Decision Trees – Part 5

Splitting

Example

- Example taken from Tan’s book.

Binary and N-way Splits

- Binary Split
  - We can split into two child nodes
- N-way split
  - Split into n (> 2) child nodes

Numeric Features

- Binary Split
  - Budget < 1m
  - Budget >= 1m
- 3-way split
  - Budget < 1m
  - Budget between 1m and 2m
  - Budget >= 2m

Semi-open Ranges

- < 10K
- [10K, 20K)
- [20K,30K)
- >= 30K
- Convention is to have the closed end “[” of the range on the left and the open one “)” on the right.
- [10K, 20K) - up to but not including 20K.
- Semi open ranges fit nicely together.

Nominal Features

- Remember this is a type of categorical feature where there is no ordering.
- For example, three types of car - family, sports and luxury.
- Binary splits include
  - {family, sports} and {luxury}
  - {family, luxury} and {sports}
- 3-way split
  - {family} {sports} {luxury}

Ordinal Features

- Binary splits
  - {low, medium} {high}
  - {low} {medium, high}

Decision Tree Algorithm

- Gererate all possible splits.
- Evaluate each split using impurity measure such as GINI
- Choose the best one.
- Lets look at how we generate all possible splits for a numeric feature.

Example - Tax Returns

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Numeric Feature, binary split

- Taxable Income is a numeric feature
- 10 instances in the training set.
- Values for the 10 instances are 125K, 100K, 70K, 120K, 95K, 60K, 220K, 85K, 75K, 90K
- Sort the instances based on Taxable Income.
- Sorted Values are
  - 60, 70, 75, 85, 90, 95, 100, 120, 125, 220

Numeric Feature, binary split

- Sorted Values are
  - 60, 70, 75, 85, 90, 95, 100, 120, 125, 220
- Choose split value between these values
  - 55, 65, 72, 80, 87, 92, 97, 110, 122, 172, 230
- These are the options and we need to find the best one.
- Get the count matrix for the first possible split.
- For the first split all instances are in the right child, none in the left child.

Numeric Feature, binary split

<table>
<tr><td>Cheat</td><td></td><td colspan="2">No</td><td colspan="2">No</td><td colspan="2">No</td><td colspan="2">Yes</td><td colspan="2">Yes</td><td colspan="2">Yes</td><td colspan="2">No</td><td colspan="2">No</td><td colspan="2">No</td><td colspan="2">No</td><td></td></tr>
<tr><td></td><td colspan="22">Taxable Income</td></tr>
<tr><td>Sorted Values →</td><td></td><td colspan="2">60</td><td colspan="2">70</td><td colspan="2">75</td><td colspan="2">85</td><td colspan="2">90</td><td colspan="2">95</td><td colspan="2">100</td><td colspan="2">120</td><td colspan="2">125</td><td colspan="2">220</td><td></td></tr>
<tr><td>Split Positions →</td><td colspan="2">55</td><td colspan="2">65</td><td colspan="2">72</td><td colspan="2">80</td><td colspan="2">87</td><td colspan="2">92</td><td colspan="2">97</td><td colspan="2">110</td><td colspan="2">122</td><td colspan="2">172</td><td colspan="2">230</td></tr>
<tr><td></td><td><=</td><td>></td><td><=</td><td>></td><td><=</td><td>></td><td><=</td><td>></td><td><=</td><td>></td><td><=</td><td>></td><td><=</td><td>></td><td><=</td><td>></td><td><=</td><td>></td><td><=</td><td>></td><td><=</td><td>></td></tr>
<tr><td>Yes</td><td>0</td><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td><td>1</td><td>2</td><td>2</td><td>1</td><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td><td>0</td><td>3</td><td>0</td></tr>
<tr><td>No</td><td>0</td><td>7</td><td>1</td><td>6</td><td>2</td><td>5</td><td>3</td><td>4</td><td>3</td><td>4</td><td>3</td><td>4</td><td>3</td><td>4</td><td>4</td><td>3</td><td>5</td><td>2</td><td>6</td><td>1</td><td>7</td><td>0</td></tr>
<tr><td>Gini</td><td colspan="2">0.420</td><td colspan="2">0.400</td><td colspan="2">0.375</td><td colspan="2">0.343</td><td colspan="2">0.417</td><td colspan="2">0.400</td><td colspan="2"><i>0.300</i></td><td colspan="2">0.343</td><td colspan="2">0.375</td><td colspan="2">0.400</td><td colspan="2">0.420</td></tr>
</table>

- Move left to right, update the count matrix, calculate the Gini value
- Choose the split position that has the least gini indexex

Best Split

- 97 gives 3/3 and 4/0
- Left child has high GINI value but the right child has low value.
- The weighted sum of these values is the best GINI value for a split.

Summary

- This illustrates how the best possible binary split can be found for a numerical feature.
- Only split values between existing values of the feature need to be considered.
- By ordering instances in increasing value of the numeric features an efficient implementation is possible.
- A count matrix is defined for the first split.
- Subsequent count matricies are obtained by updating the existing matrix on a scan from left to right.