DSS
Decision Sciences & Systems

Technische Universität München

Tutorial Business Analytics

Tutorial 6: Decision Trees

Decision Sciences & Systems (DSS)

Department of Informatics

TU München

Tutorial Business Analytics

Classifiers

Classifiers from previous lectures:

- Zero-Rule: class with the most instances (rule)
- One-Rule: rules for one attribute
- Naïve Bayes: conditional probability attribute – class

What is the difference between classification and regression?

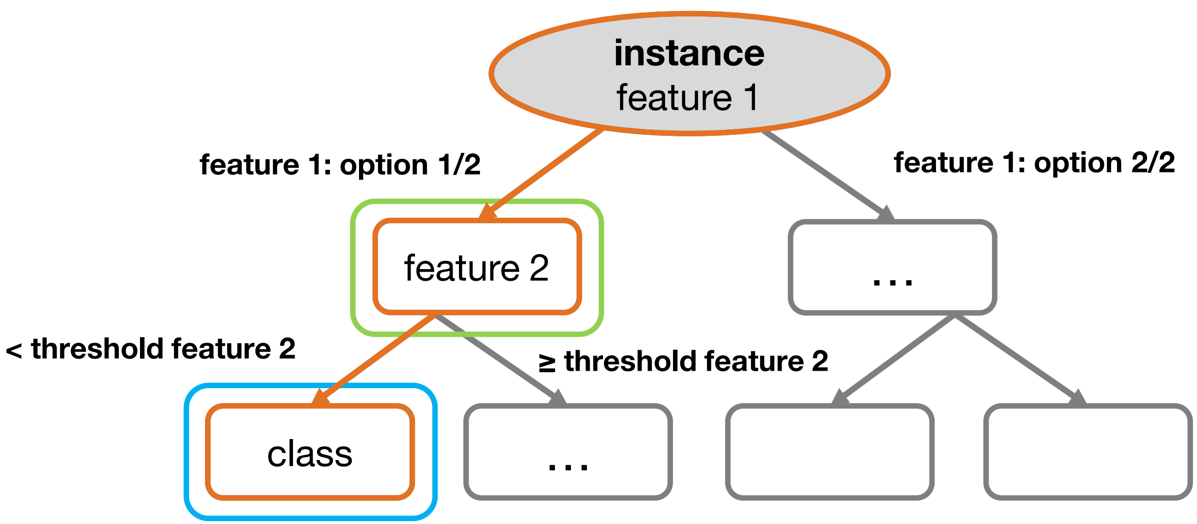
Classification	Regression
Prediction of a class label by means of the attributes	Prediction of a numerical value by means of the attributes

DSS
Decision Sciences & Systems

Tutorial Business Analytics

Classification Decision Trees

- A decision tree for n different classes is created based on some training data
- An internal node is a test on an attribute
- A branch represents an outcome of the test
- A leaf node represents a class
- A new instance is classified by following a matching path to a leaf node



Tutorial Business Analytics

Optimal Tree

For m attributes and $n = 2$ classes, there are 2^{2^m} possible trees already

- That is equal to the number of Boolean functions

Finding the optimal tree is NP-complete

- Not feasible for data mining applications

Solution: **Greedy algorithm** for tree construction

- Top down approach: The tree is created recursively from the root node
- Every possible split is assessed with a measure
- The best split is chosen
- Repeat until all leaf nodes are pure or all attributes have been used

Tutorial Business Analytics

Evaluating splits

Which split is better?

- Instances should be classified as easy as possible
- Good separation of classes (ideally leaf nodes contain instances of a single class only)
- In the worst case the separation does not affect the class distribution
- Possible measure: information

Tutorial Business Analytics

Information and entropy

- Let us denote c_i to be the absolute number of training examples being in class i at the current stage
- The probability (relative frequency) of class i then is $p_i = \frac{c_i}{C}$ with $C = \sum_{i=1}^{n} c_i$

Entropy measures information content in bits (uncertainty of a node):

$$\text{entropy}(p_1, \dots, p_n) = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i \, .$$

Information necessary to classify:

$$\text{info}([c_1, \dots, c_n]) = \text{entropy}\left(\frac{c_1}{C}, \dots, \frac{c_n}{C}\right).$$

Represents the expected amount of information that would be needed to specify the class of this node.

Tutorial Business Analytics

Information gain

The quality of a split is equal to the **gained information**

gain(attribute) = info(before split by attribute) – info(after split by attribute)

DSS
Decision Sciences & Systems

Tutorial Business Analytics

Formulas

Entropy:

$$\text{entropy}(p_1, \ldots, p_n) = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i$$

Information for $C = \sum c_i$:

$$\text{info}([c_1, \ldots, c_n]) = \text{entropy}\left(\frac{c_1}{C}, \ldots, \frac{c_n}{C}\right)$$

Average information for a numeric split into m branches, with $L_i = [c_{i,1}, \ldots, c_{i,n}]$ being the set of class counts in this split, $C_i = \sum_k c_{i,k}$ the corresponding number of instances, and $L = \sum C_i$:

$$\text{info}(L_1, \ldots, L_m) = \sum_{i=1}^{m} \frac{C_i}{L} \cdot \text{info}(L_i)$$

Information gain:

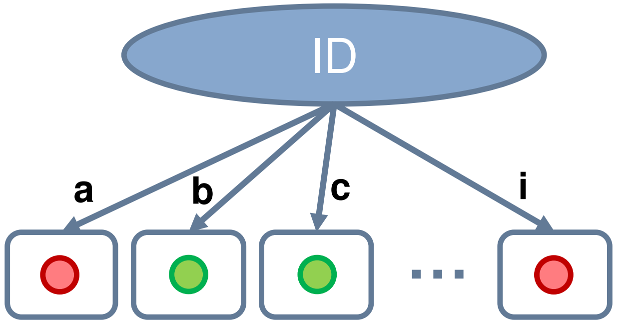
gain(attribute) = info(before split by attribute) – info(after split by attribute)

Tutorial Business Analytics

Information gain problems

Biased against attributes with a lot of edges

- For example: ID attribute
- Highest information gain because every leaf is pure
- Results in overfitting



Solution

- Take number and size of leafs into account: Intrinsic Information
- **Intrinsic information**: s is the size of a leaf (number of affected instances)

$$\text{intrinsicInfo}([s_1, \ldots, s_n]) = \text{info}([s_1, \ldots, s_n])$$

New criterion: **Gain ratio**

$$\text{gainRatio}(\text{attribute}) = \frac{\text{gain}(\text{attribute})}{\text{intrinsicInfo}(\text{attribute})}$$

Tutorial Business Analytics

Numerical attributes

Considering nominal attributes
- one edge per attribute value works well
- bad in case of numerical values

Solution: **Binary Splits**
- values are separated into two sections: below (<) and above (≥) some chosen threshold
- the split is evaluated with the information gain: set threshold to a value, s.t. information gain is maximized
- common practice to place numeric thresholds halfway between the values that delimit the boundaries
- Numeric attributes may be tested several times in a tree