A gene expression programming method for multi-target regression

Oscar Reyes
Department of Computer Science and Numerical Analysis, University of Córdoba, Spain
ogreyes@uco.es

Jose María Moyano
Department of Computer Science and Numerical Analysis, University of Córdoba, Spain
jmoyano@uco.es

José María Luna
Department of Computer Science and Numerical Analysis, University of Córdoba, Spain
jmluna@uco.es

Sebastián Ventura
Department of Computer Science and Numerical Analysis, University of Córdoba, Spain
sventura@uco.es

ABSTRACT

The study of problems that involve data examples associated with multiple targets at the same time has gained a lot of attention in the past few years. In this work, a method based on gene expression programming for the multi-target regression problem is proposed. This method solves the symbolic regression problem for multi-target contexts, allowing the construction of a model, without previous knowledge of any of its elements, that fits a set of cases. Our proposal directly handles the multi-target data, encoding the individuals with a chromosome of several genes, where each gene constructs a mathematical expression that is related to a target variable. The operators used into the evolutionary process enable the constant creation of new genetic material, and some of them may favour the detection of the existing dependencies between target variables. The experimental stage showed the benefits of the gene expression programming paradigm to solve the multi-target regression problem.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning by regression**; **Genetic algorithms**; **Genetic programming**;

KEYWORDS

Gene expression programming, Multi-target regression, Symbolic regression

ACM Reference Format:
Oscar Reyes, Jose María Moyano, José María Luna, and Sebastián Ventura. 2018. A gene expression programming method for multi-target regression. In *LOPAL '18: International Conference on Learning and Optimization Algorithms: Theory and Applications, May 2–5, 2018, Rabat, Morocco.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3230905.3230910

LOPAL '18, May 2–5, 2018, Rabat, Morocco

ACM ISBN 978-1-4503-5304-5/18/05…$15.00
https://doi.org/10.1145/3230905.3230910

1 INTRODUCTION

In the last decade, the study of problems where data examples are associated with multiple targets at the same time has gained the attention of the machine learning community, due to the numerous real-world applications that fit into this framework. Multi-target regression concerns the task of predicting multiple continuous target variables using a common set of input variables [3, 18]. Particular applications involving multi-target regression include ecological modeling [8], energy efficiency [20], signal processing [21] and more.

Up to date, a large number of methods have been proposed to solve the multi-target regression problem. Multi-target regression methods can be categorized into two groups: *problem transformation methods* and *algorithm adaptation methods* [3]. Problem transformation methods transform a multi-target regression problem into several single-target regression problems. Every single-target task is solved, and finally, an aggregation strategy is conducted to obtain the final predictions. Motivated by the tight connection between the multi-target regression problem and multi-label classification one, recent works have applied various transformations methods, which were originally designed for multi-label learning[1] [16], to solve the multi-target regression problem. Spyromitros-Xioufis et al. [18] analysed how several multi-label approaches, such as the binary relevance, stacked generalization and classifier chains, are straightforward of applying in multi-target regression. As for algorithm adaptation category, it comprises the algorithms designed to directly handle the multi-target data, i.e. the multi-target regression problem is not decomposed into several single-target regression tasks. In this category, a large number of methods have been proposed, such as statistical methods [17], support vector machines [12], kernel-based approaches [2], regression trees [8], rule-based algorithms [1], and locally weighted regression methods [15].

A well-known type of regression analysis is the symbolic regression, where a mathematical expression that best fits a given dataset is searched. No particular model is provided as a starting point, and expressions are formed by randomly combining mathematical building blocks. Symbolic regression techniques have been extensively applied to numerous real-world applications [19], and the evolutionary algorithms have been widely used to study this type of regression analysis [9, 10, 14]. One main evolutionary paradigm for

[1] In multi-label learning, the target variables are binary (a.k.a. labels).

studying the symbolic regression is Gene Expression Programming (hereafter, referred as GEP). GEP is a variant of genetic programming for the automatic generation of computer programs [5, 23]. GEP uses a population of individuals, selects parents according to fitness and evolves the population using genetic operators. However, the main difference between GEP, genetic programming and genetic algorithms resides in the representation of the individuals. In GEP, the individuals are encoded as linear strings of fixed length (like genetic algorithms) which are afterwards expressed as trees of different sizes and shapes (like genetic programming) [5]. GEP has been widely used in many applications, including classification problems [24], time series predictions [25], and others.

GEP has been effectively employed to study the symbolic regression problem [4, 5, 11, 14], but the existing works are restricted to the regression problem having a single target variable. To the best of our knowledge, the study of GEP for symbolic regression in multi-target contexts has not been studied yet. Additional difficulties appear when studying the multi-target regression problem, such as the statistical dependency that may exist between target variables; several recent studies have shown that it is important to correctly detect and exploit these dependencies to boost the predictive performance of regression algorithms [12, 13, 15, 18].

In this work, a method based on GEP is proposed to solve the multi-target regression problem. The proposed method solves the symbolic regression problem for multi-target data. Our proposal does not break down the multi-target problem into several single-target tasks. Instead, it handles directly the multi-target data, so a method with an acceptable computational cost is obtained. The method encodes the chromosome of an individual by a multigenic representation, where each gene represents a mathematical expression that is used to predict the value of a target variable. The creative power of GEP allows the constant creation of new genetic material, enabling a better exploration of the search space. The method may detect the existing dependencies between target variables by means of the transposition operators or the gene recombinations between chromosomes.

The main contribution of the present work is the introduction of a GEP-based method for the multi-target regression problem. This work corresponds to an initial study, where we want to analyse the benefits of GEP paradigm for constructing multi-target regression models. An experimental study was carried out on eight datasets, and the results showed that our proposal obtained promising predictive performances.

The rest of this paper is arranged as follows: Section 2 describes the basis of our GEP-based method. Section 3 describes the experimental set-up and discusses the results. Finally, Section 4 provides some concluding remarks.

2 A GEP-BASED METHOD

Let us say $S = \{(\mathbf{x}^1, \mathbf{y}^1), (\mathbf{x}^2, \mathbf{y}^2), \ldots, (\mathbf{x}^n, \mathbf{y}^n)\}$ represents a dataset of n training instances. An instance $i \in S$ is represented as a tuple $(\mathbf{x}^i, \mathbf{y}^i)$, where $\mathbf{x}^i \in \mathcal{X}$ and $\mathbf{y}^i \in \mathcal{Y}$ are the input and target vectors of i, respectively. $\mathcal{X}$ represents the input space[2], which contains d input variables $x_1, x_2 \ldots, x_d$, and $\mathcal{Y}$ is the output space[3], which comprises q target variables $y_1, y_2 \ldots, y_q$. On the other hand, x_ℓ^i denotes the value of the ℓ-th input variable for the instance i, whereas y_ℓ^i represents the value of the ℓ-th target variable. Also, $\hat{\mathbf{y}}$ represents the predicted target vector for an unseen input vector $\mathbf{x}$.

In this work, we proposed a method to solve the multi-target regression problem by using the symbolic regression analysis. Next, we described our GEP-based method to perform a symbolic regression in multi-target data.

Let F={$-, +, /, *, \ldots$} be a set of non-terminal symbols (functions), and T={$x_1, x_2, \ldots, x_d$} is the set of terminal symbols, where each terminal symbol represent an input variable $x_j \in \mathcal{X}$.

In GEP paradigm, a gene is composed by a head and tail. The head can contain symbols that belong to the sets F and T, whereas the tail contains only terminal symbols. The length of the head (h) is chosen according to the problem, whereas the length of the tail (t) is calculated as t=$h(a-1)+1$, where a is the number of arguments of the function with maximum arity in F. Figure 1 represents a gene of a chromosome. In this case, $h = 10$ and $t = 11$, and the expression of the gene (phenotype) is an expression tree (ET). For each problem, the length of the head (h) may be chosen, and it determines the size of the resulting ETs.

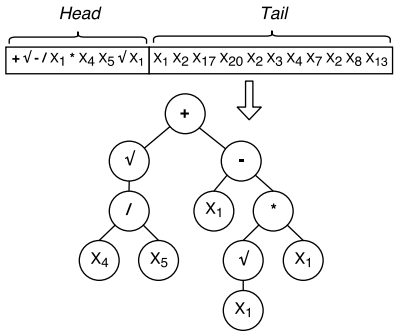

Figure 1: A GEP gene and its corresponding phenotype viewed as an ET.

Let us say we have a population with p individuals, each one coded by a multi-genic chromosome, i.e. a chromosome composed of more than one gene of equal length. A chromosome comprises q genes, $c = [g_1, g_1, \ldots, g_q]$, one gene for each target variable of the multi-target problem. Therefore, the ℓ-th gene of a chromosome codes a mathematical function that can be used posteriorly as a predictive model to estimate the value of the target variable y_ℓ^i of a test instance i. This form of codification allows that each individual of the population represents a complete solution of the multi-target problem.

One drawback of most GEP-based methods resides in the continuous transformation of the chromosomes into ETs and traversing them to obtain the fitness of the individuals. To avoid this, we used prefix notation, following the approach proposed by Peng et al. [14], which allows the evaluation of individuals without constructing ETs, and improves the computational efficiency of GEP.

The fitness function used to evaluate the adaptation level of an individual computes the *average relative root mean square error* (aRRMSE) over the training set S,

$$\frac{1}{q}\sum_{\ell=1}^{q}\sqrt{\frac{\sum_{i\in S}(y_\ell^i-\hat{y}_\ell^i)^2}{\sum_{i\in S}(y_\ell^i-\bar{y}_\ell)^2}}\,, \tag{1}$$

[2]The domain of the input variables can be continuous, discrete or mixed type.
[3]The domain of the target variables is continuous.

where y_ℓ^i and $\hat{y}_\ell^i$ are the values of the ℓ-th target variable in the known ($\mathbf{y}^i$) and predicted ($\hat{\mathbf{y}}^i$) target vectors of the instance i, respectively. On the other hand, $\bar{y}_\ell$ is the mean of the ℓ-th target variable in the set of training instances S. Summarizing, given an individual of the population, the fitness function measures the root mean squared error (RMSE) for the first target variable, evaluating the mathematical function coded into the first gene of the chromosome on each training instance $i \in S$, and afterwards, RMSE is divided by the RMSE of predicting the average value of that target in the training set. This same calculus is made for the second target variable, third target variable, and so on, and afterwards, the values obtained along all target variables are averaged.

The initial population was created randomly, but controlling the diversity between individuals. A chromosome is formed by creating each of its genes as follows: (I) the head is created by randomly picking symbols from the sets F and T, but assuring that the first element (root) of the head is a non-terminal symbol, and (II) the tail is created by randomly selecting terminal symbols from the set T. An individual is added to the initial population if it has a similarity respect to the rest of the population lower than a specific threshold. The similarity between two individuals is computed as

$$s(c^i, c^j) = 1 - \frac{\sum_{\ell=1}^{q} h(c_\ell^i, c_\ell^j)}{q^2(t+h)}, \tag{2}$$

where c^i and c^j are the chromosomes of the individuals i and j, respectively. The function $h(c_\ell^i, c_\ell^j)$ measures the number of distinct symbols, summed along all of the $t+h$ positions, in the ℓ-th gene of the chromosomes c^i and c^j. This method enables the creation of an initial population with enough diversity, which can avoid the early convergence of the method to local minima.

A binary tournament selection was used to create the intermediate population of parents. We employed a low selection pressure for not only favour the better individuals in the selection process, and therefore, to stimulate the diversity between the individuals selected.

The parents were mutated with a mutation rate p_m. We employed various mutation points per chromosome (n_{mp}), and the mutation operator performs the following steps for each mutation point: (I) a gene is randomly selected, (II) a position into the gene is randomly picked, (III) if the position belongs to the head, then the symbol in the position is changed by a symbol that belongs to the sets F or T (the symbols are randomly selected), except in the case that the position selected is the root, that in this case the replacement can be only done with a symbol of F, (IV) if the position belongs to the tail, then the symbol in the position selected is changed by other symbol of T. This mutation operator always obtains valid individuals.

A feature of GEP that differentiates it from other evolutionary paradigms is that fragments of the genome can be activated and jump to another place in the chromosome (transposable elements). The insertion sequence transposition operator (IS operator) was used with a low probability rate p_{is}, and it enables that short fragments with a function or terminal in the first position transpose to the head of genes, except to the root position in the head. This operator performs the following steps: (I) given a chromosome, a gene is selected as source from which a fragment with a length between $[1, l_{is}]$ (l_{is}-maximum length of a fragment) is randomly picked, and (II) a gene is selected as target and the fragment is copied downstream starting at a random position (distinct to the root position) of the gene head. During transposition, the sequence upstream from the insertion site stays unchanged, whereas the sequence downstream from the copied fragment loses, at the end of the gene head, as many symbols as the length of the fragment. This operator is performed over a chromosome n_{is} times.

The root IS transposition (RIS operator) was used with a low probability p_{ris}, and it allows that short fragments with a function in the first position transpose to the root of genes. This operator performs the following steps: (I) given a chromosome, a gene is randomly selected, a point is randomly picked in the head and the gene is scanned downstream until a function symbol is found, (II) this fragment is copied downstream starting at the root of the gene. During transposition, the whole head shifts to accommodate the fragment, losing a number of symbols of the head as many as the fragment length. As with IS operator, the tail of the gene subjected to transposition and all nearby genes stay unchanged. RIS operator is performed over a chromosome n_{ris} times.

The gene transposition (GT operator) was used with a low probability p_{gt}, and it allows that genes transpose to the beginning of chromosomes. This operator performs the following steps: (I) given a chromosome, a gene is randomly selected and deleted in the place of origin, and (II) the gene is copied to the beginning of the chromosome while the rest of genes are shifted to the right; this way, the length of the chromosome is maintained.

Despite the actions made by IS, RIS and GT operators, the structural organization of chromosomes is maintained, and therefore all created individuals are valid. Transposition operators can drastically reshape the expression of a gene, and the more upstream the insertion point the more profound the change [5]. Note that, mutation and transposition operators have a tremendous transforming power and they create genetic variation.

We also applied three kinds of recombination, one-point operator, two-points operator, and gene recombination operator, with probability rates equal to p_{op}, p_{tp} and p_{gp}, respectively. Given two chromosomes, the one-point operator crosses over a randomly chosen point. In the two-points recombination, two points of recombination are randomly chosen and the material between the recombination points is exchanged between the two parent chromosomes. In gene recombination, a gene position is randomly chosen and the two genes occupying this position in the parent chromosomes are exchanged. In all cases, two parent chromosomes are crossed and two new valid individuals are formed. Two-points operator has a greater transforming power than one-point recombination, and it is most useful to evolve solutions for more complex problems, especially when multigenic chromosomes are used. Gene recombination operator is unable to create new genes, since the individuals created are different arrangements of existing genes. However, it is worth to note that, this operator can introduce new material into the population, most of the time, since the exchanged genes may be very different.

Finally, we follow a generational schema to update the population passed from one generation to the next one, where the worst individual of the new population is replaced by the best individual of the old population.

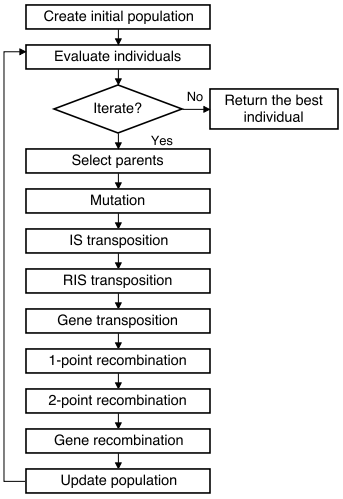

Figure 2: Flowchart of the GEPMTR method.

The proposed GEP-based method (hereafter, refereed as GEPMTR) follows the steps portrayed in Figure 2. Note that, these steps are performed n_g times (number of generations). At the end of the process the best individual found along all the generations is returned, and it can be posteriorly used as a multi-target regression model.

3 EXPERIMENTAL STUDY

In this section, first the experimental settings used in the study are explained and then, the experimental results, analysis and discussion are presented.

3.1 Experimental setting

The experiments have been executed on eight datasets for multi-target regression[4]. Table 1 show the datasets with their characteristics, such as the number of instances (n), input variables (d) and target variables (q).

Table 1: Multi-target regression datasets.

Dataset	n	d	q
slump	103	7	3
jura	359	15	3
sf1	323	10	3
atp1d	337	411	6
osales	639	413	12
wq	1060	16	14
oes97	334	263	16
oes10	403	298	16

In the experimental study, our proposal -GEPMTR- is compared with two state-of-the-art algorithms for multi-target regression, Single Target (ST) and Regressor Chains (RC) [18]. Both ST and RC use regression trees as single-target regressor, as recommended by Spyromitros-Xioufis et al. [18]. The parameters used to assess GEPMTR are listed in Table 2; most of these parameter are equal to the values proposed by Ferreira [5]. A global crossover rate of 0.8 was used, which is the sum of the rates of the three kinds of recombination.

Table 2: Parameters of the GEPMTR algorithm.

Parameter	Value
Set of functions (F)	$\{-, +, /, *, \sqrt{}, sin, cos, log\}$
Set of terminals (T)	$\{x_1, x_2, \ldots, x_d\}$
Population size (p)	100
Number of generations (n_g)	[100, 500]
Number of mutation points (n_{mp})	2
Mutation rate (n_{mp})	0.1
Number of IS fragments (n_{is})	3
Maximum length of IS fragments (l_{is})	3
IS transposition rate (p_{is})	0.1
Number of RIS fragments (n_{ris})	3
RIS transposition rate (p_{ris})	0.1
Gene transposition rate (p_{gt})	0.1
One-point recombination rate (p_{op})	0.2
Two-points recombination rate (p_{tp})	0.5
Gene recombination rate (p_{gp})	0.1
Length of the gene head (h)	[5, 10, 15, 20]

For the evaluation of the algorithms, the aRRMSE measure was computed on the test set. The aRRMSE is calculated as shown in Equation 1. This metric has been widely used to assess the multi-target regression methods [1, 3, 18]. In all experiments, to estimate aRRMSE values a 10-fold cross-validation was performed. The error metric was measured on the test set of each fold execution, and finally, these values are averaged.

In order to perform multiple comparisons between algorithms (more than two), the Friedman's test [6] was conducted. If Friedman's test indicated that there were significant differences in the performance of the algorithms, the Holm post-hoc test [7] was used to perform multiple comparisons with a control method. In the case that two independent methods were compared, we employed the Wilcoxon's signed-rank test [22] to analyse the existence of significant differences.

3.2 Results and discussion

In the experimental study, we first studied the behaviour of GEPMTR by using different values of the parameter h and number of generations. Table 3 shows the results; in each row, the best error values for each h are highlighted in bold typeface.

Table 3: Results of GEPMTR depending on h and number of generations.

	$h = 5$		$h = 10$		$h = 15$		$h = 20$	
	100 g.	**500 g.**	**100 g.**	**500 g.**	**100 g.**	**500 g.**	**100 g.**	**500 g.**
slump	**0.860**	0.872	0.848	**0.772**	0.864	**0.767**	0.845	**0.752**
jura	0.692	**0.669**	0.681	**0.646**	0.686	**0.644**	0.684	**0.642**
sf1	1.129	**1.083**	1.128	**1.128**	**1.142**	1.164	1.156	**1.141**
atp1d	0.555	**0.446**	0.541	**0.453**	0.545	**0.450**	0.544	**0.450**
osales	0.954	**0.860**	0.910	**0.861**	0.908	**0.881**	0.921	**0.888**
wq	0.976	**0.968**	0.974	**0.963**	0.975	**0.961**	0.976	**0.960**
oes97	0.660	**0.589**	0.650	**0.601**	0.658	**0.598**	0.666	**0.623**
oes10	0.530	**0.471**	0.530	**0.520**	0.546	**0.484**	0.541	**0.487**
p-**value**	0.023		0.022		0.030		0.014	

We can see that in almost all cases, when the method is executed with 500 generations the individuals evolved towards a better predictive model, achieving a better performance than when it was executed with 100 generations. For each different value of h,

[4]The datasets are available to download at http://mulan.sourceforge.net/datasets-mtr.html

the Wilcoxon's test was conducted to compare the performance of GEPMTR with 100 and 500 generations. The p-values of the Wilcoxon's test are shown in the last row of Table 3. The statistical test rejected the null hypothesis in all cases at the significance level α=0.05, meaning that the algorithm with 500 generations performed significantly better than with 100 generations.

Once verified that GEPMTR performed better with a greater number of generations, we focused the study on the value of h. Table 4 shows the rankings of GEPMTR executed with different values of h over each dataset. This table is a summary of Table 3, but it focuses on ranking values instead of aRRMSE values. In each row, the best performing configuration receives a ranking of 1, the second a ranking of 2, and so on. The last row indicates the average rankings computed by the Friedman's test.

Table 4: Ranking values of GEPMTR with different h values and 500 generations.

	h=5	h=10	h=15	h=20
slump	4.00	3.00	2.00	**1.00**
jura	4.00	3.00	2.00	**1.00**
sf1	**1.00**	2.00	4.00	3.00
atp1d	**1.00**	4.00	2.50	2.50
osales	**1.00**	2.00	3.00	4.00
wq	4.00	3.00	2.00	**1.00**
oes97	**1.00**	3.00	2.00	4.00
oes10	**1.00**	4.00	2.00	3.00
Average ranking	**2.13**	3.00	2.44	2.44

Although the Friedman's test did not find significant differences among the different values of the parameter h, the configuration with $h = 5$ was the one with the best average ranking, so the simpler the model, the greater was its predictive performance.

In the second phase of our experimental study, a comparison between GEPMTR and two state-of-the-art multi-target regressors was carried out. Table 5 shows the results; for this comparison, the results that are shown for GEPMTR are the minimum of the different configurations of h with 500 generations. The last row of the table shows the average rankings computed by the Friedman's test. In each row, the best error values are highlighted in bold typeface. The results showed that our proposal outperformed the rest of the algorithms in seven out of eight datasets, obtaining the best average ranking of the Friedman's test.

Table 5: Comparing GEPMTR with two state-of-the-art multi-target regressors.

Dataset	GEPMTR	ST	RC
slump	**0.752**	0.814	0.829
jura	**0.642**	0.696	0.704
sf1	1.083	1.127	**1.046**
atp1d	**0.446**	0.479	0.484
osales	**0.860**	0.925	0.965
wq	**0.960**	0.966	0.974
oes97	**0.589**	0.716	0.719
oes10	**0.471**	0.594	0.595
Average ranking	1.125	2.125	2.750

The Frieman's statistic was equal to 10.75, and the null hypothesis was rejected with a p-value equal to 0.005 at α=0.05, meaning that there were significant differences among the algorithms. We conducted the Holm's post-hoc test to encounter specific significant differences. The results of the Holm's test are shown in Table 6, indicating that our approach was significantly better than the rest of algorithms at α=0.05. Our GEP-based approach showed a promising predictive performance.

Table 6: Adjusted p-values computed by the Holm's test.

	ST	RC
GEPMTR vs.	0.045	0.002

3.3 Discussion

The multigenic representation was an effective way to represent the chromosomes, allowing that each individual of the population represents a different and complete solution of the multi-target regression problem. The way of creating the initial population, as well as the use of a selection operator with low pressure, enable the evolution of a population with enough diversity and may prevent the method to be trapped in local minima or flat valleys in early generations.

Operators like IS transposition, one-point and two-points recombination can introduce beneficial genetic material. Good building blocks existing in more adapted genes can be transferred to other genes, favouring the interchange of information between genes, and so, the modelling of statistical relationships between target variables.

As for the computational efficiency of GEPMTR, a prefix notation was used and the genes can be directly evaluated without constructing its corresponding ET, thus speeding up the evaluation of the individuals significantly. GEPMTR has the capacity to be parallelized without major difficulties, and therefore, the computational time of this method can be reduced. On the other hand, no time is spent decomposing the multi-target regression problem into several single-target tasks, our approach directly handles the multi-target data. It has an implicitly feature selection process that can be very beneficial in the learning process of regression models; a mathematical expression coded by a gene implicates a number of variables very fewer than the total of input variables. These features enable to use our proposal in large-scale datasets.

Our proposal performed well with the experimental settings used in this work. However, one main drawback of GEP-based methods is that they imply the analysis of a considerable number of hyperparameters. It is important for future works to further expand the experimental study; for instance by analysing different values for the mutation, transposition and recombination rates.

4 CONCLUSIONS

In this work, a novel multi-target regression method based on GEP is presented. The results showed the benefits of using GEP paradigm for resolving the multi-target regression problem. GEPMTR uses a symbolic regression analysis, coding mathematical expressions that can be used posteriorly as predictive models. The way the individuals are codified and evaluated, and also the possibilities of parallelism, result in a method with an acceptable computational cost and enable it to be used in large-scale datasets.

This work corresponds to an initial study, where we wanted to analyse the benefits of GEP paradigm for constructing multi-target regression models. In future works, we will improve some of the components of the algorithm, such as a selection operator able to tuning the selection pressure. On the other hand, it would be interesting the development of ensemble-based methods using GEPMTR members.

ACKNOWLEDGEMENTS

This research was supported by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund, project TIN2017-83445-P. This research was also supported by the Spanish Ministry of Education under FPU Grant FPU15/02948.

REFERENCES

[1] T. Aho, S. Dzžeroski B. Ženko, and T. Elomaa. 2009. Multi-target regression with rule ensembles. *Journal of Machine Learning Research* 373 (2009), 2055–2066.
[2] L. Baldassarre, L. Rosasco, A. Barla, and A. Verri. 2012. Multi-output learning via spectral filtering. *Machine Learning* 87, 3 (2012), 259–301.
[3] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga. 2015. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5, 5 (2015), 216–233.
[4] Y. Chen, D. Chen, S. U. Khan, J. Huang, and C. Xie. 2013. Solving symbolic regression problems with uniform design-aided gene expression programming. *Journal of Supercomputing* 66, 1553 (2013).
[5] C. Ferreira. 2001. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. *Complex Systems* 13 (2001), 87–129.
[6] Milton Friedman. 1940. A Comparison of Alternative Tests of Significance for the Problem of m Rankings. *Annals of Mathematical Statistics* 11, 1 (1940), 86–92.
[7] Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics* (1979), 65–70.
[8] D. Kocev, S. Džeroski, M. D. White, G. R. Newell, and P. Griffioen. 2009. Using single and multi-target regression trees and ensembles to model a compound index of vegetation condition. *Ecological Modelling* 220, 8 (2009), 1159–1168.
[9] J. R. Koza. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press.
[10] Li.Zhao, L. Wang, and D. Cui. 2012. Hoeffding bound based evolutionary algorithm for symbolic regression. *Engineering Applications of Artificial Intelligence* 25, 5 (2012), 945–957.
[11] H. S. Lopes and W. R. Weinert. 2004. Egipsys: An enhanced gene expression programming approach for symbolic regression problems. *International Journal of Applied Mathematics and Computer Science* 14, 3 (2004), 375–384.
[12] G. Melki, A. Cano, V. Kecman, and S. Ventura. 2017. Multi-Target Support Vector Regression Via Correlation Regressor Chains. *Information Sciences* 415-416 (2017), 53–69.
[13] J. M. Moyano, E. Gibaja, and S. Ventura. 2017. An evolutionary algorithm for optimizing the target ordering in Ensemble of Regressor Chains. In *IEEE Congress on Evolutionary Computation.* 2015–2021.
[14] Y. Peng, C. Yuan, X. Qin, J. Huang, and Y. Shi. 2014. An improved Gene Expression Programming approach for symbolic regression problems. *Neurocomputing* 137 (2014), 293–301.
[15] O. Reyes, A. Cano, H. Fardoun, and S. Ventura. 2018. A locally weighted learning method based on a data gravitation model for multi-target regression. *International Journal of Computational Intelligence Systems* 11 (2018), 282–295.
[16] O. Reyes, C. Morell, and S. Ventura. 2014. Evolutionary feature weighting to improve the performance of multi-label lazy algorithms. *Integrated Computer-Aided Engineering* 21, 4 (2014), 339–354.
[17] T. Similä and J. Tikka. 2007. Input selection and shrinkage in multiresponse linear regression. *Computational Statistics & Data Analysis* 52, 1 (2007), 406–422.
[18] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, and I. Vlahavas. 2016. Multi-target regression via input space expansion: Treating targets as inputs. *Machine Learning* 104, 1 (2016), 55–98.
[19] S. Stijven, E. Vladislavleva, A. Kordon, L. Willem, and M. E. Kotanchek. 2016. *Genetic Programming Theory and Practice XIII.* Springer, Cham, Chapter Prime-Time: Symbolic Regression Takes Its Place in the Real World, 241–260.
[20] A. Tsanas and A. Xifara. 2012. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings* 49, 560-567 (2012).
[21] D. Tuia, J. Verrelst, L. Alonso, F. Pérez-Cruz, and G.Camps-Valls. 2011. Multioutput support vector regression for remote sensing biophysical parameter estimation. *IEEE Geoscience and Remote Sensing Letters* 8, 4 (2011), 804–808.
[22] Frank Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin* 1, 6 (1945), 80–83.
[23] J. Zhong, L. Feng, and Y. S. Ong. 2017. Gene Expression Programming: A Survey. *IEEE Computational Intelligence Magazine* 12, 3 (2017), 54–72.
[24] C. Zhou, W. Xiao, T. M. Tirpak, and P. C. Nelson. 2003. Evolving accurate and compact classification rules with gene expression programming. *IEEE Transactions on Evolutionary Computation* 7, 6 (2003), 519–531.
[25] J. Zuo, C. J. Tang, C. Li, C. A. Yuan, and A. L. Chen. 2004. Time series prediction based on gene expression programming. In *Proocedings of the International Conference on Web-Age Information Management.* Berlin Heidelberg: Springer-Verlag, 55–64.