EvoFeat: Genetic Programming based Feature Engineering Approach to Tabular Data

Hengzhe Zhang, Qi Chen, Bing Xue, Yan Wang, Aimin Zhou, Mengjie Zhang

Victoria University of Wellington

02/12/2024

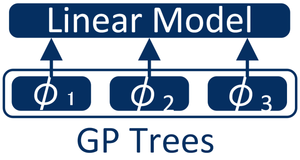
Table of Contents

Introduction

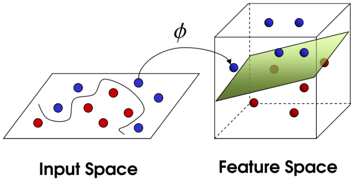
Motivation

- Linear models assume linear relationships.
- Decision trees assume axis-parallel decision boundaries.
- Real-world data often violates these assumptions.

Objective: Construct a set of new features, $\{\phi_1, \ldots, \phi_m\}$, to *enhance learning* on the dataset $\{\{x_1, y_1\}, \ldots, \{x_n, y_n\}\}$ compared to learning on the original features $\{x^1, \ldots, x^p\}$.



(a) Feature Construction on Linear Regression



(b) New Feature Space

- **Manual Design:** Based on domain knowledge.
- **Kernel Methods:** Use kernel tricks to transform data into higher dimensions.
- **Deep Learning:** Leverages neural networks to learn features automatically [1].

Limitations

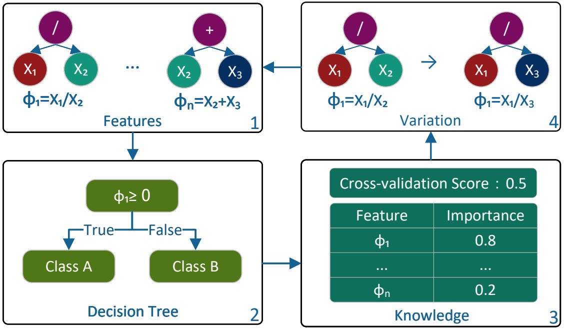
- Manual Design: Labor-intensive.
- Kernel Methods: Hard to integrate with tree-based methods.
- Deep Learning: Requires large datasets, effectiveness debatable for small, heterogeneous datasets [2].

[1] Jianxun Lian et al., *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018)

[2] Yury Gorishniy et al., *Advances in Neural Information Processing Systems* (2021)

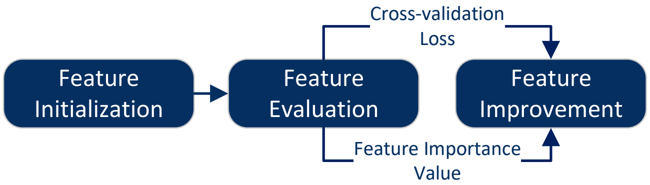
Our Approach: EvoFeat

- Constructs nonlinear features with genetic programming (GP).
- Advantages: Gradient-free, interpretable, and flexible.
- Enhances ensemble learning models.
- Uses cross-validation and feature importance for evaluation.



Preliminaries

- **Feature Initialization:**
  - Construct initial features based on domain knowledge or randomly.
- **Feature Evaluation:**
  - Evaluate features using cross-validation and calculate feature importance.
- **Feature Improvement:**
  - Discard ineffective features and replace them with new ones derived from important features.



Feature engineering workflow.

- **Cross-Validation:**
  - Evaluates generalization performance.
- **Feature Importance:**
  - Identifies useful features.
  - Risky to rely solely on feature importance.

Key Insight

Constructing multiple sets of features and evaluating them using cross-validation can provide better insights into their generalization capabilities.

The Proposed Algorithm

- **Symbolic Trees:**
  - Each individual has *k* GP trees representing *k* new features.
- **Tree Structure:**
  - Non-leaf nodes: Functions (e.g., $+$, $-$, $*$, $\log$, $\sin$).
  - Leaf nodes: Original Features.
- **Base Learners:**
  - Decision trees or linear regression models.

Initialization

Randomly initialize N individuals, each with k symbolic trees.

Evaluation

- Evaluate individuals using cross-validation loss.
- Calculate feature importance for each feature.

Algorithm Framework

Selection

Use lexicase selection [1] to select parent individuals based on cross-validation losses.

Generation

Generate new individuals using self-competitive crossover and guided mutation [2].

Archive Update

Update archive with top-performing models using reduce-error pruning [3].

---

[1] William La Cava et al., *Evolutionary Computation* (2019)
[2] Hengzhe Zhang et al., *IEEE Transactions on Evolutionary Computation* (2023)
[3] Rich Caruana et al., *Proceedings of the Twenty-First International Conference on Machine Learning* (2004)

Feature Initialization

- **Initialization Strategy:**
  - Ramped-half-and-half for symbolic trees.
  - Half full trees, half random depth.
- **Base Learner Assignment:**
  - Randomly assign **decision tree** or **linear regression model**.

- **Cross-Validation:**
  - Partition the training set into five folds.
  - Train on four folds, validate on one fold.
- **Loss Function:**
  - Cross entropy:

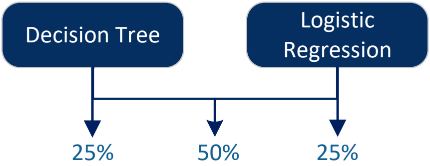
$$\sum_{c \in C} p_c * \log(q_c), \tag{1}$$

  - Where p_c is the true probability, q_c is the predicted probability.

- **Decision Tree:**
  - Calculated by the total reduction of Gini impurity contributed by each feature ϕ.
- **Logistic Regression:**
  - Calculated by the absolute value of the model coefficients.
  - Features are standardized to ensure equal influence on the coefficients.

- Three selection operators in EvoFeat:
  - Base Learner Selection
  - Individual Selection: Lexicase Selection
  - Feature Selection: Softmax Selection

- Divide population into two subgroups (decision trees, logistic regression).
- Random mating probability ($rmp = 0.5$):
  - 50%: Select parents from different subgroups.
  - 50%: Select parents from the same subgroup.



Multitask GP

- Selects individuals based on a vector of cross-validation losses, one for each instance.
- Constructs filters based on each loss value [1]:

$$\tau_j = \min_i \mathcal{L}_j^i + \epsilon_j, \tag{2}$$

- Where:
  - τ_j is the threshold,
  - $\mathcal{L}_j^i$ is the loss of the i-th individual on the j-th instance,
  - ϵ_j is the median absolute deviation.

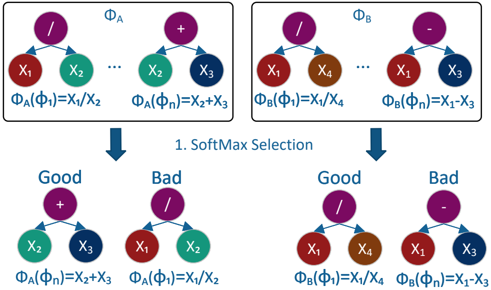
[1] William La Cava et al., *Evolutionary Computation* (2019)

SOFTMAX SELECTION

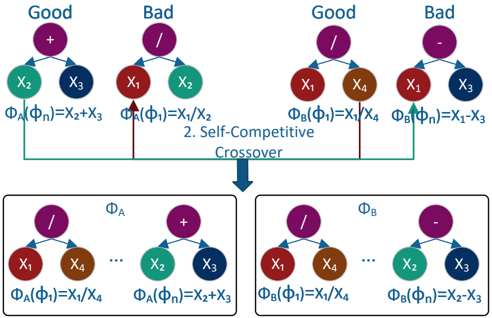
- Select features based on importance values $\{\theta_1, \ldots, \theta_k\}$.
- Uses softmax function:

$$P(\theta_i) = \frac{e^{\theta_i/T}}{\sum_{j=1}^{k} e^{\theta_j/T}}, \tag{3}$$

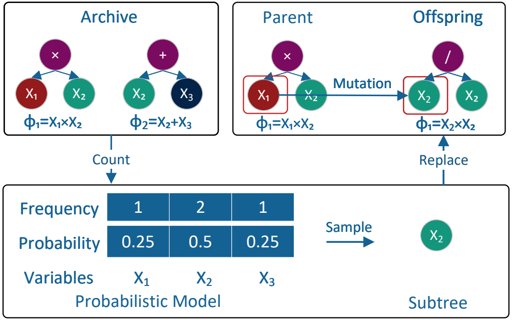
- Good features are sampled by $P(\theta_i)$, bad features by $P(-\theta_i)$.



- **Self-Competitive Crossover:**
  - Transfers beneficial material from good features to bad features.
  - Biased crossover, only modifies bad features, preserving good features.
  - Ensures top-performing features are preserved.



- **Guided Mutation:**
  - Replaces a subtree with a randomly generated subtree.
  - Uses a guided probability vector for terminal variable selection.
  - The probability vector corresponds to the terminal usage of archived individuals.



Experiments

- **Objective:** Compare EvoFeat with popular machine learning and deep learning methods.
- **Datasets:** 130 datasets from DIGEN and PMLB benchmarks.
  - **DIGEN** [1]**:**
    - A total of 40 diverse synthetic datasets generated using genetic programming.
  - **PMLB** [2]**:**
    - Collection of real-world datasets from OpenML.
    - Focus on classification tasks with more than 200 instances.
    - A total of 90 datasets selected where the product of the number of instances and the number of features is less than 10^5 due to memory constraints.

[1] https://github.com/EpistasisLab/digen
[2] https://github.com/EpistasisLab/pmlb

Experimental Settings

- **Evaluation Protocol:**
  - 80% training, 20% testing.
  - 5-fold cross-validation on the training set for parameter tuning.
  - Repeat experiments with 30 random seeds.
- **Hyperparameter Tuning:**
  - Use Heteroscedastic Evolutionary Bayesian Optimization (HEBO) [1] for tuning baseline algorithms.

[1] Alexander I Cowen-Rivers et al., *Journal of Artificial Intelligence Research* (2022)

- The detailed parameter space is shown in the paper.
- Below is an example parameter space for tuning.

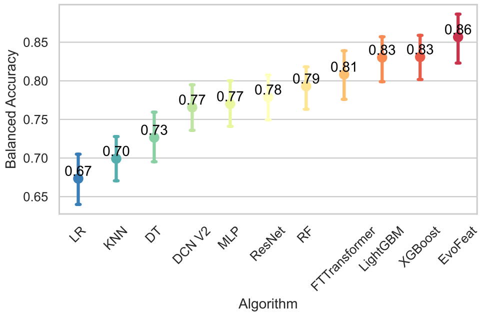
Parameter Space of FTTransformer

Hyperparameter	Range
Attention Dropout	Uniform[0,0.5]
Residual Dropout	Uniform[0,0.2]
FFN Dropout	Uniform[0,0.5]
FFN Factor	Uniform[$\frac{2}{3}$,$\frac{8}{3}$]
Token Dimension	UniformInt[64,512]
Layers	UniformInt[1,4]
Learning Rate	UniformLog[1e-4,1e-1]
Weight Decay	UniformLog[1e-6,1e-3]

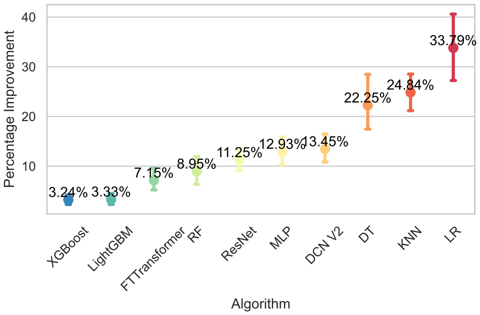
- **Machine Learning:**
  - XGBoost [1], LightGBM [2], Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), K-Nearest Neighbors (KNN).
- **Deep Learning:**
  - Multilayer Perceptron (MLP), ResNet, DCN V2 [3], FT-Transformer [4].

[1] Tianqi Chen and Carlos Guestrin, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016)
[2] Guolin Ke et al., *Advances in Neural Information Processing Systems* (2017)
[3] Ruoxi Wang et al., *Proceedings of the Web Conference 2021* (2021)
[4] Yury Gorishniy et al., *Advances in Neural Information Processing Systems* (2021)

- **Comparison:**
  - Evaluate EvoFeat against traditional and deep learning methods.
- **Results:**
  - EvoFeat outperforms state-of-the-art methods in average accuracy.
  - Demonstrates significant improvements in predictive performance.

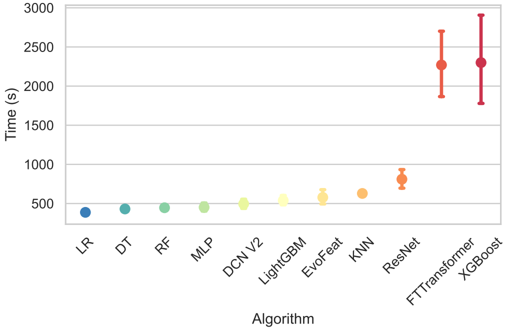


(a) Balanced testing accuracy.



(b) Improvement in accuracy.

- **Training Time:**
  - EvoFeat has comparable training time to a fine-tuned LightGBM.
  - EvoFeat is much faster than a fine-tuned FT-Transformer.



Training Time (seconds).

- **Baseline:** XGBoost, LightGBM, RF, DT, LR, KNN.
- **Results:**
  - EvoFeat achieves the best accuracy.
  - Significant improvements over XGBoost and LightGBM.

Statistical results of balanced testing accuracy on 90 PMLB and 40 DIGEN datasets.

	XGBoost	**LightGBM**	**RF**	**LR**	**KNN**	**EvoFeat**
DT	0/48/82	2/47/81	0/43/87	60/36/34	60/27/43	0/34/96
XGBoost	—	13/107/10	43/79/8	72/50/8	107/16/7	4/67/59
LightGBM	—	—	45/75/10	74/42/14	107/15/8	5/72/53
RF	—	—	—	73/47/10	102/20/8	7/62/61
LR	—	—	—	—	54/13/63	7/44/79
KNN	—	—	—	—	—	3/15/112

Comparison with Deep Learning Methods

- **Baseline:** MLP, ResNet, DCN V2, FT-Transformer.
- **Results:**
  - Deep learning methods perform comparably to RF.
  - EvoFeat outperforms these deep learning methods significantly.

Statistical results of balanced testing accuracy on 90 PMLB and 40 DIGEN datasets.

	ResNet	DCN V2	FT-Transformer	EvoFeat
MLP	18/96/16	9/118/3	10/76/44	4/33/93
ResNet	—	8/99/23	46/73/11	3/32/95
DCN V2	—	—	45/79/6	2/35/93
FT-Transformer	—	—	—	4/34/92
EvoFeat	—	—	—	—

Ablation Studies

- **Objective:** Validate improvements from heterogeneous base learners and feature importance-guided search.
- **Components:**
  - **Heterogeneous base learners:** Compare EvoFeat with different combinations of base learners.
  - **Feature importance-guided search:** Evaluate the effectiveness of feature importance-guided operators.

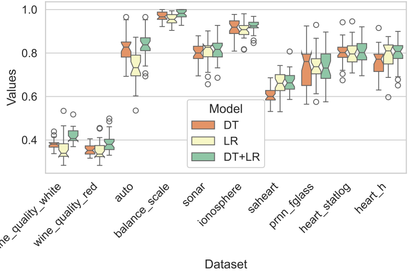
Base Learners

- **Objective:** Compare heterogeneous base learners (DT+LR) with single base learners (DT, LR).
- **Results:**
  - DT+LR achieves better average performance.
  - Significant improvements over single learners.

Comparison of balanced testing accuracy across different base learners on 90 PMLB datasets.

	LR	**DT+LR**
DT	12(+)/47($\sim$)/31(-)	0(+)/62($\sim$)/28(-)
LR	—	5(+)/70($\sim$)/15(-)

- **Objective:** Compare heterogeneous base learners (DT+LR) with single base learners (DT, LR).
- **Results:**
  - DT+LR achieves better average performance.
  - Significant improvements over single learners.



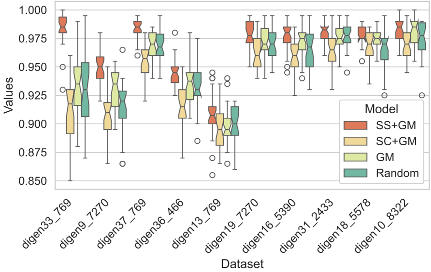
Balanced testing accuracy with different base learners.

- **Objective:** Evaluate the effectiveness of feature importance-guided operators.
- **Methods:**
  - Compare random crossover and mutation (Random) with softmax-based self-competitive crossover and guided mutation (SS+GM).
- **Results:**
  - Feature importance-guided search achieves better performance.

Comparison of balanced testing accuracy across different selection operators on 40 DIGEN datasets.

	SC+GM	**GM**	**Random**
SS+GM	12(+)/26(∼)/2(-)	5(+)/34(∼)/1(-)	12(+)/28(∼)/0(-)
SC+GM	—	0(+)/30(∼)/10(-)	5(+)/30(∼)/5(-)
GM	—	—	5(+)/35(∼)/0(-)

- **Objective:** Evaluate the effectiveness of feature importance-guided operators.
- **Methods:**
  - Compare random crossover and mutation (Random) with softmax-based self-competitive crossover and guided mutation (SS+GM).
- **Results:**
  - Feature importance-guided search achieves better performance.



Balanced testing accuracy with different selection operators.

- **Summary:**
  - EvoFeat outperforms state-of-the-art methods.
  - Heterogeneous base learners and feature importance-guided search improve performance.
- **Future Work:**
  - Investigate modularization techniques for improved interpretability.
  - Use diversity optimization to enhance ensemble performance.

Thanks for listening!

Email: Hengzhe.zhang@ecs.vuw.ac.nz

GitHub Project: https://github.com/hengzhe-zhang/EvolutionaryForest/