**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 really 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 size of dataset while still capture its generalization and the representativeness of original distributions. 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 provide as much accuracy as possible while achieve a good compression rate of the dataset.

Here, we find several public datasets. Then, we use histogram to convert continuous features into integers depending on the number of 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 “0” 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 each of these three datasets with two different settings to train the same 8 machine learning models, 96 models in total, and compared their performance of many trials, including 'LogisticRegression', 'RandomForest', 'NeuralNetwork', 'LinearSVC', 'SGD', 'Gaussian Naïve Bayes', 'AdaBoost', 'GaussianProcess'.

**Result comparison**

First, the discrete and binary datasets obtained fairly well performance comparing to the original datasets.

For example, intuitively we would expect the discrete and binary datasets would lose some accuracies. In these plots, the left side are the models using 10 bins for discrete datasets and 6 bits for binary datasets. On the right side, they are using more accurate 16 bins for discrete ones and 8 bits for binary.

From the random forest and gaussian process model, the original datasets have the highest accuracy. The discrete and binary dataset are lower. However, when we using more accurate representations for discrete and binary datasets, the scores increase dramatically. Especially the 8 bits binary dataset, it is only 2% lower while only have 1/4 of the original dataset size, which is quite amazing.

Then we move on to the linear models, they offer more interesting results. The linear models do not work well while the linear separability is not obvious on the original dataset. But the discrete and binary datasets significantly boosted the accuracies on the LinearSVC, SGD and Logistic Regression models. Note this, these models are not Neural Network, since the gradients are not backpropagated to the original floating point values. The gradient descent stopped at the discrete and binary digits.

For the Neural Network, all three datasets use the same NN structure, the discrete and binary dataset still exceed original dataset.

Remarkably, binary datasets all surpass the discrete datasets in all these linear models, which means that the linear models probably prefer binaries over integers. In addition, we must keep in mind that the binary and discrete datasets have their size significantly smaller than the original.

**Analysis**

There are three things worth attention

First, the discretization and binarization would reduce the noises in the dataset, because all floating point numbers are normalized into integers or binaries. This is quite good for models are sensitive towards the noises, like those linear models. Nonetheless, the non-linear models are not sensitive towards the noises, like the random forest and gaussian process. but these models also sometimes tend to be overfitting.

Second is the Information Bottleneck Theory, which 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. This is why discrete dataset and binary dataset sometimes could perform well or better even though we lost floating point accuracy of the original data.

Lastly, for the linear model and the Neural Network, we could convert them into a Binarized Neural Network, where their inputs, outputs and weights are all binaries. These networks are gaining popularity since 2017 or 18, that they could be deployed to devices with limited memory and computation powers, like smart watches, UAVs, etc. The obvious reason is that, currently, a normal Neural Network model could become extremely large and running them on those devices is also challenging. 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 normal CPU chips. Also, it requires very little memory for binaries. In our experiment, the binary neural network model has already exceeded the model using original data. Nevertheless, this is still a ongoing research.

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