Genetic Programming-based Evolutionary Feature Construction for Heterogeneous Ensemble Learning (IEEE TEVC)

Hengzhe Zhang

Supervisor: Mengjie Zhang, Bing Xue, Qi Chen

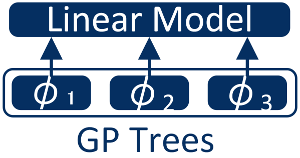
Victoria University of Wellington

29/06/2023

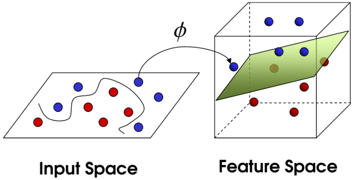
Table of Contents

Background

- The general idea of feature construction is to construct a set of new features $\{\phi_1, \ldots, \phi_m\}$ that improve the learning performance on a given dataset $\{\{x_1, y_1\}, \ldots, \{x_n, y_n\}\}$ compared to learning on the original features $\{x^1, \ldots, x^p\}$.
- Genetic programming (GP) has been widely used for automatic feature construction due to its flexible representation and gradient-free search mechanism.

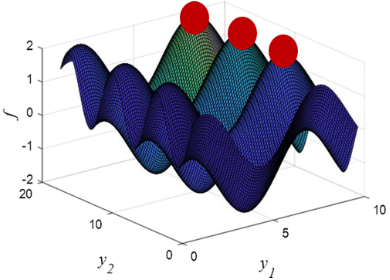


(a) Feature Construction on Linear Regression



(b) New Feature Space

- Motivation: An ensemble of multiple simple/weak GP trees is better than a single complex/strong GP tree.
- GP is naturally suited for ensemble learning because it can generate a diverse set of candidate solutions (models) through genetic operations in a single run.



Multi-modal Landscape on Training Data

Key Questions:

- How to define a base learner?
- How to select base learners?
- How to learn efficiently?

Algorithm Framework

Base Learner

Motivation:

- Decision trees are good at fitting piecewise data.
- Linear regression is good at fitting continuous data.
- Genetic programming is good at constructing features.

How to combine them?

- Combine decision tree and linear regression using gradient boosting.
- Use GP for feature construction.

Feature construction on a mixed base learner.

Gradient boosting:

- Train a linear regression model first.
- Learn the residual using a decision tree.

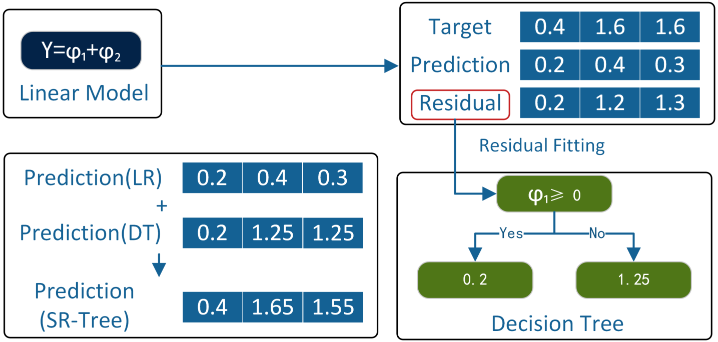


Illustration of the heterogeneous base learner.

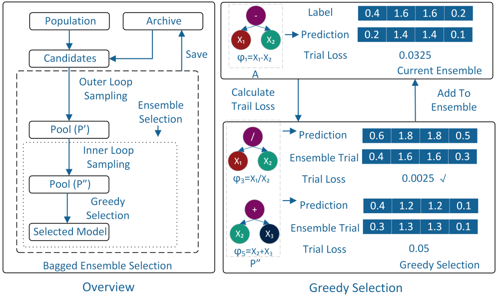
Greedy Ensemble Selection

Why select a subset?

- Not all learners in an ensemble model are accurate and diverse.

How to select a subset:

- Select Top-5 models.
- Select a model that minimizes training error on top of selected models.
- Repeat step 2 until reaching a termination criterion.



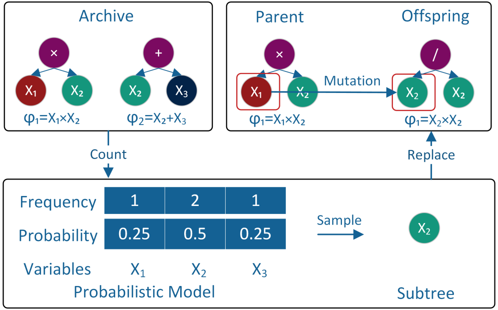
Ensemble Selection

Why need terminal variable selection?

- Not all variables in training data are useful!

How to select?

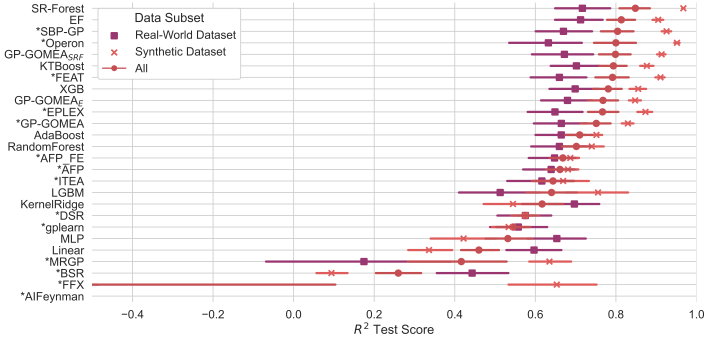
- Calculate the importance of each constructed feature.
- Calculate the relative frequency of all variables.
- Weight frequency by feature importance values.



Variable Selection

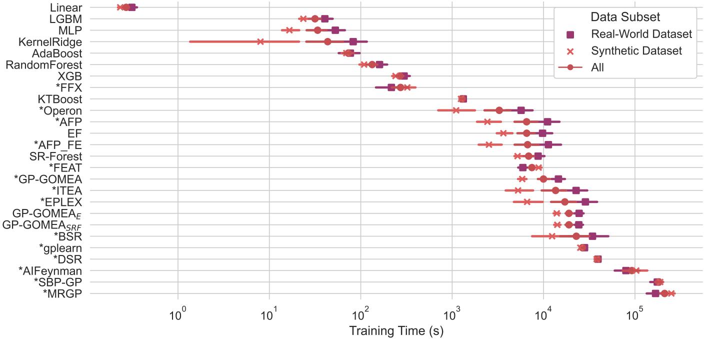
Experimental Results

SR-Forest is the best on 120 datasets on average in terms of R^2 scores.



Average R^2 scores on 120 regression datasets.

The cost of running time for SR-Forest ranks in the middle among algorithms.



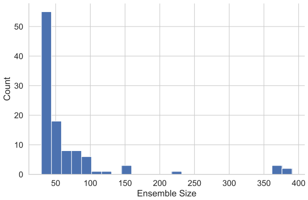
Average training time on 120 regression datasets.

- Feature construction on SR-Tree (DT+LR) is better than construction on Ridge (LR) in 22 out of 106 datasets.
- Feature construction on SR-Tree (DT+LR) is better than construction on RDT in 82 out of 106 datasets.

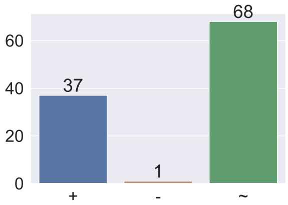
	Random-DT	**DT**
Random-DT	–	5(+)/89(∼)/12(-)
DT	12(+)/89(∼)/5(-)	–
Ridge	78(+)/20(∼)/8(-)	78(+)/21(∼)/7(-)
SR-Tree	82(+)/23(∼)/1(-)	84(+)/22(∼)/0(-)
	Ridge	**SR-Tree**
Random-DT	8(+)/20(∼)/78(-)	1(+)/23(∼)/82(-)
DT	7(+)/21(∼)/78(-)	0(+)/22(∼)/84(-)
Ridge	–	0(+)/84(∼)/22(-)
SR-Tree	22(+)/84(∼)/0(-)	–

Statistical Comparison on 106 regression datasets.

- Ensemble selection can reduce ensemble size from 100 to 30 on average.
- Ensemble selection delivers better performance in 37 datasets.



(a) Ensemble Size

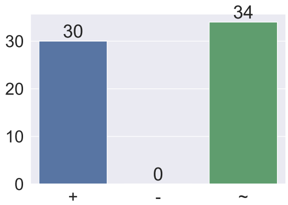


(b) Statistical Comparison on 106 datasets

- Importance-based Terminal selection (GM) delivers better performance in 30 datasets.
- Selecting only terminal variables is sufficient.

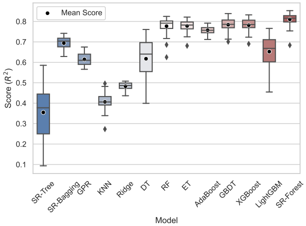
	GM	GM (F)
GM	–	15(+)/48(∼)/1(-)
GM (F)	1(+)/48(∼)/15(-)	–
GM (P+T)	1(+)/61(∼)/2(-)	20(+)/44(∼)/0(-)
Random	0(+)/34(∼)/30(-)	6(+)/38(∼)/20(-)
	GM (P+T)	**Random**
GM	2(+)/61(∼)/1(-)	30(+)/34(∼)/0(-)
GM (F)	0(+)/44(∼)/20(-)	20(+)/38(∼)/6(-)
GM (P+T)	–	33(+)/31(∼)/0(-)
Random	0(+)/31(∼)/33(-)	–

(a) Importance-based GM outperforms frequency-based GM in 15 datasets.

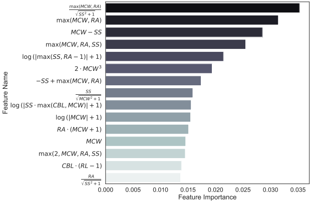


(b) GM outperforms Random in 30 datasets.

- Feature construction on heterogeneous ensemble can make it outperform other machine learning algorithms.
- Thanks to base learners providing feature importance values, we can visualize which constructed features are important.



(a) Performance Comparison



(b) Feature Importance

Conclusions

- Feature construction on a heterogeneous ensemble outperforms that on a homogeneous ensemble.
- Ensemble selection effectively reduces ensemble size while enhancing prediction performance.
- Utilizing feature importance-based variable selection (guided mutation) improves search effectiveness.

Thanks for listening!

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

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