b_{1igu}, \ldots, b_{Digu}]$ is a binary vector with the genetic code of individual i of generation g in universe $u \in \{1, \ldots, U\}$.

Fitness: Since we use out-of-sample validation, the fitness function in parallel universes will be based on different hold-out datasets. The notation f_u shows this dependence of the fitness function on the universe.

Selection, crossover and mutation: The genetic operators are identical in the parallel universes. They are applied independently in each of the parallel populations, from initial generations that are created in the same fashion previsouly described.

With parallel worlds, the output is not a single fittest individual, but an importance score based on the frequency of each gene in the final population. For feature d, the importance score is the proportion

$$\hat{\pi}_d = \frac{1}{IU} \sum_{i=1}^{I} \sum_{u=1}^{U} b_{diGu}$$

and we note $\hat{\boldsymbol{\pi}} = [\hat{\pi}_1, \ldots, \hat{\pi}_d]$ the vector of those values for all features. The bubble and order plots of Figure 1 are a graphical representation of $\hat{\boldsymbol{\pi}}$.

In the next section, we describe how to derive a threshold for the values of $\hat{\boldsymbol{\pi}}$ to determine which features should be kept, and which should be dismissed.

3 Bagged genetic algorithms (BGA)

Using PGA for feature selection yields importance score for the variables rather than a single solution as does SGA. The final decision of which variables to include is however based on the visual inspection of a figure. The purpose of this paper is to offer an objective and automatizable way to make that final decision.

Let us focus on a linear model setting where predictor d has a parameter β_d and where $\beta_d = 0$ means that the feature has no effect on the target. To establish a threshold value, we derive the distribution of $\hat{\pi}_d$ for $d \in \{1, \dots, D\}$ under a null hypothesis that implies some symmetry between individuals. Namely, under $H_0 : \beta_1 = \cdots = \beta_D = 0$, the fitness of any individual is assumed to be approximately equal since they all have an equally low ability to predict the target variable. At the selection step, this means that all individuals are equally fit, hence have an equal probability of being selected. This hypothesis of symmetry is reinforced through careful choices in the design of the genetic algorithm which are described next.

3.1 Hold out sample and bootstrap

In each universe, the dataset with N data is split into a training and a validation set. The $\tilde{n}$ data from the training set are stored in the vector of response $\tilde{\mathbf{Y}}_u$, and the predictors in the $\tilde{n} \times (D+1)$ design matrix $\tilde{\mathbf{X}}_u$ whose first column is filled with ones to account for the intercept. The rest of the data become the validation set and we denote the correspoding values n, $\mathbf{Y}_u$ and $\mathbf{X}_u$. When working with a very large dataset, sample sizes may be chosen so that $n + \tilde{n} < N$. For smaller samples, $n + \tilde{n}$ is likely equal to N, so we may use bootstrap instead to increase the diversity between universes. Sampling with replacement is then used to generate bootstraped versions of the data (of the same sizes $\tilde{n}$ and n) that are also denoted by $\tilde{\mathbf{Y}}_u$, $\tilde{\mathbf{X}}_u$, $\mathbf{Y}_u$ and $\mathbf{X}_u$. Since we combine the output of all universes in the end, this is a form of bagging (see (Breiman, 1996b)) and is likely to boost the performance of the algorithm. This was observed, e.g. by by Zhang et al. (2015) and Wang and Zhang (2015) who add noise to the parallel universes in a PGA.

An additional benefit of bagging is the dilution of the spurious correlations that could occur just by chance. Such correlations could be amplified through the generations of the genetic algorithm, but the bootstrapping makes it unlikely to occur simultaneously for the same variables in multiple parallel universes.

3.2 Choice of fitness function

The derivation of the distribution of the importance in the next section is based on the assumption that the fitness of all models is approximately equal under the null hypothesis. Using out-of-sample validation yields a fitness that is not influenced by the number of active features. The fitness we use is based on a rescaled sum of residuals calculated on the validation sample, but other choices would also be acceptable.

Consider an arbitrary individual B with p active features, i.e. $||B||^2 = p$. Let $\mathbf{B}$ be a $(D+1)\times(p+1)$ matrix generated by removing some columns from the $(D+1)\times(D+1)$ identity matrix. Namely, column one always remains for the intercept, and the following are matched to the binary values in B, all columns associated to a null value being removed. The $\mathbf{B}$ matrix allows to select appropriate columns from the design matrices. In universe u, the out-of-sample predictions from the linear regression model associated with B are then

$$\hat{\mathbf{Y}}_u(B) = \mathbf{X}_u\mathbf{B}(\mathbf{B}^\intercal\tilde{\mathbf{X}}_u^\intercal\tilde{\mathbf{X}}_u\mathbf{B})^{-1}\tilde{\mathbf{B}}^\intercal\tilde{\mathbf{X}}_u^\intercal\mathbf{Y}_u$$

which yields the unscaled residuals $\hat{\boldsymbol{\varepsilon}}(B) = \mathbf{Y}_u - \hat{\mathbf{Y}}_u(B)$. By the independence of the error in the training and the validation datasets, this vector of residuals is a multivariate normal with mean zero and covariance

$$\Sigma_u(B) = \sigma^2\left\{\mathbf{I} + \mathbf{X}_u\mathbf{B}(\mathbf{B}^\intercal\tilde{\mathbf{X}}_u^\intercal\tilde{\mathbf{X}}_u\mathbf{B})^{-1}\mathbf{B}^\intercal\mathbf{X}_u^\intercal\right\} \tag{1}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix. Since the selection is based on the relative value of the fitness, the actual value of σ^2 is irrelevant in the calculation of the fitness function

$$f_u(B) \propto \left[\hat{\boldsymbol{\varepsilon}}(B)^\intercal\left\{\Sigma_u(B)\right\}^{-1}\hat{\boldsymbol{\varepsilon}}(B)\right]^{-\gamma} \tag{2}$$

where γ is a positive scalar that can enhance the peakedness of the function. We found empirically that $\gamma = 1.5$ seems to work well.

Since the residuals are asymptotically normal, the distribution of the expression inside the brackets in Equation 2 up to a multiplicative constant is chi-square with n degrees of freedom. Most importantly, the degrees of freedom do not depend on p, the number of active features.

There are numerous other possible fitness functions in the literature. As long as a fitness function is sufficiently unbiased to offer an approximately equal fitness to all models under the null hypothesis, then the results of the next section should hold.

3.3 Distribution under the null hypothesis

In generation $g = 0$, all genes are generated at random, hence yielding DIU independent Bernoulli variates with probability of success π_0. As the populations evolve, these genes are recombined into new offsprings. We look into the distribution of those genes at generation G, when all populations are fully evolved.

Note that under the null hypothesis, the expected value of the genes of an embryo B^* is equal to that of its parents since all generated genes have the same probability of being selected. The mutation step, however, has an effect on the expected proportion of ones. Fixing d and u, then conditioning on the presence of a mutation, we get the recurrence

$$\pi_g = \mathrm{E}(b_{digu}) = (1 - \pi_{g-1})\theta_g + \pi_{g-1}(1 - \theta_g)$$

that may be written alternatively as $\Delta_{g+1} = (1 - 2\theta_g)\Delta_g$ if we define $\Delta_g = 1/2 - \pi_g$. Iterating yields $\Delta_G = \Delta_0 \prod_{g=1}^{G}(1 - 2\theta_g)$, or

$$\pi_G = \frac{1}{2}\left\{1 - (1 - 2\pi_0)\prod_{g=1}^{G}(1 - 2\theta_g)\right\} \tag{3}$$

which converges to $1/2$ if $G \to \infty$ and $\theta_g \in (0, 1/2)$ infinitely often.

Remark 1 *If the probability of mutation is fixed for all generations such that $\theta_g = \theta$, then equation 3 simplifies into $\pi_G = \{1 - (1 - 2\pi_0)(1 - 2\theta)^G\}/2$.*

Since there is no exchange between parallel universes, bits b_{diGu_1} and b_{diGu_2} will always be independent when they come from different universes. By the virtues of uniform crossover as well as the assumption that all models are equally likely to be chosen, genes in different positions, b_{d_1iGu} and b_{d_2iGu} should also be uncorrelated. For fixed d and u, two different individuals may however be correlated as they are likely to share common ancestors. In the process of creating generation $g = 1$, we may condition on having inherited gene d from a common parent to get $c_0^* = \mathrm{cov}(b^*_{di_10u}, b^*_{di_20u}) = \pi_0(1-\pi_0)/I$ for the embryo, or more generally, when creating embryos from generation g,

$$c_g^* = \mathrm{cov}(b^*_{di_1gu}, b^*_{di_2gu}) = \frac{1}{I}v_g + \frac{I-1}{I}c_g \tag{4}$$

where $v_g = \mathrm{var}(b_{digu}) = \pi_g(1 - \pi_g)$ and $c_g = \mathrm{cov}(b_{di_1gu}, b_{di_2gu})$ for arbitrary i, d and u, and with $c_0 = 0$ for the initial population. Let m_{digu} identify the diagonal elements of $\mathbf{M}_{igu}$, and note for simplicity $m_1 = m_{di_1gu}$, $b_1^* = b^*_{di_1gu}$ and similarly for m_2 and b_2^*. We can then write explicitely

$$\begin{aligned} c_{g+1} &= \mathrm{cov}\{(1 - m_1)b_1^* + m_1(1 - b_1^*), (1 - m_2)b_2^* + m_2(1 - b_2^*)\} \\ &= (1 - 2\theta_g)^2 c_g^* \end{aligned} \tag{5}$$

after simplifications due to the independence of the mutation binary markers and properties of the covariance. Substituting Equation 4 in 5 yields the recurrence

$$c_{g+1} = (1 - 2\theta_g)^2\left\{\frac{1}{I}\pi_g(1 - \pi_g) + \frac{I-1}{I}c_g\right\}.$$

Despite their unwieldy expressions, the sequences c_g and v_g are easy to determine numerically with a simple loop.

The decision of which features to keep is made from the importance scores, $\hat{\pi}_d$ who can be seen as the average of U independent and identically distributed random variables. As the number of parallel universes U increases, the central limit theorem (see e.g. (Casella and Berger, 2002)) guarantees the convergence of $\sqrt{U}(\hat{\pi}_d - \pi_G)$ to a normal distribution with mean 0 and variance

$$\sigma^2_{\hat{\pi}} = \frac{1}{I}\pi_G(1-\pi_G) + \frac{I-1}{I}c_G.$$

The decision to retain any given variable may therefore be made from a normal quantile. Since all variables are simultaneously tested, and since their indivdual tests are uncorrelated, a Šidák correction (see (Šidák, 1967)) is applied to account for multiple comparisons. After choosing a global level α, the user should thus keep those variables for which

$$\hat{\pi}_d > \pi_d + z_{1-\alpha_S}\sigma_{\hat{\pi}}/\sqrt{U}$$

where $\alpha_S = 1-(1-\alpha)^{1/D}$ is the Šidák corrected level.

Except for $\hat{\pi}_d$ itself, none of those values need to be estimated. They are deterministic functions of the parameters of the genetic algorithm. This result does not depend on the fitness function either, as long as it has the ability to make all models approximately equally fit under the null hypothesis. In particular, a constant fitness function will yield this result, but would display no ability for detecting relevant variables when they are present. A more useful extension is to determine a fitness function with appropriate properties for Generalized Linear Models (GLM), which we do next.

3.4 Extension to generalized linear models

Let us consider GLM as described in McCullagh and Nelder (1989) for a response variable Y that follows an exponential family whose density may be written as

$$f_Y(y;\theta,\phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\}$$

where θ is called the canonical or location parameter, ϕ the dispersion parameter, and the functions a, b and c are family specific known functions. Following simple calculations (see e.g. Section 2.2.2 of (McCullagh and Nelder, 1989)), the following expressions for the mean and variance may be derived, namely, $\mathrm{E}(Y) = \mu = b'(\theta)$ and $\mathrm{var}(Y) = b''(\theta)a(\phi)$. These equations are used to find the variance function, $V(\mu) = \mathrm{var}(Y)/a(\phi)$ which must be expressed as a function of μ. Actual values of those functions for known families may be found in different references, including Table 2.1 of McCullagh and Nelder (1989).

In the definition of GLM, a linear combination η of the predictors is linked to an independent observation Y from an exponential family through a link function. The expression can also be reversed to have $\mu = \ell(\eta)$ where ℓ is the inverse link function. With the convention that ℓ is applied componentwise to a vector, we can write $\boldsymbol{\mu} = \ell(\mathbf{X}\boldsymbol{\beta})$ for a model with all features on the whole dataset, where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_D)$ is the vector of parameters of the model. For some families, ϕ may be a known constant, but in other cases, it is a nuisance parameter.

For exponential family GLM, the estimate of $\boldsymbol{\beta}$ may be found through maximum likelihood and usual properties thereof are retained. In the context of this paper, we only consider canonical link functions that provide additional simplifications, including the fact that $\theta = \eta$.

Moving to the context of BGA with the notation previsouly introduced, universe u has the bootstrapped samples $\tilde{\mathbf{Y}}_u$ and $\tilde{\mathbf{X}}_u$ and the p active features are indicated by B. The maximum likelihood equation is then based on the relation $\mathrm{E}(\tilde{\mathbf{Y}}) = \ell(\tilde{\mathbf{X}}\mathbf{B}\mathbf{B}^\intercal\boldsymbol{\beta})$ which depends only on a subset of $p+1$ elements of $\boldsymbol{\beta}$, namely $\boldsymbol{\beta}\mathbf{B}$. The MLE $\hat{\boldsymbol{\beta}}_u(B)$ is therefore an asymptotically normal vector of $p+1$ elements with limiting mean $\boldsymbol{\beta}\mathbf{B}$ and variance $\{\mathbf{B}^\intercal\tilde{\mathbf{X}}_u^\intercal\tilde{\mathbf{V}}_u(B)\tilde{\mathbf{X}}_u\mathbf{B}\}^{-1}$ where $\tilde{\mathbf{V}}_u(B)$ is a diagonal matrix with the variance function V applied componentwise to $\tilde{\boldsymbol{\mu}} = \ell(\tilde{\mathbf{X}}\mathbf{B}\mathbf{B}^\intercal\boldsymbol{\beta})$ on its diagonal. The dependence on B does not change the dimension of $\tilde{\mathbf{V}}_u(B)$, but it affects the values therein. The same applies to $\mathbf{V}_u(B)$ who is based on $\boldsymbol{\mu} = \ell(\mathbf{X}\mathbf{B}\mathbf{B}^\intercal\boldsymbol{\beta})$. The distributional result follows from the properties of the MLE and arithmetics to determine the Fisher information for $\boldsymbol{\beta}\mathbf{B}$. Applying the chain rule for second-order derivatives helps those calculations that are

further simplified by properties emerging from the choice of the canonical link. In practice, $\tilde{\boldsymbol{\mu}}$ may need to be estimated by replacing $\boldsymbol{\beta}$ with its MLE $\hat{\boldsymbol{\beta}}_u(B)$.

Building towards a fitness function based on a rescaled residual sum of squares, we may write out of sample predictions as

$$\hat{\mathbf{Y}}_u(B) = \ell\left\{X_u\mathbf{B}\hat{\boldsymbol{\beta}}_u(B)\right\}.$$

From an application of the multivariate delta method, $\hat{\mathbf{Y}}_u(B)$ is asymptotically multivariate normal with mean $\boldsymbol{\mu}_u(B)$ and variance

$$a(\phi)\mathbf{V}_u(B)\mathbf{X}_u\mathbf{B}\{\mathbf{B}^\intercal\tilde{\mathbf{X}}_u^\intercal\tilde{\mathbf{V}}_u(B)\tilde{\mathbf{X}}_u\mathbf{B}\}^{-1}\mathbf{B}^\intercal\mathbf{X}_u\mathbf{V}_u(B).$$

The jacobian of the link function appears as $\mathbf{V}_u(B)$ due to their equality resulting from the use of the canonical link. The unscaled residuals $\hat{\boldsymbol{\varepsilon}}(B) = \mathbf{Y}_u - \hat{\mathbf{Y}}_u(B)$ then have mean 0 and variance

$$\Sigma_u(B) = a(\phi)\left[\mathbf{V}_u(B) + \mathbf{V}_u(B)\mathbf{X}_u\mathbf{B}\left\{\mathbf{B}^\intercal\tilde{\mathbf{X}}_u^\intercal\tilde{\mathbf{V}}_u(B)\tilde{\mathbf{X}}_u\mathbf{B}\right\}^{-1}\mathbf{B}^\intercal\mathbf{X}_u\mathbf{V}_u(B)\right] \tag{6}$$

which will be required up to a multiplicative constant, hence the nuisance parameter ϕ may be ignored. Pearsons X^2 statistic uses a sum of squared rescaled residuals to measure the goodness-of-fit. With out-of-sample validation, Equation 6 provides the appropriate factor. For GLM, we therefore use the same fitness function shown in Equation 2, but with modified values for $\hat{\boldsymbol{\varepsilon}}(B)$ and $\Sigma_u(B)$ which must be calculated up to a multiplicative constant.

The results of Section 3 do not depend on the actual models, only on the assumption that the fitness of any model is approximately equal under the null hypothesis of no predictive power. No modifications need be made to the threshold once an appropriate fitness function has been developped. Similarly, it would be possible to use other goodness-of-fit methods as fitness functions, as long as they treat fairly models that have different number of variables.

3.5 Notable differences with PGA

While we adopt the parallel populations of Zhu and Chipman (2006), some of the choices they make in the parameters of their genetic algorithms are not suitable for BGA as they violate the assumptions that we use to derive the null distribution of the importance scores. Namely, the following elements are different.

Crossover: Zhu and Chipman (2006) use one-point crossover, but we explained in the description of crossover our preference for uniform crossover to make the arbitrary order of the features in the algorithm irrelevant.

Early-stopping: Using a finite predetermined number of generations yields importance scores that are averages of U independent values of equal variance. Early-stopping means that the the number of generation depends on their diversity. While the theoretical complications are significant, the benefits of early-stopping are less clear.

Elitism: Zhu and Chipman (2006) copy the fittest half of a generation directly to the following generation. The derivation of the null distribution for the importance scores assumes no such elitism.

Selection: At each generation of Zhu and Chipman (2006), half the population survives, and all parents are picked at random from that survival pool. The expected value and variance of the importance score are based on a different mechanism where the probability of selection is proportionnal to the fitness.

Fitness function: To enhance the symmetry between all possible models under the null, we use out-of-sample validation. One challenge with in-sample measures of fitness is to ensure that it does not display systematic preference (e.g. have bias toward models with more active features).

Bootstrap: We use bootstrap to enhance the symmetry between models under the null by diluting spurious links that could appear out of luck and hence avoiding to reproduce them in all the universes.

In the next section, case studies and Monte Carlo methods are used to explore the behaviour of the importance scores under the null hypothesis.

4 Empirical exploration

We use real datasets and Monte Carlo studies to explore the behaviour of BGA in practice. The definition of "best" model is subject to debates and could be based, for instance, on the predictive abilities of the selected model according to different metrics. We rather adopt the same view as Zhu and Chipman (2006), and look at the ability of the models to detect the true active features. Part of the exploration is also designed to verify that the distribution of $\hat{\boldsymbol{\pi}}$ under the null derived in Section 3.3 is observed empirically.

4.1 Illustrative data and harder problem

In the illustrative example of Zhu and Chipman (2006), 20 features labeled x_1 to x_{20} are simulated as independent normal variates with mean zero and variance one. We took the liberty to increase the sample size from their 40 to $N = 200$ points who were generated along with the response

$$Y = x_5 + 2x_{10} + 3x_{15} + \varepsilon \tag{7}$$

where ε are independent normal variables with mean zero and variance one. We use the same parameters as Zhu and Chipman (2006) for the genetic algorithm, namely: $I = 20$, $G = 8$, $U = 25$, and $\theta_g = 1/D$ for all g. The value of the activation rate does not seem to be reported in their paper; we used $\pi_0 = 0.3$. The validation set is drawn as 20% of the sample. Figure 2 displays the boxplots of the importance scores obtained by one run of BGA on each of 1000 datasets generated following Equation 7. The horizontal plain line shows π_G, the expected importance. The plain dots are for the empirical 95$^{\text{th}}$ quantile of the importance scores for each feature and can be compared to the dashed line who show their theoretical values under the null on the left panel. On the right panel, the correction for multiple comparisons is applied. The null model was obtained by setting all regression coefficients to zero in the same equation when simulating Y.

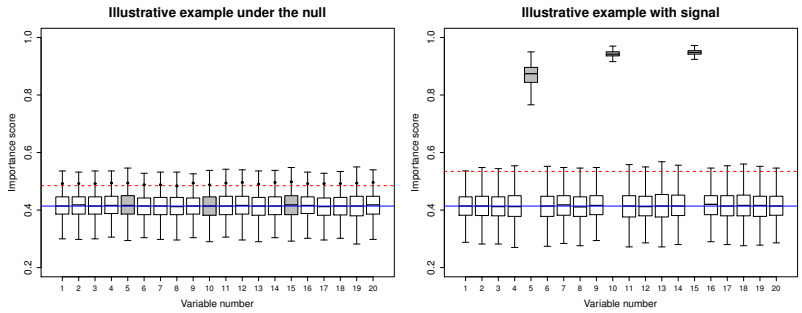

Figure 2: Importance scores obtained from the BGA of 1000 different samples generated from the same "Illustrative Example" scenario. The left panel shows the importance scores under the null hypothesis, where Y is independent from the features. The right panel shows the importance score when the signal described in Equation 7 is present. While the plain line shows the expected importance π_G, the dashed line displays the threshold $\pi_d + z_{0.95}\sigma_{\hat{\pi}}/\sqrt{U}$ on the left panel, and its Šidák adjusted equivalent on the right panel. The plain dots show the empirical 95$^{\text{th}}$ quantile of the importance scores for each variable to compare against that theoretical bound. The boxplot of the active features are coloured in gray.

Under the null, both the expected value and the 95$^{\text{th}}$ quantile of the importance scores match their theoretical values remarquably well. When a signal is present, we note that the correct relevant variables are always detected, but also, that the values of the empirical 95$^{\text{th}}$ quantiles are reasonnably close to their expected values under the null.

Let us now consider the hard problem that Zhu and Chipman (2006) attribute to George and McCulloch (1993). We preserve the sample size of 120 for the training set by simulating samples of 150 data of which a proportion of 20% is dedicated to validation. A total of 60 features of variance 2 with a compound symmetry correlation structure are simulated. A correlation of 0.5 is present between any two variables. The error term has mean 0 and variance 4. The regression parameters have four different values with $\beta_i = \lfloor (i-1)/15 \rfloor$, making 45 of the 60 features active. Figure 3 displays the same boxplots values as before, under the null and with a signal. The parameters of the genetic algorithm were set to $I = 60$, $G = 15$, $U = 100$, $\pi_0 = 0.9$ and $\theta_g = 1/D$ for all g.

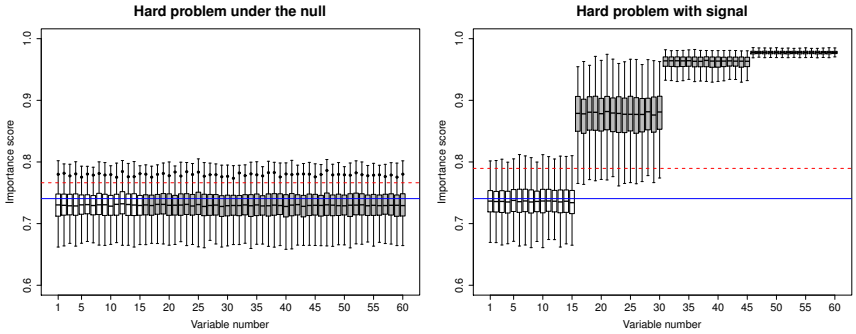

Figure 3: Importance scores obtained from the BGA of 1000 different samples generated from the same "Hard problem" scenario. The left panel shows the importance scores under the null hypothesis, where Y is independent from the features. The right panel shows the importance score when $\beta_i = \lfloor (i-1)/15 \rfloor$. While the plain line shows the expected importance π_G, the dashed line displays the threshold $\pi_d + z_{0.95}\sigma_{\hat{\pi}}/\sqrt{U}$ on the left panel, and its Šidák adjusted equivalent on the right panel. The plain dots show the empirical 95^{th} quantile of the importance scores for each variable to compare against that theoretical bound. The boxplot of the active features are coloured in gray.

Under the null, the values of the 95^{th} quantile and their expected value match fairly well as do the expected scores. On the right panel, the 45 active features are found all the time with very few exceptions of the smallest signals being very occasionnally left out – less than 4% of the time. The inactive variables sometimes get picked up, but not more often than under the null, with a false positive rate of about 3% when using the Šidák corrected level. These results are as good as Figure 3 of Zhu and Chipman (2006). Interestingly, those figures are obtained with large π_0 and smaller values show excellent performances that are however less stellar for variables 1 to 30 who see their rates of errors increases. The ideal π_0 seems to be linked with the true number of active features – if the initial population has too few or too many active features, the genetic algorithm is more likely to miss its target as the required genes take longer to become avaialbel, if they do. Note that we ran the PGA of Zhu and Chipman (2006), with their crossover and selection mechanism, and we obtained similar peformances for all values of π_0 we tried. Intuitively, their choice of pointwise crossover could be thought to help detect the unactive features that all come first, but it did not seem to be a driving factor.

4.2 Pollution data

Let us consider the pollution data used in Example 4.1 of Zhu and Chipman (2006) and based on data used by Miller (2002) but initially published by McDonald and Schwing (1973). Let us assume that the three-variable model suggested by Figure 1 is correct and use the estimates of regression with these three variables as the ground truth. The same genetic parameters are used as for Figure 1, namely $I = 50$, $G = 20$, $U = 75$, $\pi_0 = .3$ and $\theta_g = 1/D$ for all g. Figure 4 displays the boxplot of the importance scores of each features obtained from 1000 datasets with the original values of the covariates and a new Y generated from a linear regression model with all regression coefficients equal 0 except for $\beta_0 = 796.5$, $\beta_1 = 2.347$, $\beta_9 = 2.961$, and $\beta_{14} = 0.3911$. The standard deviation of the error term is set to 38.58. Those parameters were determined from a fit on the whole original dataset, then deemed true values for the simulation. The lines and solid points have the same meaning as Figure 2 presented in Section 4.1. The left panel under the null is obtained by randomly permuting the values of Y.

The theoretical mean under the null matches the empirical median closely, but contrarily to the illustrative example, we observe discrepancies between the theoretical and empirical values of the 95^{th} quantiles. While the correlation between the features and the presence of some outliers may play a role, an investigation led us to conclude that the small sample size was the real culprit here. We first ran the same simulation, but with features simulated as multivariate normal with the same mean and covariance as the original data. The ensuing plot was qualitatively identical to Figure 4. However, with a sample size of 300 instead of 60, the empirical 95^{th} quantiles are aligned on their theoretical value similarly as Figure 2. Digging further, we observed that the small sample size caused even smaller training and validation sets that are a lot more

prone to spurious correlations, making the assumption of equal fitness fail strongly within some universes. Although the variables are equally likely to be favoured by chance, they are globally picked more often in their respective wrongly favourable universes, hence the increased importance. In cases where the sample is small, it may be advisable to consider a fitness function that does not rely on a validation set. The distributional results that we obtained depend on the approximate equality of the fitness under the null, not on the specific choice that we proposed.

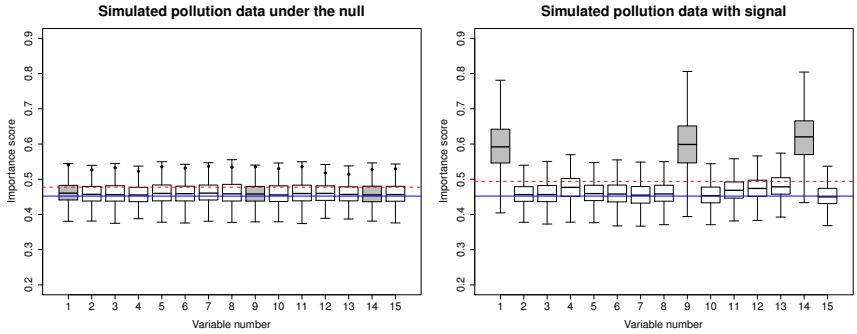

Figure 4: Importance scores obtained from the BGA of 1000 different samples generated from the same scenario based on the Pollution dataset. The left panel shows the importance scores under the null hypothesis, where Y is independent from the features. The right panel shows the importance score when the signal depends on covariates 1, 9 and 14. While the plain line shows the expected importance π_G, the dashed line displays the threshold $\pi_d + z_{0.95}\sigma_{\hat{\pi}}/\sqrt{U}$ on the left, and the Šidák adjusted equivalent on the right. The plain dots show the empirical 95^{th} quantile of the importance scores for each variable to compare against that theoretical bound. The boxplot of the active features are coloured in gray.

Looking at the right panel of Figure 4, we note that the discrepancies under the null did not affect notably the ability of BGA to detect the active features who are rightly selected over 90% of the time. Other variables however get wrongly picked between 12% and 37% of the time as a consequence of the artificially increased importance.

4.3 Wisconsin breast cancer dataset

Consider now the Wisconsin breast cancer dataset from Street et al. (1993) that we obtained from Dheeru and Karra Taniskidou (2017). A binary response variable indicating that the tumor is malignant (rather than bening) is explained with logistic regression using 30 features derived from medical imaging. The parameter α of BGA can be used to control the number of features that are selected. To simulate a simpler model, we choosee a conservative value of $\alpha = 0.005$. The other parameters of the BGA are $I = 20$, $G = 15$, $U = 50$, $\pi_0 = .3$ and $\theta_g = 1/D$ for all g. Figure 5 shows the bubble plot of this BGA with the theoretical mean importance and threshold as dotted and plain lines respectively. Variables 8 and 21 are selected: they are above the objective threshold, but we may also note that there is a gap between them and the following variable, so the subjective decision from a bubble plot would have been the same in this case.

To assess the ability of BGA to detect the active features in a realistic logistic regression setting, we use the original features of the Wisconsin breast cancer dataset, but we generate new responses that follow a logistic regression model with parameters $\beta_i = 0$, except for $\beta_0 = -18.38$, $\beta_8 = 75.25$ and $\beta_{21} = 0.8797$ that we obtained from a fit on the original dataset. We create 1000 simulated dataset reusing the same features, but generating a new response. We use the same genetic parameters as for Figure 5. The right panel of Figure 6 shows the boxplots of the importance scores obtained. The left panel was obtained under the null by randomly permuting the binary values in the simulated response, hence breaking any predictive power of the features.

The left panel shows that the importance scores behave as expected under the null, even in the case of a GLM. On the right panel, the two active features are the ones who get picked up most often, for 79% and 32% of the simulated repetitions. Variable 24 wrongly gets picked up 29% of the time, but its correlation is 0.83 with X_8 and 0.984 with X_{21}. All other features are wrongly selected less than 9% of the time even

though some have correlations exceeding 0.96 with the active features. In some way, this is a doubly hard problem: the covariates are correlated, and the model is a GLM. Yet, the BGA that we proposed shows a good performance.

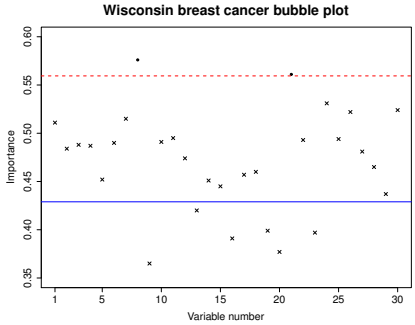

Figure 5: Bubble plot for the Wisconsin breast cancer dataset obtained from a BGA with parameters $I = 20$, $G = 15$, $U = 50$, $\pi_0 = .3$ and $\theta_g = 1/D$ for all g. The level global level was set to $\alpha = 0.005$. The plain line shows π_G, the theoretical mean of the importance when no features are active. The dashed line shows the threshold that was derived to decide what features to retain. The retained factors are shown as a dot rather, the dismissed as a cross.

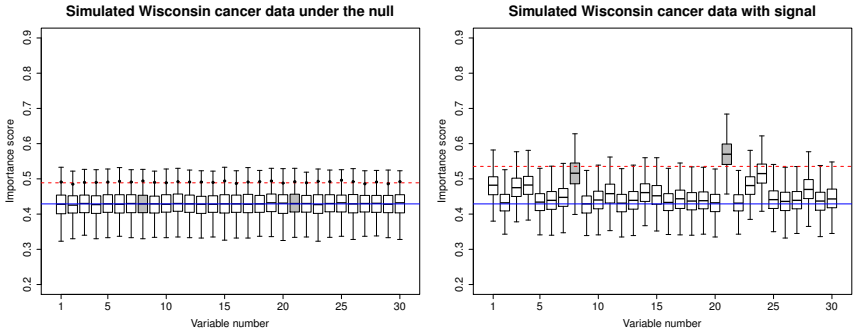

Figure 6: Importance scores obtained from the BGA of 1000 different samples generated from a scenario based on the Wisconsin breat cancer dataset. The left panel shows the importance scores under the null hypothesis, where the malingness of the tumor is independent from the features. The right panel shows the importance score when the signal depends on covariates 8 and 21. While the plain line shows the expected importance π_G, the dashed line displays the threshold $\pi_d + z_{0.95}\sigma_{\hat{\pi}}/\sqrt{U}$ on the left, and the Šidák adjusted equivalent on the right. The plain dots show the empirical 95^{th} quantile of the importance scores for each variable to compare against that theoretical bound.

5 Tuning of parameters

Genetic algorithms in general depend on many parameters that must be somewhat arbitrarily fixed and so does BGA. Experience helps in chosing appropriate values, as does occasional simulations to explore the behaviour of the method. There are however general guidelines that may help.

5.1 Fitness function

Under the null, the fitness function must yield approximately equal probability of selection to all possible models. The construction of Equation 2 was designed with this property in mind and the Monte Carlo studies in section 4 verified that the assumption is reasonnably valid in a realistic setting. A similar empirical endeavour could contribute to validating newly proposed fitness functions. As long as this property is preserved, the choice of fitness has no bearings on the distribution of selection ratios under the null hypothesis. Of course, the ability of the fitness to identify desirable models outside to the null hypothesis is also key to good performances.

The peakedness parameter (γ) highlights fitter individuals further, helping to speed up evolution. As such, it may help when computational resources are limited. A large γ may however highlight naturally occuring fortuitous fluctuations as well, so larger values are not systematically better. We mostly used values of γ in the neighborhood of 1.5, although we experimented with values as high as $\gamma = 4$ which reminded us of the ARCX4 algorithm of (Breiman, 1996a), but a fourth power did not have the same beneficial effect for the fitness.

5.2 Rates

The generation and evolution of populations depend on random elements that appear at some rates.

Activation rate (π_0): The values of π_0 has an effect on the expected values of the importance scores. Values close to 0 or 1 should be avoided as they lower the diversity of the initial population who then relies on mutation to explore numerous models. A rate of $\pi_0 = 0.5$ seems a good choice that maximizes that initial diversity, but in practice, best results are obtained when π_G is close to the true proportion of active features in the models. If prior information is known about the expected number of active features, it would make most sense to start π_0 just below that porportion when it is less than 0.5, and larger than the expected proportion otherwise. Since models are oftent expected to be sparse, smaller values of π_0 seem more appropriate.

Mutation rates (θ_g, $g \in \{1, \ldots, G\}$): Mutation breaks stagnation by infusing diversity in each generation. It helps exploring the space of all possible models, but also precludes the convergence of a population. While a lack of convergence is detrimental for single-thread genetic algorithms, it helps BGA who seeks diversity between the universes and look for a signal in the aggregation of all individuals. With a finite number of generations, a large mutation is approriate, but it should stay far below $\theta_g = 1/2$ who would make the next generation random, with no genetic memory. As Zhu and Chipman (2006) pointed out, many instances in the literature suggest $\theta_g = 1/D$, which we also found to work properly. Although we allow for a changing rate of mutation, those seem especially appropriate for single-thread genetic algorithm where diversity has value for the initial generations, but convergence eventually requires low mutation rates. Hence for BGA, we typically kept the mutation rate constant.

5.3 Number of individuals, universes and generations

The computational cost of a BGA depends highly on the number of individuals that will be generated, namely IGU.

Number of parallel universes (U): Universes are independent by design of the algorithm, and the accuracy of the selection ratios is improved by a larger number of universes. To justify the central limit theorem, using no less than 30 universes is advised, but that number could be far greater if the computational resources are available. Note also that the computation of the universes is embarassingly parallel, so it provides a trivial way to scale the algorithm on multi-core platforms.

Number of generations (G): A large number of generations may be detrimental as the covariance between two individuals (c_g) will tend to increase as g increases, yielding a larger threshold. On the other hand, too few generations might not give enough time to the genetic operators to detect the truly best models under the alternative when some features are good predictors. To complicate things, the speed of convergence to fit individuals also depends on the probability of activation and the mutations rates, especially if the mutation varies with g. Values of 10 to 15 seemed to work well for G.

Population size (I): Contrary to the intuition, larger population sizes do not yield better results. In a very large population, the probability of selection of an individual twice as fit as anybody else may be so diluted that his genes will not be passed on to the next generation. In a smaller population, his relative competitive advantage will be much larger. The population size (I) should thus be as small as possible, but not to the point of being detrimental to the exploration of the possible models. Zhu and Chipman (2006) suggested having a population size (I) of the same magnitude as the dimension of de data (D) and we found that guideline worked well, although $I < D$ also has merits when the number of dimensions gets large.

With the total computational cost of BGA being $\mathcal{O}(IGU)$, and G which is losely linked to D, it seems advisable to find a moderate value of I, then make U as large as possible.

6 Conclusion

The parallel genetic algorithm of Zhu and Chipman (2006) is an embarassingly parallel approach to feature selection. It computes importance scores for each variable from a summary of all models in a number of universes that are purposefully not fully evolved. We derive the distribution of the importance score when none of the predictors are active, which yields an objective criterion for variable selection. Simulations show that the distributional results hold well under the null hypothesis, both for linear regression and GLM. Telling apart which variables are active or not when the predictors are higly correlated is more challenging, as inactive features may have a proxy effect – they are a good alternatives when an active feature is missing in the model – but BGA performs well in those circumstances as well. The distribution of the importance scores under the null do not depend on the actual fitness function. The out-of-sample RSS that we use proposed works well for linear regression and GLM. Better fitness functions are certainly possible and could be the topic of future research.

References

Breiman, L. (1996a). Arcing classifiers. Annals of Statistics 26, 801–849.

Breiman, L. (1996b). Bagging predictors. Machine learning 24(2), 123–140.

Cantú-Paz, E. (2000). Efficient and accurate parallel genetic algorithms, Volume 1. Springer.

Casella, G. and R. L. Berger (2002). Statistical inference, Volume 2. Duxbury Pacific Grove, CA.

Dheeru, D. and E. Karra Taniskidou (2017). UCI machine learning repository.

Draper, N. R. and H. Smith (2014). Applied regression analysis, Volume 326. John Wiley & Sons.

George, E. I. and R. E. McCulloch (1993). Variable selection via gibbs sampling. Journal of the American Statistical Association *88*(423), 881–889.

Haupt, R. L. and S. E. Haupt (2004). Practical genetic algorithms. John Wiley & Sons.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983). Optimization by simulated annealing. science 220(4598), 671–680.

McCullagh, P. and J. A. Nelder (1989). Generalized linear models, Volume 37. CRC press.

McDonald, G. C. and R. C. Schwing (1973). Instabilities of regression estimates relating air pollution to mortality. Technometrics 15(3), 463–481.

Miller, A. (2002). Subset selection in regression. Chapman and Hall/CRC.

Mitchell, M. (1998). An Introduction to Genetic Algorithms. MIT Press.

Poli, R., W. W. B. Langdon, N. F. McPhee, and J. R. Koza (2008). A field guide to genetic programming. Lulu.com.

Shonkwiler, R. W. and F. Mendivil (2009). Explorations in Monte Carlo Methods. Springer Science & Business Media.

Šidák, Z. (1967). Rectangular confidence regions for the means of multivariate normal distributions. Journal of the American Statistical Association 62(318), 626–633.

Street, W. N., W. H. Wolberg, and O. L. Mangasarian (1993). Nuclear feature extraction for breast tumor diagnosis. In Biomedical Image Processing and Biomedical Visualization, Volume 1905, pp. 861–871. International Society for Optics and Photonics.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) 58(1), 267–288.

Wang, G.-W. and C.-X. Zhang (2015). Building variable selection ensembles for linear regression models by adding noise. In Machine Learning and Cybernetics (ICMLC), 2015 International Conference on, Volume 2, pp. 554–559. IEEE.

Zhang, C.-X., G.-W. Wang, and J.-M. Liu (2015). Randga: injecting randomness into parallel genetic algorithm for variable selection. Journal of Applied Statistics 42(3), 630–647.

Zhu, M. and H. A. Chipman (2006). Darwinian evolution in parallel universes: a parallel genetic algorithm for variable selection. Technometrics 48(4), 491–502.