Decision Trees and Categorical Independent Variables with Many Levels

Chao Guo

Abstract
This paper discusses a common concern regarding decision tree models, which is categorical independent variables with many levels. By dissecting the decision tree algorithm and running numerical simulations in R, we conclude that categorical variables with many levels do not always cause trouble, and any preprocessing on them should consider sample size, correlation among response and independent variables, and many other factors.

Keywords. decision tree; machine learning; categorical variables; credibility.

1. INTRODUCTION

Decision trees and other tree-based machine learning algorithms have gained popularity in the actuarial community. Due to their complexity and opaqueness, many practitioners rely on certain "rules of thumbs" to build their models. In this essay, we examine one particular well-circulated advice: categorical variables with many levels can cause trouble for tree-based models, so practitioners have to pre-process their data and trim down the number of levels. It turns out that there is much subtlety to this advice.

2. BACKGROUND AND METHODS

Decision trees and their related machine learning algorithms (e.g., random forest, gradient boosting tree) have gained popularity in the actuarial community in the past few years. This is partly due to the fact that tree-based models are capable of modeling more complicated relationship between the response and the independent variables. Also, computing power has become more accessible, and open source software has become more mature and reliable.

However, unlike generalized linear models (GLMs), tree-based models are often viewed as "black-box models". Indeed, tree-based models do not have coefficient estimates, so it is tricky to examine marginal effect of each individual independent variable. Their statistical properties are not as well-studied as GLMs either, making it difficult to perform statistical inference using tree-based models.

Perhaps because of the scarcity of reliable resources, practitioners often rely on and, in some cases, follow well-circulated "rules of thumb" without any question. This habit has two issues. First, many of the "the rules of thumb" simply do not hold. Second, even if they do hold, their context often gets lost. As a result, practitioners might be misguided and create a biased rating plan that is hard to explain to business partners as well as regulators.

One of the common beliefs is this: categorical variables with many levels can cause trouble for tree-based models. As a result, practitioners often feel compelled to do some kind of pre-processing, such as one-hot encoding. But how many levels is too many? Why does large cardinality cause trouble? Does large cardinality always cause trouble?

2.1 Decision Trees Revisited

To answer these questions, we first need to understand how a decision tree makes a split. For a detailed description, please refer to section 9.2 of reference 2. We only provide a sketch here.

```
For each variable $X_i$ in $X_1, X_2, \cdots, X_p$:
    1. If $X_i$ is numeric and has distinct values $x_1, x_2, \ldots, x_{i_m}$
        a. Examine all partitions $L(x_{i_k}) = \{y_i \mid x_i \le x_k\}$ and $R(x_{i_k}) = \{y_i \mid x_i > x_k\}$
        b. Choose the $x_{i_k}$ that minimizes the impurity of partition $L(x_{i_k}), R(x_{i_k})$ to make a split.
    2. If $X_i$ is categorical and has levels $a, b, c$
        a. Generate a subset $\{a\}$ and its complement $\{b, c\}$ from all levels.
           Examine partition on the response $L = \{y_i \mid x_i \in \{a\}\}$ and $R = \{y_i \mid x_i \in \{b, c\}\}$. Compute the impurity of this pair $L, R$.
        b. Generate another subset $\{b\}$ and its complement $\{a, c\}$ and do the same thing.
        c. Repeat the process with all possible subset and its complement.
    3. Choose the variable that gives you the partition with the smallest overall impurity.
```

If we examine the process of growing a decision tree, we see that, in fact, it is not that the cardinality of $\boldsymbol{X_i}$ that matters. Rather, it is the number of different partitions on the response variable the tree can induce matters. Specifically, if $\boldsymbol{X_j}$ is a categorical variable with $\boldsymbol{q}$ levels, then it can induce $\boldsymbol{2^{q-1}}$ different partitions on the response $\boldsymbol{y_i}$'s. On the other hand, if $\boldsymbol{X_j}$ is numeric with $\boldsymbol{q}$ distinct values, then it induces only $\boldsymbol{q}$ partitions. As a side note, that is why randomForest package in R cannot handle categorical variables with more than 32 categories. A categorical variable with 32 categories induces 2,147,483,647 partitions on $\boldsymbol{y_i}$'s. See reference 3.

From this perspective, if $\boldsymbol{X_i}$ has enough levels, the tree may tend to include $\boldsymbol{X_i}$ to the model's detriment. However, that is not the full story. We also need to consider how strong the relationship is between the response and other variables, say, $\boldsymbol{X_j}$.

2.2 A Simulation Study

For example, assume that the true model is $Y = 3X_1$ without any noise term. In this case, even if we feed the decision tree with an irrelevant X_2 with 100 levels, we will reach a tie and the decision tree will split using X_1 50% of the time. On the other hand, if the true model now includes a noise term $Y = 3X_1 + \epsilon$ where $\epsilon \sim N(0, 0.5)$, the tree will always favor the irrelevant X_2.

```
set.seed(45)    # R version 4.1.2
library(rpart)  # Version 4.1-15
library(rpart.plot) # Version 3.1.0

par(mfrow = c(2, 1))
x1 <- rnorm(mean = 1, n=100)
y <- 3*x1
x2 <- as.factor(seq(1, 100))
df <- data.frame(x1, x2 = sample(rep_len(x2, length.out = 100)), y)

fitted_cart <- rpart(y ~ x1 + x2, data=df, method = “anova”,
          control = rpart.control(minsplit = 2, maxdepth = 1))
rpart.plot(fitted_cart, main = “Case 1: 100 levels with no noise. Tie.”)
print(fitted_cart$splits)
```

Output:

```
  count ncat  improve  index adj
x1  100  -1 0.5994785 1.552605  0 # Decision tree chose x1 to split
x2  100 100 0.5994785 1.000000  0
```

```
# Number of level equals to sample size, with a little bit of noise
x1 <- rnorm(mean = 1, n=100)
y <- 3*x1 + rnorm(n = 100, sd = 0.5)
x2 <- as.factor(seq(1, 100))
df <- data.frame(x2 = sample(rep_len(x2, length.out = 100)), x1, y)

fitted_cart <- rpart(y ~ x2 + x1, data=df, method = “anova”,
          control = rpart.control(minsplit = 2, maxdepth = 1))
rpart.plot(fitted_cart, main = “Case 2: 100 levels with very little noise”)
print(fitted_cart$splits)
```

Output:

```
  count ncat  improve   index    adj
x2  100 100 0.6414503 1.0000000 0.0000000 # Decision tree chooses x2 to split
x1  100  -1 0.6371685 0.9917295 0.0000000
x1   0  -1 0.9800000 0.9079153 0.9565217
```

3. RESULTS AND DISCUSSION

The key takeaways from the simulation are the following:

1. Even though we might have a categorical variable with lots of levels, when the noise level is

small, the tree can still choose the right variable X_1 to make a split.

2. When the noise level increases, it becomes more and more difficult to make a "good split" using X_1. Since X_2 has many levels and can be used to create more partitions on y_i's, the decision tree tends to use X_2 to make a split.

To better understand this behavior, we should realize that decision trees have a very interesting property: Although a decision tree forces all variables to interact with each other (see footnote 1), it essentially does a sequence of "univariate analysis" at each node.

For example, at any given node, it might be the case that two variables, *gender* and *year_of_education*, combined can explain the response pretty well and lead to a good partition on the response. But when we look at them individually, none of them is highly correlated with the response In such a situation, a third unrelated categorical variable with lots of level might be selected. This type of situation is probably very common in practice. As a result, it is very understandable that many practitioners believe that categorical variables with many levels tend to cause problems.

Another thing to consider is the sample size. For example, a categorical variable might have 20 levels. If we have 10 million data points, then 20 levels might not be "too many". However, in practice, even though we have a large data set, we might still run into trouble. Because as the tree grows deeper, the sample size within each partition becomes smaller and smaller. For example, if we start with 100,000 data points and we grow our tree 3 levels deep perfectly balanced, we end up with only 12,500 data points.

4. CONCLUSIONS

In conclusion, a categorical variable's influence over the decision tree depends on many factors, including its number of levels, its "correlation" with the response variable, sample size of the data, etc. We advise against rushing to dropping certain levels of a categorical variable, or dropping the variable itself completely. We suggest understanding the model's behavior related to the categorical variables using tools such as variable importance plot (i.e., VIP), preprocess the variables using business context if interpretability is desired, and clarify the goal of the model.

5. REFERENCES

[1] https://chaoguo14.github.io/2021/02/27/Decision-tree-and-variables-having-lots-of-levels/

[2] Hastie, T. and Tibshirani, R. and Friedman, J. "The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition". 2009. Page 305 – 310.

[3] https://stats.stackexchange.com/questions/49243/rs-randomforest-can-not-handle-more-than-32-levels-what-is-workaround

Biography of the Author

Chao Guo is a data scientist at Amazon. Before that, he worked at Munich Re as a senior data scientist. He has a master's degree in statistics from Rutgers University as well as a bachelor's degree in mathematics from University of California-Los Angeles.