Introduction

Hi everyone, our project is about the binarization and discretization. This is relatively new concept that researchers try to find out whether the machine learning models need full floating point accuracy data to obtain good performance.

In our project, our goal is to find out that if the discretization and binarization feature engineering techniques could reduce the noises of real world data or capture highly skewed or non-standard distribution. However, due the disruptive transformation process, the transformed data could lose its characteristics and become meaningless. We propose a unique binarization process using binary trees, which could potentially capture the original data distribution as much as possible, while the generalization and the representativeness of original distributions and characteristics.

Result comparison

First, we find several public datasets and we use histogram to convert features into integers depending on the bins, we call these discrete datasets. Next, we use our method, a binary tree, to convert features into binaries. As you can see the example binary tree here, each branch of the tree represents one digit “-1” or “+1”. Eventually, each number would be represented by 3 digits and indicate which intervals would that number falls into. We call these binary datasets.

Then, we used three types of dataset, each trained the 8 same models and compared their performance of hundreds of trials, including 'LogisticRegression', 'RandomForest', 'NeuralNetwork', 'LinearSVC', 'SGD', 'Gaussian Naïve Bayes', 'AdaBoost', 'GaussianProcess'.

Overall, the discrete and binary datasets obtained fairly well performance comparing to the original datasets. Most of them were about 3% - 8% lower accuracy, but considering their computation time, it was reasonable. We will talk about the computation time later. Apart from that, surprisingly, for the linear models and the neural network, the discretization and binarization actually provided better outcomes than the original dataset using the same settings.

Analysis

There are two things worth attention, first is the information bottleneck

First, the Information Bottleneck Theory

is about understanding and specifying which features of example X play a role in the making a prediction. The discretization and binarization could be considered as finding a short code for X that preserves the maximum information about Y. That is, we squeeze the information that X provides about Y through a ‘bottleneck’ formed by a limited set of codewords. For example, like these two distributions here, in order to make a fairly good prediction, you actually only need to know whether the point falls into the left or right side of this line. The discretization process is trying to find this threshold, while only using 3-bit integers, which is the number of bins from the histogram, or for binarization, just 3 binary bits from the binary tree. This is why discrete dataset and binary dataset could perform well or better even though we lost floating point accuracy of the original data.

Second, for the linear model and the Neural Network, there is something called Binarized Neural Network which are gaining popularity. Currently, running a normal Neural Network model could become extremely large and challenging for devices with limited memory or computation power. However, if all of the inputs, outputs and weights are all binaries. Then, the computation of the forward function and backpropagation process would be kind of atomic, because things could easily be done as binaries for chips. In addition, it requires very little memory for binaries. In our experiment, the binary neural network model has already exceeded the model using original data.

This is our project for this class, and if you have any question, please contact us via emails. Thank you for listening and have a good summer.

Over the past two or three years,

Bipolar Binary