

# Chapter VI: Classification

## Knowledge Discovery in Databases

Luciano Melodia M.A.

Evolutionary Data Management, Friedrich-Alexander University Erlangen-Nürnberg

Summer semester 2021



## Chapter VI: Classification

### **Classification: basic concepts.**

Decision-tree induction.

Bayes classification methods.

Rule-based classification.

Model evaluation and selection.

Techniques to improve classification accuracy: ensemble methods.

Summary.

# Supervised vs. unsupervised learning

## Supervised learning (classification).

Supervision:

The **training data** (observations, measurements, etc.) are accompanied by **labels** indicating the **class** of the observations.

New data is classified based on a **model** created from the training data.

## Unsupervised learning (clustering).

The class labels of training data are unknown.

Or rather, there are no training data.

Given a set of measurements, observations, etc., the goal is to find classes or clusters in the data.

See next chapter.

## Prediction problems: classification vs. numerical prediction

### Classification:

Predicts **categorical class labels** (discrete, nominal).

Constructs a model based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data.

### Numerical prediction:

Models **continuous-valued functions**.

I.e. predicts missing or unknown (future) values.

### Typical applications of classification:

Credit/loan approval: Will it be paid back?

Medical diagnosis: Is a tumor cancerous or benign?

Fraud detection: Is a transaction fraudulent or not?

Web-page categorization: Which category is it?

## Classification – a two-step process

### Model construction: describing a set of predetermined classes:

Each tuple/sample is assumed to belong to a predefined class, as determined by the **class-label attribute**.

The set of tuples used for model construction is the **training set**.

The **model** is represented as classification rules, decision trees, or mathematical formulae.

### Model usage, for classifying future or unknown objects:

Estimate **accuracy** of the model:

The known label of **test samples** is compared with the result from the model.

**Accuracy rate** is the percentage of test-set samples that are correctly classified by the model.

Test set is independent of training set (otherwise overfitting).

If the accuracy is acceptable, **use the model** to classify data tuples whose class labels are not known.

## Classification – a two-step process



## Process (II): using the model in prediction



## Chapter VI: Classification

Classification: basic concepts.

**Decision-tree induction.**

Bayes classification methods.

Rule-based classification.

Model evaluation and selection.

Techniques to improve classification accuracy: ensemble methods.

Summary.

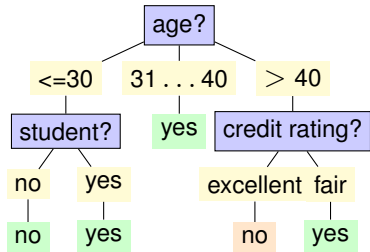


## Decision-tree induction: an example

### Training dataset: buys\_computer.

The dataset follows an example of Quinlan's ID3 (playing tennis).

### Resulting tree:



age	income	student	credit_rating	buys_coputer
$\leq 30$	high	no	fair	no
$\leq 30$	high	no	excellent	no
31 ... 40	high	no	fair	yes
$> 40$	medium	no	fair	yes
$> 40$	low	yes	fair	yes
$> 40$	low	yes	excellent	no
31 ... 40	low	yes	excellent	yes
$\leq 30$	medium	no	fair	no
$\leq 30$	low	no	fair	yes
$> 40$	medium	yes	fair	yes
$\leq 30$	medium	yes	excellent	yes
31 ... 40	medium	no	excellent	yes
31 ... 40	high	yes	fair	yes
$> 40$	medium	no	excellent	no

## Algorithm for decision-tree induction

### Basic algorithm (a greedy algorithm):

Tree is constructed in a **top-down recursive divide-and-conquer manner**.

Attributes are categorical.

If not: discretize in advance.

At start, all the training examples are at the root.

Examples are **partitioned recursively** based on selected attributes.

Test attributes are selected on the basis of a heuristic or statistical measure.

E.g. information gain – see on the next slide.

### Conditions for stopping partitioning:

All samples for a given node belong to the same class.

There are no remaining attributes for further partitioning.

Majority voting is employed for classifying the leaf.

There are no samples left (i.e. partition for particular value is empty).

## Attribute-selection measure: information gain (ID3/C4.5)

**Select the attribute with the highest information gain.**

Let  $p_i$  be the probability that an arbitrary tuple in  $D$  belongs to class  $C_i$ , estimated by  $\frac{|C_i|}{|D|}$ , such that  $1 \leq i \leq m$ .

**Expected information** (entropy) needed to classify a tuple in  $D$ :

$$\text{Info}(D) = - \sum_{i=1}^m p_i \log_2(p_i). \quad (1)$$

**Information** needed (after using attribute  $A$  to split  $D$  into  $v$  partitions) to classify  $D$ :

$$\text{Info}_A(D) = \sum_{j=1}^v \left( \frac{|D_j|}{|D|} \text{Info}(D_j) \right). \quad (2)$$

**Information gained** by branching on  $A$ :

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D). \quad (3)$$

## Attribute selection: information gain

**Class P:** buys\_computer = "yes"

**Class N:** buys\_computer = "no"

$$\text{Info}(D) = I(9, 5) = -\frac{9}{14} \log_2\left(\frac{9}{14}\right) - \frac{5}{14} \log_2\left(\frac{5}{14}\right) = 0.94$$

age	p	n	$I(p, n)$
$\leq 30$	2	3	0.971
31 ... 40	4	0	0
$> 40$	3	2	0.971

Similarly,

$$\text{Gain}(\text{income}) = 0.029,$$

$$\text{Gain}(\text{student}) = 0.151,$$

$$\text{Gain}(\text{credit\_rating}) = 0.048.$$

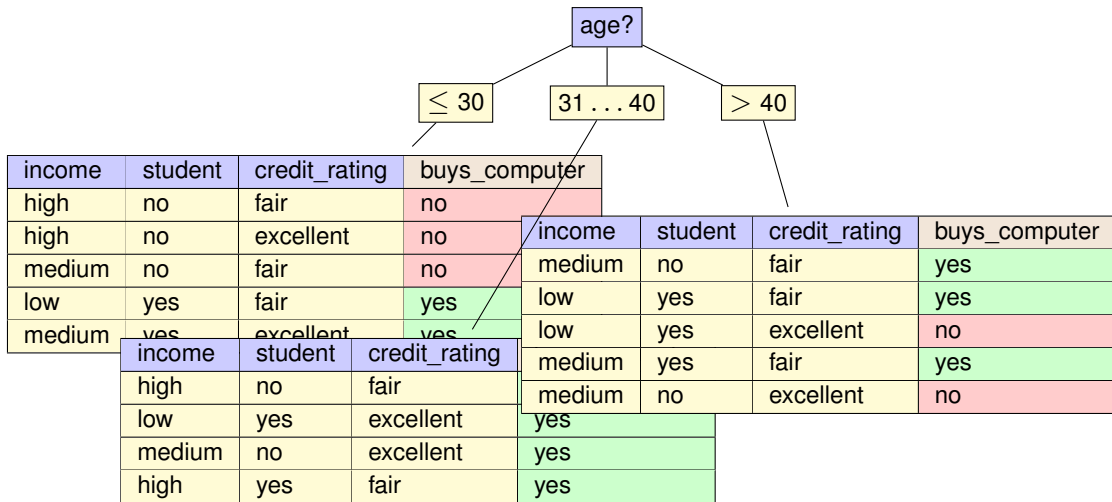
$$\text{Info}_{\text{age}}(D) = \frac{5}{14} I(2, 3) + \frac{4}{14} I(4, 0) + \frac{5}{14} I(3, 2) = 0.694.$$

$\frac{5}{14} I(2, 3)$  means "age  $\leq 30$ " has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence,

$$\text{Gain}(\text{age}) = \text{Info}(D) - \text{Info}_{\text{age}}(D) = 0.246.$$

age	income	student	credit_rating	buys_computer
$\leq 30$	high	no	fair	no
$\leq 30$	high	no	excellent	no
31 ... 40	high	no	fair	yes
$> 40$	medium	no	fair	yes
$> 40$	low	yes	fair	yes
$> 40$	low	yes	excellent	no
31 ... 40	low	yes	excellent	yes
$\leq 30$	medium	no	fair	no
$> 40$	medium	yes	fair	yes
$\leq 30$	medium	yes	excellent	yes
31 ... 40	medium	no	fair	yes
31 ... 40	high	yes	fair	yes
$> 40$	medium	no	excellent	no

## Partitioning in the example



## Computing information gain for continuous-valued attributes

**Let attribute A be a continuous-valued attribute.**

**Must determine the best split point for A.**

Sort the values of A in increasing order.

Typically, the midpoint between each pair of adjacent values is considered as a possible split point.

$\frac{a_i + a_{i+1}}{2}$  is the midpoint between the values of  $a_i$  and  $a_{i+1}$ .

The point with the minimum expected information requirement for A is selected as the split point for A.

**Split:**

$D_1$  is the set of tuples in  $D$  satisfying  $A \leq \text{split point}$ ,  
and  $D_2$  is the set of tuples in  $D$  satisfying  $A > \text{split point}$ .

**So to say: Discretization as you go along.**

For this particular purpose.

## Gain ratio for attribute selection (C4.5)

Information-gain measure is biased towards attributes with a large number of values. C4.5 (a successor of ID3) uses gain ratio to overcome the problem (normalization to information gain):

$$\text{SplitInfo}_A(D) = - \sum_{j=1}^v \frac{|D_j|}{|D|} \log_2 \left( \frac{|D_j|}{|D|} \right), \quad (4)$$

$$\text{GainRatio}(A) = \frac{\text{Gain}(A)}{\text{SplitInfo}_A(D)}. \quad (5)$$

Example:

$$\text{SplitInfo}_{\text{income}}(D) = -\frac{4}{14} \log_2 \left( \frac{4}{14} \right) - \frac{6}{14} \log_2 \left( \frac{6}{14} \right) - \frac{4}{14} \log_2 \left( \frac{4}{14} \right) = 1.557, \quad (6)$$

$$\text{GainRatio}(\text{income}) = \frac{0.029}{1.557} = 0.019. \quad (7)$$

The attribute with the maximum gain ratio is selected as the splitting attribute.

## Gini index

### **Corrado Gini (1884 – 1965).**

Italian statistician and sociologist.

### **Also called Gini coefficient.**

### **Measures statistical dispersion.**

Zero expresses perfect equality where all values are the same.

One expresses maximal inequality among values.

### **Based on the Lorenz curve.**

Plots the proportion of the total sum of values ( $y$ -axis) that is cumulatively assigned to the bottom  $x\%$  of the population.

Line at 45 degrees thus represents perfect equality of value distribution.

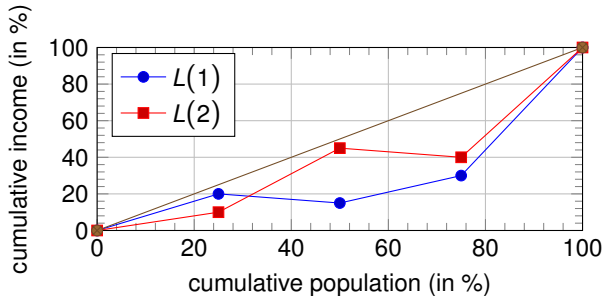
### **Gini coefficient then is . . .**

. . . the ratio of the area that lies between the line of equality and the Lorenz curve over the total area under the line of equality.



## Gini index (II)

Example: Distribution of incomes.



## Gini index (CART, IBM IntelligentMiner)

If a dataset  $D$  contains examples from  $n$  classes, Gini index  $\text{gini}(D)$  is defined as:

$$\text{gini}(D) = 1 - \sum_{j=1}^n p_j^2, \quad (8)$$

where  $p_j$  is the relative frequency of class  $j$  in  $D$ .

If a dataset  $D$  is split on  $A$  into two subsets  $D_1$  and  $D_2$ , the Gini index  $\text{gini}(D)$  is defined as

$$\text{gini}_A(D) = \frac{|D_1|}{|D|} \text{gini}(D_1) + \frac{|D_2|}{|D|} \text{gini}(D_2). \quad (9)$$

**Reduction in impurity:**

$$\Delta \text{gini}_A(A) = \text{gini}(D) - \text{gini}_A(D). \quad (10)$$

**The attribute  $A$  provides the smallest  $\text{gini}_A(D)$  (or the largest reduction in impurity) is chosen to split the node.**

Need to enumerate all the possible splitting points for each attribute.

## Computation of Gini index

### Example:

$D$  has 9 tuples in  $\text{buys\_computer} = \text{"yes"}$  and 5 in  $\text{"no"}$ , thus

$$\text{gini}(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459. \quad (11)$$

Suppose the attribute  $\text{income}$  partitions  $D$  into 10 in  $D_1 : \{\text{low}, \text{medium}\}$  and 4 in  $D_2 : \{\text{high}\}$ :

$$\text{gini}(D|_{D[\text{income}] = \text{"medium"}, \text{"low"}}) \quad (12)$$

$$= \left(\frac{10}{14}\right) \text{gini}(D_1) + \frac{4}{14} \text{gini}(D_2) \quad (13)$$

$$= \frac{10}{14} \left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14} \left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) = \quad (14)$$

$$= 0.443 = \text{gini}(D|_{D[\text{income}] = \text{"high"}}). \quad (15)$$

## Computation of Gini index (II)

**Example (cont.):**

$$\text{gini}(D|_{D[\text{income}] = \text{"low"}, \text{"high"}}) = 0.458,$$

$$\text{gini}(D|_{D[\text{income}] = \text{"medium"}, \text{"high"}}) = 0.450.$$

Thus, split on the {"low", "medium"} and {"high"}, since it has the lowest gini index.

**All attributes are assumed continuous-valued.**

**May need other tools, e.g. clustering, to get the possible split values.**

**Can be modified for categorical attributes.**

## Computation of Gini index (II)

**The three measures, in general, return good results, but**

**Information gain:**

Biased towards multi-valued attributes.

**Gain ratio:**

Tends to prefer unbalanced splits in which one partition is much smaller than the others.

**Gini index:**

Biased to multi-valued attributes.

Has difficulty when number of classes is large.

Tends to favor tests that result in equal-sized partitions and purity in both partitions.

## Other attribute-selection measures

### **CHAID:**

A popular decision-tree algorithm, measure based on  $\chi^2$  test for independence.

### **C-SEP:**

Performs better than information gain and Gini index in certain cases.

### **G-statistic:**

Has a close approximation to  $\chi^2$  distribution.

### **MDL (Minimal Description Length) principle:**

I.e. the simplest solution is preferred.

The best tree is the one that requires the fewest number of bits to both (1) encode the tree and (2) encode the exceptions to the tree.

### **Multivariate splits:**

Partitioning based on multiple variable combinations.

CART: finds multivariate splits based on a linear combination of attributes.

### **Which attribute-selection measure is the best?**

Most give good results, none is significantly superior to others.

## Overfitting and tree pruning

### **Overfitting: An induced tree may overfit the training data.**

Too many branches, some may reflect anomalies due to noise or outliers.

Poor accuracy for unseen samples.

### **Two approaches to avoid overfitting:**

#### **Prepruning:**

Halt tree construction early.

Do not split a node, if this would result in the goodness measure falling below a threshold.

Difficult to choose an appropriate threshold.

#### **Postpruning:**

Remove branches from a "fully grown" tree.

Get a sequence of progressively pruned trees.

Use a set of data different from the training data to decide which is the "best pruned tree."

## Enhancements to Basic Decision-Tree Induction

### **Allow for continuous-valued attributes.**

Dynamically define new discrete-valued attributes that partition the values of continuous-valued attributes into a discrete set of intervals.

### **Handle missing attribute values.**

Assign the most common value of the attribute.

Assign probability to each of the possible values.

### **Attribute construction.**

Create new attributes based on existing ones that are sparsely represented.

This reduces fragmentation, repetition, and replication.



## Classification in large databases

**Classification – a classical problem extensively studied by statisticians and machine-learning researchers.**

**Scalability:**

Classifying datasets with millions of examples and hundreds of attributes with reasonable speed.

**Why is decision-tree induction popular?**

Relatively fast learning speed (compared to other classification methods).

Convertible to simple and easy-to-understand classification rules.

Can use SQL queries for accessing databases.

Classification accuracy comparable with other methods.

**RainForest (Gehrke, Ramakrishnan & Ganti, VLDB'98).**

Builds an AVC-list (attribute, value, class label).

## Scalability framework for RainForest

**Separates the scalability aspects from the criteria that determine the quality of the tree.**

**Builds an AVC-list:**

AVC (Attribute, Value, Class\_label).

**AVC-set (of an attribute X):**

Projection of training dataset onto the attribute  $X$  and class label where counts of individual class label are aggregated.

**AVC-group (of a node  $n$ ):**

Set of AVC-sets of all predictor attributes at the node  $n$ .

## RainForest: training set and its AVC-sets

age	income	student	credit_rating	buys_computer
$\leq 30$	high	no	fair	no
$\leq 30$	high	no	excellent	no
31 ... 40	high	no	fair	yes
$> 40$	medium	no	fair	yes
$> 40$	low	yes	fair	yes
$> 40$	low	yes	excellent	no
31 ... 40	low	yes	excellent	yes
$\leq 30$	medium	no	fair	no
$\leq 30$	low	yes	fair	yes
$> 40$	medium	yes	fair	yes
$\leq 30$	medium	yes	excellent	yes
31 ... 40	medium	no	excellent	yes
31 ... 40	high	yes	fair	yes
$> 40$	medium	no	excellent	no

AVC-set on age:

age	yes	no
$\leq 30$	2	3
31 ... 40	4	0
$> 40$	3	2

AVC-set on income:

income	yes	no
high	2	2
medium	4	2
low	3	1

## RainForest: training set and its AVC-sets (II)

age	income	student	credit_rating	buys_computer
$\leq 30$	high	no	fair	no
$\leq 30$	high	no	excellent	no
31 ... 40	high	no	fair	yes
$> 40$	medium	no	fair	yes
$> 40$	low	yes	fair	yes
$> 40$	low	yes	excellent	no
31 ... 40	low	yes	excellent	yes
$\leq 30$	medium	no	fair	no
$\leq 30$	low	yes	fair	yes
$> 40$	medium	yes	fair	yes
$\leq 30$	medium	yes	excellent	yes
31 ... 40	medium	no	excellent	yes
31 ... 40	high	yes	fair	yes
$> 40$	medium	no	excellent	no


AVC-set on student:

student	yes	no
yes	6	1
no	3	4

AVC-set on credit\_rating:

credit_rating	yes	no
fair	6	2
excellent	3	3

Thank you for your attention.  
**Any questions about the sixth chapter?**

Ask them now, or again, drop me a line:  
 `luciano.melodia@fau.de`.