# The Reborn Decision Trees

Jinxiong Zhang

December 5, 2019

## Overview

1. Motivation

2. Parameterization of Decision Trees

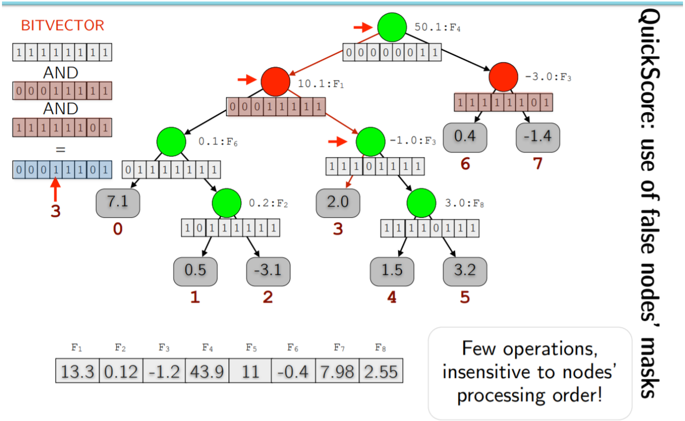
3. Extension

4. Current Results

# Decision Trees are not described in the language of computational graph

- Different inputs, different path and depth;
- Difficult to optimize;
- Logical rather than arithmetical;
- Natural to deal with categorical features and missing data;
- Many ensemble methods applied to decision trees.

# Bitvectors in QuickScorer



| F₁ | F₂ | F₃ | F₄ | F₅ | F₆ | F₇ | F₈ |
|---|---|---|---|---|---|---|---|
| 13.3 | 0.12 | -1.2 | 43.9 | 11 | -0.4 | 7.98 | 2.55 |

Few operations, insensitive to nodes' processing order!

QuickScore: use of false nodes' masks

# Parameterization of Decision Trees

There are 3 phases of such parameterization as following:

- Test phase: find the false nodes of a specific input sample;
- Traversal phase: apply the logical AND $\wedge$ to the bitvectors of the false nodes of the sample;
- Output phase: find the terminal node according to the leftmost element of the result at the last step.

In the test phase, it is the 'if-then' sentence or 'yes or no' question that matters rather than the features are numerical or not.
If it is false, we select the corresponding bitvectors of false node.

# How to digitalize the test phase?

For simplicity, we only consider the numerical features. We call the node is true if the feature $x_i$ is less than the threshold value $v_i$.

- If $x_i \leq v_i$, the node is true and we do not select its bitvector $b_i$.
- Otherwise we do select its bitvector $b_i$.
- In short, we can express it as $\sigma(x_i - v_i)b_i$ where $\sigma$ is the binarized ReLU or step function.
- The test phase is $B\sigma(Sx - t)$ where $S$ is the selection matrix and $t$ is the threshold vector; $B$ is the bitvector matrix.
- Each column of $S$ is elementary vector(also called one-hot vector); each column of $B$ is the bitvector of the corresponding node.

# The representation of decision trees in the computational graph

$$v[i], i = \arg\max(B\sigma(Sx - t)) \tag{1}$$

# The Research Goals

- How can we deal with the categorical attribute?
- Why it is the step function?
- Can we optimize the decision trees with gradient-based methods?
- What is the bitvector matrix $B$?
- Can we extend the binary matrix $S$ to more general real matrices?
- Can we replace the $\arg\max$ operator?
- What are the boosted decision trees?

# Some solution to the questions in last slide.

- We can consider the categorical attribute as numb variable.
- ReLU can play the role of step function $\sigma$
- In some special case, we can optimize the decision trees with gradient-based methods.
- We also can describe the oblique decision tree in the similar way.

Thanks.