# Econometric Genetic Programming in Binary Classification: Evolving Logistic Regressions Through Genetic Programming

André Luiz Farias Novaes[1](✉), Ricardo Tanscheit[2],
and Douglas Mota Dias[3]

[1] Informatics Department, University of Lisbon, Lisbon, Portugal
andrelfnovaes@gmail.com
[2] Electrical Engineering Department, PUC-Rio, Rio de Janeiro, RJ, Brazil
ricardo@ele.puc-rio.br
[3] Electronics and Telecommunications Engineering Department,
UERJ, Rio de Janeiro, RJ, Brazil
douglasmota.uerj@gmail.com

**Abstract.** Logistic Regression and Genetic Programming (GP) have already been compared to each other in classification tasks. In this paper, Econometric Genetic Programming (EGP), first introduced as a regression methodology, is extended to binary classification tasks and evolves logistic regressions through GP, aiming to generate high accuracy classifications with potential interpretability of parameters, while uses statistical significance as a feature-selection tool and GP for model selection. EGP-Classification (or EGP-C), the name of this proposed EGP's extension, was tested against a large group of algorithms in three cross-sectional datasets, showing competitive results in most of them. EGP-C successfully competed against highly non-linear algorithms, like Support Vector Machines and Multilayer Perceptron with Back Propagation, and still allows interpretability of parameters and models generated.

**Keywords:** Genetic programming · Binary classification · Logistic regression · Model selection

## 1 Introduction

Logistic Regression (LR or logit regression) and Genetic Programming (GP) have already been compared to each other in classification tasks [1–3].

The work on [4] is pioneer in evolving LR models through GP. As they state, their approach merges the ability of LR to deal with dichotomous data and provide quantitative results with the optimization characteristic of GP to search the entire hypothesis space for the "most fit" hypothesis. GP modifies, using an iterative trial and error process, LR models formed by vegetation indices built from basic function blocks defined in the function and terminal sets. Each candidate model is refined with a stepwise backward elimination using the level of significance associated with Chi-square test of each term and then evaluated based on the fitness function which is

E. Oliveira et al. (Eds.): EPIA 2017, LNAI 10423, pp. 382–394, 2017.
DOI: 10.1007/978-3-319-65340-2_32

defined by the model's Kappa statistics and the number of terms in the model. Figure 1 shows a possible individual generated by its algorithm.

Kaizen Programming (KP) [18] is an interesting evolutionary tool based on concepts of continuous improvement from Kaizen methodology, which was successfully tested against traditional SR benchmark functions. In [19], KP was coupled with LR models to extract useful features from a widely studied credit scoring dataset, aiming at improving the prediction performance of LR.

EGP, which was first introduced in [5] for regression tasks and tested against traditional feature-selection econometric algorithms, is carefully constructed considering econometric theory on cross-sectional datasets, aiming to generate high accuracy regressions with potential interpretability of parameters.

EGP is now extended to binary classification tasks, evolving logistic regressions through GP, aiming to improve the approach proposed by its predecessors, [4, 19], particularly on interpretation of parameters. Predictors 1, 2 and 3, in Fig. 1, when components in a logit model, offer just a few or even none interpretation of parameters. To perceive this, it is sufficient to try an interpretation on $(\mathrm{B}_5\mathrm{B}_3)/(\mathrm{B}_3 + \mathrm{B}_1)$, a coefficient in a LR model of [4]. EGP-C uses just polynomials in the $X\beta$ part of LR and to see why this kind of approach is beneficial to parameter interpretation, see [6]. EGP-C is interpretation-oriented and also aims to generate high accuracy models.

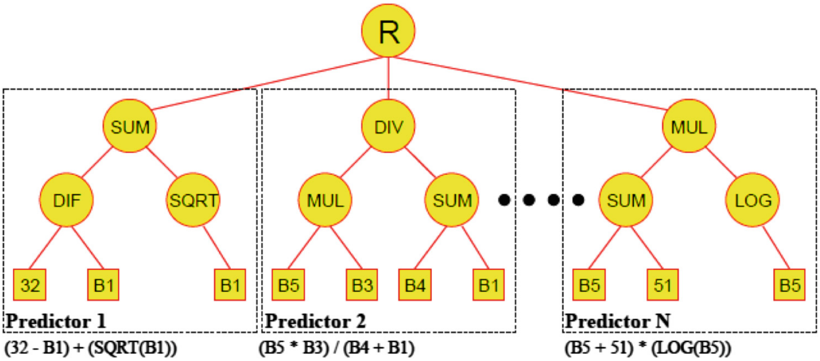

**Fig. 1.** Example of a proposed candidate model representation for the GP and LR integrated model. (Source: [4])

This paper is organized as follows: Sect. 2 describes the elements of econometrics used by EGP-C: there is no intention to fully exhaust the theme; justification on these elements is presented when necessary. Section 3 succinctly describes EGP-C. Sections 4 proposes experiments and discuss results. Conclusion is done in Sect. 5, with mention to future work.

# 2 Econometrics

## 2.1 Logistic Regression, Maximum Likelihood, Newton's Method

The Logistic Regression (LR) aims to model $P_t$, the conditional probability of $y_t = 1$ to $\boldsymbol{X}$, with $t \in [1, n]$. Only possible outcomes for $y_t$ are 0 or 1. The matrix $\boldsymbol{X} = [\boldsymbol{x}_1 \ldots \boldsymbol{x}_i \ldots \boldsymbol{x}_k]$ is constructed by $\boldsymbol{x}_i$ $n \times 1$ vectors, $i \in [1, k]$. Mathematically:

$$P_t \equiv \Pr(y_t = 1|\boldsymbol{X}) = E(y_t|\boldsymbol{X}) \tag{1}$$

Multiple linear regressions are inadequate to model (1) and [7] shows the reason. Logit or probit models [8] are well recognized methods for binary classification tasks. Both methods consist of modelling $E(y_t|\boldsymbol{X})$ with a transformation function, $F(x)$, applied to an index function, $h(\boldsymbol{X}_t, \boldsymbol{\beta})$:

$$E(y_t|\boldsymbol{X}) = F(h(\boldsymbol{X}_t, \boldsymbol{\beta})) = F(\boldsymbol{X}_t\boldsymbol{\beta}) \tag{2}$$

with $h(\boldsymbol{X}_t, \boldsymbol{\beta}) = \boldsymbol{X}_t\boldsymbol{\beta}$. The expected value $E(y_t|\boldsymbol{X})$ is a typical cumulative probability distribution, a monotonically growing linear transformation that maps from the real line to $[0, 1]$, with properties $F(-\infty) = 0$, $F(\infty) = 1$ and $(\partial F(x)/\partial x) > 0$.

Logit and probit models are usually preferred over other econometric classification models mainly due linearity on $h(\boldsymbol{X}_t, \boldsymbol{\beta})$. For probit regressions, $E(y_t|\boldsymbol{X}) = \phi(\boldsymbol{X}_t\boldsymbol{\beta})$, the cumulative normal probability distribution, which has not closed formula but is easily calculated numerically. For the logit regression:

$$E(y_t|\boldsymbol{X}) = \Lambda(\boldsymbol{X}_t\boldsymbol{\beta}) = \frac{e^{\boldsymbol{X}_t\boldsymbol{\beta}}}{1 + e^{\boldsymbol{X}_t\boldsymbol{\beta}}} \tag{3}$$

which has closed formula. $\Lambda(\boldsymbol{X}_t\boldsymbol{\beta})$ is called logistic function.

Maximum Likelihood (ML) is commonly used to estimate $\widehat{\boldsymbol{\beta}}$ on (3) [9]. ML estimation proposes the maximization of the ML function, which gives the likelihood of the sample $\boldsymbol{y}$ to be observed as realizations of $n$ independent Bernoulli random variables. The vector $\widehat{\boldsymbol{\beta}}$ is the solution of this maximization, which usually occurs on the logarithm of ML function, because it involves a sum instead of a product:

$$l(\boldsymbol{y}, \boldsymbol{\beta}) = \sum\nolimits_{t=1}^{n} [y_t \log(\Lambda(\boldsymbol{X}_t\boldsymbol{\beta})) + (1 - y_t)\log(1 - \Lambda(\boldsymbol{X}_t\boldsymbol{\beta}))] \tag{4}$$

which is globally concave whenever $\log(\Lambda(\boldsymbol{X}_t\boldsymbol{\beta}))$ and $\log(1 - \Lambda(\boldsymbol{X}_t\boldsymbol{\beta}))$ are concave functions of $\boldsymbol{X}_t$: in such case, $\widehat{\boldsymbol{\beta}}$ is unique. However, [10] states that the presence of non-linear elements, crossed feature terms (like $x_3 x_{11}^2$), will not permit oneness of $\widehat{\boldsymbol{\beta}}$. First order conditions of (4) are:

$$\sum\nolimits_{t=1}^{n} (y_t - \Lambda\left(\boldsymbol{X}_t\widehat{\boldsymbol{\beta}}\right))X_{ti} \tag{5}$$

Conditions in (5) are just solved numerically, due non-linearity in parameters $\widehat{\boldsymbol{\beta}}$, and Newton's Method (NM) is an interactive method that possibly solves it, performing as follows:

$$\boldsymbol{\beta}^{(s+1)} = \boldsymbol{\beta}^{(s)} - \boldsymbol{H}^{-1}\left(\boldsymbol{\beta}^{(s)}\right)\nabla l\left(\boldsymbol{\beta}^{(s)}\right) \tag{6}$$

with $\boldsymbol{H}$ the Hessian Matrix and $\nabla l$ the gradient of $l(\boldsymbol{\beta})$. Even if there is no global maximum for $l(\boldsymbol{\beta})$, [11] guarantees it always increases by NM.

### 2.2 Hypothesis Test

Hypothesis Test (HT) is applied in EGP-C in the same way it is applied in EGP and it is only possible due satisfiability of three regularity conditions, as described in [12]. Under these conditions and $n$ sufficiently large, the following verifies:

$$\widehat{\boldsymbol{\beta}} \xrightarrow{d} N\left(\boldsymbol{\beta}, [I(\boldsymbol{\beta})]^{-1}\right) \tag{7}$$

$$\frac{\widehat{\beta}_i - \beta_i}{SE\left(\widehat{\beta}_i\right)} \sim N(0,1) \tag{8}$$

with $I(\boldsymbol{\beta})$, the Fisher Information, equals to $\boldsymbol{\sigma}^2_{\boldsymbol{\beta}}$, the asymptotic variance. For a brief review on HT and how it is applied in EGP, see [5].

## 3 Econometric Genetic Programming – Classification: EGP-C

EGP-C is the EGP algorithm applied to classifications tasks, when logit models are evolved. The main difference between EGP and EGP-C lies in Accuracy and related metrics, showed in Sect. 3.3.

EGP-C evolves models in format of (3) through GP, which is responsible for model selection. GP is mainly based in [13] configuration.

### 3.1 Representation

Individuals are multigenic. Any constant in any program comes from NM in (6), i.e. there are no ephemeral constants. The terminal set, namely $\Omega$, is purely composed by variables. The primitive set, namely $\vartheta$, is composed just by variables and operations of sum and multiplication, due $X\beta$ format.

Search space for EGP-C is the number of models, $n_{mod}$, which is function of the number of regressors created for an individual, $n_{reg}$.

$$n_{reg} = \sum\nolimits_{q_{var}=1}^{K} \frac{(K - 1 + q_{var})!}{(K - 1)!q_{var}!} \tag{9}$$

$$n_{mod} = \sum\nolimits_{q_{reg}=1}^{n_{reg}} \frac{n_{reg}!}{\left(n_{reg} - q_{reg}\right)!} \tag{10}$$

In (9), $q_{var}$ is the number of variables on Ω necessary to create a regressor; in (10), $q_{reg}$ is the number of regressors required to build a model. (9) is the sum of possible combinations with repetitions, $q_{var}$ to $q_{var}$. (10) is the sum of possible arrangements of $n_{reg}$ regressors, $q_{reg}$ to $q_{reg}$. Supposing $K = 3$ for a particular dataset, $n_{mod}$ rounds $10^{17}$.

### 3.2 Initial Population

EGP-C uses a probabilistic version of ramped half-and-half method. Figure 2 shows a possible individual generated by EGP-C.

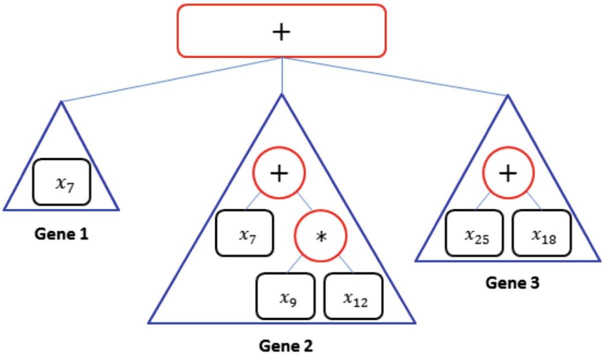

**Fig. 2.** A possible individual generated by EGP-C.

Set Ω is composed by $K$ features (independent variables). Every individual has its own set of regressors, forming its own $\boldsymbol{X}$, composed by simple or combined elements of Ω. As an example, it is possible that $x_1$, $x_3x_{11}^2$ and $x_3x_4x_6$ are regressors of a particular individual, formed by features $x_1$, $x_3$, $x_4$, $x_6$ and $x_{11}$.

### 3.3 Accuracy

In EGP, RMSE is the objective function and $\overline{\mathrm{R}}^2$ is used to compare models. In EGP-C, the percentage of correctly classified instances ("%_*corr*") has been chosen as objective function (accuracy measure), due benchmarks were evaluated using such metric. Experiments and Results will fully describe the comparison methodology.

To calculate accuracy in an EGP-C individual, the following procedure needs to be done (Fig. 3).

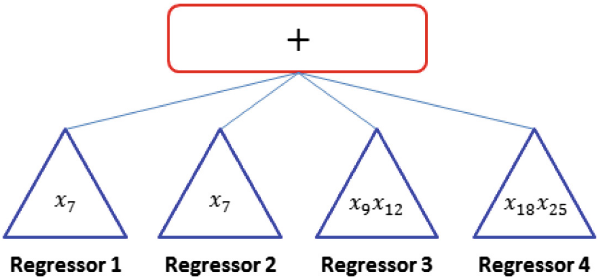

**Fig. 3.** The multigenic individual is written as a set of regressors. Repetitions will be discarded.

EGP-C will solve (5) for $X\beta = \beta_1 x_7 + \beta_2 x_9 x_{12} + \beta_3 x_{18} x_{25}$. If any of the regressors are not statistically significant, they will be removed from (5). In sequence, (6) is recalculated just with statistically significant regressors from (5). $\%\_corr$ is finally calculated using $\widehat{\boldsymbol{\beta}}$ after these steps. This routine is traditional in econometric studies, ensuring statistical significance over a determined significance level $\alpha$. Modifications described are necessary just for accuracy calculation, therefore individuals will keep their multigene structure to mutation, crossover and elitism (Fig. 4).

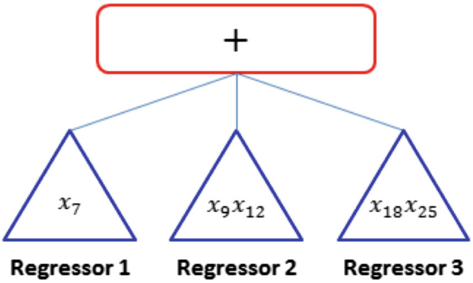

**Fig. 4.** Individual ready for accuracy calculation.

EGP-C does not estimate on genes, just on regressors, by two main reasons: possible multicollinearity problem, interfering on HT for $\beta_i$, and lack of interpretation for $\widehat{\beta}_i$ when it is related to a gene.

### 3.4 Selection

Tournament selection with $n_{tourn} = 7$ and repetitions allowed, with a variation on lexicographic parsimony pressure of [14], is used. Individuals with a large number of statistically significant regressors will be preferred over others with a few number, in the same range of fitness. Therefore, EGP-C is parsimonious in its nature, because it avoids the individuals with a large amount of *introns* (in this case, non statistically significant regressors).

### 3.5 Mutation, Crossover and Elitism

Types of mutation used: traditional mutation proposed by [15] and mutation by regressors' substitution. Types of crossover used: intergenic and intragenic crossovers. Mutation and crossover rates vary through evolution following automatic adaptation of operators as described in [16]. Elitism rate is set to 5% of individuals by generation.

### 3.6 Tools and Parameters

EGP-C is implemented through a modification on GPTIPS, a *Matlab* toolbox, presented in [17]. Information on EGP-C parameters are shown in Table 1.

**Table 1.** EGP-C Parameters

| Parameters | |
|---|---|
| - Population size | 150 |
| - Generations | 50 |
| - Maximum gene depth | 5 |
| - Maximum number of genes by individual | 5 |
| - Probability of traditional mutation [15] | 95%. |
| - Probability of intragenic crossover | 50%. |
| - Threshold for classification in LR | $E(y_t \| \boldsymbol{X}) = 0.5$ |

## 4 Experiments and Results

EGP-C can be used in different forms, e.g. for model selection or interpretation of parameters. For the reasoning of this article, EGP-C was submitted to generate high accuracy models, with potential interpretability of parameters, and tested against a large group of algorithms in three classification cross-sectional datasets, namely: "Breast Cancer **Wisconsin** (Original) Data Set"; "Pima Indians **Diabetes** Data Set"; "**Ionosphere** Data Set". All information on datasets can be found in UCI Machine Learning Repository [20].

To fully test EGP-C's capability to generate high accuracy models, a generous list of algorithms to compare it was required. The Computational Intelligence Laboratory in Informatics' Department of Nicolaus Copernicus University holds results on a list of algorithms for datasets used in this article. Their comparison methodology is based on a 10-fold cross validation on the entire dataset and that is the reason EGP-C is evaluated in the same manner. Authors of each algorithm are responsible for every result divulgated at Computational Intelligence Laboratory in Informatics' Department of Nicolaus Copernicus University domain [21].

Tables 2, 3 and 4 present the results. Results for EGP-C are identified as "EGP-Classification" in Tables. Algorithms are ordered by $\%\_corr$, the percentage of correct hits, while standard deviation works as the next sorting criteria.

**Table 2.** Results for Wisconsin Dataset

| Position | Algorithm | % of correct hits ± standard deviation | Reference |
|---|---|---|---|
| 1 | NB + kernel est | 97,5 ± 1,8 | WD, WEKA, 10×10CV |
| 2 | SVM (5xCV) | 97,2 | Bennet and Blue |
| 3 | kNN with DVDM distance | 97,1 | our (KG) |
| 4 | GM k-NN, k = 3, raw, Manh | 97,0 ± 2,1 | WD, 10×10CV |
| 5 | GM k-NN, k = opt, raw, Manh | 97,0 ± 1,7 | WD, 10CV only |
| 6 | VSS, 8 it/2 neurons | 96,9 ± 1,8 | WD/MK; 98.1% train |
| 7 | FSM-Feature Space Mapping | 96,9 ± 1,4 | RA/WD, a = .99 Gaussian |
| 8 | Fisher linear discr. anal | 96,8 | Ster, Dobnikar |
| 9 | MLP + BP | 96,7 | Ster, Dobnikar |
| 10 | MLP + BP (Tooldiag) | 96,6 | Rafał Adamczak |
| 11 | LVQ | 96,6 | Ster, Dobnikar |
| 12 | kNN, Euclidean/Manhattan f. | 96,6 | Ster, Dobnikar |
| 13 | SNB, semi-naive Bayes (pairwise dependent) | 96,6 | Ster, Dobnikar |
| **14** | **EGP-Classification** | **96,43 ± 2,88** | |
| 15 | SVM lin, opt C | 96,4 ± 1,2 | WD-GM, 16 missing with −10 |
| 16 | VSS, 8 it/1 neuron! | 96,4 ± 2,0 | WD/MK, train 98.0% |
| 17 | GM IncNet | 96,4 ± 2,1 | NJ/WD; FKF, max. 3 neurons |
| 18 | NB - naive Bayes (completly independent) | 96,4 | Ster, Dobnikar |
| 19 | SSV opt nodes, 3CV int | 96,3 ± 2,2 | WD/GM; training 96.6 ± 0.5 |
| 20 | IB1 | 96,3 ± 1,9 | Zarndt |
| 21 | DB-CART (decision tree) | 96,2 | Shang, Breiman |
| 22 | GM SSV Tree, opt nodes BFS | 96,0 ± 2,9 | WD/KG (beam search 94.0) |
| 23 | LDA - linear discriminant analysis | 96 | Ster, Dobnikar |
| 24 | OC1 DT (5xCV) | 95,9 | Bennet and Blue |
| 25 | RBF (Tooldiag) | 95,9 | Rafał Adamczak |
| 26 | GTO DT (5xCV) | 95,7 | Bennet and Blue |
| 27 | ASI - Assistant I tree | 95,6 | Ster, Dobnikar |
| 28 | MLP + BP (Weka) | 95,4 ± 0,2 | TW/WD |
| 29 | OCN2 | 95,2 ± 2,1 | Zarndt |
| 30 | IB3 | 95,0 ± 4,0 | Zarndt |
| 31 | MML tree | 94,8 ± 1,8 | Zarndt |

(*continued*)

**Table 2.** (*continued*)

| Position | Algorithm | % of correct hits ± standard deviation | Reference |
|---|---|---|---|
| 32 | ASR - Assistant R (RELIEF criterion) tree | 94,7 | Ster, Dobnikar |
| 33 | C4.5 tree | 94,7 ± 2,0 | Zarndt |
| 34 | LFC, Lookahead Feature Constr binary tree | 94,4 | Ster, Dobnikar |
| 35 | CART tree | 94,4 ± 2,4 | Zarndt |
| 36 | ID3 | 94,3 ± 2,6 | Zarndt |
| 37 | C4.5 (5xCV) | 93,4 | Bennet and Blue |
| 38 | C4.5 rules | 86,7 ± 5,9 | Zarndt |
| 39 | Default, majority | 65,5 | – |
| 40 | QDA - quadratic discr anal | 34,5 | Ster, Dobnikar |

**Table 3.** Results for Diabetes Dataset

| Position | Algorithm | % of correct hits ± standard deviation | Reference |
|---|---|---|---|
| 1 | Logdisc | 77,7 | Statlog |
| 2 | IncNet | 77,6 | Norbert Jankowski |
| 3 | DIPOL92 | 77,6 | Statlog |
| 4 | Linear Discr. Anal. | 77,5 − 77,2 | Statlog; Ster & Dobnikar |
| 5 | SVM, linear, C = 0.01 | 77,5 ± 4,2 | WD-GM, 10XCV averaged 10x |
| 6 | SVM, Gauss, C, sigma opt | 77,4 ± 4,3 | WD-GM, 10XCV averaged 10x |
| **7** | **EGP-Classification** | **76,95 ± 6,00** | |
| 8 | SMART | 76,8 | Statlog |
| 9 | GTO DT (5xCV) | 76,8 | Bennet and Blue |
| 10 | kNN, k = 23, Manh, raw, W | 76,7 ± 4,0 | WD-GM, feature weighting 3CV |
| 11 | kNN, k = 1:25, Manh, raw | 76,6 ± 3,4 | WD-GM, most cases k = 23 |
| 12 | ASI | 76,6 | Ster & Dobnikar |
| 13 | Fisher discr. analysis | 76,5 | Ster & Dobnikar |
| 14 | MLP + BP | 76,4 | Ster & Dobnikar |
| 15 | MLP + BP | 75,8 ± 6,2 | Zarndt |
| 16 | LVQ | 75,8 | Ster & Dobnikar |
| 17 | LFC | 75,8 | Ster & Dobnikar |

(*continued*)

**Table 3.** (*continued*)

| Position | Algorithm | % of correct hits ± standard deviation | Reference |
|---|---|---|---|
| 18 | RBF | 75,7 | Statlog |
| 19 | NB | 75,5 − 73,8 | Ster & Dobnikar; Statlog |
| 20 | kNN, k = 22, Manh | 75,5 | Karol Grudziński |
| 21 | MML | 75,5 ± 6,3 | Zarndt |
| 22 | FSM stand. 5 feat. | 75,4 ± 4,9 | WD, 10x10 test, CC > 0.15 |
| 23 | SNB | 75,4 | Ster & Dobnikar |
| 24 | BP | 75,2 | Statlog |
| 25 | SSV DT | 75,0 ± 3,6 | WD-GM, SSV BS, node 5CV MC |
| 26 | kNN, k = 18, Euclid, raw | 74,8 ± 4,8 | WD-GM |
| 27 | CART DT | 74,7 ± 5,4 | Zarndt |
| 28 | CART DT | 74,5 | Stalog |
| 29 | DB-CART | 74,4 | Shang & Breiman |
| 30 | ASR | 74,3 | Ster & Dobnikar |
| 31 | FSM standard | 74,1 ± 1,1 | WD, 10x10 test |
| 32 | ODT, dyadic trees | 74,0 ± 2,3 | Blanchard |
| 33 | Cluster means, 2 prototypes | 73,7 ± 3,7 | MB |
| 34 | SSV DT | 73,7 ± 4,7 | WD-GM, SSV BS, node 10CV strat |
| 35 | SFC, stacking filters | 73,3 ± 1,9 | Porter |
| 36 | C4.5 DT | 73 | Stalog |
| 37 | C4.5 DT | 72,7 ± 6,6 | Zarndt |
| 38 | Bayes | 72,2 ± 6,9 | Zarndt |
| 39 | C4.5 (5xCV) | 72 | Bennet and Blue |
| 40 | CART | 72,8 | Ster & Dobnikar |
| 41 | Kohonen | 72,7 | Statlog |
| 42 | C4.5 DT | 72,1 ± 2,6 | Blanchard (average in 100 runs) |
| 43 | kNN | 71,9 | Ster & Dobnikar |
| 44 | ID3 | 71,7 ± 6,6 | Zarndt |
| 45 | IB3 | 71,7 ± 5,0 | Zarndt |
| 46 | IB1 | 70,4 ± 6,2 | Zarndt |
| 47 | kNN, k = 1, Euclides, raw | 69,4 ± 4,4 | WD-GM |
| 48 | kNN | 67,6 | Statlog |
| 49 | C4.5 rules | 67,0 ± 2,9 | Zarndt |

**Table 4.** Results for Ionosphere Dataset

| Position | Algorithm | % of correct hits ± standard deviation | Reference |
|---|---|---|---|
| 1 | 3-NN + simplex | 98,7 | Our own weighted kNN |
| 2 | VSS 2 epochs | 96,7 | MLP with numerical gradient |
| 3 | 3-NN | 96,7 | KG, GM with or without weights |
| 4 | IB3 | 96,7 | Aha, 5 errors on test |
| 5 | 1-NN, Manhattan | 96 | GM kNN (our) |
| 6 | MLP + BP | 96 | Sigillito |
| 7 | SVM Gaussian | 94,9 ± 2,6 | GM (our), defaults, similar for C = 1 − 100 |
| 8 | C4.5 | 94,9 | Hamilton |
| 9 | 3-NN Canberra | 94,7 | GM kNN (our) |
| 10 | RIAC | 94,6 | Hamilton |
| 11 | C4 (no windowing) | 94 | Aha |
| 12 | C4.5 | 93,7 | Bennet and Blue |
| 13 | SVM | 93,2 | Bennet and Blue |
| 14 | Non-lin perceptron | 92 | Sigillito |
| 15 | FSM + rotation | 92,8 | our |
| 16 | 1-NN, Euclidean | 92,1 | Aha, GM kNN (our) |
| 17 | DB-CART | 91,3 | Shang, Breiman |
| 18 | Linear perceptron | 90,7 | Sigillito |
| 19 | OC1 DT | 89,5 | Bennet and Blue |
| 20 | CART | 88,9 | Shang, Breiman |
| 21 | SVM linear | 87,1 ± 3,9 | GM (our), defaults |
| **22** | **EGP-Classification** | **86,9 + 5,21** | |
| 23 | GTO DT | 86 | Bennet and Blue |

EGP-C was competitive in Wisconsin and Diabetes datasets, performing in 14th (40 algorithms in total) and 7th (49 in total), respectively. Support Vector Machines (SVMs) and Multilayer Perceptron with Back Propagation (MLP + BP), which are highly non-linear in structure, using more complex non-linear functions like trigonometric ones, presented results just a little better than EGP-C (in average, 0.5% above). Additionally, SVMs and MLP + BP permits low or none interpretability of parameters.

In logit models, the coefficient $\widehat{\beta}_i$ can be interpreted as the effect of a unit of change in $x_i$ on the predicted logits with other regressors considered constants in the model. (11) is a model generated by EGP-C.

$$\frac{e^{-7.12+0.14x_4-0.25x_6}}{1+e^{-7.12+0.14x_4-0.25x_6}} \tag{11}$$

with $X\beta = -7.12 + 0.14x_4 - 0.25x_6$. The effect on the odds of a 1-unit increase in $x_4$ is $e^{0.14} = 1.15$, meaning the odds of an instance to be classified as $y_t = 1$ increase approximately 15% when $x_4$ is added by one unit (of $x_4$), regardless of the $x_6$ value.

In Ionosphere dataset, EGP-C was not competitive. Ionosphere has a binary attribute with 89.17% of its values equals to 1 and the rest, 10.83%, equals 0. When combining features to generate regressors, this attribute (suppose it is $x_1$) can easily form a regressor that is collinear with other. For example, $x_1$ and $x_1^2$ composing the same individual. In such cases, $\boldsymbol{X}$ has not full rank and thus the variant of NM used to solve (5) will fail to find a maximum (even it is local) to (4), because this version of NM has not a protection against linear dependent columns. That is why EGP-C, in some circumstances like highly unbalanced datasets, is purposely set to do not generate too large individuals, consequently compromising accuracy. EGP-C controls individuals' growth by parsimoniously regulating GP parameters as population size, number of generations or maximum gene height for trees.

## 5 Conclusion and Future Work

EGP-C was successful in achieving its objective of generating high accuracy logit models. Although non-linear, logit models generated by EGP-C hold a linear portion on its structure, $X\beta$, which permits potential interpretability of parameters.

Future work points out in designing EGP for time series forecast. High collinearity between columns of $\boldsymbol{X}$ requires a distinct approach to prediction, depending on the model someone is interested in.

## References

1. Nourani, V., Pradhan, B., Ghaffari, H., Sharifi, H.: Landslide susceptibility mapping at Zonouz plain, Iran using genetic programming and comparison with frequency ratio, logistic regression, and artificial neural network models. Commun. J. Int. Soc. Prev. Mitig. Nat. Hazards **71**(1), 523–547 (2014)
2. Ritchie, M.D., Motsinger, A.A., Bush, W.S., Coffey, C.S., Moore, J.H.: Genetic programming neural networks: a powerful bioinformatics tool for human genetics. Commun. Appl. Soft Computing J. **7**(1), 471–479 (2007)
3. Ong, C.-S., Huang, J.-J., Tzeng, G.-H.: Building credit scoring models using genetic programming. Commun. Expert Syst. Appl.: Int. J. **29**(1), 41–47 (2005)
4. Momm, H.G., Easson, G., Kuszmaul, J.: Integration of logistic regression and genetic programming to model coastal Louisiana land loss using remote sensing. In: Proceedings of the American Society for Photogrammetry and Remote Sensing 2007 Annual Conference, ASPRS 2007, Tampa, FL, USA (2007)
5. Novaes, A.L.F., Tanscheit, R., Dias, D.M.: Programação Genética Econométrica Aplicada a Problemas de Regressão em Conjuntos de Dados Seccionais. In: Proceedings of XIII Encontro Nacional de Inteligência Artificial, ENIAC 2016, Recife, PE, Brazil (2016)
6. Wooldridge, J.: Introductory Econometrics: A Modern Approach, 4th edn. Cengage Learning, Boston (2009)

7. Novaes, A.L.F.: Programação Genética Econométrica: uma Nova Abordagem para Problemas de Regressão e Classificação em Conjuntos de Dados Seccionais. Master's thesis. Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brazil (2015)
8. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning, 1st edn. Springer New York Inc., New York (2001)
9. Davidson, R., MacKinnon, J.: Estimation and Inference in Econometrics, 1st edn. Oxford University Press, Oxford (1993)
10. Pratt, J.W.: Concavity of the log likelihood. J. Am. Stat. Assoc. **76**(1), 103–106 (1981)
11. Murray, W.: Newton-Type Methods. 1st edn. Stanford (2010)
12. Greene, W.H.: Econometric Analysis, 7th edn. Prentice Hall, Upper Saddle River (2011)
13. Poli, R., Langdon, W.B., McPhee, N.F.: A Field Guide to Genetic Programming, 1st edn. Lulu Enterprises, Raleigh (2008)
14. Luke, S., Panait, L.: Lexicographic parsimony pressure. In: Proceedings of the 2002 Conference on Genetic and Evolutionary Computation, GECCO 2002, pp. 829–836. ACM, San Francisco (2002)
15. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems), 1st edn. The MIT Press, Cambridge (1992)
16. Silva, S., Almeida, J.: Gplab – a genetic programming toolbox for matlab. In: Proceedings of the Nordic MATLAB Conference, pp. 273–278 (2003)
17. Searson, D.P., Leahy, D.E., Willis, M.J.: GPTIPS: an open source genetic programming toolbox for multigene symbolic regression. In: Proceedings of The International Multiconference of Engineers and Computer Scientists 2010, IMECS 2010, Hong Kong, pp. 77–80 (2010)
18. De Melo, V.V.: Kaizen programming. In: Proceedings of the 2014 Conference on Genetic and Evolutionary Computation, GECCO 2014, pp. 895–902. ACM, New York (2014)
19. De Melo, V.V., Banzhaf, W.: Improving logistic regression classification of credit approval with features constructed by Kaizen programming. In: Proceedings of the 2016 Conference on Genetic and Evolutionary Computation, GECCO 2016, pp. 61–62. ACM, New York (2016)
20. UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA. http://archive.ics.uci.edu/ml. Accessed 24 Feb 2015
21. Datasets used for classification: comparison of results. Nicolaus Copernicus University, Department of Informatics, Computational Intelligence Laboratory, Toruń, Poland. http://www.springer.com/lncs. Accessed 24 Feb 2015