

1 Finding If Someone Will Default

A credit default is when someone has spent money using their credit card but is unable to pay back the credit card company. Naturally, credit card companies are eager to avoid this, and try to find many mechanisms to find out which people are about to default, so they can prevent them from spending money.

This exercise includes data from Taiwanese credit card companies, and in it you will use a decision tree to predict if someone will default on their credit card debt next month. In order to do this, you shall use the data file `DefaultOfCreditCardClients.csv`, which you can download from the course Moodle. The description of the data file is found in the `Default of credit card clients Data Set.pdf` file you can also download from the course Moodle.

Broadly, the last variable indicates if the person defaulted on their debt. The other variables show various other parameters of the person and their financial situation (ignore the first attribute, which is just a counter). Since some of these variables are continuous, you will need to set thresholds yourself to “bucket” them – which means each of your trees might break different than someone else’s (which is fine!).

When you build your decision trees, you are expected to use entropy to calculate the more meaningful attributes, and to use the χ^2 test to prune vertices.

Your Python code will include the following functions. You can assume the file `DefaultOfCreditCardClients.csv` is found in the same directory as your code.

`build_tree(<float> k)` $k \in (0, 1]$. You need to build a decision tree, using k ratio of the data (so if $k = 0.6$, you arbitrarily choose 60% of the data), and validate it on the remainder. The outcome is printing out the decision tree, and reporting the error.

`tree_error(<int> k)` You need to report the quality of the decision tree by building k -fold cross validation, and reporting the error.

`will_default<array>()` You receive an an input from the bank of a loan they are considering. You can assume it is in the same order as the data file (but without the bit saying if the person defaults, of course). You return 1 if you think the person will default, and 0 if not.

2 Hebbian Learning

In class, we saw a symmetric, one-neuron-at-a-time updating neural net converges to a value.

- (a) Show an example of a non-symmetric (yet still one-neuron-at-a-time updating) neural net that does not converge.
- (b) Show an example of a symmetric neural net that updates simultaneously (i.e., not one-neuron-at-a-time updating) that does not converge.

3 Learning

Could you use both neural nets and decision trees together in a boosting algorithm? How would you use the boosting weights in the training of a deep neural net?