Saturday, December 3, 2016

Decision tree with pruning

Preprocess

In the preprocess step, we processed on continuous typed features so that they can be handled in the following tree node splitting step. We discretized each continuous attribute to form an ordinal binary attribute by checking if a sample is smaller or bigger than the median of all samples of the attribute. This way, continuous attributes were labeled either 0(no bigger than the median of the attribute) or 1(bigger than the median).

For nominal attributes, we numbered them to make implementation easier. For dataset2, “Absent” was labeled as 0 and “Present” was labeled as 1.

In new version, we made each continuous attribute into k bins and is labeled 0 if it’s in the 1st bin, 1 if it’s in the 2nd bin, 2 if it’s in the 3rd bin and so on so forth, until k-1 if it’s in the kth bin. And it becomes an ordinal attribute because samples labeled 0(landed in the 1st bin) are smaller than those labeled 1(landed in the 1st bin). There order is:

samples labeled 0 < samples labeled 1 < …….. < samples labeled k-1

The choice of k can be inferred by cross validation.

Note:

when k = 2, it’s performance is different from my older version.

this version: since I am using bins, data needs to be compared to is (max(data) + min(data)) / 2

older version: data is compared to average(data)

Tree splitting criterion

We used 2-way splitting to split tree nodes and we used classification error to choose the best split.

choose_i =

arg min{ p_i*classification_error_i(left_child) + (1-p_i)*classification_error_i(right_child) }

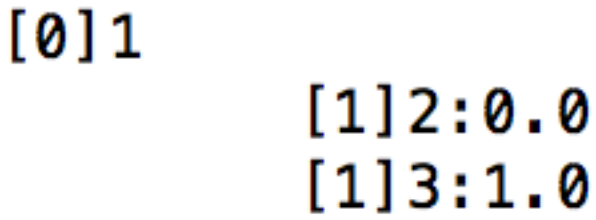
If a node doesn’t meet the stoping criterion(not leaf node), we followed the splitting criterion to split.

Stoping criterion: check if a node is leaf node

1. has only one point in the node
2. all points in the node have same label(impurity is 0)
3. all splits's impurity is bad: wighted sum of children > parent
4. the node's impurity is smaller than the threshold impurity (pruning)

Tree presentation(visualization)

An example:



each column is a level of the tree

format: [parent_id] my_id

'[]' shows the ID of the parent's. if its [0], this node is the root

if the current node is a leaf node, the number after ':' is the label

In the example, node 1 is the root and it has 2 children node 2 and node 3;

node 1 is not a leaf node, so it doesn't have the part ':0.0' or ':1.0';

node 2 is the leaf node and all data that goes to this node is labeled as 0;

node 3 is also a leaf node and all data that goes to this node is labeled as 1.