Feature learning for interpretable, Performant Decision Trees

January 8, 2024

Reviewr : Park Seok Hun

Table of Contents

Table of Contents

Property of Decision Tree

- Decision Tree :

  depth $\downarrow$ ▶ Performance $\downarrow$, Interpretable $\uparrow$

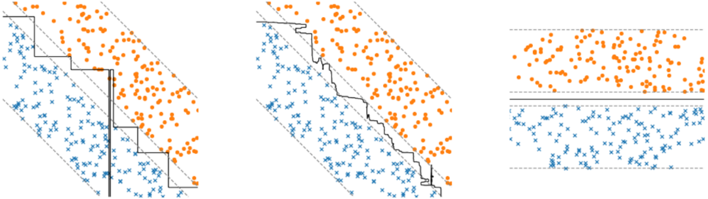
  depth $\uparrow$ ▶ Performance $\uparrow$, Interpretable $\downarrow$

- No matter how much the depth is increased, the performance on test data does not significantly outperform other models.

Introduction

- They propose an algorithm that, through **feature learning**, generates a single tree with an appropriate depth for a Decision Tree while achieving good performance.
- **Feature learning** means that in the decision tree training, instead of $X_j \leq c$, the algorithm contemplates $f(\mathbf{X}) \leq c$, and learning f during the training process where $\mathbf{X} = (X_1, ..., X_p)'$.

Example of feature transformation



(a) A decision tree is relatively complex and generalizes poorly.

(b) A random forest is very complex and generalizes better, but not perfectly.

(c) After rotation, a decision tree is simple and generalizes perfectly.

- Above picture demonstrates that through feature transformation, a single tree can achieve good performance.

Proposed method

- They propose alternating between learning a tree, similar to the CART, and performing feature learning based on gradients.
- For the feature learning, they consider Kernel Density Decision Tree(KDDT) which is differentiable.

Table of Contents

Fuzzy Decision Tree(FDT)

Problem of conventional Decision Tree(CART) :

Rule : If $X <= 3000$, X is classified as group A else B.

Then, $X = 2999$ and $X = 3001$ are classified as different group.

- This causes prediction errors for Decision Trees near the boundaries
- If CART deterministically split data into child node, FDT reflect the possibility of data being split into each child node (e.g., using probabilities)

Crisp Decision Tree

- Crisp Decision Tree : CART

Table of Contents

Kernel Density Decision Tree(KDDT)

- KDDT is a model that expresses the likelihood of splitting into child nodes using a kernel to represent probabilities.
- Note that unlike CART, KDDT is the differentiable model.

Example
Let $X \in \mathbb{R}$ be a input vector. Then, we have
$\mathbb{I}(X \in [a_j, b_j]) \to F(X - a_j) - F(X - b_j)$ where F is cdf of normal.

Table of Contents

Feature learning

- Any differentiable parameterized class of feature transforms can be used.
- example : Linear transformation $\mathbf{X} \to A\mathbf{X} + b$
- When training rule in the KDDT, feature learning is conducted based on gradient method for the impurity measure.

Table of Contents

Fuzzy into Crisp

- In the KDDT paper, they proposed the method for converting fuzzy decision trees to crisp decision trees, and it seems to be employed here.
- However, the details are not elaborated upon, and there is no code available.
- Note that the performance slightly degrades during the converting.

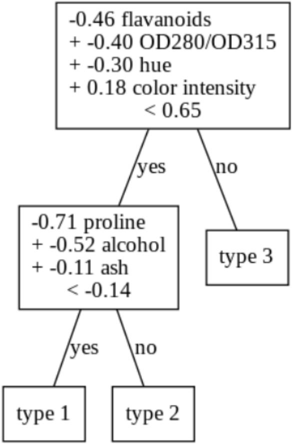
Table of Contents

Experiments

data n, p, q	LR	MLP	DT	RF	ET	XGB	ours: linear fuzzy	 crisp
iris [18] 150, 4 (4), 3	**0.960** -	0.953 -	0.947 6.4	0.947 7.2e2	0.953 2.1e3	0.947 4.3e2	**0.960** 6.1	**0.960** 7.6
heart-disease [30] 303, 13 (20), 2	**0.822** -	0.792 -	0.707 13.9	0.802 4.8e3	0.795 1.1e4	0.792 7.9e2	0.812 21.6	0.812 19.4
dry-bean [31] 13611, 16 (16), 7	0.925 -	**0.934** -	0.912 99.8	0.923 6.7e4	0.921 2.0e5	0.928 1.3e4	0.920 1.1e2	0.913 45.8
wine [1] 178, 13 (13), 3	0.983 -	**0.989** -	0.904 8.5	0.977 9.4e2	**0.989** 3.3e3	0.955 2.4e2	0.983 2.0	0.983 2.0
car [5] 1728, 6 (21), 4	0.926 -	0.992 -	0.977 95.3	0.964 2.3e4	0.971 3.1e4	**0.994** 4.5e3	0.991 29.0	0.992 29.0
wdbc [44] 569, 30 (30), 2	0.974 -	**0.975** -	0.935 13.0	0.965 1.9e3	0.970 6.0e3	0.968 2.7e2	0.972 1.3	0.972 1.3
sonar [38] 208, 60 (60), 2	0.755 -	0.879 -	0.735 14.1	0.826 2.0e3	**0.880** 5.6e3	0.855 3.0e2	0.818 5.7	0.799 3.9
pendigits [2] 10992, 16 (16), 10	0.952 -	**0.994** -	0.964 3.2e2	0.993 3.8e4	**0.994** 9.8e4	0.991 8.5e3	0.981 2.6e2	0.976 2.4e2
ionosphere [39] 351, 34 (34), 2	0.875 -	0.917 -	0.892 15.5	0.934 2.2e3	**0.943** 5.9e3	**0.943** 3.4e2	0.932 3.9	0.920 5.5

- Even as a single tree, proposed model performs well.

Experiments



- The above results is from converted crisp proposed model.