University
Politehnica
of Bucharest

Faculty of
Automatic
Control and
Computers

Computer
Science and
Engineering
Department

Supervised Leaning
Decision Trees

Ciprian-Octavian Truică
ciprian.truica@cs.pub.ro

Overview

- Supervised learning
- How to read a data set
- Classification
- Evaluation Methods for Classification
- Decision Trees
- Summary

Overview

- Supervised learning
- How to read a data set
- Classification
- Evaluation Methods for Classification
- Decision Trees
- Summary

Supervised Learning

- Supervised learning is one of the most studied subdomains of Data Mining and Machine Learning, a branch of Artificial Intelligence.
- Means that a new model can be built starting from past experiences (data)

Supervised Learning

- Supervised learning includes:
  - Classification:
    - Results are discrete values
    - Goal: identify group/class membership
  - Regression:
    - Results are continuous or ordered values
    - Goal: estimate or predict a response

Overview

- Supervised learning
- How to read a data set
- Classification
- Evaluation Methods for Classification
- Decision Trees
- Summary

How to read a dataset

- Given a set of n observations

$$X = \{x_1, x_2, \dots, x_n\}$$

- Each observation contains m attributes

$$A = \{A_1, A_2, \dots, A_m\}$$

- Each attribute has an domain of values

$$dom(A_i) = \{a_{i1}, a_{i2}, \dots, a_{ip}\}$$

- This means that an observation x_i can be written as

$$x_i = \left\{a_1^{(i)}, a_2^{(i)}, \dots, a_m^{(i)}\right\}$$

$$a_1^{(i)} \in dom(A_1), a_2^{(i)} \in dom(A_2), \dots, a_m^{(i)} \in dom(A_m)$$

How to read a dataset

- Given a set of k labels (or classes)

$$C = \{c_1, c_2, \dots, c_k\}$$

- Sometimes classes are denoted by

$$Y = \{y_1, y_2, \dots, y_k\}$$

- Labeled dataset

$$D = \{(x_1, c_{1i}), (x_2, c_{2i}), \dots, (x_n, c_{ni})\}$$

How to read a dataset

Observations	Outlook	Temperature	Humidity	Windy	Class
x_1	Sunny	Hot	High	FALSE	No
x_2	Sunny	Hot	High	TRUE	No
x_3	Overcast	Hot	High	FALSE	Yes
x_4	Rainy	Mild	High	FALSE	Yes
x_5	Rainy	Cool	Normal	FALSE	Yes
x_6	Rainy	Cool	Normal	TRUE	No
x_7	Overcast	Cool	Normal	TRUE	Yes
x_8	Sunny	Mild	High	FALSE	No
x_9	Sunny	Cool	Normal	FALSE	Yes
x_{10}	Rainy	Mild	Normal	FALSE	Yes
x_{11}	Sunny	Mild	Normal	TRUE	Yes
x_{12}	Overcast	Mild	High	TRUE	Yes
x_{13}	Overcast	Hot	Normal	FALSE	Yes
x_{14}	Rainy	Mild	High	TRUE	No

- The number of observations $n = 14$
- The number of attributes $m = 4$
- The number of classes $k = 2$
- The classes are $C = \{Yes, No\}$
- For element (x_4, c_{4i}) the class $c_{4i} = Yes$

- The attributes are $A = \{Outlook, Temperature, Humidity, Windy\}$
- The domain of A_1 is $dom(A_1) = \{Sunny, Overcast, Rainy\}$
- For $i = 3$ then $x_i = \{a_1^{(i)}, a_2^{(i)}, \dots, a_m^{(i)}\}$
- Becomes $x_i = \{Overcast, Hot, High, False\}$
- In other words:

$$a_1^{(3)} = Overcast$$
$$a_2^{(3)} = Hot$$
$$a_3^{(3)} = High$$
$$a_4^{(3)} = False$$

Overview

- Supervised learning
- How to read a data set
- Classification
- Evaluation Methods for Classification
- Decision Trees
- Summary

Classification

- Classification is the process of identifying to which set of categories (classes) a new observation belongs by constructing a model using past labeled observations
- For constructing the model, classification uses:
  - A training set – a set already labeled with classes used for building the model
  - A validation set – a set already labeled with classes that was not used for building the model
  - A test set - a set with no labels for which the class of each element are identified by the model

Classification

- **Input:**
  - A set of k classes $C = \{c_1, c_2, \dots, c_k\}$
  - A set of n labeled items $D = \{(x_1, c_{i1}), (x_2, c_{i2}), \dots, (x_n, c_{in})\}$.
  - The items $x_1, x_2, \dots, x_n$ are each labeled with a class $c_j \in C$.
  - Each item constrains a set m attributes $A = \{A_1, A_2, \dots, A_m\}$, in other words $x_i = \left\{a_1^{(i)}, a_2^{(i)}, \dots, a_m^{(i)}\right\}$
  - D is called the **training set**.
  - For calibration of some algorithms, a **validation set** is also required. This validation set contains also labeled items not included in the training set.
- **Output:**
  - A model or method for classifying new items.
  - The set of new items that will be classified using this model/method is called the **test set**

Classification

- Classification can be seen as 4 separate problems:
  - Binary Classification
    - Only 2 classes are involved in classification
  - Multiclass Classification
    - Multiple classes are involved in classification
  - Multilabel Classification
    - Each observation can belong to multiple classes
  - Hierarchical Classification
    - Each class can be divided into multiple subclasses.
    - The goal is to classify the input into one of these subclasses.

Classification

- Based on the dataset, classification can be divided into 2 other separate problems:
  - Balanced
    - The number of records in the dataset are almost similar for all the classes
  - Imbalanced
    - The number of records in the datasets varies widely for each class
- Real world problems fall into the imbalanced multiclass/multilabel problems

Classification

- Dealing with imbalanced datasets:

1. Undersampling
   - Removing elements from the dataset for the classes with the larger number of elements to achieve balance
2. Oversampling
   - Duplicating elements from the dataset for the classes with a small number of elements to achieve balance

Classification

- Classification Algorithms :
  - Class association rules (CAR)
  - Decision Trees
  - Rule Induction Systems – uses a decision tree to infer a set of rules that can replaces the decision tree.
  - Naïve Bayesian classification: for every observation $x_i \in D$ the method computes the probability $P(c_j|x_i)$ for each class $c_j \in C$.

Classification

- Classification Algorithms :
  - Support vector machines (SVM)
    - Used for binary classification
    - Can be extended to multiclass classification
    - Also used for outlier detection (one class SVM)
  - K-Nearest Neighbor (KNN)
    - Classifies an observation using the classes of its neighbors

Classification

- Classification Algorithms :
  - Ensemble methods
    - Bootstrap
    - Bagging
    - Random Forest
    - Extremely Randomized Trees
    - Boosting
    - AdaBoost

Overview

- Supervised learning
- How to read a data set
- Classification
- Evaluation Methods for Classification
- Decision Trees
- Summary

Evaluation Methods for Classification

- For estimating the efficiency of a classifier several methods can be used:
  - Accuracy
  - Error Rate
  - Precision
  - Recall
  - Specificity
  - F-score

Evaluation Methods for Classification

- These measures have different formulas depending on the type of classification problem solved:
  - Binary Classification
  - Multiclass Classification
  - Imbalanced Multiclass Classification
  - Multilabel classification – not discussed see [Sokolova 2009]

Binary Classification Evaluation

- Binary Classification Evaluation
  - The confusion table is used for evaluating classification

Data Class	Classified as Positive	Classified as Negative
Actual Positive	TP (True Positive)	FN (False Negative)
Actual Negative	FP (False Positive)	TN (True Negative)

Binary Classification Evaluation

- Accuracy (A) – measures the overall effectiveness of a classifier

$$A = \frac{TP + TN}{TP + FP + TN + FN}$$

- Error Rate (ER) – is the proportion of incorrectly classified test examples

$$ER = 1 - A = \frac{FP + FN}{TP + FP + TN + FN}$$

Binary Classification Evaluation

- Precision (P) measures the class agreement of the data labels with the positive labels given by the classifier.

$$P = \frac{TP}{TP + FP}$$

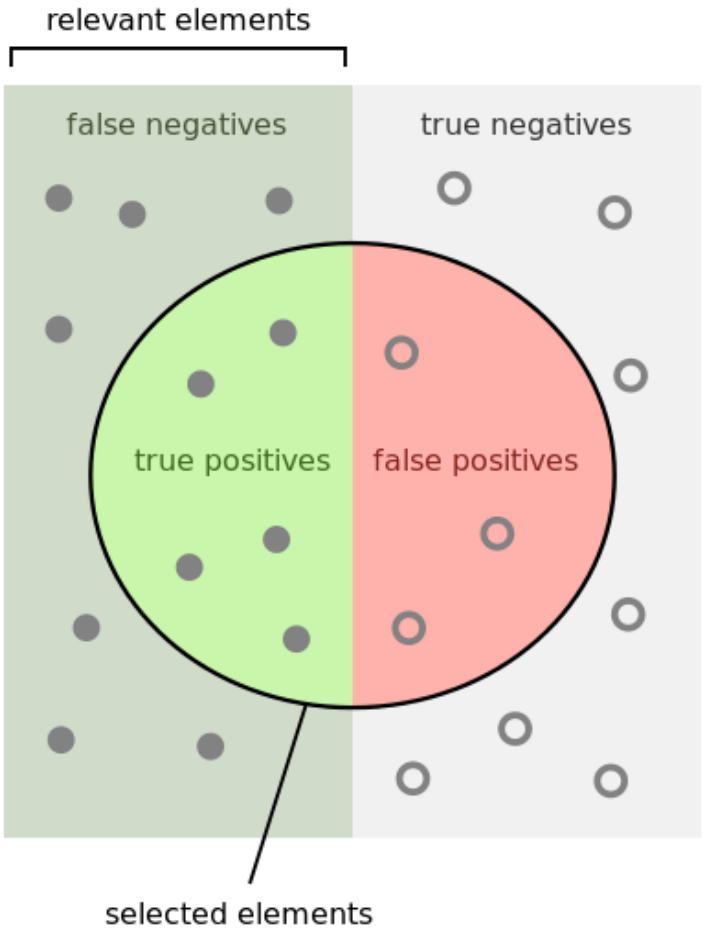
- Recall (R) or sensitivity measures the effectiveness of a classifier to identify positive labels.

$$R = \frac{TP}{TP + FN}$$

- Specificity (S) measures how effectively a classifier identifies negative labels.

$$S = \frac{TN}{TN + FP}$$

Binary Classification Evaluation



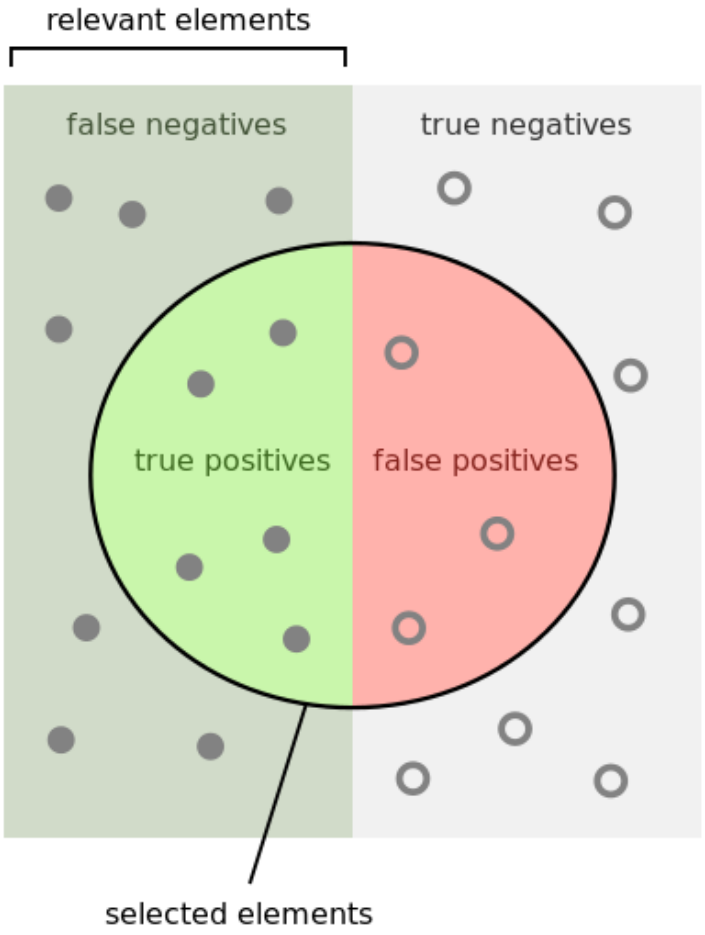
How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

Binary Classification Evaluation



How many relevant items are selected? e.g. How many sick people are correctly identified as having the condition.

Sensitivity=

How many negative selected elements are truly negative? e.g. How many healthy peple are identified as not having the condition.

Specificity =

Binary Classification Evaluation

- F-Score (F_β) – measures the relations between data's positive classes and those given by a classifier

$$F_\beta = (\beta^2 + 1) \cdot \frac{P \cdot R}{\beta^2 P + R}$$

- The F_1 and F_2 are usually used

$$F_1 = (1^2 + 1) \cdot \frac{P \cdot R}{1^2 P + R} = \frac{2 \cdot P \cdot R}{P + R}$$

$$F_2 = (2^2 + 1) \cdot \frac{P \cdot R}{2^2 P + R} = \frac{5 \cdot P \cdot R}{4 \cdot P + R}$$

Binary Classification Evaluation

- We have 30 animals 16 cats and 14 dogs:
- After classification:
  - 14 cats are classified as cats
  - 2 cats are classified as dogs
  - 10 dogs are classified as dogs
  - 4 dogs are classified as cats

	Classified as cats	Classified as dogs
Actual cats	14	2
Actual dogs	4	10

$$A = \frac{24}{30} = 0.8 \qquad ER = 1 - A = 0.2$$

$$P = \frac{14}{14+4} \cong 0.778 \qquad R = \frac{14}{14+2} = 0.875$$

$$S = \frac{10}{10+4} \cong 0.714 \qquad F_1 \cong 0.824 \qquad F_2 \cong 0.854$$

Multiclass Classification Evaluation

- Multiclass Classification Evaluation for N classes uses modified versions of the measures presented.
- These measures are also the generalized versions for evaluating classification.
- For binary classification $N = 2$

Multiclass Classification Evaluation

- Average Accuracy ($avgA$) – measures the per-class effectiveness of a classifier

$$avgA = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i}$$

- Error Rate ($avgER$) – measures the average per-class classification error

$$avgER = \frac{1}{N}\sum_{i=1}^{N}\frac{FP_i + FN_i}{TP_i + FP_i + TN_i + FN_i}$$

Multiclass Classification Evaluation

- Micro Precision (μP) measures the per-class agreement of the data labels with the positive labels given by the classifier.

$$\mu P = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} (TP_i + FP_i)}$$

- Macro Precision (MP) measures the average per-class agreement of the data class labels with those of the classifiers

$$MP = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i}{TP_i + FP_i}$$

Multiclass Classification Evaluation

- Micro Recall (μR) or Micro Sensitivity measures the per-class effectiveness of a classifier to identify positive labels.

$$\mu R = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + FN_i}$$

- Macro Recall (MR) or Macro Sensitivity measures the average per-class effectiveness of the classifier to identify labels.

$$MR = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i}{TP_i + FN_i}$$

Multiclass Classification Evaluation

- Micro Specificity (μS) measures the per-class effectiveness of a classifier to identifies negative labels.

$$\mu S = \frac{\sum_{i=1}^{N} TN_i}{\sum_{i=1}^{N} (TN_i + FP_i)}$$

- Macro Specificity (MS) measures the average per-class effectiveness of the classifier to identify negative labels.

$$MS = \frac{1}{N} \sum_{i=1}^{N} \frac{TN_i}{TN_i + FP_i}$$

Multiclass Classification Evaluation

- Micro F-Score (μF_β) – measures the per-class relations between the data's positive classes and those given by a classifier

$$\mu F_\beta = (\beta^2 + 1) \cdot \frac{\mu P \cdot \mu R}{\beta^2 \cdot \mu P + \mu R}$$

- Macro F-Score (MF_β) – measures average per-class relations between the data's positive classes and those given by a classifier

$$MF_\beta = (\beta^2 + 1) \cdot \frac{MP \cdot MR}{\beta^2 \cdot MP + MR}$$

Multiclass Classification Evaluation

- We have 50 animals: 18 cats, 17 dogs, 15 rabbits:
- After classification:
  - 14 cats are classified as cats
  - 2 cats are classified as dogs
  - 2 cats are classified as rabbits
  - 10 dogs are classified as dogs
  - 4 dogs are classified as cats
  - 3 dogs are classified as rabbits
  - 13 rabbits are classified as rabbits
  - 2 rabbits are classified as cats
  - 0 rabbits are classified as dogs

	Classified as cats	Classified as dogs	Classified as rabbits
Actual cats	14	2	2
Actual dogs	4	10	3
Actual rabbits	2	0	13

$$avgA = \frac{1}{3} \cdot \left(\frac{14+26}{50} + \frac{10+31}{50} + \frac{13+30}{50} \right) \cong 0.827$$

Imbalanced Multiclass Classification Evaluation

- The evaluation methods of a classier must be adapted to take into account the weights of each class
- The weight w_i of class c_i is computed as the total number of observations n divided by the number of observations n_i for that class c_i multiplied by the total number of classes N

$$w_i = \frac{n}{N \cdot n_i} \Leftrightarrow \frac{1}{w_i} = \frac{N \cdot n_i}{n}$$

Imbalanced Multiclass Classification Evaluation

- All the Micro measures remain the same for Imbalanced Multiclass Classification
- These weighted measures alter the generalized Macro (Average)
- The weighted versions calculates metrics for each label, and their average, weighted by support

Imbalanced Multiclass Classification Evaluation

- Weighted Average Accuracy (wA) – measures the per-class effectiveness of a classifier divided by the class weight

$$wA = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{1}{w_i} \cdot \frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i}$$

$$= \sum_{i=1}^{N} \frac{n_i}{n} \cdot \frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i}$$

Imbalanced Multiclass Classification Evaluation

- Error Rate (wER) – measures the average per-class classification error divided by the class weight

$$wER = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{1}{w_i} \cdot \frac{FP_i + FN_i}{TP_i + FP_i + TN_i + FN_i}$$
$$= \sum_{i=1}^{N} \frac{n_i}{n} \cdot \frac{FP_i + FN_i}{TP_i + FP_i + TN_i + FN_i}$$

Imbalanced Multiclass Classification Evaluation

- Macro Precision (wP) measures the average per-class divided by the weight of the class agreement of the data class labels with those of the classifiers

$$wP = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{1}{w_i} \cdot \frac{TP_i}{TP_i + FP_i}$$

$$= \sum_{i=1}^{N} \frac{n_i}{n} \cdot \frac{TP_i}{TP_i + FP_i}$$

Imbalanced Multiclass Classification Evaluation

- Macro Recall (wR) or Macro Sensitivity measures the average per-class effectiveness of the classifier to identify labels divided by the weight of the class.

$$wR = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{1}{w_i} \frac{TP_i}{TP_i + FN_i}$$

$$= \sum_{i=1}^{N} \frac{n_i}{n} \frac{TP_i}{TP_i + FN_i}$$

Imbalanced Multiclass Classification Evaluation

- Macro Specificity (wS) – measures the weighted average per-class effectiveness of the classifier to identify negative labels.

$$wS = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{1}{w_i} \cdot \frac{TN_i}{TN_i + FP_i}$$
$$= \sum_{i=1}^{N} \frac{n_i}{n} \cdot \frac{TN_i}{TN_i + FP_i}$$

Imbalanced Multiclass Classification Evaluation

- Macro F-Score (wF_β) – measures weighted average per-class relations between the data's positive classes and those given by a classifier

$$wF_\beta = (\beta^2 + 1) \cdot \frac{wP \cdot wR}{\beta^2 \cdot wP + wR}$$

Multiclass Classification Evaluation

- We have 10,000 animals: 7,000 cats, 2,500 dogs, 500 rabbits:
- Cats weight: $\frac{10{,}000}{3 \cdot 7{,}000}$
- Dogs weight: $\frac{10{,}000}{3 \cdot 2{,}500}$
- Rabbits weight: $\frac{10{,}000}{2 \cdot 500}$
- After classification:
  - 6,500 cats are classified as cats
  - 400 cats are classified as dogs
  - 100 cats are classified as rabbits
  - 2,250 dogs are classified as dogs
  - 200 dogs are classified as cats
  - 50 dogs are classified as rabbits
  - 450 rabbits are classified as rabbits
  - 35 rabbits are classified as cats
  - 15 rabbits are classified as dogs

	Classified as cats	Classified as dogs	Classified as rabbits
Actual cats	6,500	400	100
Actual dogs	200	2,250	50
Actual rabbits	35	15	350

$$wA = 0.7 \cdot \frac{6500 + 2665}{10000} + 0.25 \cdot \frac{2250 + 6985}{10000} + 0.05 \cdot \frac{350 + 9350}{10000} \cong 0.921$$

Classification Evaluation

- A training and a test set are used for evaluation
- Usually the labeled data set is split into multiple sets to obtain the training and test sets
- Methods:
  - Holdout method
  - K-folds Cross validation
  - Stratified cross validation

Classification Evaluation

- Holdout method
  - The dataset is split into 2 sets
  - The training set contains usually two thirds of the data set
  - The test set contains the remaining elements

Classification Evaluation

- K-fold cross validation
  - The dataset is split into K disjoint subsets with the same size
  - For each subset a model is build using it as the training set and the rest K-1 sets are used as test sets
  - In this way K values are obtained for each evaluation measure
  - The mean of each measure is the final measure used for evaluation

Classification Evaluation

- Stratified cross validation
  - An adaptation of K-folds validation
  - Each fold has the same distribution of labels
  - Used for imbalanced datasets

Overview

- Supervised learning
- How to read a data set
- Classification
- Evaluation Methods for Classification
- Decision Trees
- Summary

Decision Trees

- A Decision Tree is used to build a classification or a regression models in the form of a tree structure
- It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.
- The final result is a tree with **decision nodes** and **leaf nodes**.

Decision Trees

- The **decision nodes** are attributes
- Branches refer to discrete values (one or more) or intervals for these attributes
- Leaves are labeled with classes.
- For each leaf, a support and a confidence may be computed:
  - Support is the proportion of examples matching the path from root to that leaf
  - Confidence is the classification accuracy for examples matching that path.

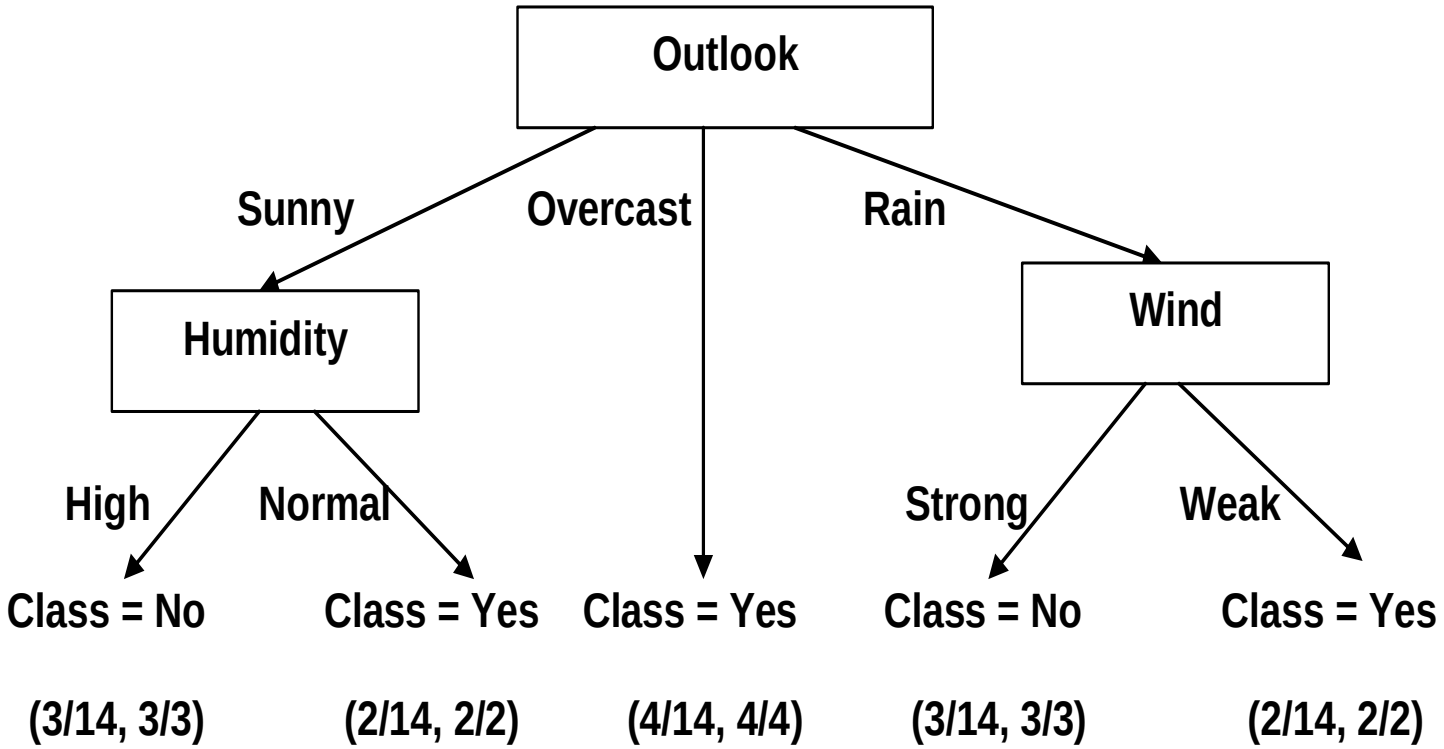
Decision Trees

- Each path from the root to a leaf can be used as a class association rule
- When passing from decision trees to rules, each rule has the same support and confidence as the leaf from where it comes.
- Any example match a single path of the tree (so a single leaf or class).

Decision Trees

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Decision Trees



Decision Trees – ID3

- ID3 stands for Iterative Dichotomiser 3 and is a greedy algorithm for building decision trees introduced by Ross Quinlan in 1986 (see [Quinlan 1986]).
- The algorithm constructs the decision tree in a top-down manner choosing at each node the ‘best’ attribute for branching.

Decision Trees – ID3

- ID3 steps:
  1. First a root attribute is chosen
     - Build a separate branch for each different value of the attribute.
     - The training set is also divided
       - Each branch inheriting the examples matching the attribute value of the branch.
     - An attribute cannot be chosen twice on the same path
     - From the moment an attribute was chosen for a node it will never be tested again for the descendants of that node
  2. Process repeats for each descendant until
     - All examples have the same class (in that case the node becomes a leaf labeled with that class)
     - All attributes have been used (the node also become a leaf labeled with the mode value – the majority class).

Decision Trees – ID3

- Given a set of k classes

$$C = \{c_1, c_2, \dots, c_k\}$$

- Let us consider a labeled dataset

$$D = \{(x_1, c_{1i}), (x_2, c_{2i}), \dots, (y_n, c_{ni})\}$$

- Each observation x_j contains m attributes

$$A = \{A_1, A_2, \dots, A_m\}$$

- If the attributes are continuous, then they should be replaced with discrete values

Decision Trees – ID3

- A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous)
- ID3 algorithm uses entropy to calculate the homogeneity of a sample
- If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one
- The entropy of D can be computed as:

$$H(D) = -\sum_{i=1}^{k} p(c_i) \log_2 p(c_i)$$

- $p(c_i)$ is the probability of class c_i in the dataset D
- The probability $p(c_i)$ is determined by counting

Decision Trees – ID3

- If attribute A_t having r distinct values is considered for branching, it will partition D in r disjoint subsets $D_1, D_2, \dots, D_r$

$$D = \bigcup_{i=1}^{r} D_i$$

- The entropy of the split is given by

$$H(D|A_t) = \sum_{i=1}^{r} \frac{|D_i|}{|D|} \cdot H(D_i)$$

- $\frac{|D_i|}{|D|}$ is the proportion of elements in D_i to the number elements in the dataset D

Decision Trees – ID3

- Because the purity of the datasets is increasing, $H(D)$ is bigger than $H(D|A_t)$.
- The difference between them is called the ***information gain***:

$$IG(D, A_t) = H(D) - H(D|A_t)$$

- The 'best' attribute is determined by the highest gain

Decision Trees – ID3 – Issues

- Because is a greedy algorithm it leads to a local optimum
- Attributes with many values leads to a higher gain
- For solving this problem the gain may be replaced with the ***gain ratio***:

$$IGR(D, A_t) = \frac{IG(D, A_t)}{-\sum_{i=1}^{r} \frac{|D_i|}{|D|} \cdot \log_2 \frac{|D_i|}{|D|}}$$

- Sometimes (when only few examples are associated with leaves) the tree overfits the training data and does not work well on test examples

Decision Trees – ID3

- Some attributes are more costly then others
- Costly attributes
  - It is better that lower-cost attributes to be closer to the root than other attributes.
  - This may be done by weighting the gain by the cost:

$$WIG(D, A_t) = \frac{\left(IG(D, A_t)\right)^2}{cost(A_t)}$$

Decision Trees – ID3

- To avoid overfitting, the tree may be simplified by pruning:
  - Pre-pruning:
    - Growing is stopped before normal end.
    - The leaves are not 100% pure and are labeled with the majority class (the mode value).
  - Post-pruning:
    - After running the algorithm some sub-trees are replaced by leaves.
    - In this case the labels are mode values for the matching training examples.
    - Post-pruning is better because in pre-pruning is hard to estimate when to stop.

Decision Trees – C4.5

- C4.5 is the improved version of ID3
- Also developed by Ross Quinlan
- Some characteristics:
  - Numeric (continuous) attributes are allowed
  - Deal sensibly with missing values
  - Post-pruning to deal with for noisy data
- An even improved version is C5.0.

Decision Trees – C4.5

- The most important improvements from ID3 are:
  1. The attributes are chosen based on gain-ratio and not simply gain.
  2. Post pruning is performed in order to reduce the tree size.
     - The pruning is made only if it reduces the estimated error.
     - There are two prune methods:
       - Sub-tree replacement: Select a sub-tree and replaces it with a single leaf.
       - Sub-tree raising: selects a subtree and replaces it with a child node one (a sub-tree replaces his parent).

Decision Trees – CART

- CART (Classification And Regression Tree) works on the same principles as ID3
- The main difference is in the function used for calculating the homogeneity of a sample (computing the split)
- CART uses the Gini Impurity:

$$GI(D) = \sum_{i=1}^{k} \frac{p(c_i)}{|D|} \cdot \left(1 - \frac{p(c_i)}{|D|}\right) = 1 - \sum_{i=1}^{k} \left(\frac{p(c_i)}{|D|}\right)^2$$

- The Gini Impurity for the split is given by the:

$$GI(D|A_t) = \sum_{i=1}^{r} \frac{|D_i|}{|D|} \cdot GI(D_i)$$

- Unlike Information Gain where we take the maximum to make the split, in this case we take the minimum

Overview

- Supervised learning
- How to read a data set
- Classification
- Evaluation Methods for Classification
- Decision Trees
- Summary

Summary

- This course presented:
  - Classification
  - Evaluation Methods for Classification
    - Accuracy
    - Error Rate
    - Precision
    - Recall
    - Sensitivity
    - F-Score
  - Decision Trees
    - ID3
    - C4.5
    - CART

References

- [Sokolova 2009] M. Sokolova and G. Lapalme (2009). A systematic analysis of performance measures for classification tasks, Information Processing & Management, 45(4), 427-437.
- [Quinlan 1986] J.R. Quinlan (1986). Induction of Decision Trees, Machine Learning, 1(1), 81-106