GA-Auto-PU: A Genetic Algorithm-based Automated Machine Learning System for Positive-Unlabeled Learning

Jack D. Saunders
School of Computing
University of Kent
Canterbury, Kent, United Kingdom
jds39@kent.ac.uk

Alex A. Freitas
School of Computing
University of Kent
Canterbury, Kent, United Kingdom
A.A.Freitas@kent.ac.uk

ABSTRACT

Positive-Unlabeled (PU) learning is a growing field of machine learning that now consists of numerous algorithms; the number is now so large that considering an extensive manual search to select the best algorithm for a given task is impractical. As such, the area of PU learning could benefit from an Automated Machine Learning (Auto-ML) system, which selects the best algorithm for a given input dataset, among a pre-defined set of candidate algorithms. This work proposes such with GA-Auto-PU, a Genetic Algorithm-based Auto-ML system that can generate PU learning algorithms. Experiments with 20 real-world datasets show that GA-Auto-PU significantly outperformed a state-of-the-art PU learning method.

CCS CONCEPTS

• Computing methodologies → Genetic algorithms; • Computational methodologies → Machine learning;

KEYWORDS

Genetic algorithms, Machine Learning, Auto-ML, Classification

ACM Reference format:

Jack D. Saunders and Alex A. Freitas. 2022. GA-Auto-PU: A Genetic Algorithm-based Automated Machine Learning System for Positive-Unlabeled Learning. In *Proceedings of the Genetic and Evolutionary Computation Conference 2022 (GECCO '22). ACM, Boston, MA, USA, 4 pages.* https://doi.org/10.1145/1234567890

1 Introduction

Positive-Unlabeled (PU) learning aims to learn classifiers from positive and unlabeled data [1], differing from binary classification due to the absence of a negative set. The *unlabeled* set consists of instances which can be positive or negative, but whose label is unknown. Thus, learning a classifier that can accurately predict the class is challenging. PU data occurs in many application domains, such as bioinformatics [2], text classification [3], pharmacology [4], etc. [1], due to the impracticality of obtaining fully labeled data.

Many PU learning algorithms have been proposed in the literature [1]. However, finding the optimal algorithm for a specific classification dataset is a challenge. Thus, the PU learning area could benefit from an Automated Machine Learning (Auto-ML) system, which selects the best algorithm for a dataset, among a set of candidate algorithms. Currently, no Auto-ML system for PU learning exists. We address this gap with GA-Auto-PU, a Genetic Algorithm-based Auto-ML system.

2 Background on PU learning and Auto-ML

Positive-Unlabeled (PU) learning is a field of machine learning that learns models from datasets of positive and unlabeled instances [1]. A standard classifier requires data with two class labels, so, when built with a PU dataset, will treat the unlabeled set as a separate class and predict the probability of an instance being labeled instead of the probability of it being positive ($\mathbf{P}(y = \mathbf{1})$) [2]. In contrast, PU learning models are trained to predict $\mathbf{P}(y = \mathbf{1})$ using PU data.

The 3 main PU learning approaches are: the two-step approach, biased learning, and class prior methods [1]. GA-Auto-PU focuses on the popular two-step approach. Step 1 finds reliable negative (RN) instances in the unlabeled set, i.e., instances that substantially differ from the labeled positives and are likely not unlabeled positive instances. Step 2 builds a classifier to distinguish labeled positive instances from the RN set [1]. Whilst the literature generally refers to Step 1 and Step 2 when discussing two-step methods, we reference the steps as phases and recognize that "Step 1" often consists of two distinct phases. We refer to Phase I-A, used to extract an initial RN set; Phase I-B, an optional step using the initial RN set to extract more RN instances from the unlabeled set; and Phase II, "Step 2" in the usual description.

Automated Machine Learning (Auto-ML) is an emerging ML field that looks to limit the human involvement in ML applications, allowing users without extensive ML knowledge to operate complex ML pipelines [5]. No algorithm achieves good performance on all learning tasks, and algorithm performance is largely dependent on input data [6]. Auto-ML can lessen these issues by searching for the best algorithm specific to the target ML task. Many Auto-ML systems have been developed [5], e.g., the Tree-based Pipeline Optimization Tool (TPOT) [7] and the Resilient Classification Pipeline Evolution System (RECIPE) [8].

3 The GA-Auto-PU System

GA-Auto-PU is a generational Genetic Algorithm (GA) that evolves a PU learning algorithm. The GA uses standard uniform crossover and mutation as search operators, but the individual representation, fitness evaluation, tournament selection and elitism are tailored to PU learning, so these are described next.

3.1 Individual Representation

An individual is a two-step PU learning method. Each method has 3 components: phase I-A, phase I-B, and phase II. Phase I-A has 3 parameters: iteration_count_1A: the number of sets to split the unlabeled set into when learning a classifier to distinguish the positive and the unlabeled set; threshold_1A: the predicted positive class probability under which an instance is considered a reliable negative (RN) instance; and classifier_1A, used to predict the RN instances. Phase I-B has 3 parameters: threshold_1B, classifier_1B, and flag_1B. The threshold and the classifier are akin to those in phase I-A. The flag_1B parameter indicates whether to skip phase I-B. Phase II has one parameter: classifier_2, trained to distinguish the positive and the RN sets. Fig. 1 shows the individual encoding.

$$\big[[\text{phase I} - \text{A}], [\text{phase I} - \text{B}], \text{classifier\_2}\big]$$
$$\text{phase I} - \text{A}: [\text{iteration\_count\_1A}, \text{threshold\_1A}, \text{classifier\_1A}]$$
$$\text{phase I} - \text{B}: [\text{threshold\_1B}, \text{classifier\_1B}, \text{flag\_1B}]$$

Fig. 1. Individual representation (linear encoding). [phase I-A] and [phase I-B] are nested components of the individual.

3 genes (Classifier_1A, Classifier_1B, and Classifier_2) can take the same set of values, representing 18 different classification algorithms (called "Candidate_classifiers"): Gaussian naïve Bayes, Random forest, Decision tree, Multilayer perceptron, Support vector machine, Stochastic gradient descent classifier, Logistic regression, K-nearest neighbor, Deep forest, AdaBoost, Gradient boosting classifier, Linear discriminant analysis, Extra tree classifier, Extra trees classifier (ensemble of Extra trees), Bagging classifier, Bernoulli naïve Bayes, Gaussian process classifier, and Histogram-based gradient boosting classification tree.

The candidate values of each gene in the individual representation (defining the GA's search space) are as follows:

Iteration_count_1A: { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }
Threshold_1A: {0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5 }
Classifier_1A: { Candidate_classifiers }
Threshold_1B: {0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5 }
Classifier_1B: { Candidate_classifiers}
Flag_1B: { True, False }
Classifier_2: { Candidate_classifiers }

The size of the GA's search space is thus 11,664,000 solutions (calculated as 10 × 10 × 18 × 10 × 18 × 2 × 18).

3.2 Fitness Evaluation

GA-Auto-PU uses a multi-objective fitness function, with two objectives to be optimized: F-measure and recall, defined as:

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \qquad (1)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (2)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad (3)$$

where TP is the number of true positives; FP is the number of false positives; and FN is the number of false negatives.

F-measure is arguably the most used metric in the PU learning literature [1] and is the primary metric to be optimized in this work as it considers the recall/precision trade-off. Yet, it considers recall and precision equally, which is sub-par for our datasets, involving in general disease prediction. We focus on correctly predicting unlabeled positives, so recall of the positive class has priority over precision. Reducing false negatives rather than false positives is the priority as false negatives represent undiagnosed patients.

Yet, it is crucial to maximize precision and recall, since too low precision is undesirable. We could alter the F-measure, weighting recall higher than precision, but this would require setting ad-hoc weights for recall and precision. Instead, we use a tournament selection process based on F-measure and recall, described below.

Prioritizing recall poses the question: why not use recall as the primary and precision as the secondary metric? Experiments with this approach led to the GA favoring individuals that classified all instances as positive, with very low precision. So, we implemented F-measure as the primary metric to ensure that precision was still considered, whilst maximizing recall as the secondary metric.

Recall that each individual encodes 3 classification algorithms, which are used in phases I-A, I-B and II of the PU learning system. Fitness evaluation involves applying these algorithms (possibly excluding the algorithm for the optional Phase I-B) to the training set. To describe this process, we will use the following notation:

RN: The set of reliable negative instances.

P: The set of labeled positive instances.

U: The set of unlabeled instances.

P+RN: The combined set of *P* and *RN*.

P(y=1): The probability of an instance belonging to the positive class, as calculated by the classifier.

Pseudocode 1 Assess Fitness (Individual, Training set)

```
1. Split Training set into 5 Learning and Validation sets;
2. For each Learning set and corresponding Validation set:
   a. P = all labeled positive instances in Learning set;
   b. U = all unlabeled instances in Learning set;
   c. RN, U = Phase I-A(P, U);  // call Pseudocode 2
   d. If Flag_1B then RN, U = Phase I-B(P+RN, U);  // call Pseudocode 3
   e. Train Classifier_2 to distinguish P and RN;
   f. Classify Validation set;
3. Individual's Fitness Values = Average F-measure, Recall;
```

Pseudocode 2 Phase I-A(P, U)

```
1. RN = { }; // RN is initialized with empty set
2. Sets = split U into Iteration_count_1A subsets;
3. For every Set in Sets:
   a. Train Classifier_1A on P and Set;
   b. Classify all unlabeled instances in Set;
   c. For each unlabeled Instance in Set:
      i. If $P(y=1) < Threshold\_1A$
            then $RN = RN \cup Instance$, $U = U - Instance$;
4. Return RN, U;
```

Pseudocode 3 Phase I-B(P+RN, U)

```
1. Train Classifier_1B on P+RN;
2. Use the trained classifier to Classify U;
3. For each Instance in U:
   a. If $P(y=1) < Threshold\_1B$
         then $RN = RN \cup Instance$,  $U = U - Instance$
4. Return RN, U;
```

Pseudocode 1 shows how an individual's fitness is computed; Pseudocodes 2 and 3 show Phase I-A and Phase I-B, respectively. Pseudocode 1 applies internal cross-validation on the training set; thus, in step 3, the individual is assigned the average values of F-measure and recall over the 5 validation sets of the cross-validation. A solution with no identified RN instances will have a fitness of 0.

3.3 Tournament selection and elitism

The proposed tournament selection process utilizes a lexicographic multi-objective optimization approach, considering F-measure as a higher-priority objective and recall as a lower-priority objective; i.e. these objectives are optimized in their order of priority [9].

Between two individuals in a tournament, if one F-measure is substantially higher, it is deemed the winner. However, if the difference is insubstantial, we consider their recall, deeming one the winner if its recall is substantially higher. Else, if the differences in both measures are irrelevant, F-measure determines the winner.

Lexicographic optimization involves a 'tolerance threshold' by which the difference between two values is deemed practically irrelevant [9]. To avoid arbitrarily setting this, we use Cohen's d measure of 'effect size' [10] to decide if the difference is relevant, as shown in Pseudocode 4 and Equation (4), where "Metric" is the F-measure or recall of two individuals (denoted by indices 1 and 2) in a tournament and SD is the standard deviation:

$$\text{Cohen's d (Metric)} = \left| \frac{\text{Metric}_1 - \text{Metric}_2}{SD_{pooled}} \right| \quad (4)$$

$$SD_{pooled} = \sqrt{\frac{SD_1^2 + SD_2^2}{2}} \quad (5)$$

Using Cohen's d allows us to use a widely accepted threshold of 0.2 to indicate relevance. Precisely, a Cohen's d < 0.2 is considered a small effect size [10], which is considered an irrelevant difference of F-measure or recall in our case; whilst a Cohen's d ≥ 0.2 is considered a relevant difference of F-measure or recall, enough to select a tournament winner based on that measure.

Pseudocode 4 Tournament_Selection(Indiv1, Indiv2)

```
1. If Cohen's_d(F-measure) ≥ 0.2
2.    Then Winner = Indiv with highest F-measure;
3.    Else If Cohen's_d(Recall) ≥ 0.2
          a. Then Winner = Indiv with highest Recall
          b. Else Winner = Indiv with highest F-measure
```

The elitism procedure is also based on lexicographic optimization, but selecting the best individual out of the entire current population, instead of between only two individuals in a tournament.

4 Datasets and Experimental Methodology

Table 1 shows the 20 datasets used to assess GA-Auto-PU, with 13 UCI benchmarks [11] and 7 biomedical datasets. The training set of each dataset consists of an unlabeled set of all negative instances and 20% (randomly sampled) of the positive instances (their label hidden), and a positive set consisting of the remaining 80% positive instances. The positive and negative class labels in the test sets remain unchanged. The last column of Table 1 shows the % of positive instances in the dataset before positive instances are hidden in the unlabeled training set.

Table 1. Dataset characteristics.

Dataset	No. instances	No. features	Positive class %
Alzheimer's [12]	354	9	10.73%
Autism [11]	288	15	48.26%
Breast cancer Coi. [11]	116	9	55.17%
Breast cancer Wis. [11]	569	30	37.26%
Breast cancer mut. [13]	1416	53	32.42%
Cervical cancer [11]	668	30	2.54%
Cirrhosis [14]	277	17	25.72%
Dermatology [11]	359	34	13.41%
Pima I. Diabetes [11]	769	8	34.90%
Early Diabetes [15]	521	17	61.54%
Heart Disease [11]	304	13	54.46%
Heart Failure [16]	300	12	32.11%
Hepatitis C [11]	590	13	9.51%
Kidney Disease [11]	159	24	27.22%
Liver Disease [11]	580	11	71.50%
Maternal Risk [11]	1014	6	26.82%
Parkinson's [11]	196	22	75.38%
Parkinson's Biom. [17]	131	29	23.08%
Spine [11]	311	6	48.39%
Stroke [18]	3427	15	5.25%

The experiments use a stratified 5-fold cross-validation procedure. For each training set, GA-Auto-PU evolves the best PU method configuration before building a classifier from the training set with that configuration. The classifier is then used to predict the class of all instances in the test set. This process is repeated for all pairs of training and test sets in the 5-fold cross-validation, and the reported results are the average over the 5 test set results.

GA-Auto-PU parameters were set as follows: generation count: 50, population size: 101, uniform crossover probability: 0.9, gene crossover probability: 0.5, gene mutation probability: 0.1, tournament size: 2. GA-Auto-PU is compared to a state-of-the-art deep forest PU learning method [19], DF for short. DF learns from the same training sets given to GA-Auto-PU and is evaluated using the same cross-validation folds. The hyperparameter settings of the DF method were kept as described in [19]. For each performance measure, we compare the performance of GA-Auto-PU against DF, using the non-parametric Wilcoxon Signed-Rank test [20].

5 Computational Results and Discussion

Here we report results across the 20 datasets. GA-Auto-PU is referred to as "GA". Table 2 shows F-measure values. GA won 18; DF won 2 – a statistically significance difference, p=0.0001.

Table 3 shows recall and precision. Despite DF's poor F-measure, its recall was the highest in all datasets with statistical significance, p=0.00001, but at a high cost to precision as GA outperformed DF 19 times with statistical significance, p=0.0001. The difference was very large: greater than 30% on 12 datasets, and greater than 50% on 8. DF's poor performance is likely due to its hyperparameters, choosing 20% of the unlabeled instances as the negative set and choosing 1% of instances as the reliable negatives (RN) in Phase I-A. This may work for large, imbalanced datasets but is not always applicable as it may identify few RN instances, resulting in a classifier which over-predicts the positive class. This shows the value of hyperparameter tuning, as done by our system.

Table 2. F-measure values obtained by the methods.

Dataset	GA F-measure	DF F-measure
Alzheimer’s	**0.587**	0.188
Autism	**0.974**	0.648
Breast cancer Coi.	**0.709**	0.697
Breast cancer Wis.	**0.963**	0.543
Breast cancer mut.	**0.889**	0.489
Cervical cancer	**0.867**	0.061
Cirrhosis	**0.534**	0.405
Dermatology	**0.860**	0.228
Pima I. Diabetes	**0.581**	0.436
Early Diabetes	**0.960**	0.762
Heart Disease	**0.742**	0.607
Heart Failure	**0.770**	0.487
Hepatitis C	**0.936**	0.176
Kidney disease	**0.964**	0.428
Liver disease	0.828	**0.833**
Maternal health	**0.856**	0.403
Parkinson’s	**0.937**	0.856
Parkinson's Biom	0.209	**0.354**
Spine	**0.951**	0.652
Stroke	**0.225**	0.086
Total wins	18	2

Table 3. Recall and precision values obtained by the methods.

Dataset	GA recall	DF recall	GA precision	DF precision
Alzheimer’s	0.579	**0.947**	**0.595**	0.104
Autism	0.957	**0.9923**	**0.993**	0.481
Breast cancer C	0.781	**0.969**	**0.649**	0.544
Breast cancer W	0.943	**1.000**	**0.980**	0.373
Breast cancer M	0.961	**0.998**	**0.827**	0.324
Cervical cancer	0.765	**0.882**	**1.000**	0.032
Cirrhosis	0.662	**0.986**	**0.448**	0.255
Dermatology	0.833	**0.958**	**0.889**	0.130
Pima I Diabetes	0.783	**1.000**	**0.461**	0.279
Early Diabetes	0.950	**0.994**	**0.974**	0.618
Heart Disease	0.904	**1.000**	**0.629**	0.436
Heart Failure	0.750	**1.000**	**0.791**	0.322
Hepatitis C	0.911	**0.982**	**0.962**	0.097
Kidney disease	0.930	**1.000**	**1.000**	0.272
Liver disease	0.973	**1.000**	**0.721**	0.715
Maternal health	0.860	**0.941**	**0.851**	0.257
Parkinson’s	0.952	**0.993**	**0.921**	0.753
Parkinson's Biom	0.233	**0.933**	0.189	**0.219**
Spine	0.973	**1.000**	**0.930**	0.484
Stroke	0.322	**0.811**	**0.173**	0.045
Total wins	0	20	19	1

6 Conclusions

This paper introduced GA-Auto-PU, the first Auto-ML system for PU learning, which automatically builds a two-step PU learning method for a given dataset. In experiments with 20 datasets, GA-Auto-PU obtained significantly higher F-measure and precision values than DF, whilst DF obtained significantly higher recall values than GA-Auto-PU. Yet, overall, DF had very poor precision and F-measure (the latter is the primary metric in PU-learning). Currently, GA-Auto-PU is limited to only two-step PU learning methods. Future versions of the system will incorporate other PU learning approaches, such as biased learning.

GA-Auto-PU code is available at github.com/jds39/GA-Auto-PU.

REFERENCES

[1] Bekker, J. and Davis, J., 2020. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109(4), 719-760.

[2] Elkan, C. and Noto, K., 2008. Learning classifiers from only positive and unlabeled data. In *Proc. 14th ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining*, 213-220.

[3] Li, X. and Liu, B., 2003. Learning to classify texts using positive and unlabeled data. In *Proc 18th Int. Joint Conf. on Artif. Intel.*. 3, 587-592.

[4] Zheng, Y., et al., 2019. DDI-PULearn: a positive-unlabeled learning method for large-scale prediction of drug-drug interactions. *BMC Bioinformatics*, *20*(19),1-12.

[5] He, X., Zhao, K. and Chu, X., 2021. AutoML: A Survey of the State-of-the-Art. *Knowledge-Based Systems*, *212*, article 106622.

[6] Brazdil, P., et al., 2008. Metalearning: Applications to data mining. Springer.

[7] Olson, R.S., et al., 2016, July. Evaluation of a tree-based pipeline optimization tool for automating data science. In *Proc. of the Genetic and Evolutionary Computation Conf. 2016*, 485-492.

[8] de Sá, A.G., et al., 2017. RECIPE: a grammar-based framework for automatically evolving classification pipelines. *Proc. European Conf. on Genetic Programming*, 246-261.

[9] Freitas, A.A., 2004. A critical review of multi-objective optimization in data mining: a position paper. *ACM SIGKDD Explorations Newsletter, 6(2)*, 77-86.

[10] Ellis, P.D., 2010. The essential guide to effect sizes. CUP.

[11] Asuncion, A., Newman, D., 2007. UCI machine learning repository.

[12] Marcus, D.S. et al., 2010. Open access series of imaging studies: longitudinal MRI data in nondemented and demented older adults. *Journal of Cognitive Neuroscience*, 22(12), 2677-2684.

[13] Pereira, B., et al., 2016. The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. *Nature Communications*, *7*(1), 1-16.

[14] Fleming, T.R. and Harrington, D.P., 1991. Counting Processes and Survival Analysis. John Wiley and Sons.

[15] Islam, M.F., et al, 2020. Likelihood prediction of diabetes at early stage using data mining techniques. In *Computer Vision and Machine Intelligence in Medical Image Analysis*, 113-125.

[16] Chicco, D., Jurman, G., 2020. Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. *BMC Medical Informatics and Decision Making*, *20*(1), 1-16.

[17] Hlavnička, J. et al., 2017. Automated analysis of connected speech reveals early biomarkers of Parkinson’s disease in patients with rapid eye movement sleep behaviour disorder. *Scientific Reports*, *7*(1), 1-13.

[18] Emon, M.U., et al. 2020. Performance Analysis of Machine Learning Approaches in Stroke Prediction. In *Proc. 4th Intern. Conf. on Electronics, Communic. and Aerospace Technol. (ICECA)*, 1464-1469.

[19] Zeng, X., et al., 2020. Predicting disease-associated circular RNAs using deep forests combined with positive-unlabeled learning methods. *Briefings in Bioinformatics*, 21(4), 1425-1436.

[20] Wilcoxon, F., et al., 1963. Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test. *Selected tables in mathematical statistics*, 1, 171-259.