Category Trees User Manual

Kieran Greer, Distributed Computing Systems, Belfast, UK.
https://distributedcomputingsystems.co.uk.
Version 1.1.

1 Introduction

‘Category Trees’ branch on category type and not feature, like Decision Trees. It uses a supervised clustering technique, where each category is represented by a separate base classifier. Each base classifier then classifies its own subset of data rows and creates a centroid to represent the category. If the classifier is subsequently associated with rows from other categories, it needs to create new classifiers for the incorrect data. The classifier therefore branches to new layers when there is a split in the data, and creates new classifiers there for the incorrectly classified rows. The schematic of Figure 1 shows the classification process, where a new layer has been added to classifier A, so that it can correctly re-classify the category A and B sub-groups that are closest to it. The second level uses a subset of the whole dataset that is specifically only the data rows assigned to the classifier at the parent level.

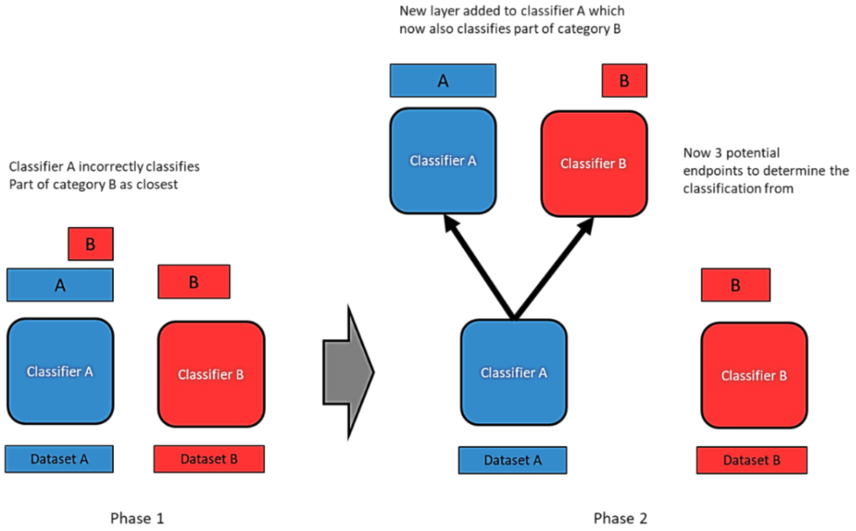

New layer added to classifier A which now also classifies part of category B
A
B
Classifier A
Classifier B
Now 3 potential endpoints to determine the classification from
Classifier A incorrectly classifies Part of category B as closest
B
A
B
Classifier A
Classifier B
Classifier A
B
Classifier B
Dataset A
Dataset B
Dataset A
Dataset B
Phase 1
Phase 2

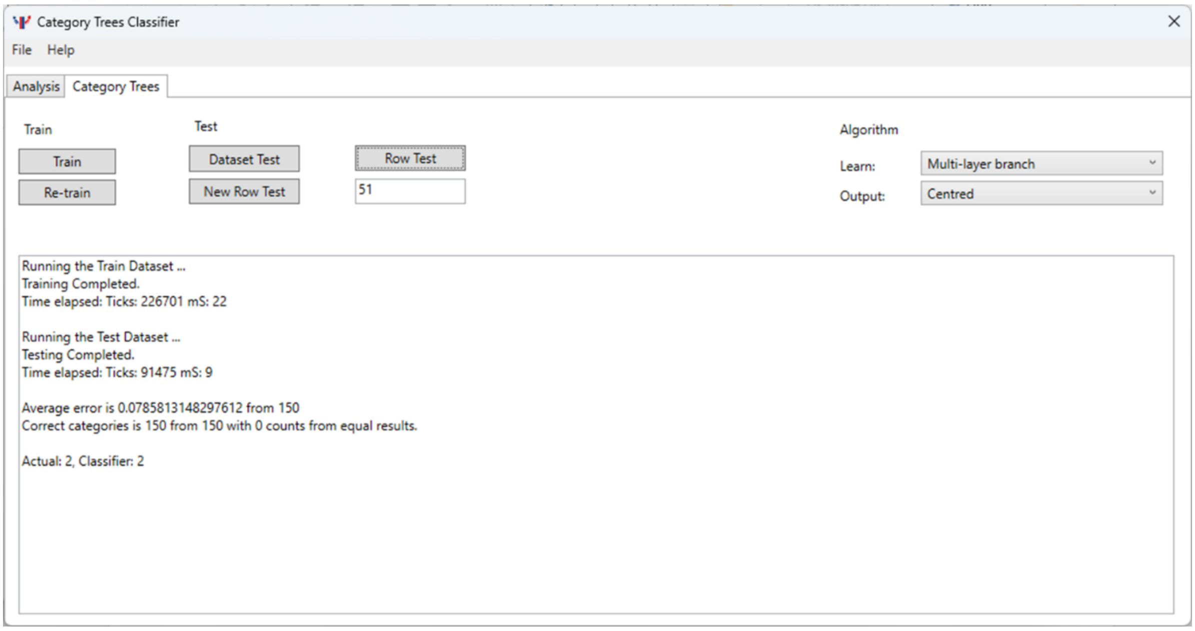
Figure 1. Schematic of the classifier in action. Phase 1 realises that classifier A also classifies part of category B better. Phase 2 then adds a new layer to classifier A, to re-classify this subset only.

2 Installation

To install the program simply unzip the files into an empty directory and then click the ‘CategoryTreesDCS’ exe file.

The program is written in C#.Net and so you will also need the .Net framework to run the program. A version of this can be downloaded from the Microsoft web site. The whole GUI is shown in Figure **2**.

This example shows the result for the benchmark Iris dataset, where the classifier has been trained on it and then tested on the whole dataset, and then a specific data row has been queried.



51

Learn: Multi-layer branch

Output: Centred

```
Running the Train Dataset ...
Training Completed.
Time elapsed: Ticks: 226701 mS: 22

Running the Test Dataset ...
Testing Completed.
Time elapsed: Ticks: 91475 mS: 9

Average error is 0.0785813148297612 from 150
Correct categories is 150 from 150 with 0 counts from equal results.

Actual: 2, Classifier: 2
```

Figure 2. Category Trees Classifier.

3 Run the Classifier

To use the classifier, you need to select a train and a test dataset. You then train the classifier on the train dataset and then test it on the test dataset. You can alternatively, enter single rows of input values to get the classifier evaluation for that row.

3.1 Dataset Format

A dataset is stored as a simple text file, but with some additional information at the start. This is the start of the benchmark Iris[1] dataset, where the additional metadata is as follows:

```
Input:0-3
Output:4
Type:java.lang.Double
Tokenizer:,
5.1,3.5,1.4,0.2,1
```

The first line of data is shown at the end, which contains 4 columns of input data and a fifth column that is the actual output category (value 1). The classifier needs to read some metadata first, shown

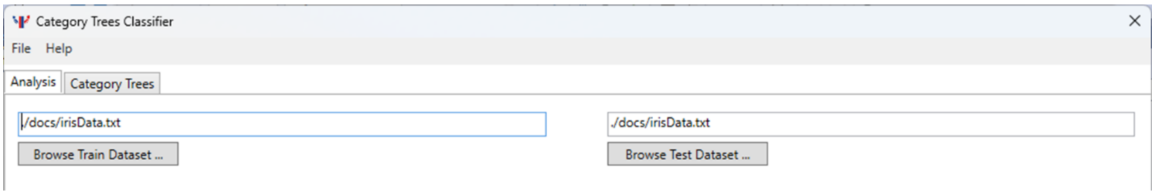
---

[1] UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/

in the lines above this. The metadata specifies that columns 0 to 3 (counting starts at 0) are the input columns and column 4 is the output column. It then specifies that the data type is Double, which should not be changed. It then specific that the token separator is a comma (‘,’). When it reads a data row, it separates the values using this token. You then add the actual data rows to the file and save it as a text file (.txt).

3.2 Analysis Panel

You firstly have to enter the path to a train and alternatively a test dataset. This is done through the analysis panel, shown in Figure **3**. When you enter the train file path using ‘Browse Train Dataset,’ it is automatically assigned to the test file as well. You can then click the ‘Browse Test Dataset’ button to enter a different test data file. After you train the classifier, this panel shows some additional information on the structure of the classifier, but you do not need to be concerned about the values.



./docs/irisData.txt

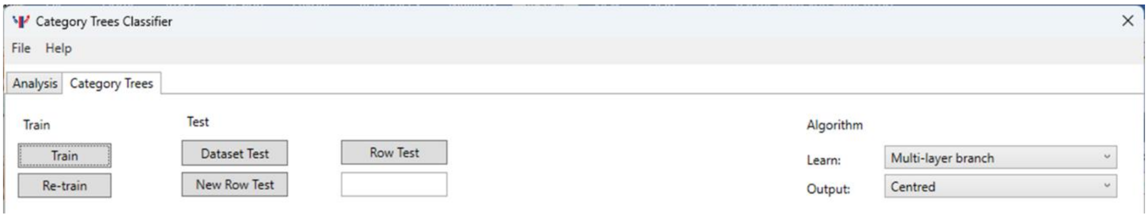
./docs/irisData.txt

Figure 3. Analysis Panel.

3.3 Category Trees Panel

You then train the classifier in the Category Trees panel, shown in Figure **4**.

The algorithm settings are on the RHS of the panel. There is only the 1 algorithm and only 1 version of it, that is the multi-layer branching mechanism. The second combo box allows you to select the output type of ‘Centred’ or ‘Banded.’ Because each category creates a separate classifier, the output value does not matter very much. If the data is normalized to be within the values 0 to 1, then an average value of 0.5 may be appropriate and this is what the Centred option uses. However, for the whole classifier, each category also has a discrete value, in the range 0 to 1, for example 0, 0.5 or 1 for 3 categories. The Banded option therefore uses this value for each individual classifier instead. But it is probably OK to just stick with the default setting.



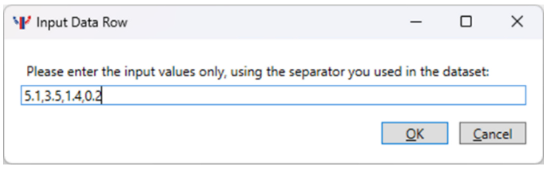
Learn: Multi-layer branch

Output: Centred

Figure 4. Category Trees Panel.

You then need to train the classifier on the train dataset before testing it. To do this you click the 'Train' button. You can then test the classifier in one of two different ways, as follows:

1. Test the whole dataset: If there is a test dataset entered, then you can click the 'Dataset Test' button to test on the whole dataset. The output will indicate the classification accuracy. Then to get a specific value, you can enter a data row number, starting at 1, and the 'Row Test' button will retrieve the category the classifier thinks the data row belongs to. To use the row test, you have to test on the whole dataset first.
2. Alternatively, you can train the classifier and then ask it to categorise some unknown data row, using the 'New Row Test' button. If you click this button, an input dialog box opens that asks you to enter the data row values, shown in Figure **5**. The user should copy the input values in the row only and paste them into the text box. The example shows the first data row of the Iris dataset pasted into the text box, for example. If you then click 'OK,' the classifier will return what it thinks is the correct category for that data row. Note that the actual category will be missing then, because it is not known.



Please enter the input values only, using the separator you used in the dataset:

5.1,3.5,1.4,0.2

Figure 5. Input a Data Row.

Results

Tests show that the method can successfully classify a diverse set of benchmark datasets and many to 100% accuracy. It seems to work well with medical or biological data, but may not work quite this well with different test datasets. Because it is quick and automatic to run, it could be an interesting first choice to test your data with.

Reference

Greer, K. (2021). Category Trees – Classifiers that Branch on Category, International Journal of Artificial Intelligence & Applications (IJAIA), Vol.12, No.6, pp. 65 - 76.