DAT630

Classification

Basic Concepts, Decision Trees, and Model Evaluation

Introduction to Data Mining, Chapter 4

30/08/2016

Krisztian Balog | University of Stavanger

Basic Concepts

Classification

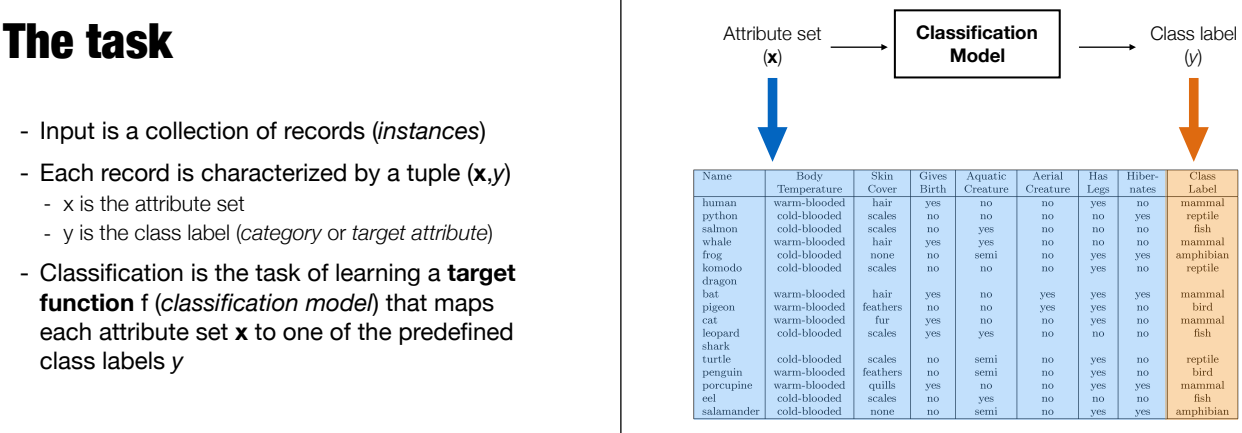
- Classification is the task of assigning objects to one of several predefined categories
- Examples
  - Credit card transactions: legitimate or fraudulent?
  - Emails: SPAM or not?
  - Patients: high or low risk?
  - Astronomy: star, galaxy, nebula, etc.
  - News stories: finance, weather, entertainment, sports, etc.

Why?

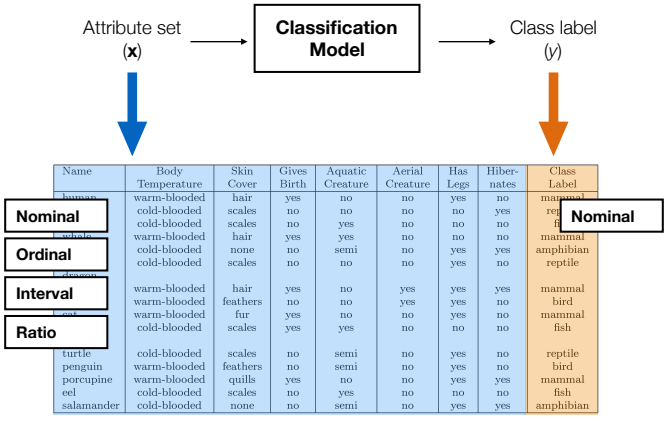
- Descriptive modeling
  - Explanatory tool to distinguish between objects of different classes
- Predictive modeling
  - Predict the class label of previously unseen records
  - Automatically assign a class label when presented with the attributes of the record

The task

- Input is a collection of records (*instances*)
- Each record is characterized by a tuple (**x**,*y*)
  - x is the attribute set
  - y is the class label (*category* or *target attribute*)
- Classification is the task of learning a **target function** f (*classification model*) that maps each attribute set **x** to one of the predefined class labels *y*

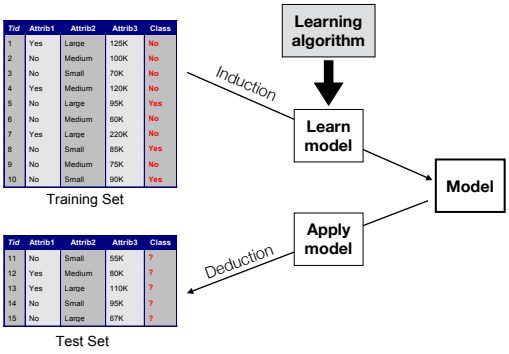


Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard shark	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

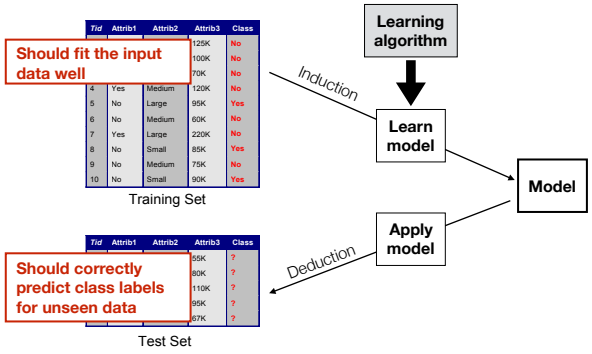


General approach

General approach



Objectives for Learning Alg.



Learning Algorithms

- Decision trees
- Rule-based
- Naive Bayes
- Support Vector Machines
- Random forests
- k-nearest neighbors
- …

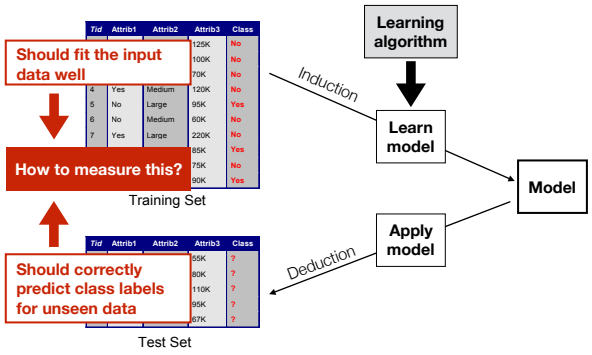
Machine Learning vs. Data Mining

- Similar techniques, but different goal
- Machine learning is focused on developing and designing learning algorithms
  - More abstract, e.g., features are given
- Data Mining is applied Machine Learning
  - Performed by a person who has a goal in mind and uses Machine Learning techniques on a specific dataset
  - Much of the work is concerned with data (pre)processing and feature engineering

Today

- Decision trees
- Binary class labels
  - Positive or Negative

Objectives for Learning Alg.



Evaluation

- Measuring the performance of a classifier
- Based on the number of records correctly and incorrectly predicted by the model
- Counts are tabulated in a table called the **confusion matrix**
- Compute various **performance metrics** based on this matrix

Confusion Matrix

		Predicted class	
		Positive	**Negative**
Actual class	**Positive**	**True Positives (TP)**	**False Negatives (FN)**
	Negative	**False Positives (FP)**	**True Negatives (TN)**

Confusion Matrix

		Predicted class: Positive	Predicted class: Negative
Actual class	Positive	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

Type II Error failing to raise an alarm

Type I Error raising a false alarm

Example

"Is the man innocent?"

		Predicted class: Positive Innocent	Predicted class: Negative Guilty
Actual class	Positive Innocent	True Positive Freed	False Negative Convicted
	Negative Guilty	False Positive Freed	True Negative Convicted

convicting an innocent person (miscarriage of justice)

letting a guilty person go free (error of impunity)

Evaluation Metrics

- Summarizing performance in a single number
- Accuracy

$$\frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{TP+TN}{TP+FP+TN+FN}$$

- Error rate

$$\frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{FP+FN}{TP+FP+TN+FN}$$

- We seek high accuracy, or equivalently, low error rate

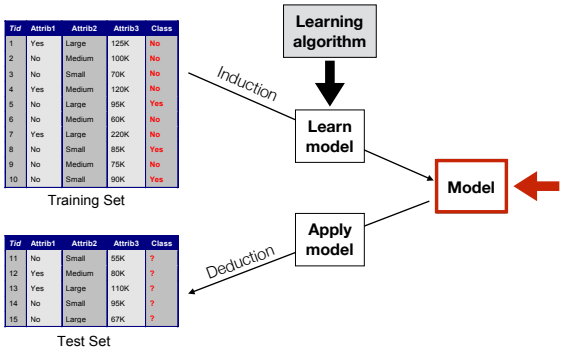
Decision Trees

Motivational Example

How does it work?

- Asking a series of questions about the attributes of the test record
- Each time we receive an answer, a follow-up question is asked until we reach a conclusion about the class label of the record

Decision Tree Model



Decision Tree

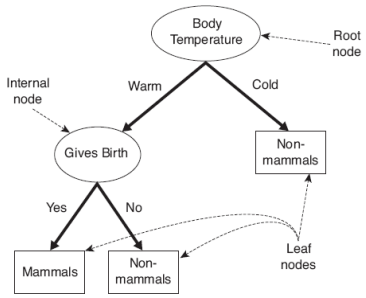


Figure 4.4. A decision tree for the mammal classification problem.

Decision Tree

Root node
no incoming edges
zero or more outgoing edges

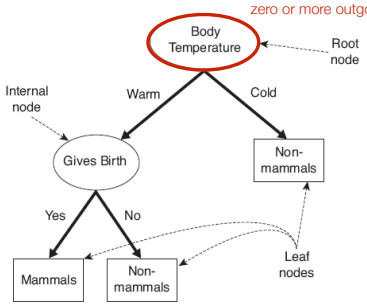


Figure 4.4. A decision tree for the mammal classification problem.

Decision Tree

Internal node
exactly one incoming edges
two or more outgoing edges

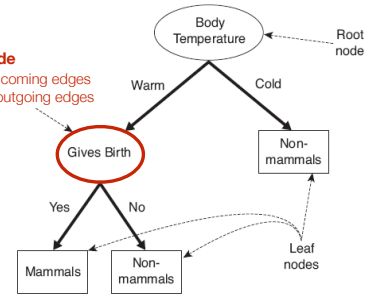
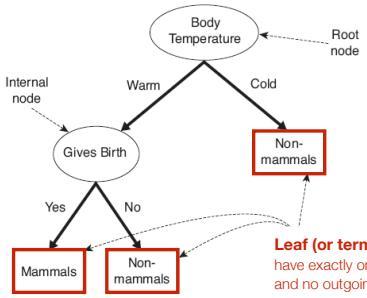


Figure 4.4. A decision tree for the mammal classification problem.

Decision Tree



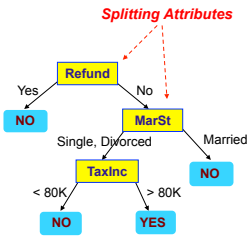
Leaf (or terminal) nodes
have exactly one incoming edges
and no outgoing edges

Figure 4.4. A decision tree for the mammal classification problem.

Example Decision Tree

Tid	Refund	Marital Status	Taxable Income	Cheat
	categorical	categorical	continuous	class
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

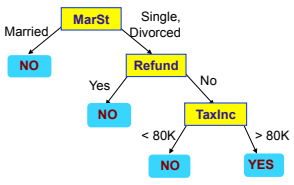
Training Data



Model: Decision Tree

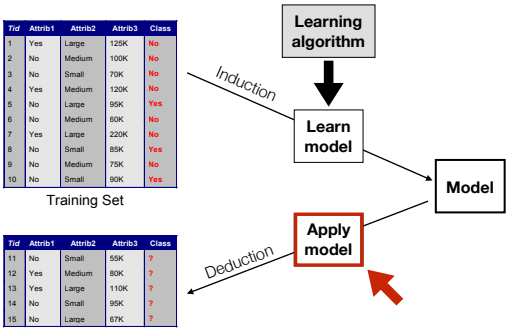
Another Example

Tid	Refund	Marital Status	Taxable Income	Cheat
	categorical	categorical	continuous	class
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

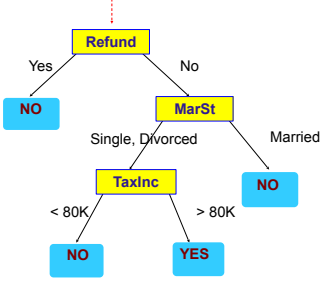


There could be more than one tree that fits the same data!

Apply Model to Test Data



Start from the root of tree.

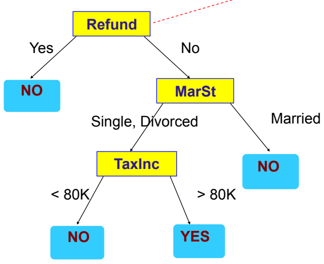


Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

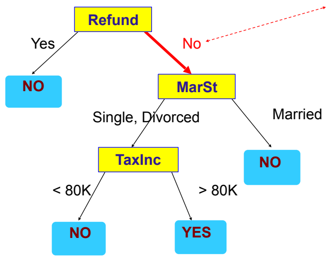
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



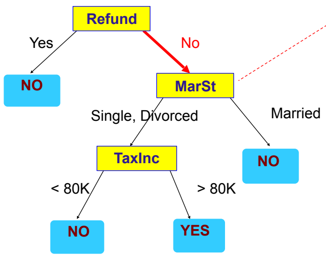
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



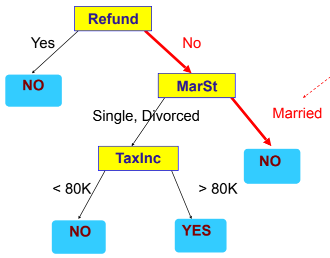
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



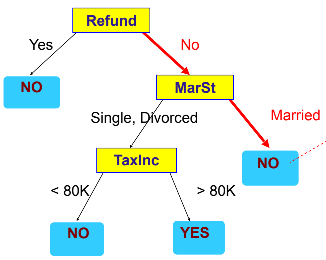
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to "No"

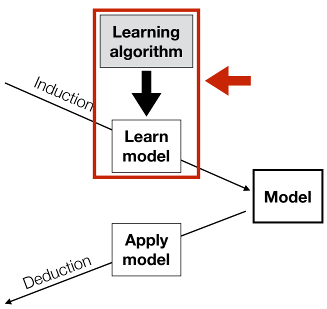
Visual explanation

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

Decision Tree Induction

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

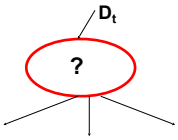
Tree Induction

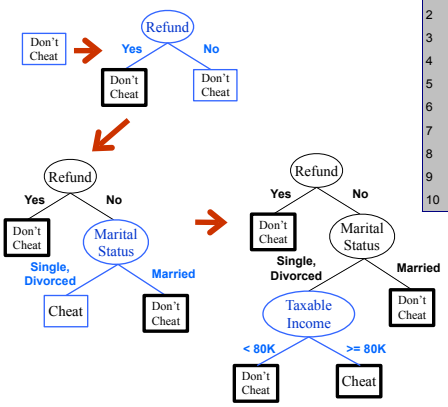
- There are exponentially many decision trees that can be constructed from a given set of attributes
- Finding the optimal tree is computationally infeasible (NP-hard)
- Greedy strategies are used
  - Grow a decision tree by making a series of locally optimum decisions about which attribute to use for splitting the data

Hunt's algorithm

- Let D_t be the set of training records that reach a node t and $y=\{y_1,\ldots y_c\}$ the class labels
- General Procedure
  - If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
  - If D_t is an empty set, then t is a leaf node labeled by the default class, y_d
  - If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

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





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

Tree Induction Issues

- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?
- Determine when to stop splitting

Tree Induction Issues

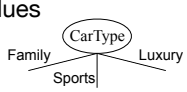
- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?
- Determine when to stop splitting

How to Specify Test Condition?

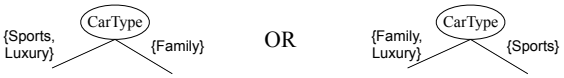
- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

Splitting Based on Nominal Attributes

- **Multi-way split**: use as many partitions as distinct values

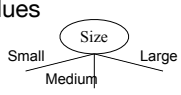


- **Binary split**: divide values into two subsets; need to find optimal partitioning

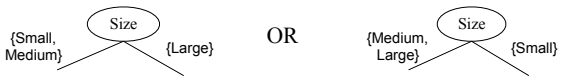


Splitting Based on Ordinal Attributes

- **Multi-way split**: use as many partitions as distinct values



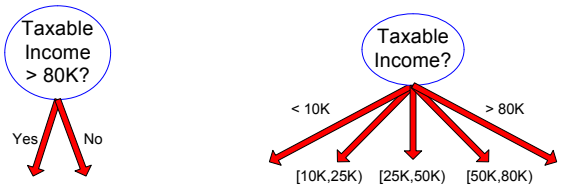
- **Binary split**: divides values into two subsets; need to find optimal partitioning



Splitting Based on Continuous Attributes

- Different ways of handling
  - **Discretization** to form an ordinal categorical attribute
    - Static – discretize once at the beginning
    - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering
- Binary Decision: $(A < v)$ or $(A \geq v)$
  - consider all possible splits and finds the best cut
  - can be more compute intensive

Splitting Based on Continuous Attributes

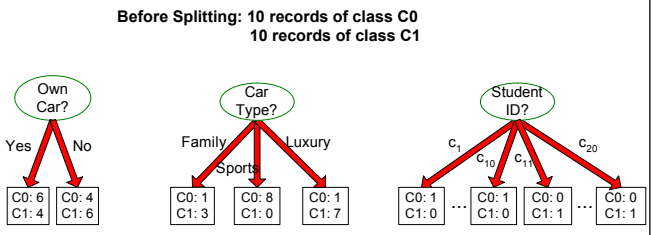


(i) Binary split (ii) Multi-way split

Tree Induction Issues

- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?
- Determine when to stop splitting

Determining the Best Split



Which test condition is the best?

Determining the Best Split

- Greedy approach:
  - Nodes with **homogeneous** class distribution are preferred
- Need a measure of node **impurity**:

C0: 5 C1: 5	C0: 9 C1: 1
Non-homogeneous, High degree of impurity	**Homogeneous, Low degree of impurity**

Impurity Measures

- Measuring the impurity of a node
  - ***P(i\|t)*** = fraction of records belonging to class *i* at a given node *t*
  - *c* is the number of classes

$$\text{Entropy}(t) = -\sum_{i=0}^{c-1} P(i|t) log_2 P(i|t)$$

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} P(i|t)^2$$

$$\text{Classification error}(t) = 1 - \max P(i|t)$$

Entropy

$$\text{Entropy}(t) = -\sum_{i=0}^{c-1} P(i|t) log_2 P(i|t)$$

- Maximum (log n_c) when records are equally distributed among all classes implying least information
- Minimum (0.0) when all records belong to one class, implying most information

Exercise

$$\text{Entropy}(t) = -\sum_{i=0}^{c-1} P(i|t) log_2 P(i|t)$$

C1	**0**
C2	**6**

C1	**1**
C2	**5**

C1	**2**
C2	**4**

Exercise

$$\text{Entropy}(t) = -\sum_{i=0}^{c-1} P(i|t) log_2 P(i|t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1
Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6
Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (5/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6
Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

GINI

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} P(i|t)^2$$

- Maximum (1 - 1/n_c) when records are equally distributed among all classes, implying least interesting information
- Minimum (0.0) when all records belong to one class, implying most interesting information

Exercise

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} P(i|t)^2$$

C1	**0**
C2	**6**

C1	**1**
C2	**5**

C1	**2**
C2	**4**

Exercise

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} P(i|t)^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1
Gini = 1 – P(C1)² – P(C2)² = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6
Gini = 1 – (1/6)² – (5/6)² = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6
Gini = 1 – (2/6)² – (4/6)² = 0.444

Classification Error

$$\text{Classification error}(t) = 1 - \max P(i|t)$$

- Maximum (1 - 1/n_c) when records are equally distributed among all classes, implying least interesting information
- Minimum (0.0) when all records belong to one class, implying most interesting information

Exercise

$$\text{Classification error}(t) = 1 - \max P(i|t)$$

C1	**0**
C2	**6**

C1	**1**
C2	**5**

C1	**2**
C2	**4**

Exercise

$$\text{Classification error}(t) = 1 - \max P(i|t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1
Error = 1 – max (0, 1) = 1 – 1 = 0

C1	**1**
C2	**5**

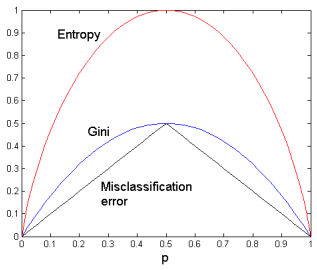
P(C1) = 1/6 P(C2) = 5/6
Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6
Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

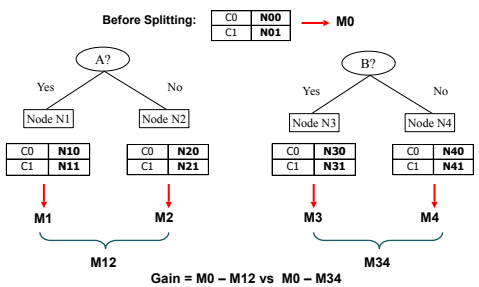
Comparison of Impurity Measures

For a 2-class problem:



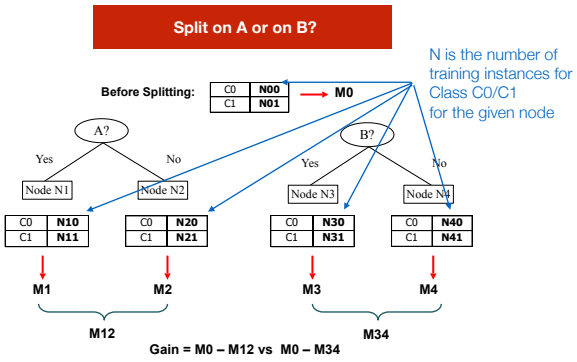
Gain = goodness of a split

Split on A or on B?

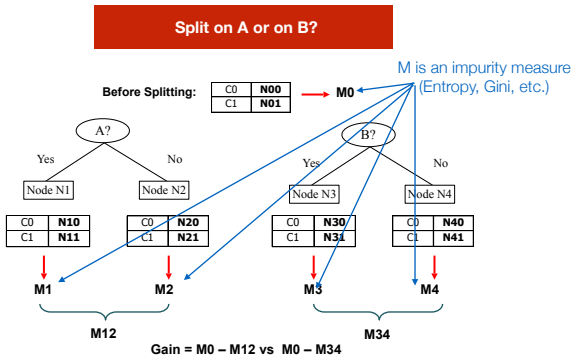


Gain = M0 – M12 vs M0 – M34

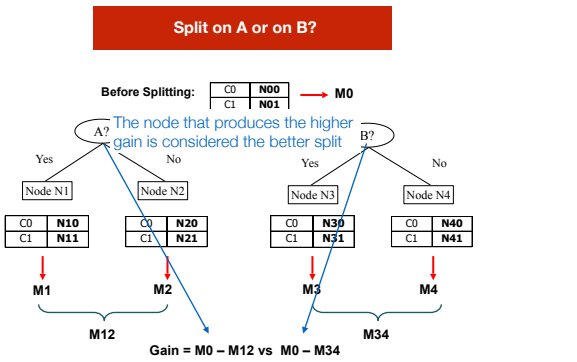
Gain = goodness of a split



Gain = goodness of a split



Gain = goodness of a split



Information Gain

- When Entropy is used as the impurity measure, it's called **information gain**
- Measures how much we gain by splitting a parent node

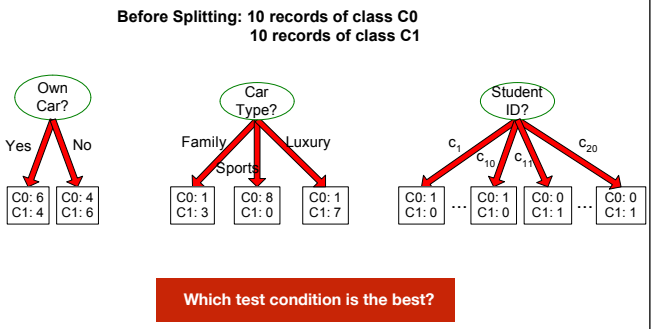
number of attribute values

number of records associated with the child node v_j

$$\Delta_{info} = \text{Entropy}(p) - \sum_{j=1}^{k} \frac{N(v_j)}{N} \text{Entropy}(v_j)$$

total number of records at the parent node

Determining the Best Split



Gain Ratio

- Can be used instead of information gain

$$\text{Gain ratio} = \frac{\Delta_{info}}{\text{Split info}}$$

$$\text{Split info} = -\sum_{i=1}^{k} P(v_i) \log_2 P(v_i)$$

- It the attribute produces a large number of splits, its split info will also be large, which in turn reduces its gain ratio

Tree Induction Issues

- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?
- Determine when to stop splitting

Stopping Criteria for Tree Induction

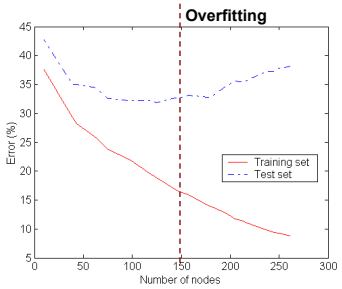
- Stop expanding a node when all the records belong to the same class
- Stop expanding a node when all the records have similar attribute values
- Early termination
  - See details in a few slides

Summary Decision Trees

- Inexpensive to construct
- Extremely fast at classifying unknown records
- Easy to interpret for small-sized trees
- Accuracy is comparable to other classification techniques for many simple data sets

Practical Issues of Classification

Underfitting and Overfitting



Underfitting: when model is too simple, both training and test errors are large

How to Address Overfitting

- **Pre-Pruning** (Early Stopping Rule): stop the algorithm before it becomes a fully-grown tree
  - Typical stopping conditions for a node
    - Stop if all instances belong to the same class
    - Stop if all the attribute values are the same (i.e., belong to the same split)
  - More restrictive conditions
    - Stop if number of instances is less than some user-specified threshold
    - Stop if class distribution of instances are independent of the available features
    - Stop if expanding the current node does not improve impurity measures (e.g., Gini or information gain)

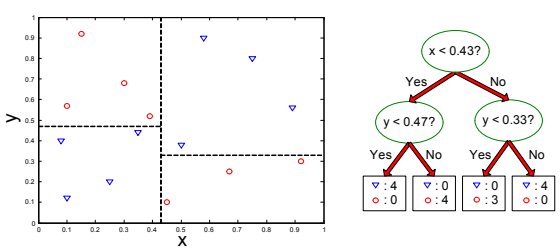
How to Address Overfitting

- **Post-pruning**: grow decision tree to its entirety
  - Trim the nodes of the decision tree in a bottom-up fashion
  - If generalization error improves after trimming, replace sub-tree by a leaf node
  - Class label of leaf node is determined from majority class of instances in the sub-tree

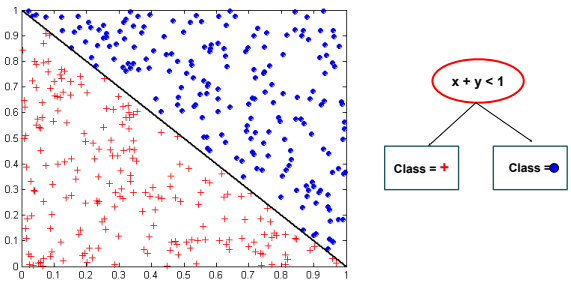
Methods for estimating performance

- Holdout
  - Reserve 2/3 for training and 1/3 for testing (validation set)
- Cross validation
  - Partition data into k disjoint subsets
  - k-fold: train on k-1 partitions, test on the remaining one
  - Leave-one-out: k=n

Expressivity



Expressivity



Exercise

Assignment 1

https://github.com/kbalog/uis-dat630-fall2016/tree/master/assignment-1