**Introduction**

Hi everyone, our project is about the discretization and binarization. In our project, our goal is to find out that how discretization and binarization could still capture the generalization and the representativeness of original full floating-point data while reducing the size of dataset. Also, 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. In these plots, the left side are the models using 10 histogram 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.

EJay would then show and explain our experiment results.

**Result comparison**

Thank you. In our experiment, for the random forest and gaussian process model, the original datasets have the highest accuracy while the discrete and binary ones are lower. However, when using more accurate representation settings, the scores of discrete and binary increase dramatically. Especially the 8 bits binary, it is only 2% lower with 1/4 of the original size, which is quite amazing.

Then linear models offer more interesting results. They 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 SVC, SGD and Logistic Regression. Note this, these are not 1-layer Neural Network, since the gradients are not backpropagated to the original floating point values. For the Neural Network, given the same structure, the discrete and binary dataset also exceed original dataset. Remarkably, our binary datasets all surpass the discrete datasets in all these models, which means that the linear models probably prefer binaries over integers.

**Analysis**

There are three takeaways.

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. Also, the non-linear models also sometimes tend to be overfitting to high floating point accuracy.

Second is the Information Bottleneck Theory, which is about understanding how features 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 and binary dataset sometimes could perform well or better even without floating point accuracy.

Lastly, we could convert the linear model and the Neural Network into a Binarized Neural Network, where their inputs, outputs and weights are all binaries. These networks are gaining popularity right now, that they could be deployed to devices with limited memory and computation powers, like smart watches. With all binary values, the computation would be kind of atomic on normal CPU chips with little memory. In our experiment, the binary neural network model has already exceeded the model using original data. Nevertheless, this is still an ongoing research.

This is our project for this class. Thank you for listening and have a good summer.

Bipolar Binary