Computer Science 477

Decision Tree Induction

Lecture 5

Overview

- Classification models in the form of decision trees.

- Equivalently, in the form of a set of decision rules

Reminder: what this class is about

- Extracting knowledge, patterns, useful information from large data sets
- Specific techniques:
  - Classification
    - Constructing a method of classifying new instances using information in a training set
  - Clustering: subsetting large datasets into meaningful grouping.
  - Association Analysis: determining whether elements tend to occur together
    - Paradigm: market baskets
      - Are beer and diapers purchased together?
  - Sequence mining
    - Finding meaningful, recurring sequences of events

Example

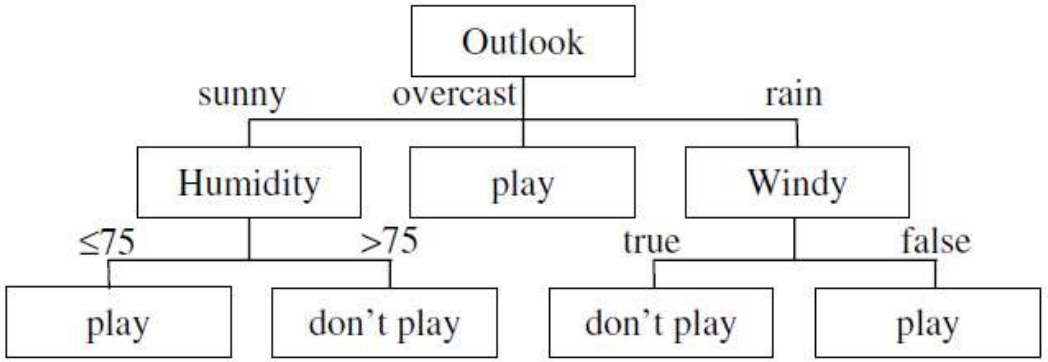
- Class
  - Play/don't play
- Attributes
  - Outlook
  - Temperature
  - Humidity
  - Windy

Outlook	Temp (°F)	Humidity (%)	Windy	Class
sunny	75	70	true	play
sunny	80	90	true	don't play
sunny	85	85	false	don't play
sunny	72	95	false	don't play
sunny	69	70	false	play
overcast	72	90	true	play
overcast	83	78	false	play
overcast	64	65	true	play
overcast	81	75	false	play
rain	71	80	true	don't play
rain	65	70	true	don't play
rain	75	80	false	play
rain	68	80	false	play
rain	70	96	false	play

- If tomorrow the values of *Outlook*, *Temperature*, *Humidity* and *Windy* were sunny, 74◦F, 77% and false respectively, what would the decision be?

Decision Tree for Golf Data

- *Outlook*, *Temperature*, *Humidity* and *Windy* were sunny, 74◦F, 77% and false respectively, what would the decision be?



- If the value of *Outlook* is sunny, next consider the value of *Humidity*.
- If the value is less than or equal to 75 the decision is *play*. Otherwise the decision is *don't play*.
- If the value of *Outlook* is overcast, the decision is *play*.
- If the value of *Outlook* is rain, next consider the value of *Windy*. If the value is true the decision is *don't play*, otherwise the decision is *play*.
- Note that the value of *Temperature* is never used.

Terminology

- Standard data representation: a universe of *objects* (people, houses etc.), each of which can be described by the values of a collection of its *attributes*.
- Attributes with a finite (and generally fairly small) set of values, such as sunny, overcast and rain, are called *categorical*.
- Attributes with numerical values, such as *Temperature* and *Humidity*, are generally known as *continuous*.
- Descriptions of a number of objects are held in tabular form in a *training set*.
  - Each row of the figure comprises an *instance*, i.e. the (non-classifying) attribute values and the classification corresponding to one object.
- The aim is to develop *classification rules* from the data in the training set.

Decision Tree Terminology

- A decision tree is created by a process known as *splitting on the value of attributes* (or just *splitting on attributes*),
  - Testing the value of an attribute such as *Outlook* and then creating a branch for each of its possible values.
- In the case of continuous attributes the test is normally whether the value is ‘less than or equal to’ or ‘greater than’ a given value known as the *split value*
- Splitting process continues until each branch can be labelled with just one classification

Degrees Dataset

- **Classes**
  - FIRST, SECOND
- **Attributes**
  - **SoftEng**
    - A,B
  - **ARIN**
    - A,B
  - **HCI**
    - A,B
  - **CSA**
    - A,B
  - **Project**
    - A,B
- What determines who is classified as FIRST or SECOND?

SoftEng	ARIN	HCI	CSA	Project	Class
A	B	A	B	B	SECOND
A	B	B	B	A	FIRST
A	A	A	B	B	SECOND
B	A	A	B	B	SECOND
A	A	B	B	A	FIRST
B	A	A	B	B	SECOND
A	B	B	B	B	SECOND
A	B	B	B	B	SECOND
A	A	A	A	A	FIRST
B	A	A	B	B	SECOND
B	A	A	B	B	SECOND
A	B	B	A	B	SECOND
B	B	B	B	A	SECOND
A	A	B	A	B	FIRST
B	B	B	B	A	SECOND
A	A	B	B	B	SECOND
B	B	B	B	B	SECOND
A	A	B	A	A	FIRST
B	B	B	A	A	SECOND
B	B	A	A	B	SECOND
B	B	B	B	A	SECOND
B	A	B	A	B	SECOND
A	B	B	B	A	FIRST
A	B	A	B	B	SECOND
B	A	B	B	B	SECOND
A	B	B	B	B	SECOND

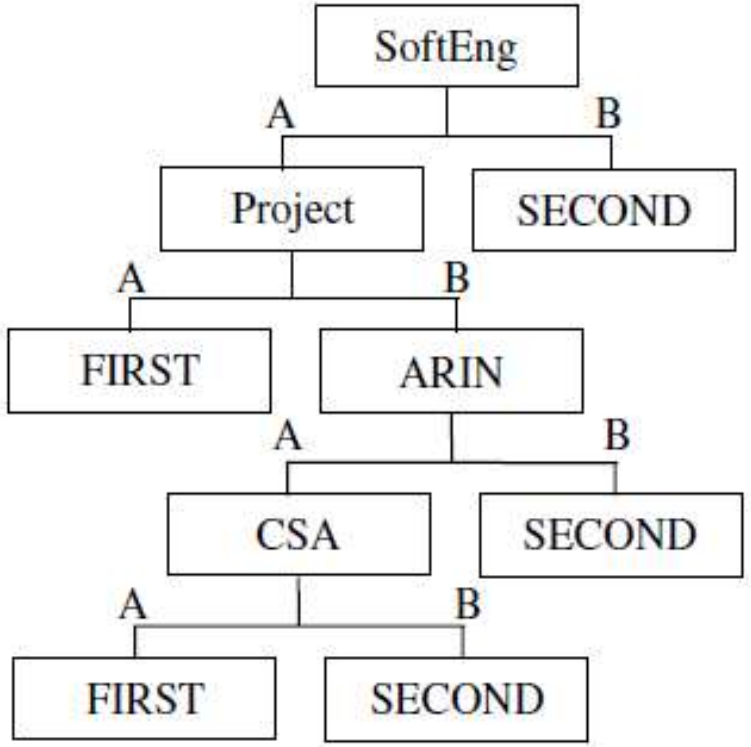
Implied Rules

- IF SoftEng = A AND Project = A THEN Class = FIRST
- IF SoftEng = A AND Project = B AND ARIN = A AND CSA = A THEN Class = FIRST
- IF SoftEng = A AND Project = B AND ARIN = A AND CSA = B THEN Class = SECOND
- IF SoftEng = A AND Project = B AND ARIN = B THEN Class = SECOND
- IF SoftEng = B THEN Class = SECOND

SoftEng	ARIN	HCI	CSA	Project	Class
A	B	A	B	B	SECOND
A	B	B	B	A	FIRST
A	A	A	B	B	SECOND
B	A	A	B	B	SECOND
A	A	B	B	A	FIRST
B	A	A	B	B	SECOND
A	B	B	B	B	SECOND
A	B	B	B	B	SECOND
A	A	A	A	A	FIRST
B	A	A	B	B	SECOND
B	A	A	B	B	SECOND
A	B	B	A	B	SECOND
B	B	B	B	A	SECOND
A	A	B	A	B	FIRST
B	B	B	B	A	SECOND
A	A	B	B	B	SECOND
B	B	B	B	B	SECOND
A	A	B	A	A	FIRST
B	B	B	A	A	SECOND
B	B	A	A	B	SECOND
B	B	B	B	A	SECOND
B	A	B	A	B	SECOND
A	B	B	B	A	FIRST
A	B	A	B	B	SECOND
B	A	B	B	B	SECOND
A	B	B	B	B	SECOND

As a Decision Tree

- IF SoftEng = A AND Project = A THEN Class = FIRST
- IF SoftEng = A AND Project = B AND ARIN = A AND CSA = A THEN Class = FIRST
- IF SoftEng = A AND Project = B AND ARIN = A AND CSA = B THEN Class = SECOND
- IF SoftEng = A AND Project = B AND ARIN = B THEN Class = SECOND
- IF SoftEng = B THEN Class = SECOND



Simplified Rules

- IF SoftEng = A AND Project = A THEN Class = FIRST
- IF SoftEng = A AND Project = B AND ARIN = A AND CSA = A THEN Class = FIRST
- IF SoftEng = A AND Project = B AND ARIN = A AND CSA = B THEN Class = SECOND
- IF SoftEng = A AND Project = B AND ARIN = B THEN Class = SECOND
- IF SoftEng = B THEN Class = SECOND

```
if (SoftEng = A) {
   if (Project = A) Class = FIRST
   else {
      if (ARIN = A) {
         if (CSA = A) Class = FIRST
         else Class = SECOND
       }
      else Class = SECOND
   }
}
else Class = SECOND
```

Top-Down Induction of Decision Trees - TDIDT

- Algorithm known since the mid-1960s
- Formed the basis for many classification systems
  - ID3 and C4.5
- The method produces decision rules in the implicit form of a decision tree.
- Decision trees generated by repeatedly splitting on the values of attributes.
- This process is also known as *recursive partitioning*.

Basic Algorithm - TDIDT

- At each non-leaf node in developing tree an attribute is chosen for splitting.
- Potentially *any* attribute, except that the same attribute must not be chosen twice in the same branch.
- Important condition: ***Adequacy Condition***:
  - No two instances with the same values of all the attributes may belong to different classes
  - I.e., must be consistent.
- Ways of dealing with inconsistent training sets later

TDIDT – Basic Algorithm

- IF all the instances in the training set belong to the same class THEN return the value of the class
- ELSE
  - (a) Select an attribute *A* to split on
  - (b) Sort the instances in the training set into subsets, one for each value of attribute *A*
  - (c) Return a tree with one branch for each *non-empty* subset
    - Each branch having a descendant subtree or a class value produced by applying the algorithm recursively

TDIDT Algorithm

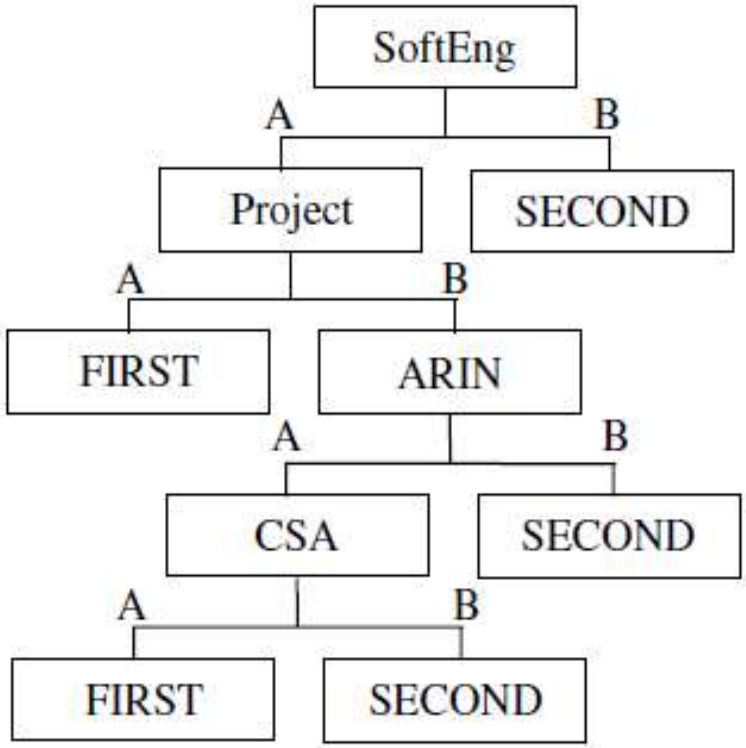
- Until No More Splitting is Possible:

TDIDT: BASIC ALGORITHM

IF all the instances in the training set belong to the same class
THEN return the value of the class
ELSE (a) Select an attribute A to split on^{+}
(b) Sort the instances in the training set into subsets, one for each value of attribute A
(c) Return a tree with one branch for each *non-empty* subset, each branch having a descendant subtree or a class value produced by applying the algorithm recursively

$^{+}$ Never select an attribute twice in the same branch

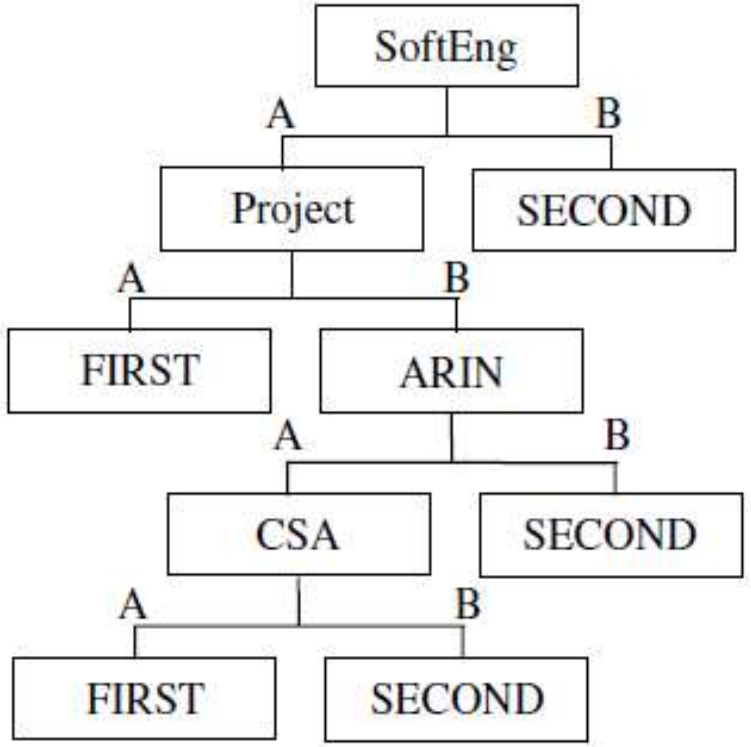
Implied Rules



SoftEng	ARIN	HCI	CSA	Project	Class
A	B	A	B	B	SECOND
A	B	B	B	A	FIRST
A	A	A	B	B	SECOND
B	A	A	B	B	SECOND
A	A	B	B	A	FIRST
B	A	A	B	B	SECOND
A	B	B	B	B	SECOND
A	B	B	B	B	SECOND
A	A	A	A	A	FIRST
B	A	A	B	B	SECOND
B	A	A	B	B	SECOND
A	B	B	A	B	SECOND
B	B	B	B	A	SECOND
A	A	B	A	B	FIRST
B	B	B	B	A	SECOND
A	A	B	B	B	SECOND
B	B	B	B	B	SECOND
A	A	B	A	A	FIRST
B	B	B	A	A	SECOND
B	B	A	A	B	SECOND
B	B	B	B	A	SECOND
B	A	B	A	B	SECOND
A	B	B	B	A	FIRST
A	B	A	B	B	SECOND
B	A	B	B	B	SECOND
A	B	B	B	B	SECOND

As a Decision Tree

- IF SoftEng = A AND Project = A THEN Class = FIRST
- IF SoftEng = A AND Project = B AND ARIN = A AND CSA = A THEN Class = FIRST
- IF SoftEng = A AND Project = B AND ARIN = A AND CSA = B THEN Class = SECOND
- IF SoftEng = A AND Project = B AND ARIN = B THEN Class = SECOND
- IF SoftEng = B THEN Class = SECOND



Train Data Set

day	season	wind	rain	class
weekday	spring	none	none	on time
weekday	winter	none	slight	on time
weekday	winter	none	slight	on time
weekday	winter	high	heavy	late
saturday	summer	normal	none	on time
weekday	autumn	normal	none	very late
holiday	summer	high	slight	on time
sunday	summer	normal	none	on time
weekday	winter	high	heavy	very late
weekday	summer	none	slight	on time
saturday	spring	high	heavy	cancelled
weekday	summer	high	slight	on time
saturday	winter	normal	none	late
weekday	summer	high	none	on time
weekday	winter	normal	heavy	very late
saturday	autumn	high	slight	on time
weekday	autumn	none	heavy	on time
holiday	spring	normal	slight	on time
weekday	spring	normal	none	on time
weekday	spring	normal	slight	on time