7th ICML Workshop on Automated Machine Learning (2020)

W-EDGE: Weight Updating in Directed Graph Ensembles to improve Classification

Xavier Fontes XAVIER.FONTES@FE.UP.PT
Faculty of Engineering of the University of Porto, Portugal

Daniel Castro Silva DCS@FE.UP.PT
Faculty of Engineering of the University of Porto, Portugal

Pedro Henriques Abreu PHA@DEI.UC.PT
Department of Informatics Engineering of the University of Coimbra, Portugal

Abstract

Ensembles constitute one of the most widely used approaches in classification tasks, for their ability to mitigate the weaknesses of several models while making a more robust and powerful prediction tool. Evolutionary Directed Graph Ensembles (EDGE) is a framework for evolving ensembles of models represented as directed acyclic graphs, where the connections between nodes represent the impact of a node's prediction in the successor's prediction. The Directed Graph Ensembles (DGEs) have their topology evolved, with the connections bootstrapped using a sample of the training data. Our work extends EDGE by evolving the weights of the connections between nodes of a DGE using Adam, an algorithm for gradient-based optimization. We also automatically decide, at each generation, whether to evolve the topology or the weights, using the Kolmogorov–Smirnov test. Two sets of tests were devised to evaluate this work, the first comparing the results directly with the ones from the EDGE framework. In contrast, the second compares the results with a baseline consisting of Decision Trees, Random Forests, and Gradient Boosting classifiers on a compilation of datasets from the KEEL dataset repository. The results for the first test set show that our implementation versus the original EDGE improves accuracy in three datasets by 4.2 percentage points and lagging 0.20 percentage points in one. The baseline comparison shows our work to achieve the pique accuracy in 20 out of the 21 tested datasets, with gains between 1 and 15 percentage points and the only decline at around 0.10 percentage points.

1. Introduction

In recent years there have been interesting developments in ensemble learning, making it ubiquitous in machine learning. Specifically, Random Forests are one of the *de facto* methods in machine learning classification tasks (Polikar, 2012). In essence, ensembles have base models that are trained independently of each other, and the predictions of all models are then combined to form the final prediction of the ensemble as a whole (Dietterich, 2000; Russell and Norvig, 2009).

In the literature, one approach to ensemble learning that has shown satisfactory results is the use of Evolutionary Algorithms (Sylvester and Chawla, 2006; Gagné et al., 2007; Kim and Cho, 2008). Evolutionary Algorithms stem from the concept of evolving populations of individuals, trying to incrementally improve the population at each generation according to some fitness function. This domain is combined with ensemble learning in various ways, one

of the most adopted being to have an evolutionary meta-model choose the weights or the base models themselves (Kwon and Moon, 2004; Chandra and Yao, 2006). The ensemble is thus evolved generation after generation to increase predictive performance.

Evolutionary Directed Graph Ensembles (EDGE) is a machine learning tool based on the evolution of a population of ensembles where each ensemble is represented as a weighted graph (Fontes and Silva, 2019). Evolutionary Algorithms are used to evolve ensembles of models arranged in a directed acyclic structure. The weights of connections between nodes map the strength each node gives its predecessors' predictions. One of EDGE's shortcomings, as it was presented, is that it only evolves the topology of the graph ensembles, keeping the weights between nodes static.

This work tackles the aforementioned shortcoming, by evolving the weights attributed to the predecessors of any given node. This work also introduces a heuristic to chose, at the start of each generation, whether to evolve the topology or the weights of the graph ensembles. We use an optimizer to update the weights between nodes in the graph with a decision heuristic based on a nonparametric statistic test to present what is, arguably, an improved implementation of EDGE. The proposed development is tested and compared with previous results from EDGE, as well as with a collection of baseline models in several different datasets.

The rest of this paper is organized as follows. In Section 2, we present a brief overview of EDGE and a more detailed description of our contribution. The experimental setup and results are discussed in Section 3. Finally, we conclude this work and mention future developments in Section 4.

2. EDGE and Our Contribution

In EDGE, every graph ensemble is referred to as a Directed Graph Ensemble (DGE) and instantiated as a directed acyclic graph. Every node in the graph has a Component Model (CM), an instance of a model used for prediction, such as a Decision Tree or Random Forest. Every node of the graph computes prediction P based on its predecessors and its CM. The prediction of node i, P_i, for a given data point X_j, is calculated according to the formula:

$$P_i(X_j) = \alpha \times pred_i(X_j) + (1 - \alpha) \times \frac{pp_i(X_j)}{\|pp_i(X_j)\|_1}$$

where

$$pp_i(X) = \sum_{n \in pr(i)} P_n(X) \circ W_i^n$$

and $\circ$ represents the Hadamard product; α represents a node's confidence in its own predictions; $pred_i(X)$ denotes the prediction made by CM_i; $pr(i)$ denotes the set of predecessors of node i; $pp_i(X)$ represents the aggregated prediction made by the predecessors of i; W_i^n represents the weight vector of node i in its predecessor, node n. The inner workings of EDGE make use of an auxiliary data structure, the Reservoir, which is tasked with having a configuration space for a given set of models. At runtime, it can supply EDGE with the necessary CMs. Every generation, the graph ensembles that constitute the population are evaluated on a portion of the training data and given a fitness value, representing how

apt each individual is. EDGE defines fitness as the accuracy with class-balanced weights. The distribution of fitness values of a population at any given generation represents how well the generation is as a whole. A full description of the original formulation of EDGE is presented by Fontes and Silva (2019).

The main contribution of this work is the implementation of a strategy to evolve the weights between nodes in the graph ensembles. The strategy consists of using Adam, a gradient-based optimization algorithm (Kingma and Ba, 2015), to update the weights of the graph ensembles, namely the previously mentioned parameters W and α. The gradients are computed with respect to a loss function that we wish to minimize. The loss function used was the sum of mean squared error (MSE) for each class probability value across a subset of data. It was assumed that, at each generation, only one type of evolution step is performed – topology or weight evolution – thus we devised a heuristic based on the statistical comparison between two probability distributions. At the start of each generation, the most suitable type of step is chosen. In the scheme of EDGE's internal process, our contribution can be seen as the red-colored components in Fig. 1.

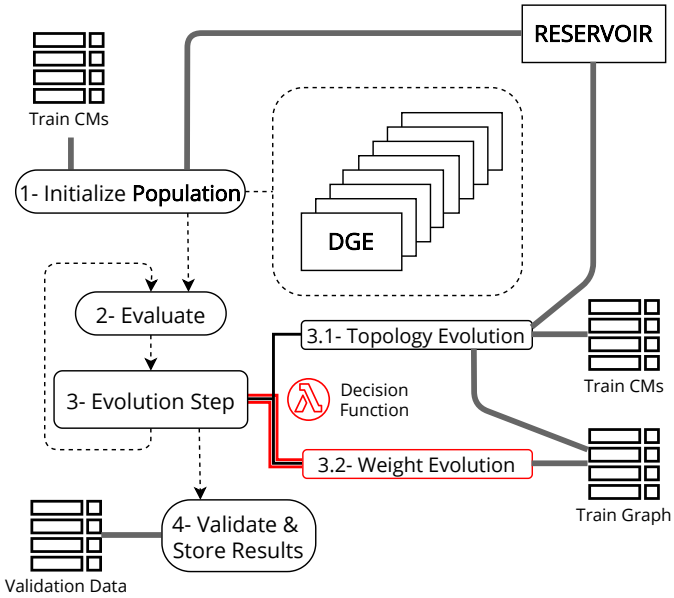

Figure 1: Inner process of EDGE, with our contribution outlined.

The decision process for the evolution step was made using the Kolmogorov–Smirnov test (K-S test) to decide whether to switch the type of evolution step or not (Conover, 1980). At each generation after the first, we use the K-S two-sample test to ascertain whether two distributions of fitness values, the current generation's and the previous', differ from one another. The devised heuristic always starts with a topology step. In subsequent generations, if we reject the null hypothesis that the two fitness distributions come from the same distribution, the evolution continues with the previous type of step, otherwise switching to the other type. The intuition behind this heuristic is that we keep performing the same step type if, following the K-S test, it continues to produce distributions of the

fitness values different from the previous generation, with the assumption that whenever we have different distributions, the population as a whole has improved.

3. Experimental Results

To evaluate our contribution, two sets of tests were designed. The first is intended to compare this version against the previously introduced version of EDGE. For a more natural distinction between the two different implementations, we refer to ours as Weighted-EDGE (W-EDGE). The second type was designed to compare this work against a baseline of models, on a bigger dataset pool. Both test sets and their results are discussed next.

3.1 W-EDGE vs EDGE

In this test set, we used the same four datasets used by Fontes and Silva (2019): the Anuran dataset, where feature variables derived from their callings are used to predict the species of anuran (Colonna et al., 2017); the MNIST dataset, for its widespread use as a benchmarking tool (Deng, 2012); an appliances energy forecasting dataset (Candanedo et al., 2017); and a parking lot occupancy dataset introduced by the authors in earlier works (Fontes and Silva, 2018)[1]. The parameters of W-EDGE matched with those of EDGE (Table A.1). The results can be seen in Table 1, where bold text outlines the best results.

Table 1: Comparison between EDGE and W-EDGE. Results from EDGE are the best results presented, per dataset. Results from W-EDGE are the average of 3 runs and values between parentheses are the best obtained.

	EDGE		W-EDGE	
Dataset	Accuracy	F1-Score	Accuracy	F1-Score
MNIST	97.78	97.78	**97.93** (98.34)	**97.93** (98.34)
Anuran	**99.17**	**99.16**	98.80 (98.94)	98.79 (98.94)
Appliances	86.27	84.29	**93.37** (93.96)	**93.26** (93.74)
Parking Lot	87.68	87.10	**93.28** (93.81)	**93.24** (93.75)

Of the 4 datasets, W-EDGE improved on 3, 2 of them by a significant margin. In the significantly dominated datasets, accuracy was improved by around 6 percentage points, whereas the dataset that EDGE dominated has a difference of only 0.2 percentage points. The two datasets in which W-EDGE improved by a bigger margin are time series with a discretized target variable. We argue that a possible explanation is the robustness of evolving both topology and weights that allows capturing patterns, even in different types of problems.

1. Dataset available online from `https://github.com/xfontes42/parking_lot_ds`

3.2 W-EDGE vs Baseline

Baseline models were defined using Decision Tree models, Random Forests and Gradient Boosting classifiers, following the configuration space presented in Table A.2. We chose these three types of models for baselines and their specific configuration because they represent the models and approximate parameters that EDGE used as CMs, arguing for a fairer comparison. The configuration space of W-EDGE is disclosed in Table A.3. We used the Anuran dataset mentioned in the previous section as a link to it and because of its size and 20 datasets from the KEEL Dataset Repository (Alcalá-Fdez et al., 2011). Each dataset's main characteristics are showcased in Table A.4.

The division between train and test was made randomly, with the test set being 20% of the whole data, and the remaining 80% being further split 60-20 into training the CMs and training the graphs, respectively.

Table 2: Comparison between baseline and W-EDGE. Only the best results for each are shown.

Dataset Name	Baseline		W-EDGE	
	Accuracy	F1-Score	Accuracy	F1-Score
Anuran	98.33	98.32	**99.17**	**99.16**
H9-car-good	99.71	99.71	**99.88**	**99.89**
H9-poker-8-9_vs_5	98.80	98.20	**99.61**	**99.58**
H9-winequality-white-9_vs_4	100.00	100.00	100.00	100.00
H9-zoo-3	100.00	100.00	100.00	100.00
L9-haberman	82.26	80.40	**89.54**	**89.14**
L9-iris0	100.00	100.00	100.00	100.00
L9-new-thyroid1	97.67	97.60	**100.00**	**100.00**
L9-newthyroid2	100.00	100.00	100.00	100.00
ST-appendicitis	90.91	89.63	**100.00**	**100.00**
ST-haberman	75.81	74.19	**89.54**	**89.37**
ST-hepatitis	100.00	100.00	100.00	100.00
ST-iris	96.67	96.66	**100.00**	**100.00**
ST-monk-2	100.00	100.00	100.00	100.00
ST-newthyroid	95.35	95.60	**100.00**	**100.00**
ST-ring	95.68	95.67	**97.46**	**97.46**
ST-shuttle-c2-vs-c4	100.00	100.00	100.00	100.00
ST-tae	74.19	73.85	**89.47**	**89.47**
ST-titanic	**79.37**	**76.48**	79.29	76.24
ST-wine	100.00	100.00	100.00	100.00
ST-zoo	95.24	93.12	**100.00**	**100.00**

The results can be examined in Table 2, all of them being the best results aggregated by the baseline models and the configurations of W-EDGE.

Overall the results are very positive, with our implementation improving the accuracy of 12 of the 21 datasets, maintaining the performance in 8 of the 21 datasets, and only losing by a small margin (0.1 percentage points) in one case. On the datasets that were improved, the improvement ranged from 0.17 to 15.28 percentage points, with an average 5.25 percentage point increase in accuracy. The perfect accuracy values should be taken critically, showcasing perhaps that some datasets are too simple. W-EDGE achieved good results, regardless of the Imbalance in the datasets, which also hints at its ability to be robust against skewed data. We consider these results to support the promise of our contribution to the further development of a framework that automates the process of model choice and the evolution of ensembles.

4. Conclusion and Future Work

EDGE is a tool for ensemble evolution that evolves a population of ensembles where each ensemble is represented as a directed graph. An auxiliary data structure is used to supply EDGE with CMs that are the nodes of the graph ensembles. This method allows the exploration of an enormous space state for the possible CMs used and possible good pairings of models, having only to setup the most basic parameters.

In this work, we set our focus on improving this existing tool with a method for updating the weights, using the Adam optimization algorithm, and a decision mechanism based on the KS-test of equality between probability distributions to choose whether to evolve the topology or the weights at each generation of the ensembles' evolution. Allowing the weights of the graphs to be updated, as well as the decision of the evolution step, we believe data can be better used to learn the weight each node should have in its predecessors and in itself.

Our implementation, dubbed W-EDGE, constitutes an improvement over the standard EDGE implementation. Two experimental setups support the argument, the first being that W-EDGE improved on 75% the datasets. The improvement was more than one order of magnitude bigger than the decline (4.2 percentage point increase versus 0.2 percentage points decrease in accuracy). The other setup where W-EDGE was compared against a baseline of models on 21 different datasets showed W-EDGE to perform worse on only one dataset, by a relatively low margin of 0.10 percentage points, while maintaining or improving in the other datasets by as much as 15 percentage points.

For future work, attention is to be given in running experiments with more configurations and different types of baseline models, as well as choosing a more significant number of datasets. The weight update can be a subset of the topology evolution instead of a different type of step altogether. In short, we might not even consider weight updating as changing from one generation to another, effectively combining both types of evolution steps. Since we are evolving the weight each node gives its predecessors and itself, we can explore heuristics that force the topology to be changed upon specific triggers, like a low self-confidence of a node triggering its removal.

References

Jesús Alcalá-Fdez, Alberto Fernández, Julián Luengo, Joaquín Derrac, Salvador García, Luciano Sánchez, and Francisco Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17:255–287, 01 2011.

Luis M. Candanedo, Véronique Feldheim, and Dominique Deramaix. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and Buildings*, 140:81–97, 2017. doi: DOI:10.1016/j.enbuild.2017.01.083.

Arjun Chandra and Xin Yao. Ensemble learning using multi-objective evolutionary algorithms. *Journal of Mathematical Modelling and Algorithms*, 5(4):417–445, 2006. DOI:10.1007/s10852-005-9020-3.

Gabriel Colonna, Eduardo Nakamura, Marco Cristo, and Marcelo Gordo. Anuran Calls (MFCCs) Data Set, 2017. `https://archive.ics.uci.edu/ml` (last seen on 2018/11/10).

W.J. Conover. *Practical nonparametric statistics*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1980. ISBN 9780471028673. URL `https://books.google.pt/books?id=m54s2puW_5AC`.

L. Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, Nov 2012. ISSN 1053-5888. doi: 10.1109/MSP.2012.2211477.

Thomas G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the Multiple Classifier Systems, First International Workshop, (MCS'00), June 21-23 2000, Cagliari, Italy*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2000. doi: 10.1007/3-540-45014-9_1. URL `https://doi.org/10.1007/3-540-45014-9_1`.

Xavier Fontes and Daniel Castro Silva. Towards Hybrid Prediction over Time Series with Non-Periodic External Factors. In *Proceedings of the 5th International Conference on Time Series and Forecasting (ITISE'18), September 19–21, Granada, Spain*, pages 1431–1442, 2018.

Xavier Fontes and Daniel Castro Silva. EDGE: Evolutionary directed graph ensembles. *International Journal of Hybrid Intelligent Systems*, 15(4):243–256, 2019. DOI:10.3233/HIS-190273.

Christian Gagné, Michèle Sebag, Marc Schoenauer, and Marco Tomassini. Ensemble learning for free with evolutionary algorithms? In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'07), July 7-11 2007, London, England, UK*, pages 1782–1789. ACM, 2007. doi: 10.1145/1276958.1277317. URL `https://doi.org/10.1145/1276958.1277317`.

Kyung-Joong Kim and Sung-Bae Cho. Evolutionary ensemble of diverse artificial neural networks using speciation. *Neurocomputing*, 71(7-9):1604–1618, 2008. DOI:10.1016/j.neucom.2007.04.008.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR'15), May 7-9 2015, San Diego, CA, USA*, page 13, 2015.

Yung-Keun Kwon and Byung Ro Moon. Evolutionary ensemble for stock prediction. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'04), June 26-30 2004, Seattle, WA, USA*, volume 3103 of *Lecture Notes in Computer Science*, pages 1102–1113. Springer, 2004. doi: 10.1007/978-3-540-24855-2_120. URL `https://doi.org/10.1007/978-3-540-24855-2_120`.

Robi Polikar. *Ensemble Learning*, pages 1–34. Springer US, Boston, MA, 2012. ISBN 978-1-4419-9326-7. doi: 10.1007/978-1-4419-9326-7_1.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, USA, 3rd edition, 2009. ISBN 0136042597.

Jared Sylvester and Nitesh V. Chawla. Evolutionary ensemble creation and thinning. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'06), part of the IEEE World Congress on Computational Intelligence (WCCI'06), July 16-21 2006, Vancouver, BC, Canada*, pages 5148–5155. IEEE, 2006. doi: 10.1109/IJCNN.2006.247245. URL `https://doi.org/10.1109/IJCNN.2006.247245`.

Appendix A. Configurations for the Experimental Results Section

A.1 Configuration values for W-EDGE and EDGE comparison

NA (Not Applicable)

Parameter	W-EDGE	EDGE
NElite	2	2
NBottom	0	0
MutationRate	0.1	0.1
PopSize	20	20
NGenerations	100	100
Adam Learning Rate	0.001	NA
Optimizer Steps	2	NA
K-S p-value	0.5	NA

A.2 Configuration space of baseline

Parameter	Random Forest	Decision Tree	Gradient Boosting
Training examples (%)	80	80	80
Estimators	40, 100	-	40, 100
Criterion	Gini Impurity	Gini Impurity	Friedman MSE
Max Depth	None	None	3
Max Features	Sqrt	All Features	All Features
Learning Rate	-	-	0.001, 0.1
Loss	-	-	Deviance
Splitter	-	Best	-

A.3 Configuration space of W-EDGE

Parameter	Value
$NElite$	1
$NBottom$	0
$PopSize$	10, 20, 40
$NGenerations$	10, 20, 50
$AdamLR$	0.001
$OptimizerSteps$	2

A.4 Dataset Description

The different prefixes (ST - standard, H9 - higher than 9, L9 - lower than 9) denote the imbalance ration category of the dataset. The choice of datasets from different imbalance ratio categories was to simulate real-world data problems and gather a larger set of datasets.

Dataset Name	No. Samples	No. Features	No. Target Classes
Anuran	7195	22	10
H9-car-good	1728	6	2
H9-poker-8-9_vs_5	2075	10	2
H9-winequality-white-9_vs_4	168	11	2
H9-zoo-3	101	16	2
L9-haberman	306	3	2
L9-iris0	150	4	2
L9-new-thyroid1	215	5	2
L9-newthyroid2	215	5	2
ST-appendicitis	106	7	2
ST-haberman	306	3	2
ST-hepatitis	80	19	2
ST-iris	150	4	3
ST-monk-2	432	6	2
ST-newthyroid	215	5	3
ST-ring	7400	20	2
ST-shuttle-c2-vs-c4	129	9	2
ST-tae	151	5	3
ST-titanic	2201	3	2
ST-wine	178	13	3
ST-zoo	101	16	7

A.5 Further Experiments - Dataset Description

Similar to Table A.4, for new datasets used in further experiments.

Dataset Name	No. Samples	No. Features	No. Target Classes
ST-saheart	462	9	2
ST-led7digit	500	7	10
ST-balance	625	4	3
L9-pima	768	8	2
ST-mammographic	830	5	2
ST-vehicle	846	18	4
ST-german	1000	20	2
ST-flare	1066	11	6
H9-flare-F	1066	11	2
ST-contraceptive	1473	9	3
ST-yeast	1484	8	10
L9-yeast1	1484	8	2
H9-abalone19	4174	8	2
ST-yeast-1-4-5-8_vs_7	693	8	2
H9-winequality-red-4	1599	11	2
H9-abalone-19_vs_10-11-12-13	1622	8	2
ST-marketing	6876	13	9
ST-cleveland	297	13	5

A.6 Further Experiments - Results

Comparison of accuracy between Baseline and W-EDGE in the datasets from Table A.5.

Dataset Name	Baseline	W-EDGE
ST-saheart	70.97	**87.16**
ST-led7digit	75.40	**76.67**
ST-balance	86.72	**93.72**
L9-pima	77.27	**91.58**
ST-mammographic	86.75	**89.24**
ST-vehicle	78.00	**90.23**
ST-german	77.00	**90.07**
ST-flare	73.55	**81.93**
H9-flare-F	95.98	**97.06**
ST-contraceptive	55.86	**78.61**
ST-yeast	63.43	**83.38**
L9-yeast1	80.47	**89.49**
H9-abalone19	99.28	**99.74**
ST-yeast-1-4-5-8_vs_7	96.40	**98.27**
H9-winequality-red-4	96.56	**98.79**
H9-abalone-19_vs_10-11-12-13	98.15	**99.18**
ST-marketing	36.12	**67.70**
ST-cleveland	62.00	**82.55**