Data Mining

1.2 Types of Learning

Fall 2008

Instructor: Dr. Masoud Yaghini

Outline

- Classification learning
- Numeric prediction
- Association learning
- Clustering
- References

Classification Learning

Classification learning

- **Classification learning**:
  - predicting a discrete class
- Classification learning is supervised
  - Process is provided with actual outcome
- Outcome is called the **class** of the example
  - In weather problem the play or don't play judgment
  - In contact lenses problem the lens recommendation

Classification learning

- The success of classification learning
  - Using an independent set of **test data** for which class labels are known but not made available to the machine.
- In practice success is often measured subjectively

Classification learning

- **The ways of representing**:
  - Decision Trees
  - Classification Rules

Decision trees

- "Divide and conquer" approach produces decision tree
- Nodes involve testing a particular attribute
- Usually, attribute value is compared to constant
- Other possibilities:
  - Comparing values of two attributes
  - Using a function of one or more attributes

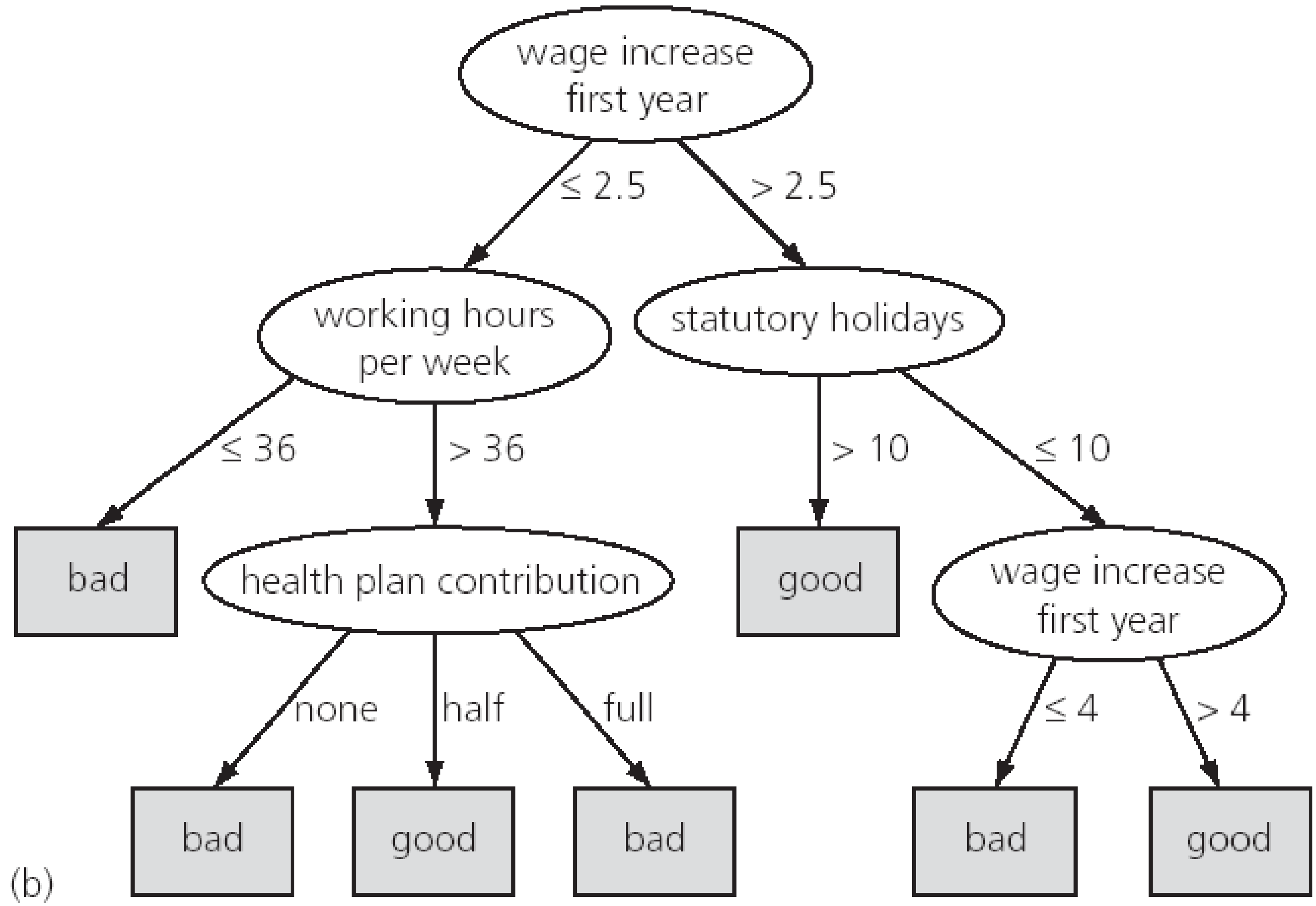
Decision trees

- Leaf nodes
  - give a classification that applies to all instances that reach the leaf
  - set of classifications
  - probability distribution over all possible classifications
- To classify an unknown instance,
  - it is routed down the tree according to the values of the attributes tested in successive nodes, and
  - when a leaf is reached the instance is classified according to the class assigned to the leaf.

Nominal and numeric attributes

- If the attribute is nominal:
  - Number of children usually equal to the number of possible values
  - Usually attribute won't get tested more than once
- If the attribute is numeric:
  - Test whether value is greater or less than constant
  - Attribute may get tested several times
  - Other possibility: three-way split (or multi-way split)
    - Integer: less than, equal to, greater than
    - Real: below, within, above

Decision tree for the labor data



Classification rules

- Popular alternative to decision trees
- Rules include two parts:
  - **Antecedent or precondition**:
    - a series of tests just like the tests at the nodes of a decision tree
    - Tests are usually logically ANDed together
    - All the tests must succeed if the rule is to fire
  - **Consequent or conclusion**:
    - The class or set of classes or probability distribution assigned by rule
- Example: A rule from contact lens problem

```
If tear production rate = reduced then recommendation = none
```

From trees to rules

- Easy: converting a tree into a set of rules
  - One rule for each leaf:
    - Antecedent contains a condition for every node on the path from the root to the leaf
    - Consequent is class assigned by the leaf
- Produces rules that are very clear
  - Doesn't matter in which order they are executed
- But: resulting rules are unnecessarily complex
  - It needs to remove redundant tests/rules

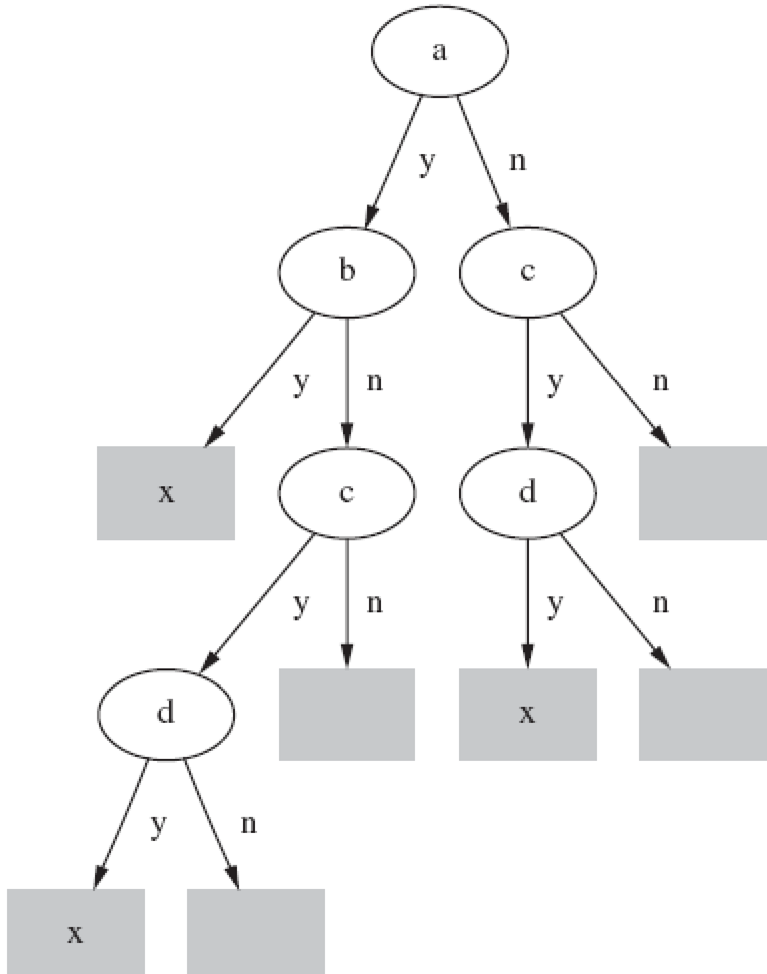
From rules to trees

- More difficult: transforming a rule set into a tree
  - Tree cannot easily express disjunction between rules
- Example: rules which test different attributes

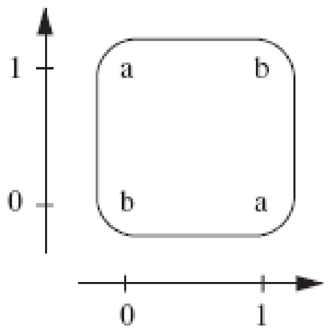
```
If a and b then x
If c and d then x
```

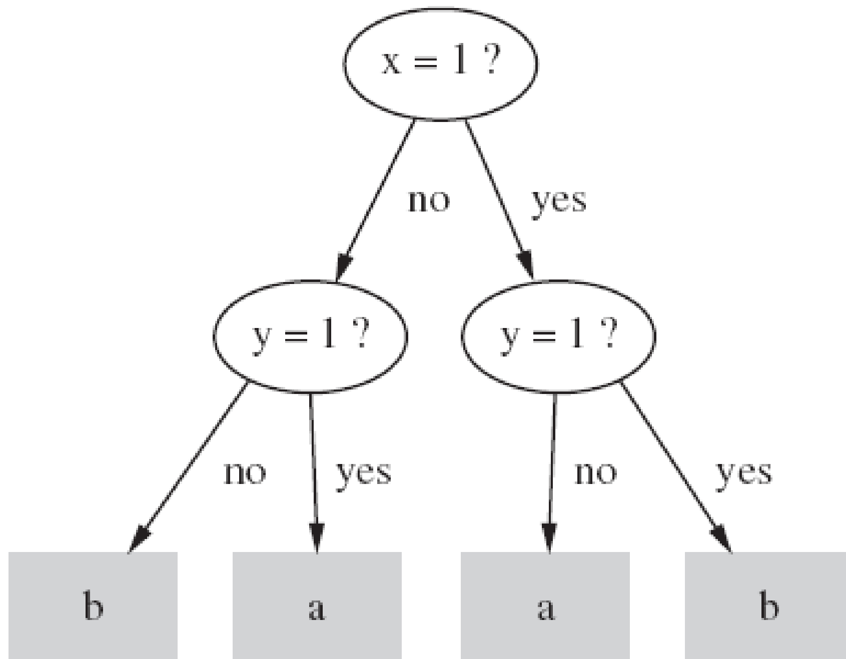
A tree for a simple disjunction

- **Replicated subtree problem:** tree contains identical subtrees



The exclusive-or problem





If x=1 and y=0 then class = a

If x=0 and y=1 then class = a

If x=0 and y=0 then class = b

If x=1 and y=1 then class = b

A tree with a replicated subtree

- If it is possible to have a “default” rule that covers cases not specified by the other rules, rules are much more compact than trees
- There are four attributes, *x, y, z,* and *w,* each can be 1, 2, or 3

If x=1 and y=1 then class = a

If z=1 and w=1 then class = a

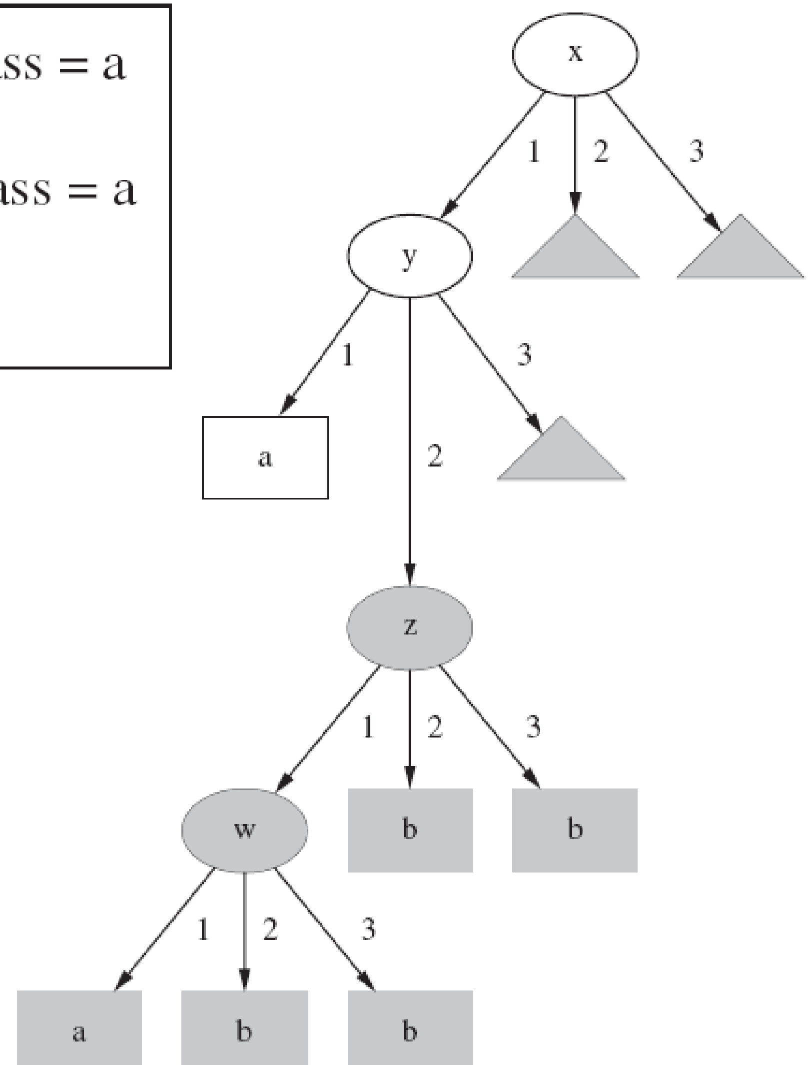
Otherwise class = b

A tree with a replicated subtree

If x=1 and y=1 then class = a

If z=1 and w=1 then class = a

Otherwise class = b



Executing a rule set

- Two ways of executing a rule set:
  - Ordered set of rules (“decision list”)
    - Order is important for interpretation
  - Unordered set of rules
    - Rules may overlap and lead to different conclusions for the same instance

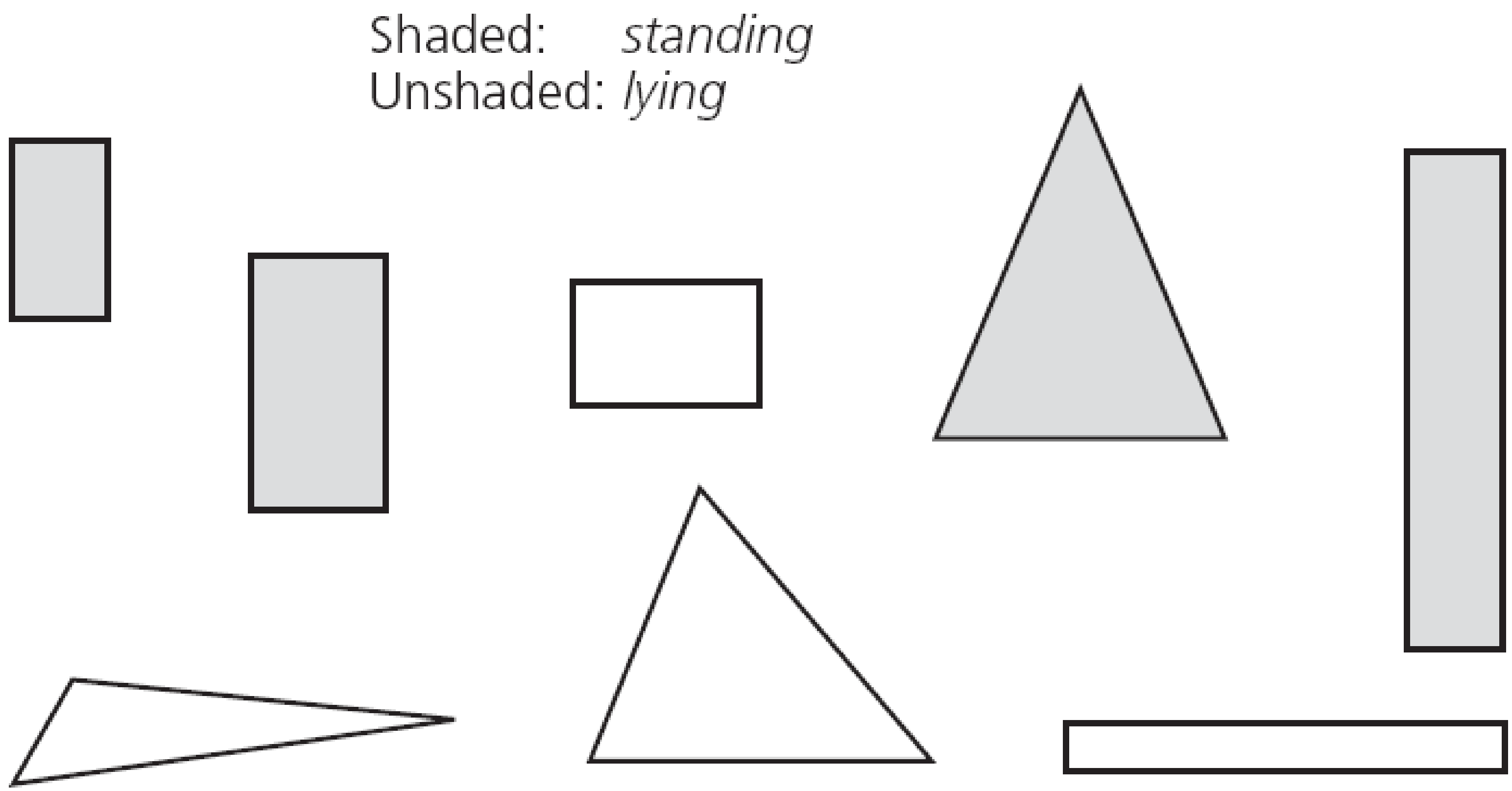
Rules involving relations

- Type of rules:
  - **Propositional**: all rules involved comparing an attribute value to a constant
    - e.g. temperature < 45
  - **Relational**: all rules involved comparing attributes with each other
    - e.g. width < height

Rules involving relations

- Example:
  - we have the set of eight building blocks of the various shapes and sizes
  - There are two classes of **standing** and **lying**.
  - we wish to learn the concept of *standing.*
  - The four shaded blocks are positive (standing) examples of the concept, and the unshaded blocks are negative *(lying) examples.*

The shapes problem



The shapes problem

- Training data for the shapes problem

Width	Height	Sides	Class
2	4	4	standing
3	6	4	standing
4	3	4	lying
7	8	3	standing
7	6	3	lying
2	9	4	standing
9	1	4	lying
10	2	3	lying

The shapes problem

- A propositional solution

```
if width ≥ 3.5 and height < 7.0 then lying
if height ≥ 3.5 then standing
```

- A relational solution

```
if width > height then lying
if height > width then standing
```

Numeric prediction

Numeric prediction

- **Numeric prediction**:
  - predicting a numeric quantity
- Numeric prediction is a variant of classification learning in which the outcome is a numeric value rather than a category.
- Learning is supervised
  - Process is being provided with target value
- Measure success on test data

Numeric prediction

- A version of the weather data in which what is to be predicted is the time (in minutes) to play

Outlook	Temperature	Humidity	Windy	Play time (min.)
sunny	85	85	false	5
sunny	80	90	true	0
overcast	83	86	false	55
rainy	70	96	false	40
rainy	68	80	false	65
rainy	65	70	true	45
overcast	64	65	true	60
sunny	72	95	false	0
sunny	69	70	false	70
rainy	75	80	false	45
sunny	75	70	true	50
overcast	72	90	true	55
overcast	81	75	false	75
rainy	71	91	true	10

Numeric prediction

- To find the important attributes and how they relate to the numeric outcome is more important than predicting value for new instances.

Numeric prediction

- **Representing numeric prediction**:
  - **Linear regression equation**: an equation to predicts a numeric quantity
  - **Regression tree**: a decision tree where each leaf predicts a numeric quantity
    - Predicted value is average value of training instances that reach the leaf
  - **Model tree**: a regression tree with linear regression models at the leaf nodes

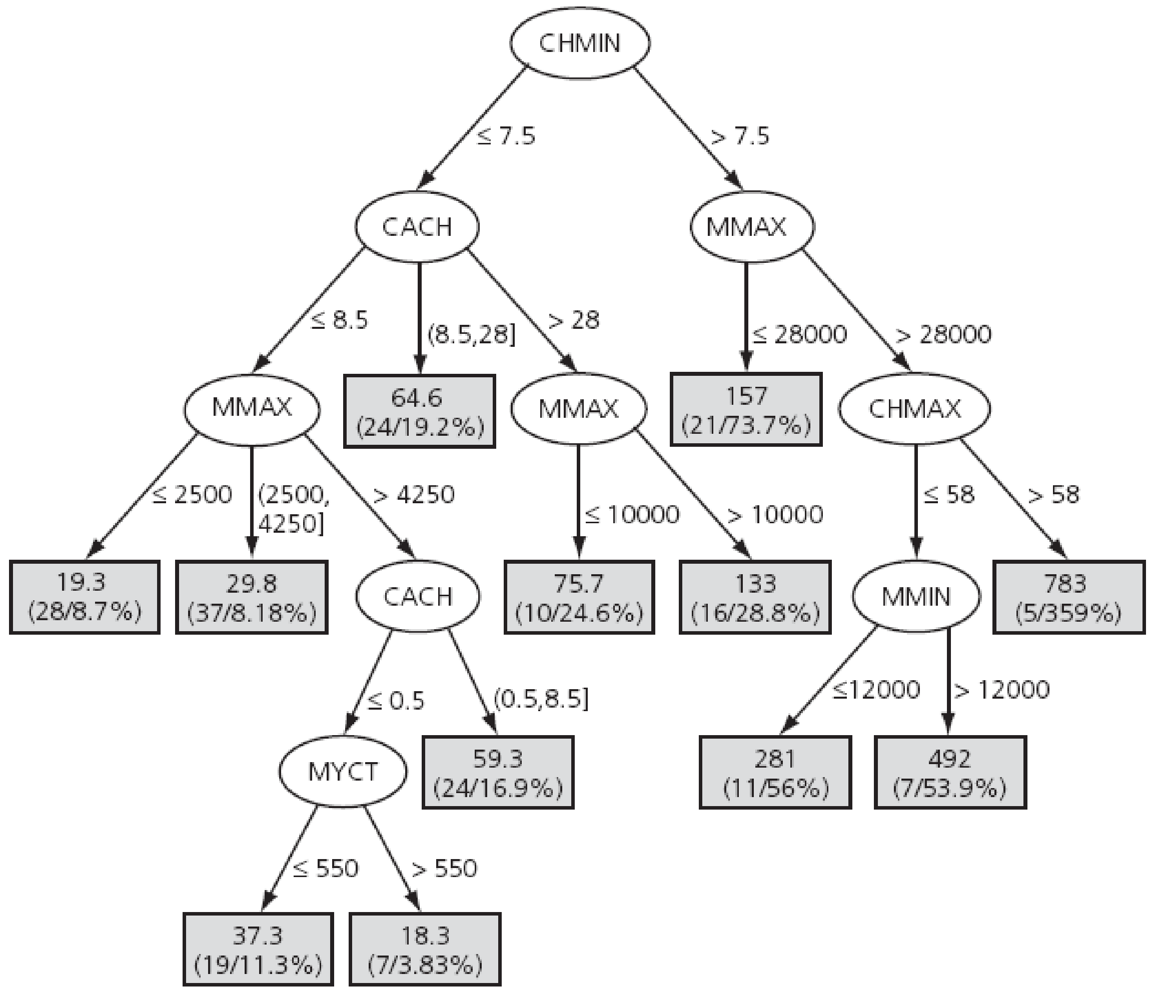
Linear regression equation

- Linear regression equation for CPU data

```
PRP =
  -56.1
  +0.049 MYCT
  +0.015 MMIN
  +0.006 MMAX
  +0.630 CACH
  -0.270 CHMIN
  +1.46 CHMAX
```

Regression tree

- Regression tree for the CPU data

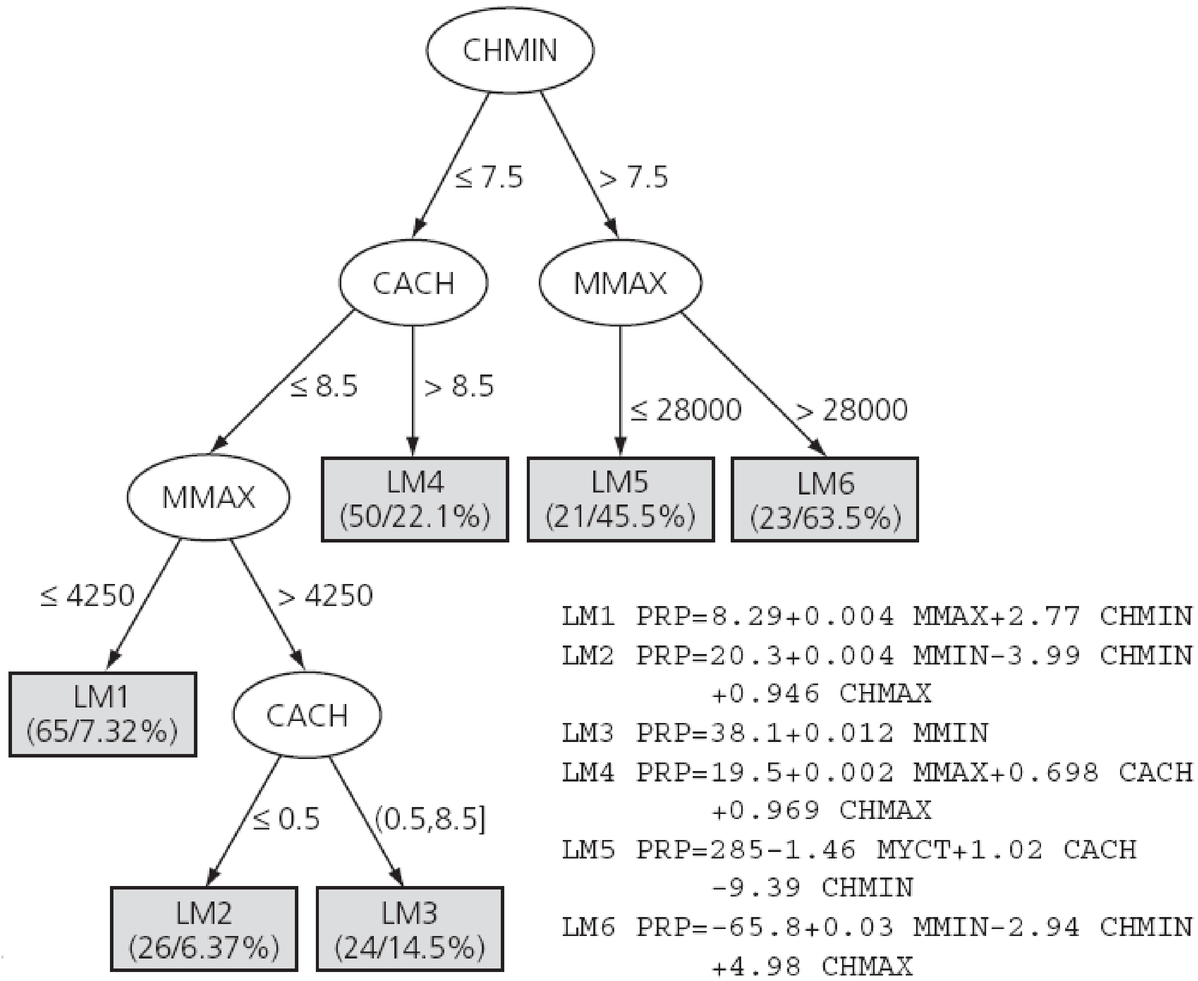


Regression tree

- We calculate the average of the absolute values of the errors between the predicted and the actual CPU performance measures,
- It turns out to be significantly less for the tree than for the regression equation.

Model tree

- Model tree for the CPU data



Association Learning

Association learning

- **Association learning**:
  - detecting associations between attributes
  - can be applied if there is no specified class
- Association learning is unsupervised
- Association rules usually involve only nonnumeric attributes
- Difference to classification learning:
  - Can predict any attribute's value, not just the class
  - More than one attribute's value at a time
  - There are more association rules than classification rules

Association rules

- Large number of possible associations
  - Output needs to be restricted to show only the most predictive associations
  - only those with high **support** and high **confidence**

Support and confidence of a rule

- **Support or coverage**:
  - number of instances predicted correctly
- **Confidence or accuracy**:
  - number of correct predictions, as proportion of all instances that rule applies to
- Example: 4 cool days with normal humidity

```
If temperature = cool then humidity = normal
```

  - Support = 4, confidence = 100%
- Normally: minimum support and confidence prespecified, e.g. support >= 2 and confidence >= 95% for weather data

Interpreting association rules

- Interpretation is not obvious:

```
If windy = false and play = no then outlook = sunny
                                    and humidity = high
```

- is **not** the same as

```
If windy = false and play = no then outlook = sunny
If windy = false and play = no then humidity = high
```

- It means that the following also holds:

```
If humidity = high and windy = false and play = no
   then outlook = sunny
```

Clustering

Clustering

- **Clustering**:
  - grouping similar instances into clusters
- Clustering is unsupervised
  - The class of an example is not known
- Example:
  - a version of the iris data in which the type of iris is omitted
  - Then it is likely that the 150 instances fall into natural clusters corresponding to the three iris types.

Iris data as a clustering problem

	Sepal length (cm)	Sepal width (cm)	Petal length (cm)	Petal width (cm)
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
. . .				
51	7.0	3.2	4.7	1.4
52	6.4	3.2	4.5	1.5
53	6.9	3.1	4.9	1.5
54	5.5	2.3	4.0	1.3
55	6.5	2.8	4.6	1.5
. . .				
101	6.3	3.3	6.0	2.5
102	5.8	2.7	5.1	1.9
103	7.1	3.0	5.9	2.1
104	6.3	2.9	5.6	1.8
105	6.5	3.0	5.8	2.2
. . .				

Clustering

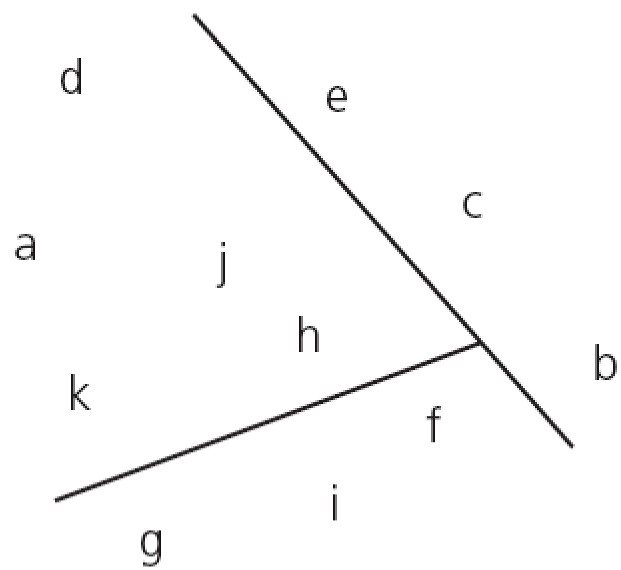
- Clustering may be followed by a second step of classification learning in which rules are learned that give an intelligible description of how new instances should be placed into the clusters.

Representing clusters

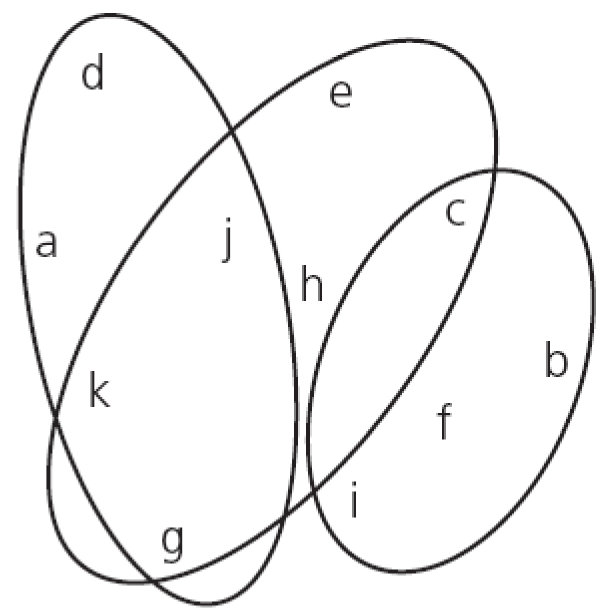
- The output takes the form of a diagram that shows how the instances fall into clusters.
- Different cases:
  - ***Simple 2D representation:*** involves associating a cluster number with each instance
  - ***Venn diagram:*** allow one instance to belong to more than one cluster
  - ***Probabilistic assignment:*** associate instances with clusters probabilistically
  - ***Dendrogram:*** produces a hierarchical structure of clusters (dendron is the Greek word for tree)

Representing clusters

Simple 2D representation



Venn diagram

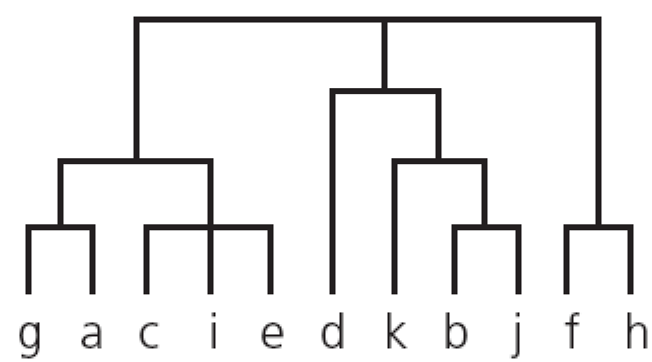


Representing clusters

Probabilistic assignment

	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
d	0.1	0.1	0.8
e	0.4	0.2	0.4
f	0.1	0.4	0.5
g	0.7	0.2	0.1
h	0.5	0.4	0.1

Dendrogram



References

References

- Ian H. Witten and Eibe Frank, **Data Mining: Practical Machine Learning Tools and Techniques**, 2nd Edition, Elsevier Inc., 2005. (Chapter 2 & 3)

The end