Data Mining

Part 1. Introduction

1.2 Data Mining Functionalities

Fall 2009

Instructor: Dr. Masoud Yaghini

Outline

- Introduction
- Mining Frequent Patterns, Associations, and Correlations
- Classification
- Numeric Prediction
- Cluster Analysis
- Interesting Patterns
- References

Introduction

Introduction

- Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks.
- Data mining tasks:
  - **Descriptive data mining**: characterize the general properties of the data in the database.
  - **Predictive data mining**: perform inference on the current data in order to make predictions.

Introduction

- Different views lead to different classifications
  - **Data view**: Kinds of data to be mined
  - **Knowledge view**: Kinds of knowledge to be discovered
  - **Method view**: Kinds of techniques utilized
  - **Application view**: Kinds of applications adapted

Mining Frequent Patterns, Associations, and Correlations

Frequent Patterns

- **Frequent patterns**,
  - are patterns that occur frequently in data.
- The kinds of frequent patterns:
  - frequent itemsets patterns
  - frequent sequential patterns
  - frequent structured patterns

Frequent Patterns

- **A frequent itemset pattern**
  - refers to a set of items that frequently appear together in a transactional data set, such as milk and bread.
- **A frequent sequential pattern**
  - such as the pattern that customers tend to purchase first a PC, followed by a digital camera, and then a memory card, is a (frequent) sequential pattern.
- **A frequent structured pattern**
  - can refer to different structural forms, such as graphs, trees, or lattices, which may be combined with itemsets or subsequences.
  - If a substructure occurs frequently, it is called a (frequent) structured pattern.

Frequent Patterns

- Mining frequent patterns leads to the discovery of interesting **associations** and **correlations** within data.

Association Analysis

- Suppose, as a marketing manager of *AllElectronics*, you would like to determine which items are frequently purchased together within the same transactions.
- An example from the AllElectronics transactional database, is:

$$buys(X, \text{“computer”}) \Rightarrow buys(X, \text{“software”}) \quad [support = 1\%, confidence = 50\%]$$

  - where X is a variable representing a customer.

Association Analysis

- A **confidence**, or **certainty**, of 50% means that if a customer buys a computer, there is a 50% chance that she will buy software as well.
- A **1% support** means that 1% of all of the transactions under analysis showed that computer and software were purchased together.
- This association rule involves **a single attribute** or predicate (i.e., buys) that repeats.
- Association rules that contain a single predicate are referred to as **single-dimensional association rules**.

Association Analysis

- We may find association rules like:

$$age(X, \text{"20...29"}) \wedge income(X, \text{"20K...29K"}) \Rightarrow buys(X, \text{"CD player"})$$

$$[support = 2\%, confidence = 60\%]$$

  - The rule indicates that of the AllElectronics customers under study, 2% are 20 to 29 years of age with an income of 20,000 to 29,000 and have purchased a CD player at *AllElectronics.*
  - There is a 60% probability that a customer in this age and income group will purchase a CD player.
- This is an association between more than one attribute (i.e., age, income, and buys).
- This is **a multidimensional association rule**.

Support and confidence of a rule

- 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
```

Association Analysis

- Large number of possible associations
  - Output needs to be restricted to show only the most predictive associations
- Association rules are interesting if they do satisfy both:
  - **A minimum support threshold:** number of instances predicted correctly
  - **A minimum confidence threshold:** number of correct predictions, as proportion of all instances that rule applies to

Association Analysis

- 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
- Additional analysis can be performed to uncover interesting **statistical correlations** between associated attribute-value pairs.

Classification

Classification

- **Classification**
  - Construct models (functions) that describe and distinguish classes or concepts to predict the class of objects whose **class label** is unknown
- Example:
  - In weather problem the play or don't play judgment
  - In contact lenses problem the lens recommendation
- **Training data set**
  - The derived model is based on the analysis of a **set of training data** (i.e., data objects whose class label is known).

Classification

- 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.
- Classification learning is supervised
  - Process is provided with actual outcome

Classification

- **Examples:**
  - Decision Trees
  - Classification Rules
  - Neural Network
  - Naïve Bayesian classification,
  - Support vector machines
  - k-nearest neighbor classification

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

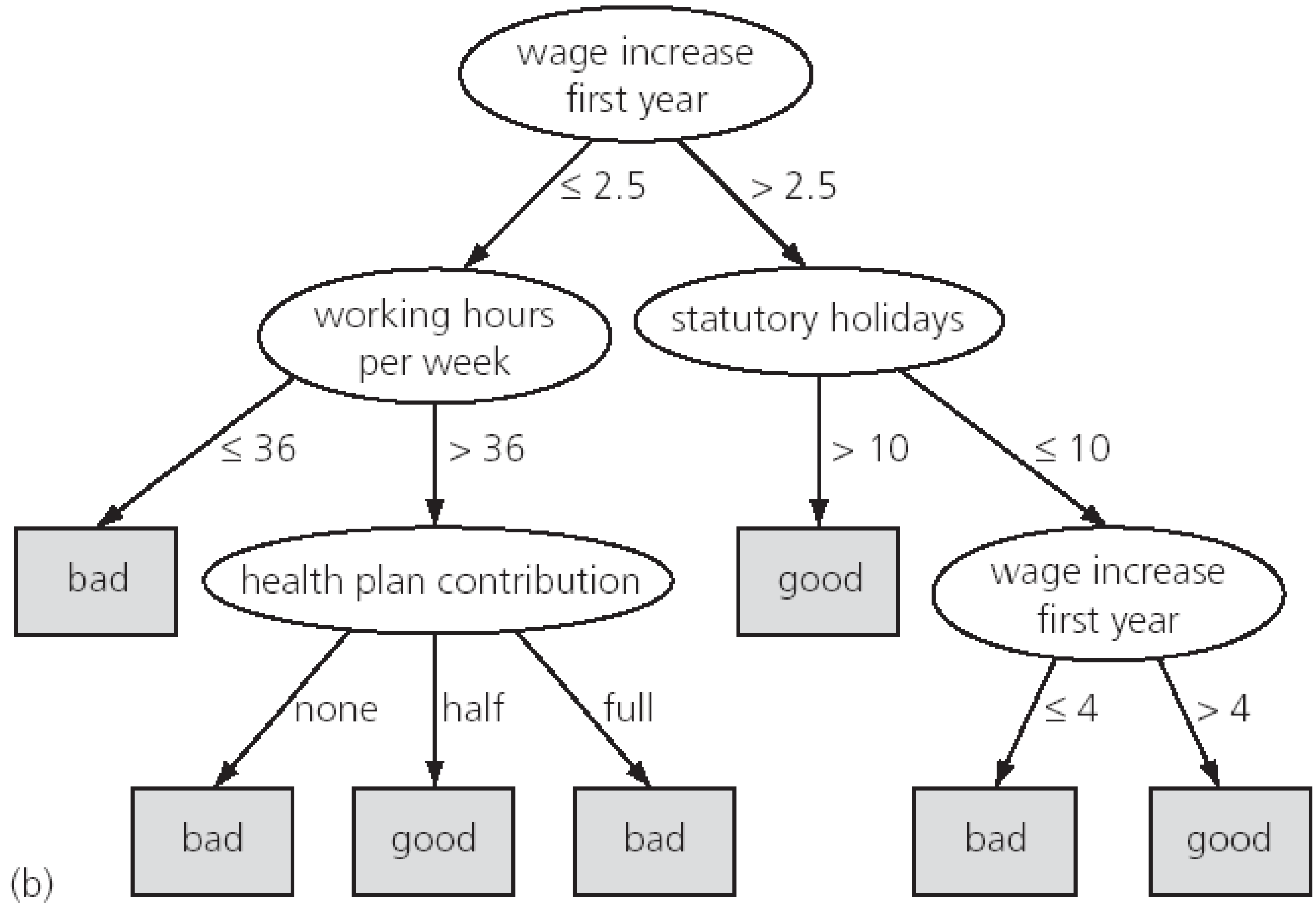
Decision trees

- 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

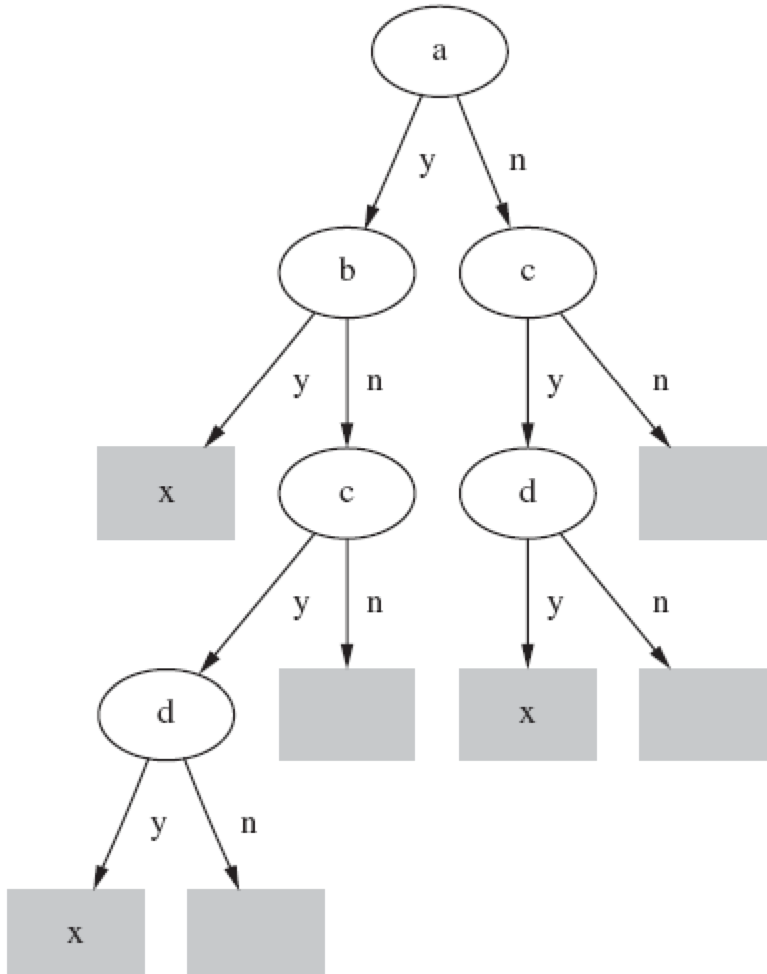
From rules to trees

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

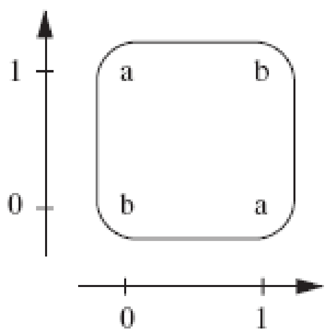
```
If a and b then x
If c and d then x
```

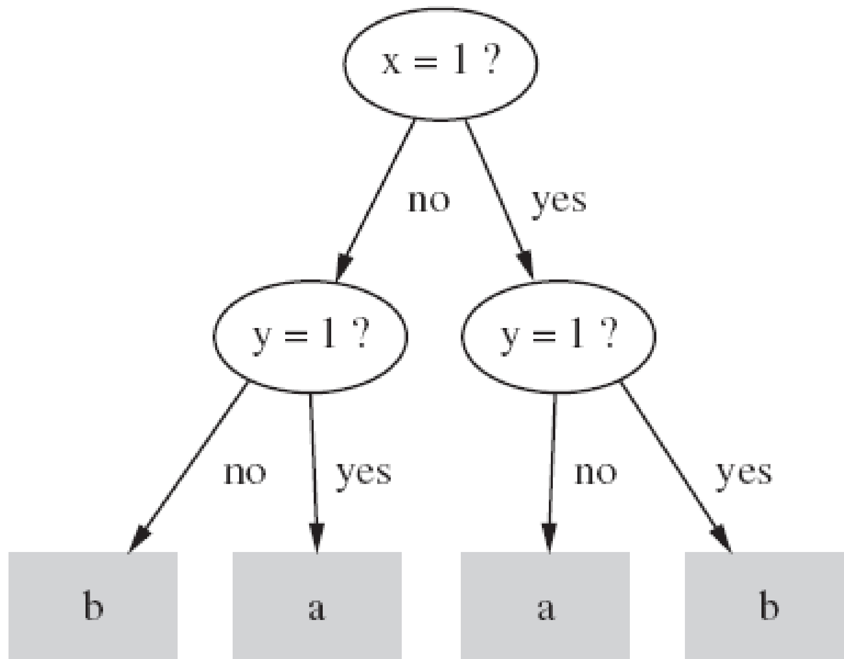
A tree for a simple disjunction

- **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

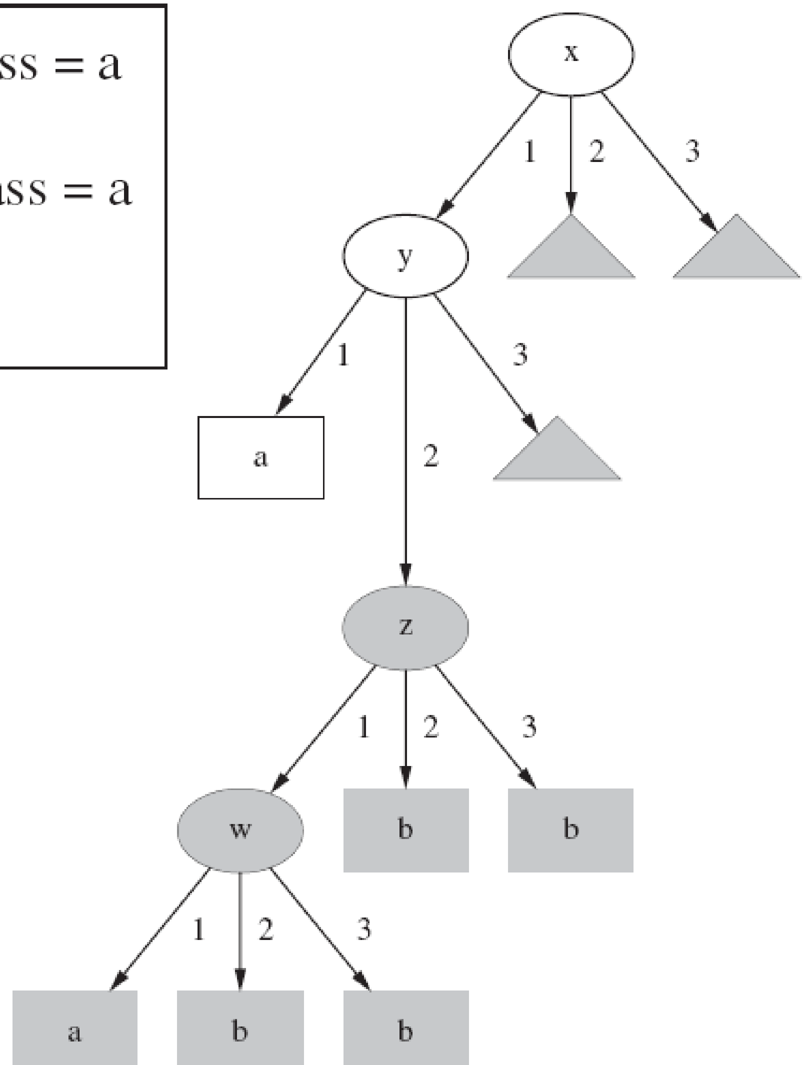
Otherwise class = b

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

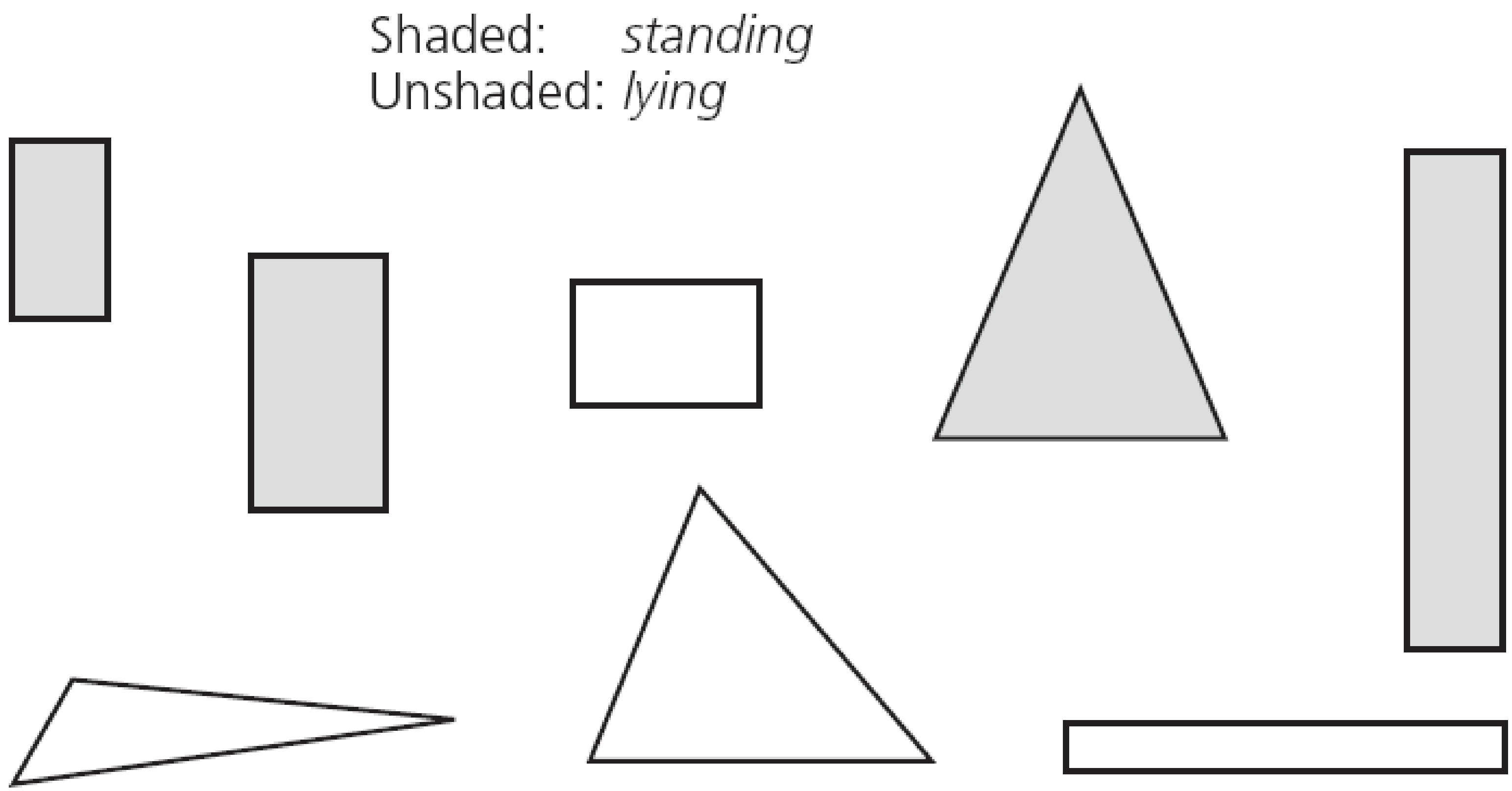
Rules involving relations

- 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

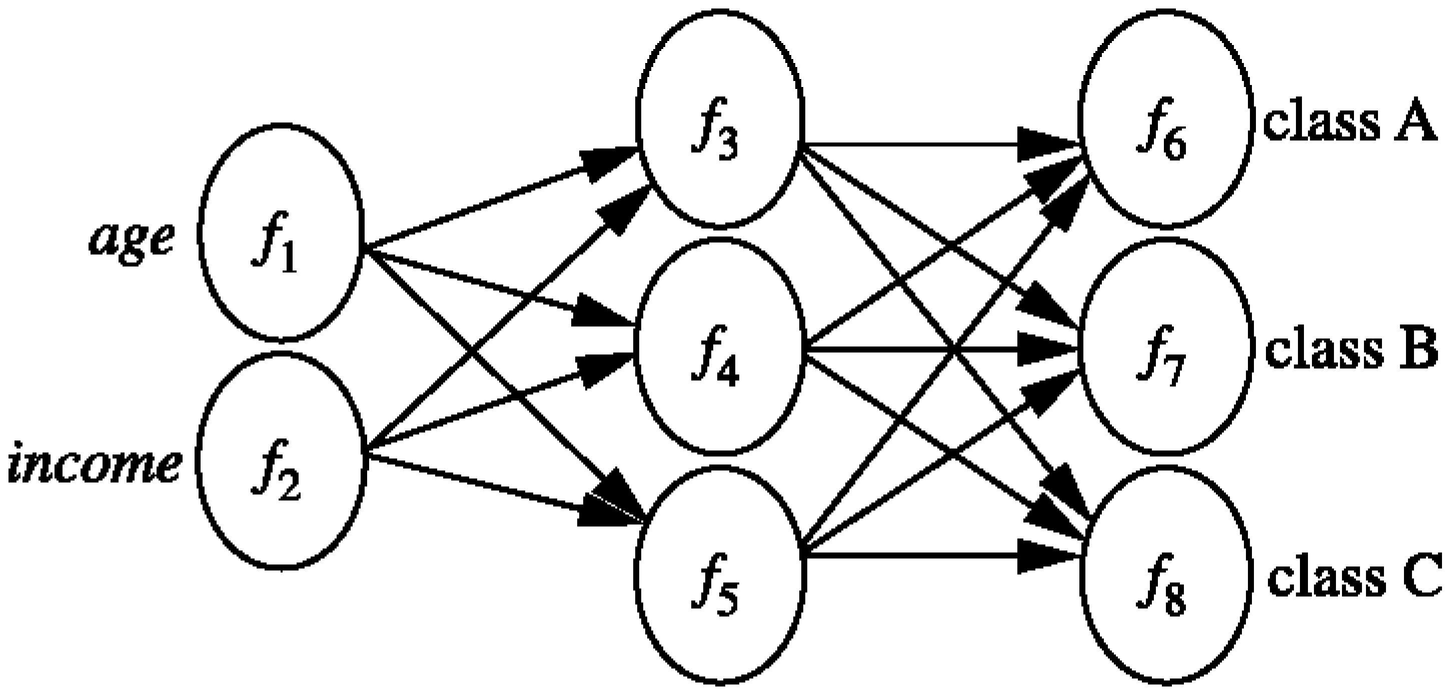
```
if width ≥ 3.5 and height < 7.0 then lying
if height ≥ 3.5 then standing
```

- A relational solution

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

Neural Network

- A neural network, when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units.



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

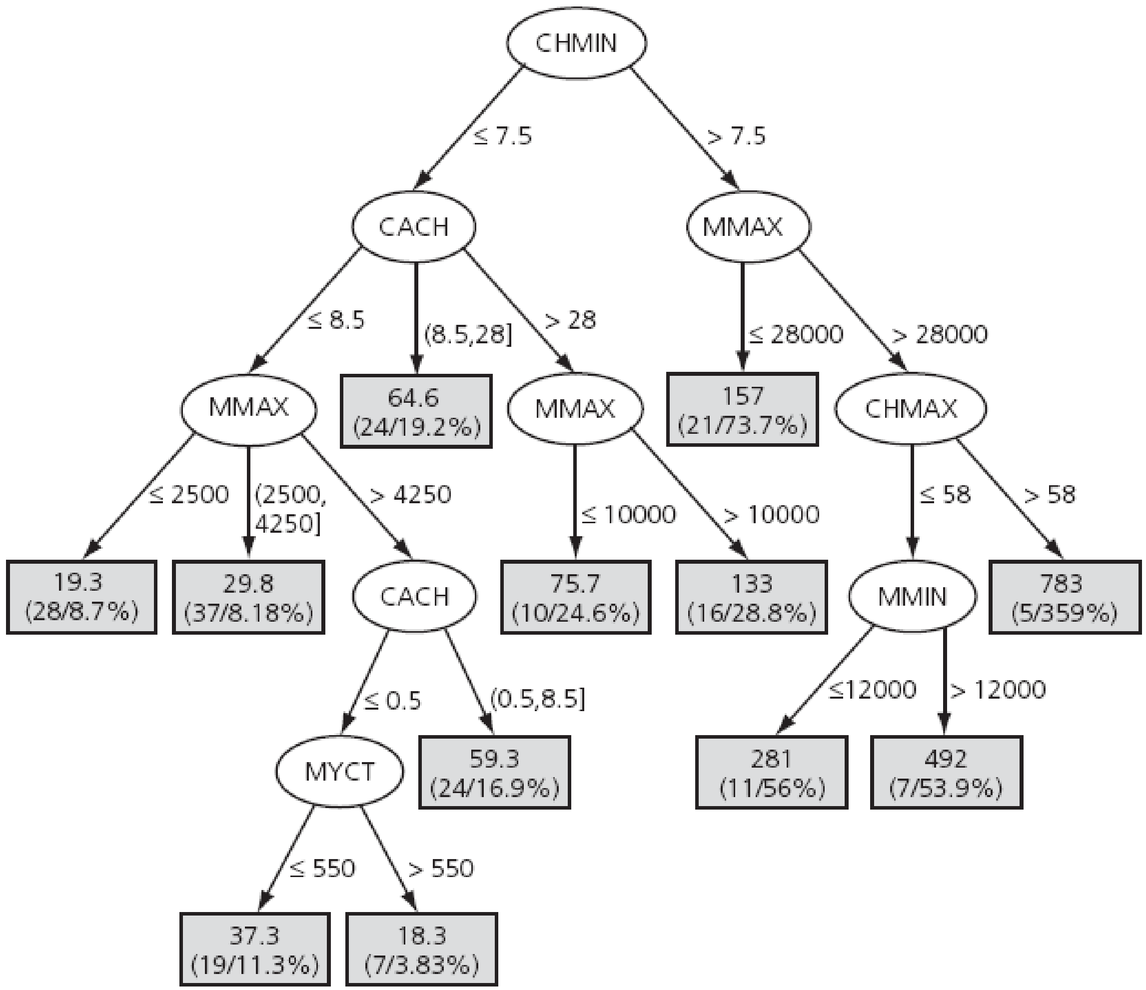
Linear regression equation

- 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

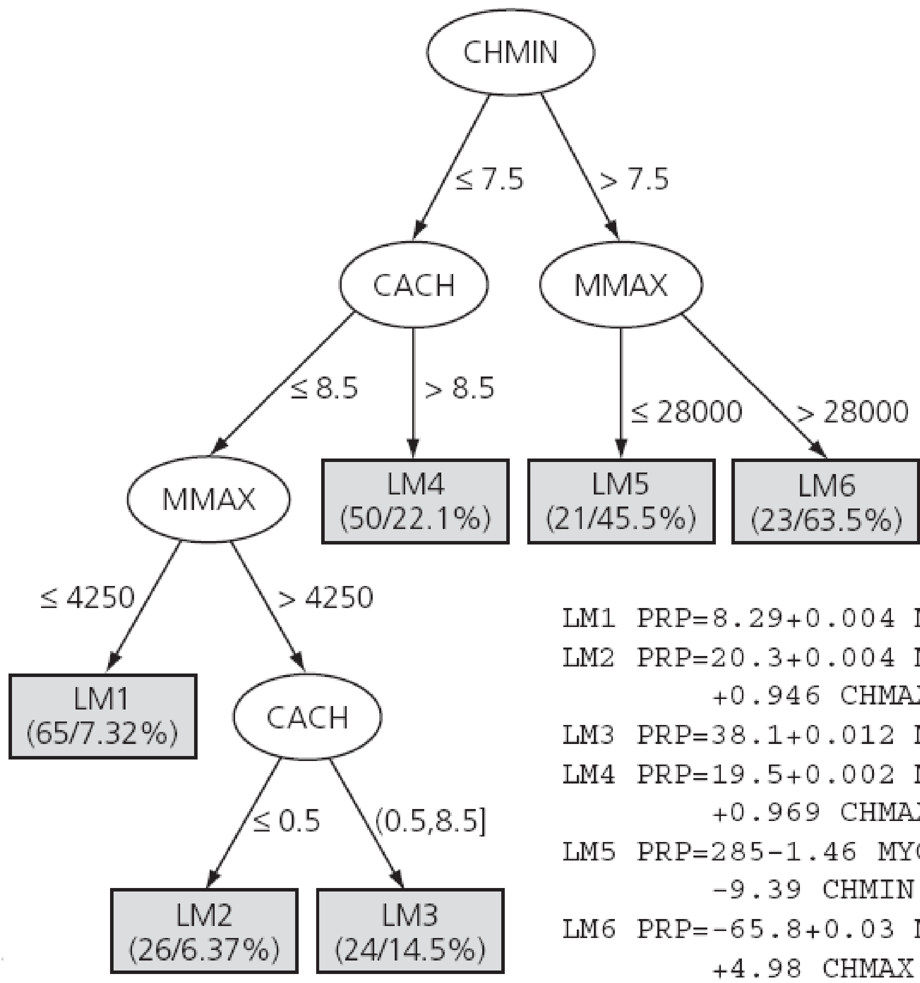


Regression tree

- 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




```
LM1 PRP=8.29+0.004 MMAX+2.77 CHMIN
LM2 PRP=20.3+0.004 MMIN-3.99 CHMIN
        +0.946 CHMAX
LM3 PRP=38.1+0.012 MMIN
LM4 PRP=19.5+0.002 MMAX+0.698 CACH
        +0.969 CHMAX
LM5 PRP=285-1.46 MYCT+1.02 CACH
        -9.39 CHMIN
LM6 PRP=-65.8+0.03 MMIN-2.94 CHMIN
        +4.98 CHMAX
```

Cluster Analysis

Cluster Analysis

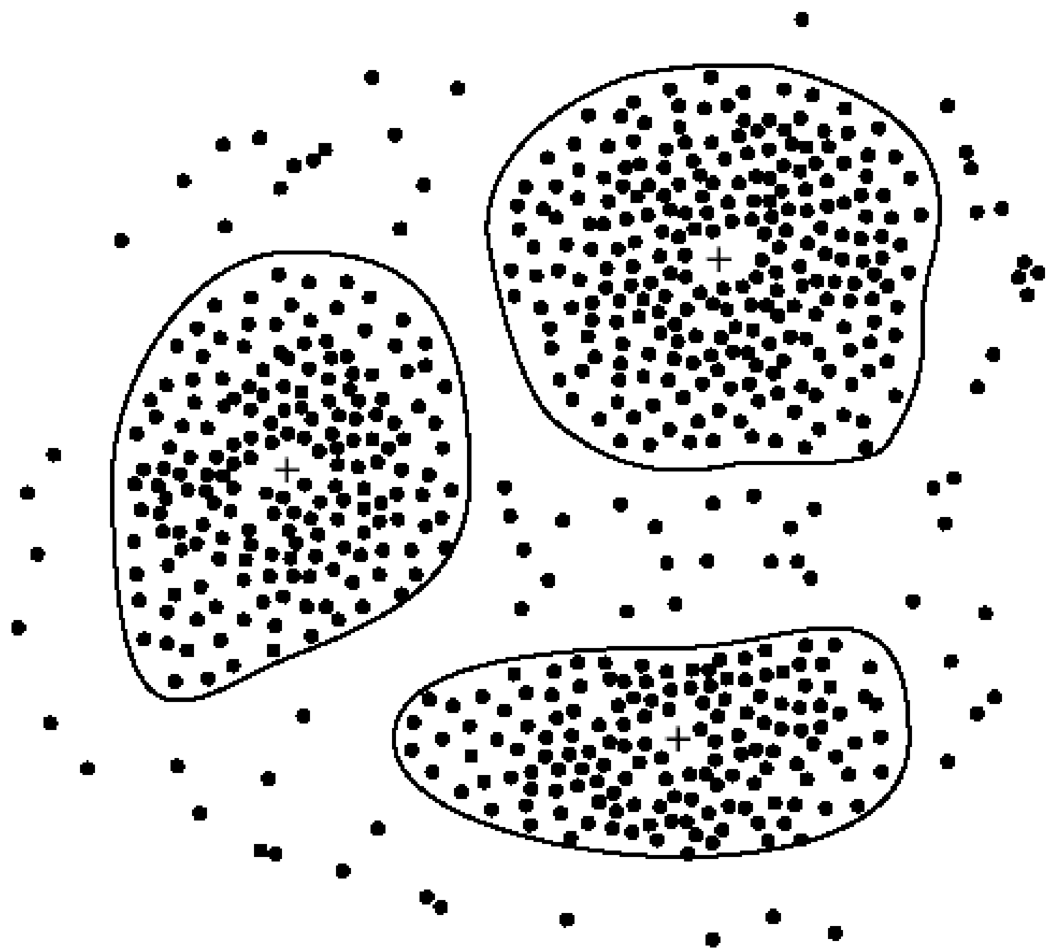
- **Clustering**:
  - grouping similar instances into clusters
- Clustering is unsupervised
  - The class of an example is not known
- Example:
  - a version of the iris data in which the type of iris is omitted
  - Then it is likely that the 150 instances fall into natural clusters corresponding to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

Cluster Analysis

- A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters. Each cluster "center" is marked with a "+".



Clustering

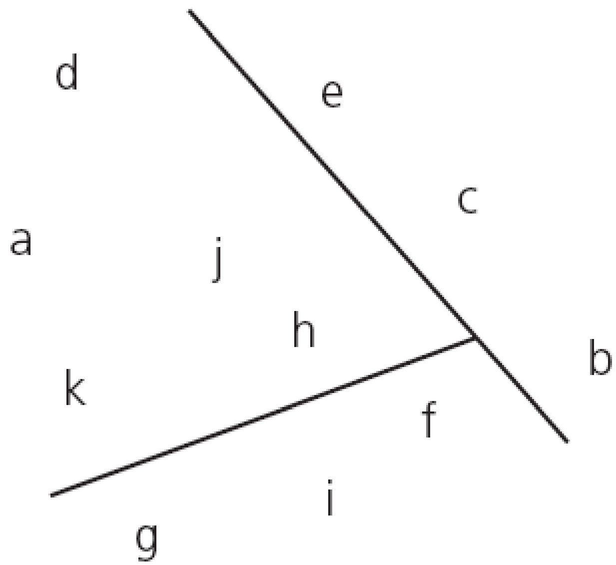
- Clustering may be followed by a second step of classification learning in which rules are learned that give an intelligible description of how new instances should be placed into the clusters.

Representing clusters

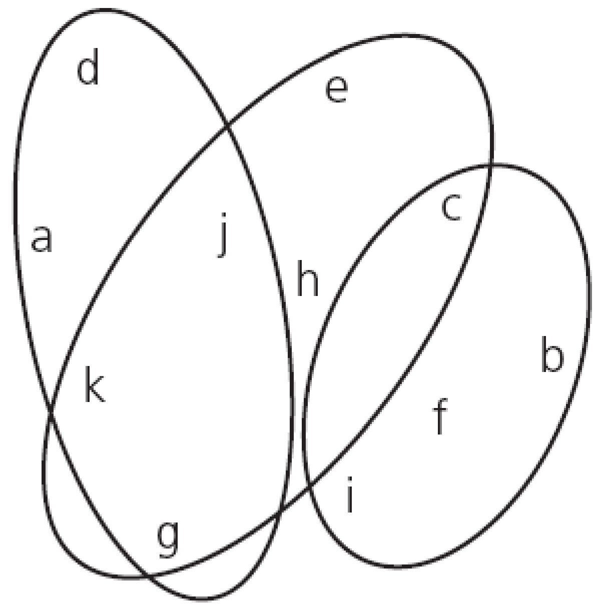
- The output takes the form of a diagram that shows how the instances fall into clusters.
- Different cases:
  - ***Simple 2D representation:*** involves associating a cluster number with each instance
  - ***Venn diagram:*** allow one instance to belong to more than one cluster
  - ***Probabilistic 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

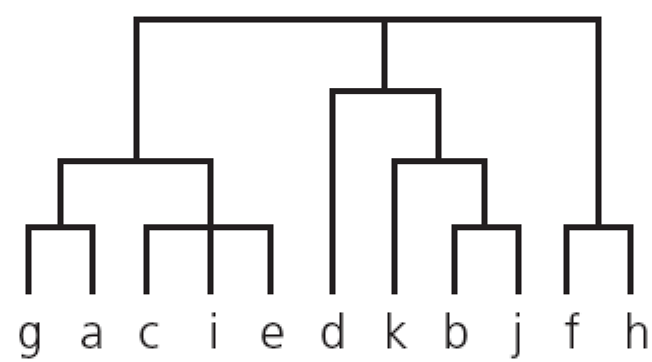


Representing clusters

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



Interesting Patterns

Interesting Patterns

- Data mining may generate thousands of patterns: Not all of them are interesting
- **What makes a pattern interesting?**
  1. Easily understood by humans,
  2. Valid on new or test data with some degree of certainty
  3. Potentially useful
  4. Novel
  5. Validates some hypothesis that a user seeks to confirm

Interesting Patterns

- Objective vs. subjective interestingness measures
  - Objective: based on statistics and structures of patterns, e.g., support, confidence, etc.
  - Subjective: based on user's belief in the data, e.g., unexpectedness, novelty, actionability, etc.

Interesting Patterns

- **Can a data mining system generate all of the interesting patterns?**
  - refers to the **completeness** of a data mining algorithm
  - Do we need to find all of the interesting patterns?
  - **Association rule** mining is an example (where the use of constraints and interestingness measures) can ensure the completeness of mining.
  - Heuristic vs. exhaustive search

Interesting Patterns

- **Can a data mining system generate only interesting patterns?**
  - First general all the patterns and then filter out the uninteresting ones
  - Generate only the interesting patterns

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