Classification

Classical problem in statistics and machine learning.

Separation of objects or ordering of objects into classes. Usually a set of classes are pre-defined. A set of training samples are used for building class models.

Each new object is then assigned to one of the classes using the model.

Typical Applications: credit approval, treatment effectiveness analysis

Apriori Classification

The classification based on training samples with class labels known is called *supervised learning*. New data is classified based on the model developed during training.

Each object has a number of attributes. One special attribute is the class label called the *output* or *dependent* attribute. Value known for the training sample but not for others. Other attributes are *independent* attributes.

Attributes may be *numerical* (e.g. income) or *categorical* (ordered e.g. course grade or unordered e.g. gender, job title, race). The class label is categorical.

Posteriori Classification

No training data with known class labels available. May be classes not known apriori.

Usually clustering is used in such situations although some *unsupervised classification* techniques do exist.

Classification

Classification is a prediction technique. Given the values of the independent attributes the class can be predicted.

Regression is another prediction technique.

Classification is used to predict categorical values while regression is used to predict continuous or ordered values.

25
20+5 – labelled instances
20 – Training - Model
5 – Testing – AV - PV

Two Phases

Model construction phase:

- Each training sample belongs to a predefined class given by the class label attribute. A model is built.

Usage and testing phase:

- Each test object (not used for training) is classified
- Accuracy of the model is estimated
  - If the class label of some test samples is known, it is compared with the result from the model
  - Accuracy rate is the percentage of test samples correctly classified by the model

Quality evaluation

Many quality measures. Han and Kambler list these:

- Predictive accuracy
- Efficiency or speed to construct model and to use
- Robustness in handling noise and missing values
- Scalability
  - efficiency in handling large disk-resident data
- Interpretability:
  - understanding and insight provided by the model
- Goodness of rules (size, compactness of model)

Decision Tree

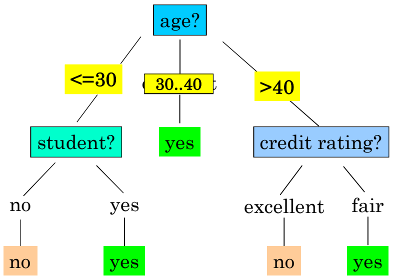
A decision tree is an approach like that needed to support the game of twenty questions where each internal node of the tree denotes a question or a test on the value of an independent attribute, each branch represents an outcome of the test, and each leaf represents a class.

Assume that each object has a number of independent attributes and a dependent attribute.

Example

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
30…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
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

A decision Tree



Decision Tree

To classify an object, the appropriate attribute value is used at each node, starting from the root, to determine the branch taken. The path found by tests at each node leads to a leaf node which is the class the model believes the object belongs to.

Decision tree is an attractive technique since the results are easy to understand. The rules can often be expressed in natural language e.g. *if the student has GPA > 3.0 and class attendance > 90% then the student is likely to get a D.*

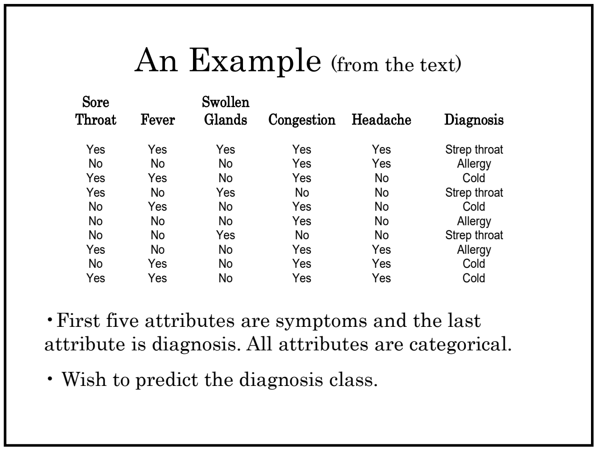
Basic Algorithm

1. The training data is S. Discretise all continuous-valued attributes. Let the root node contain S.
2. If all objects S in the root node belong to the same class then stop.
3. Split the next leaf node by selecting an attribute A from amongst the independent attributes that best divides or splits the objects in the node into subsets and create a decision tree node.
4. Split the node according to the values of A.
5. Stop if any of the following conditions is met otherwise continue with 3.
   -- data in each subset belongs to a single class.
   -- there are no remaining attributes on which the sample may be further divided.

Building A Decision Tree

The aim is to build a decision tree consisting of a *root* node, a number of *internal* nodes, and a number of *leaf* nodes. Building the tree starts with the root node and then splitting the data into two or more children nodes and splitting them in lower level nodes and so on until the process is complete.

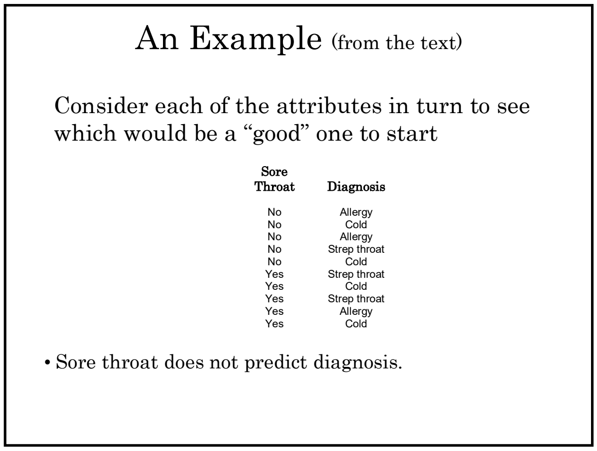
The method uses induction based on the training data. We illustrate it using a simple example.



An Example (from the text)

Sore Throat	Fever	Swollen Glands	Congestion	Headache	Diagnosis
Yes	Yes	Yes	Yes	Yes	Strep throat
No	No	No	Yes	Yes	Allergy
Yes	Yes	No	Yes	No	Cold
Yes	No	Yes	No	No	Strep throat
No	Yes	No	Yes	No	Cold
No	No	No	Yes	No	Allergy
No	No	Yes	No	No	Strep throat
Yes	No	No	Yes	Yes	Allergy
No	Yes	No	Yes	Yes	Cold
Yes	Yes	No	Yes	Yes	Cold

- First five attributes are symptoms and the last attribute is diagnosis. All attributes are categorical.
- Wish to predict the diagnosis class.

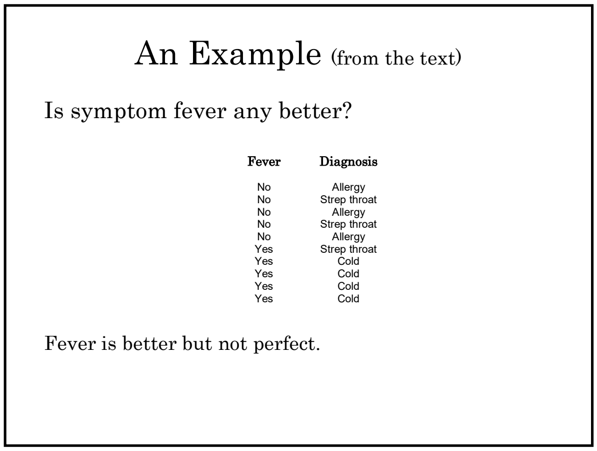


An Example (from the text)

Consider each of the attributes in turn to see which would be a “good” one to start

Sore Throat	Diagnosis
No	Allergy
No	Cold
No	Allergy
No	Strep throat
No	Cold
Yes	Strep throat
Yes	Cold
Yes	Strep throat
Yes	Allergy
Yes	Cold

- Sore throat does not predict diagnosis.

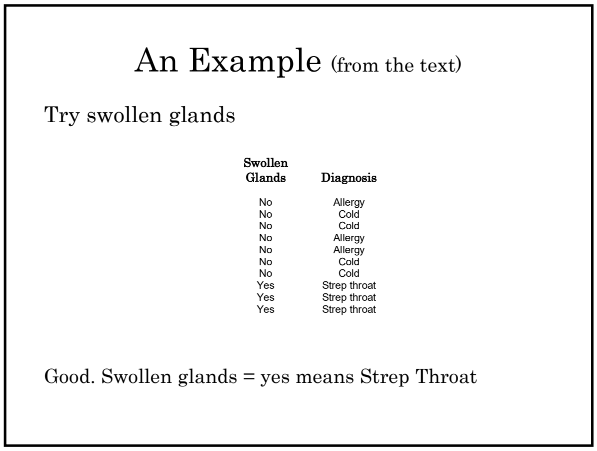


An Example (from the text)

Is symptom fever any better?

Fever	Diagnosis
No	Allergy
No	Strep throat
No	Allergy
No	Strep throat
No	Allergy
Yes	Strep throat
Yes	Cold
Yes	Cold
Yes	Cold
Yes	Cold

Fever is better but not perfect.



An Example (from the text)

Try swollen glands

Swollen Glands	Diagnosis
No	Allergy
No	Cold
No	Cold
No	Allergy
No	Allergy
No	Cold
No	Cold
Yes	Strep throat
Yes	Strep throat
Yes	Strep throat

Good. Swollen glands = yes means Strep Throat

An Example (from the text)

Try congestion

Congestion	Diagnosis
No	Strep throat
No	Strep throat
Yes	Allergy
Yes	Cold
Yes	Cold
Yes	Allergy
Yes	Allergy
Yes	Cold
Yes	Cold
Yes	Strep throat

Not helpful.

An Example (from the text)

Try the symptom headache

Headache	Diagnosis
No	Cold
No	Cold
No	Allergy
No	Strep throat
No	Strep throat
Yes	Allergy
Yes	Allergy
Yes	Cold
Yes	Cold
Yes	Strep throat

Not helpful.

An Example

This approach does not work if there are many attributes and a large training set. Need an algorithm to select an attribute that best discriminates among the target classes as the split attribute.

How do we find the attribute that is most influential in determining the dependent attribute?

The tree continues to grow until it is no longer possible to find better ways to split the objects.

Finding the Split

One approach involves finding the data's diversity (or uncertainty) and choosing a split attribute that minimises diversity amongst the children nodes or maximises the following:

diversity(before split) - diversity(left child) - diversity(right child)

We discuss two approaches. One is based on information theory and the other is based on the work of Gini who devised a measure for the level of income inequality in a country.

Finding the Split

Since our aim is to find nodes that belong to the same class (called *pure*), a term *impurity* is sometime used to measure how far the node is from being pure.

The aim of the split then is to reduce impurity:

impurity(before split) - impurity(left child) - impurity(right child)

Impurity is just a different term. Information theory or the Gini index may be used to find the split attribute that reduces impurity by the largest amount.

Information Theory

value x or value y.

- If s is going to be always x then there is no information and there is no uncertainty.
- What about at p(x) = 0.9 and p(y) = 0.1?
- What about p(x) = 0.5 and p(y) = 0.5?

The measure of information is

Suppose there is a variable s that can take either a

$$I = -\text{sum}\ (p_i \log(p_i))$$

Information

Information is defined as $-p_i * \log(p_i)$ where p_i is the probability of some event.

p_i is always less than 1, so $\log(p_i)$ is always negative and $-p_i * \log(p_i)$ is always positive.

Note that log of 1 is always zero, the log of any number greater than 1 is always positive and the log of any number smaller than 1 is always negative.

Information Theory

$$I = (-0.5 \log(0.5) - 0.5 \log(0.5))$$

This comes out to 1.0 and is the max information for an event with two possible values. Also called entropy. A measure of the minimum number of bits required to encode the information.

Consider a dice with six possible outcomes with equal probability. The information is:

$$I = 6 * (-(1/6) \log(1/6)) = 2.585$$

Therefore three bits are required to represent the outcome of rolling a dice.

Information Theory

Why is information lower if a toss is more likely to get a head than a tail?

If a loaded dice had much more chance of getting a 6, say 50% or even 75%, does the roll of the dice has less or more information?

The information is:
50% I = 5 * (– (0.1) log (0.1)) – 0.5*log(0.5)

75% I = 5 * (– (0.05) log (0.05)) – 0.75*log(0.75)
How many bits are required to represent the outcome of rolling the loaded dice?

Information Gain

- Select the attribute with the highest information gain
- Assume the training data S has two classes, P and N
  - Let S contain a total of s objects, p of class P and n of class N *(so p + n = s)*
  - The amount of information in S given the two class P and N is

$$I(p,n) = -\frac{p}{s}\log_2\frac{p}{s} - \frac{n}{s}\log_2\frac{n}{s}$$

Information Gain

- Assume that using an attribute A the set S is partitioned into $\{S_1, S_2, \ldots, S_v\}$
  - If S_i contains p_i examples of P and n_i examples of N, the entropy, or the expected information needed to classify objects in all subtrees S_i is

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

  - The encoding information that would be gained by branching on A

$$Gain(A) = I(p,n) - E(A)$$

Back to the Example

There are 10 (s = 10) samples and three classes.

Strep throat = t = 3
Cold = c = 4
Allergy = a = 3

Information = – 3/10 log(3/10) – 4/10 log(4/10) – 3/10 log(3/10) = 1.57

Let us now consider using the various symptoms to split the sample

Example

Sore Throat
Yes has t = 2, c = 2, a =1, total 5
No has t = 1, c = 2, a = 2, total 5
I(y) = 2*(–2/5 log(2/5)) – (1/5 log(1/5)) = 1.52
I(n) = 2*(–2/5 log(2/5)) – (1/5 log(1/5)) = 1.52
Information = 0.5*1.52 + 0.5*1.52 = 1.52

Fever
Yes has t = 1, c = 4, a =0, total 5
No has t = 2, c = 0, a = 3, total 5
I(y) = –1/5 log(1/5)) – 4/5 log(4/5)
I(n) = –2/5 log(2/5)) – 3/5 log(3/5)
Information = 0.5 I(y) + 0.5 I(n) = 0.846

Example

Swollen Glands
Yes has t = 3, c = 0, a =0, total 3
No has t = 0, c = 4, a = 3, total 7
I(y) = – 3/3 log(3/3)) = 0
I(n) = – 4/7 log(4/7)) – (3/7 log(3/7))
Information = 0.3*I(y) + 0.7*I(n) = 0.69

Congestion
Yes has t = 1, c = 4, a =3, total 8
No has t = 2, c = 0, a = 0, total 2
I(y) = –1/8 log(1/8) – 4/8 log(4/8) – 3/8 log(3/8)
I(n) = – 2/2 log(2/2)) = 0
Information = 0.8 I(y) + 0.2 I(n) = 1.12

Example

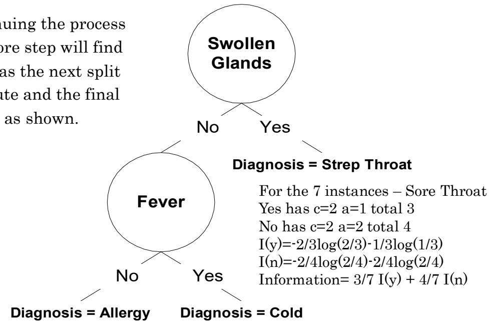
Headache
Yes has t = 2, c = 2, a =1, total 5
No has t = 1, c = 2, a = 2, total 5
I(y) = 2*(–2/5 log(2/5)) – (1/5 log(1/5)) = 1.52
I(n) = 2*(–2/5 log(2/5)) – (1/5 log(1/5)) = 1.52
Information = 0.5*1.52 + 0.5*1.52 = 1.52

So the values for information are:

Sore Throat	1.52
Fever	0.85
Swollen Glands	0.69
Congestion	1.12
Headache	1.52

Decision Tree

Continuing the process one more step will find Fever as the next split attribute and the final Result as shown.



For the 7 instances – Sore Throat
Yes has c=2 a=1 total 3
No has c=2 a=2 total 4
I(y)=-2/3log(2/3)-1/3log(1/3)
I(n)=-2/4log(2/4)-2/4log(2/4)
Information= 3/7 I(y) + 4/7 I(n)