Imbalanced Classification with TPG Genetic Programming: Impact of Problem Imbalance and Selection Mechanisms

Nicolas Sourbier, Justine Bonnot, Karol Desnos, Frédéric Majorczyk, Olivier Gesny, Thomas Guyet, Maxime Pelcat

Imbalanced Classification with TPG Genetic Programming: Impact of Problem Imbalance and Selection Mechanisms

Nicolas Sourbier
nicolas.sourbier@insa-rennes.com
Univ Rennes, INSA Rennes, IETR,
UMR CNRS 6164
Rennes, France

Justine Bonnot*
justine.bonnot@insa-rennes.fr
Univ Rennes, INSA Rennes, IETR,
UMR CNRS 6164
Rennes, France

Frédéric Majorczyk
DGA-MI - CIDRE
Bruz, France
frederic.majorczyk@intradef.gouv.fr

Olivier Gesny
Silicom
3 E rue de Paris, Cesson-Sevigné
France
ogesny@silicom.fr

Thomas Guyet
Inria - Centre de Lyon, France
Villeurbanne, France
thomas.guyet@inria.fr

Maxime Pelcat*
maxime.pelcat@insa-rennes.fr
Univ Rennes, INSA Rennes, IETR,
UMR CNRS 6164
Rennes, France

ABSTRACT

Recent research advances on Tangled Program Graphs (TPGs) have demonstrated that Genetic Programming (GP) can be used to build accurate classifiers. However, this performance has been tested on balanced classification problems while most of the real world classification problems are imbalanced, with both over-represented classes and rare classes.

This paper explores the effect of imbalanced data on the performance of a TPG classifier, and proposes mitigation methods for imbalance-caused classifier performance degradation using adapted GP selection phases. The GP selection phase is characterized by a fitness function, and by a comparison operator. We show that adapting the TPG to imbalanced data significantly improves the classifier performance. The proposed adaptations on the fitness make the TPG agent capable to fit a model even with 10^4 less examples than the majority class whereas the revised selection phase of the GP process increases the robustness of the method for moderate imbalance ratios.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Artificial intelligence**; **Genetic programming**; *Genetic algorithms.*

KEYWORDS

classification, machine learning, genetic programming, imbalanced data, tangled program graphs, selection

ACM Reference Format:
Nicolas Sourbier, Justine Bonnot, Frédéric Majorczyk, Olivier Gesny, Thomas Guyet, and Maxime Pelcat. 2022. Imbalanced Classification with TPG Genetic Programming: Impact of Problem Imbalance and Selection Mechanisms. In *Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3520304.3529008

GECCO '22 Companion, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9268-6/22/07.
https://doi.org/10.1145/3520304.3529008

1 INTRODUCTION

Imbalanced classification is the problem of training a classifier where one or several classes are represented by fewer samples than other classes. Imbalanced classification is particularly necessary when rare events [26], or anomalies [2] need to be detected in a large amount of *normal* data. Imbalance can reach orders of magnitude, making it difficult to identify minority classes while avoiding false positives - false identification of the minority class.

While most real world use cases are imbalanced, studies on learning based classification tend to focus on balanced databases. Genetic Programming (GP) has been shown to perform well on balanced image classification [12] by using a framework called Tangled Program Graph (TPG). With respect to competitors, TPG offer lightweight inference and a support for both high data cardinality and efficient diversity maintenance [19]. TPG are based on a versatile graph of teams of programs, modeling complex relationships between input data and output *actions*. When used for classification, TPG actions translate into classes while they originally refer to environment modifying actions in a Reinforcement Learning (RL) environment.

This paper aims at adapting the TPG GP framework so as to improve its capacity to perform imbalanced classification. We propose new semantics to be integrated within TPG in the form of an adapted selection system. Paper contibution are the folowing: we evaluate the classification performance degradation induced by the database imbalance, we compare fitness functions and evaluation metrics, and evaluate their effect on training and evaluation of a TPG-based classification, and we demonstrate on an example that GP can adapt to imbalanced classification problems, provided that the fitness and selection phase is adapted.

2 RELATED WORK

Methods for dealing with imbalanced learning problems can be classified into *data level methods* that use database over- and under-sampling or specialized feature selection, and *algorithm level methods*, including e.g. cost sensitive methods and hybrid/ensemble learning [15]. Data level methods modify the input data to mitigate

performance degradation due to imbalanced databases whereas algorithm level methods modify the agent to make it resilient to imbalanced data. Ensemble methods are a combination of both data level and algorithm level mitigation.

Over-sampling [28] and under-sampling [27] methods re-balance the database to make an unmodified classifier robust to imbalance. Synthetic Minority Oversampling Technique (SMOTE) [3, 8] combines over-sampling and under-sampling. These methods have drawbacks, as over-sampling artificially duplicates samples and often causes over-fitting while under-sampling deletes potentially important components from the majority class. At algorithm level, the mitigation of the imbalance issue is mostly addressed through adapting the fitness function. In [16–18, 24], authors use custom or cost-based functions to counteract the effects of data imbalance. In [22], a fitness function is derived from the F1-Score and adapted to analyzing textual and biological data.

In these works, no comparison is performed on the performance of the training using one fitness function or another. There exist generic fitness functions [10] that are relevant candidates for learning in an imbalanced environment such as Matthew's Correlation Coefficient (MCC) [29], G-mean [1] or Cohen's Kappa (κ) [23]. Section 4.2 will discuss these functions and other Machine Learning (ML) metrics, for their use as training fitness functions.

Previous work on ensemble methods has shown that custom or cost based fitness functions [16, 17, 24] are able to make machine learning converge on imbalance ratios between 1:8 and 1:129. Other studies use algorithmic level mitigation and base their fitness on more generic functions such as per class accuracy [13], F-score [7, 22] or custom fitness functions [18, 21]. In particular, [18] was able to learn on a binary classification problem with an imbalance ratio of 1:4500. Classification with GP is an active research field. In [19], a TPG is specialized for performing balanced classification through the comparison of learning schemes. The framework GEGELATI [6] provides an implementation of the TPG that has been studied on the imbalanced learning context of network intrusion detection [20]. With respect to these previous works, the main objective of this paper is to study the effect of imbalance ratio on classification performances of a GP technique, and to compare algorithm level methods that are likely to counteract the negative effects of imbalance.

3 TANGLED PROGRAM GRAPHS MODEL AND LEARNING ALGORITHM

This study is based on the TPG genetic programming framework, as introduced by Kelly and Heywood [11], that consists of three elements composing a directed graph: *programs*, *teams* and *actions*. *Teams* are the internal vertices of the graph while *actions* are leaves of the graph. *programs*, composed of arithmetic *instructions* such as additions or exponents, are associated to edges of the graph that each connect a source *team* to either a destination *team* or a destination *action* vertex. Starting from a unique *root* team, The execution of a TPG progresses through programs and teams until an action is reached, corresponding to a class in a classification TPG. To choose a path in the graph, programs compete by each returning a value, called bid, while the highest bid is systematically selected. The evolution process of a TPG relies on the generation of several root *teams* by genetic programming. Worst-performing root *teams* are deleted from the TPG while new teams and programs, are generated through evolution. Mutations favor the emergence of long-living valuable sub-graphs of interconnected *teams*. Hence, complexity is added to the TPG adaptively, only if this complexity leads to a better evaluation. An extension of the TPG has been proposed to support classification [12]. This proposal motivates the current study on imbalanced classification.

4 PROBLEM DEFINITION

4.1 The Imbalanced Classification Problem

We consider the following problem: an ML classifier m is trained on a training set s_{tr} and tested on a test set s_{te}. Each sample $\sigma \in s_{tr} \cup s_{te}$ belongs to one of two classes $\{P, N\}$ where P denotes the positive class and N the negative class. The oracle function $o : s_{tr} \cup s_{te} \rightarrow \{P, N\}$ associates to each sample σ its true class $o(\sigma)$ while the classifier $m : s_{tr} \cup s_{te} \rightarrow \{P, N\}$ associates to each sample σ its predicted class $m(\sigma)$. Given a set of samples s, the subset of true positive samples is expressed as $s^P = \{\sigma \in s | o(\sigma) = P\}$. Conversely, the subset of true negative samples is expressed as $s^N = \{\sigma \in s | o(\sigma) = N\}$. In the rest of this paper, we will use TP, TN, FP and FN as intuitive cardinality expressions of respectively true positives, true negatives, false positives and false negatives in the classifier produced results, on the test set. We formally define these expressions as $TP = card(\{\sigma \in s_{te} | o(\sigma) = P \wedge m(\sigma) = P\})$, $TN = card(\{\sigma \in s_{te} | o(\sigma) = N \wedge m(\sigma) = N\})$, $FP = card(\{\sigma \in s_{te} | o(\sigma) = N \wedge m(\sigma) = P\})$, $FN = card(\{\sigma \in s_{te} | o(\sigma) = P \wedge m(\sigma) = N\})$.

A binary classification problem is imbalanced when the training and test sets contain less positive samples than negative samples. If data is received as samples, imbalance corresponds to a low frequency of appearance of samples of one class, making each of them of a high importance. In order to compactly express the degree of imbalance that can reach very high values, we propose the notion of Imbalance Order of Magnitude (IOM). We define *iom* of a sample set s as $iom : s \rightarrow \mathbb{N}, iom(s) = log_{10}(card(s^N)/card(s^P))$.

For example, an IOM of 3 in a training set s_{tr} means that there are 1000 times more negative samples than positive samples in the set. The same imbalance is expressed as a ratio $1 : 1000$, where a ratio is defined as $1 : \frac{card(s^N)}{card(s^P)}$. In this paper, we hypothesize that the IOM of both the training and test sets are equivalent: $iom = iom(s_{tr}) \approx iom(s_{te})$ and we thus refer to this IOM as the global imbalance of the problem. Intuitively, the imbalance of a problem will systematically tend to complicate the task of the classifier. This paper studies this hypothesis with a GP classifier, and characterize the link between imbalance ratios and classification degradation.

4.2 Choosing fitness and evaluation metrics

Two functions need to be selected to train and test a TPG: the fitness function and the evaluation function. While the fitness function generates the inputs to the comparison of two teams, the evaluation function evaluates the capacity of the model to perform classification. For both fitness and evaluation, state-of-the-art functions can be used, such as classification accuracy and F1-score. We discuss here the metrics that will be evaluated in experimental results.

On imbalanced problems, accuracy, precision, recall and F1-score shadow the minority class mis-predictions and profit to the majority class [25]. As a consequence, **accuracy** is not an adequate method for evaluating learning-based methods in highly imbalanced contexts. Similarly, **F1-score** does not take into account TN. F1-score is arguably more adapted to imbalanced classification than accuracy [9] but a model mistaking half of the time on the prediction of the minority class results in a seemingly fair F1-score of 0.667. **G-mean** [14] (geometric mean of the product of the sensitivity and the specificity) is more adapted to imbalance cases. The value of G-mean is equally sensitive to the ratio of TP versus FN and to the ratio of TN versus FP. **MCC** [4], also known as the phi-coefficient), is the computation of the correlation between true and predicted values, leading to a maximal correlation coefficient of 1 in the case of a perfect classifier. Similarly to G-mean, high MCCs can be reached only for good predictions of both majority and minority classes. Finally, the **Cohen's Kappa** score measures the reliability of a decision by taking into account the probability that a good decision happened by chance. Similarly to MCC, the computation of the κ score is sensitive to the amount of available positives in the database. The *kappa* score will tend toward zero when the amount of positives lowers, unless in the specific case of a perfect classifier.

Considering fitness functions, a simple fitness strategy for the training of the classification policy is to return $+1$ for a good classification and -1 when the policy is mistaking [19]. To mitigate the effect of imbalance, this fitness system can be weighted using **balanced accuracy**, weighting the accuracy fitness function with coefficients inversely proportional to the imbalance ratio.

In an imbalanced context, the fitness function and evaluation metric must be seen as functions to maximize and do not tell the whole truth on the accuracy of the classifier. Indeed, each problem comes with specific *FN* or *FP* requirements. The metrics discussed here-over are assessed as both fitness and evaluation functions in the next section.

5 EXPERIMENTAL RESULTS

5.1 Experimental Setup

To study the effect of imbalance and test the selection proposition, we use the MNIST database [5], a balanced collection of $50k$ training images and $10k$ test gray-scale images of handwritten digits (from zero to nine). In order to use the database for imbalanced training, we adjust it by selecting 1 Positive class among the 10 available classes, selecting an imbalance ratio, filling a training subset of 10 000 images with images from negative and positive classes until imbalance ratio is reached, and extracting a test set of 2 000 images.

The imbalance ratio ranges from 0 IOM to 4 IOM with steps of 0.5 IOM. A minimum of 1 Positive sample is forced into the test set if not present.

5.2 Fitness function and Evaluation metric

In this subsection, we investigate which fitness function and evaluation metric to use for training on an imbalanced database. In practice, the ML learning process tends to reach the highest fitness when the best model is defined by the model reaching the highest evaluation. As a baseline, a legacy TPG is trained on a balanced classification task. It can be observed that TPG converges for each positive class, a mean convergence of $\kappa = 0.61$ is reached, with quite a large variance suggesting classification problems with diverse complexities. Convergence is reached in most cases at generation 180, we thus fix this number of generations in next experiments.

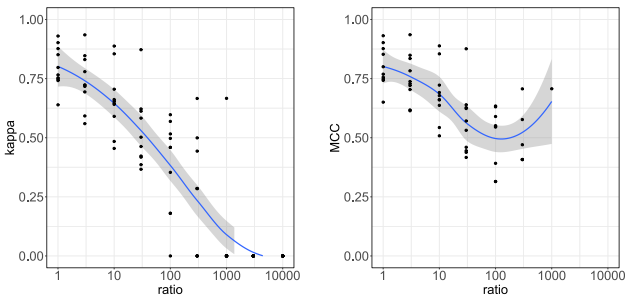

Figure 1: κ (a) and MCC (b) versus imbalance ratio of a model. A point represents an evaluation for a fixed minority class of the database, for a fixed imbalance ratio. The blue line and gray area represent conditional mean of the evaluation associated with confidence intervals. IOM goes from 0 to 4. MCC is shown to badly capture classification degradation.

5.2.1 Evaluation metric: An evaluation metric for imbalanced classification shall (i) reach a maximum for the considered best model, (ii)] reach a minimum for the considered worst model, (iii) take into account the imbalance of the database, and (iv) be humanly understandable. As explained in Section 4.2, F1-score, State-of-the-art TPG fitness and balanced accuracy are poor evaluation metrics on an imbalanced problem. Furthermore, it has been shown that G-mean is not a good evaluation metric as the influence of the Positive class is shadowed when the number of positive samples is low. Figure 1 thus compares the evaluation results of the same models in terms of the two remaining considered metrics: κ and MCC. One expects for the mean evaluation of the classification quality to decrease when the IOM increases. The right curve representing the evolution of the MCC stops decreasing at an IOM of 3 as the other computed points are irrelevant. **The adequate evaluation metric for imbalanced problems among the standard metrics is the Cohen's Kappa function.** Each of these functions merge several parameters into a single value and, as so, only give a partial view of the problem. The class-wise accuracy or confusion matrix are still more complete indicators of learning.

5.2.2 Fitness function. A good fitness function shall(i)] Reach a maximum for the best model's predictions, (ii) reach a minimum for the worst model's predictions, and (iii) produce a model that maximizes the aforementioned evaluation metric. For the same reasons evoked on evaluation, F1-score and State-of-the-art TPG fitness are poor evaluation metrics for high IOM. Figure 2 shows the influence of κ, MCC and G-mean used as fitness functions. **For low IOM, from 0 to 1, the G-mean fitness function is the most performing. The MCC Fitness function is to be preferred when the IOM is above 1. Figure 2 also shows that a TPG can train, though with degraded performance, on problems with 4 IOMs of imbalance.**

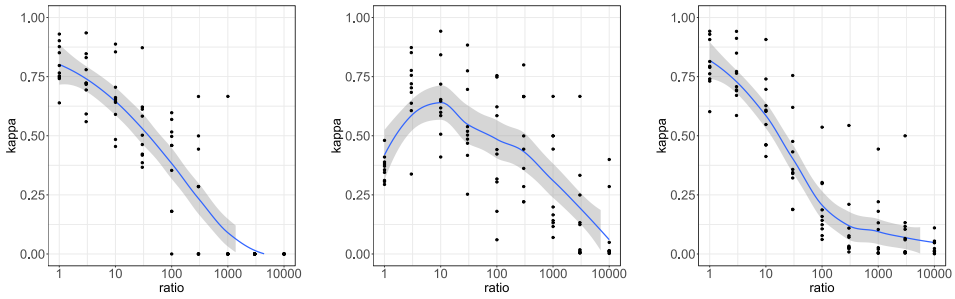

Figure 2: κ versus imbalance for fitness functions κ (a), MCC (b) and G-mean (c). A point represents an evaluation at a fixed imbalance ratio. The blue line and gray area represent the conditional mean of the evaluation results associated with confidence intervals. IOM goes from 0 to 4. G-mean is the best fitness function for low imbalance, and MCC for high imbalance.

6 CONCLUSION

This paper has studied the effect of imbalance on a binary classification task based on genetic programming. We demonstrate that the Cohen's Kappa is, among standard metrics, the one to use to evaluate a classifier on imbalanced problems. We also propose algorithm modifications on fitness and selection phases to support imbalance in genetic programming. Indeed, this problem is bound to a training phase where selection and fitness functions are key components, as the genetic programming algorithm uses selection phases to keep the individuals that perform correctly, based on their respective fitness. This paper shows that the G-mean fitness function is a good candidate for low IOM while MCC fits better the highly imbalanced problems. In practice, it is shown that learning in an environment with an IOM of 4 is possible, but with very degraded performances.

REFERENCES

[1] Y. S. Aurelio, C. L. de Almeida, G. M.and de Castro, and A. P. Braga. Learning from imbalanced data sets with weighted cross-entropy function. *Neural processing letters*, 2019.

[2] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[3] N. V Chawla, K. W Bowyer, L. O. Hall, and W P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 2002.

[4] D. Chicco and G. Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 2020.

[5] L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 2012.

[6] K. Desnos, N. Sourbier, P.-Y. Raumer, O. Gesny, and M. Pelcat. Gegelati: Lightweight artificial intelligence through generic and evolvable tangled program graphs. In *DASIPg (14th edition)*, pages 35–43, 2021.

[7] D. Devarriya, C. Gulati, V. Mansharamani, A. Sakalle, and A. Bhardwaj. Unbalanced breast cancer data classification using novel fitness functions in genetic programming. *Expert Systems with Applications*, 2020.

[8] A. Fernandez, Salvador G., F. Herrera, and Nitesh V C. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, 2018.

[9] M. Grandini, E. Bagli, and G. Visani. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*, 2020.

[10] L. A Jeni, J. Cohn, and F. De La Torre. Facing imbalanced data–recommendations for the use of performance metrics. In *2013 Humaine association conference on affective computing and intelligent interaction*. IEEE.

[11] Stephen K., R. J. Smith, and M. I. Heywood. Emergent Policy Discovery for Visual Reinforcement Learning Through Tangled Program Graphs: A Tutorial. In *Genetic Programming Theory and Practice XVI*. Springer, 2019.

[12] S. Kelly and M. I Heywood. Emergent tangled graph representations for atari game playing agents. In *EuroGP*. Springer, 2017.

[13] S. Khanchi, A. Vahdat, M. I Heywood, and A N. Zincir-Heywood. On botnet detection with genetic programming under streaming data label budgets and class imbalance. *Swarm and evolutionary computation*, 2018.

[14] A. Kulkarni, D. Chong, and F. A Batarseh. Foundations of data imbalance and solutions for a data democracy. In *data democracy*. Elsevier, 2020.

[15] J. L Leevy, T. M Khoshgoftaar, R. A Bauder, and N. Seliya. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 2018.

[16] V. López, A. Fernández, M. J. Del Jesus, and F. Herrera. A hierarchical genetic fuzzy system based on genetic programming for addressing classification with highly imbalanced and borderline data-sets. *Knowledge-Based Systems*, 2013.

[17] W; Pei, B. Xue, L. Shang, and M. Zhang. Genetic programming for high-dimensional imbalanced classification with a new fitness function and program reuse mechanism. *Soft Computing*, 2020.

[18] T. Perry, M. Bader-El-Den, and S. Cooper. Imbalanced classification using genetically optimized cost sensitive classifiers. In *2015 IEEE CEC*. IEEE, 2015.

[19] R. J Smith, R. Amaral, and M. I Heywood. Evolving simple solutions to the cifar-10 benchmark using tangled program graphs. In *2021 IEEE CEC*. IEEE, 2021.

[20] N. Sourbier, K. Desnos, T. Guyet, F. Majorczyk, O. Gesny, and M. Pelcat. Secure-gegelati always-on intrusion detection through gegelati lightweight tangled program graphs. *JSPS*, 2021.

[21] M. A. U.H. Tahir, S. Asghar, A. Manzoor, and M. A. Noor. A classification model for class imbalance dataset using genetic programming. *IEEE Access*, 2019.

[22] F. Viegas, L. Rocha, M. Gonçalves, F. Mourão, G. Sá, T. Salles, G. Andrade, and I. Sandin. A genetic programming approach for feature selection in highly dimensional skewed data. *Neurocomputing*, 2018.

[23] S. Vieira, U. Kaymak, and J. Sousa. Cohen's kappa coefficient as a performance measure for feature selection. In *International Conference on Fuzzy Systems*. IEEE, 2010.

[24] C. Y. Wang, L. Hu, MZ Guo, X. Liu, and Q Zou. imdc: an ensemble learning method for imbalanced classification with mirna data. *Genetics and Molecular Research*, 2015.

[25] N. W. S. Wardhani, M. Y. Rochayani, A. Iriany, A. D. Sulistyono, and P. Lestantyo. Cross-validation metrics for evaluating classification performance on imbalanced data. In *2019 ic3ina*. IEEE, 2019.

[26] J. Wu, J. M Rehg, and M. D Mullin. Learning a rare event detection cascade by direct feature selection. Technical report, Georgia Institute of Technology, 2003.

[27] S. Yen and Y. Lee. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In *Intelligent Control and Automation*. Springer, 2006.

[28] H. Zhang and M. Li. Rwo-sampling: A random walk over-sampling approach to imbalanced data classification. *Information Fusion*, 2014.

[29] Q. Zhu. On the performance of matthews correlation coefficient (mcc) for imbalanced dataset. *Pattern Recognition Letters*, 2020.