Overview of Machine Learning
  Rote Learning
  Supervised Learning
  Reinforcement Learning
  Unsupervised Learning

In-depth case study on Decision Tree Learning
  What is a decision tree?
  What kinds of problem are suitable for Decision Tree Learning?
  What is a parsimonious solution?
  Entropy and the ID3 algorithm
Different kinds of learning

Rote Learning: the new knowledge is implanted directly with no inference at all e.g. simple memorisation of past events, or a knowledge engineer’s direct programming of rules elicited from a human expert into an expert system

Supervised Learning: the system is supplied with a set of training examples consisting of inputs and corresponding outputs, and is required to discover the relation or mapping between them e.g. as a series of rules, or a neural network

Reinforcement Learning: the system takes actions (without initially knowing the utility of those actions), and then after the actions are taken receives feedback in the form of a reward that enables the system to infer utility values for available actions in each state of the world

Unsupervised Learning: the system is supplied with a set of training examples consisting or inputs and is required to discover for itself what appropriate outputs should be e.g. a Kohonen Network or Self Organising Map
More on rote learning

- A rote learning system does not need to do any processing to understand or interpret the information supplied by the environment. All it must do is memorise the incoming information for later use. Virtually every computer system can be said to do rote learning in so far as it stores instructions for performing task. Rote learning can be very simple. It may not appear to involve any sophisticated problem-solving capabilities. But even this simple form of learning demonstrates the need for some capabilities that will become increasingly important in more complex learning systems.

- These capabilities include, organised storage of information in order for it to be faster to use a stored value. The number of distinct objects that might potentially be stored can be very large. To keep the number of stored object down to manageable lever, some kind of generalisation may be necessary.
More on supervised learning

Supervised learning often occurs in batches, for example, a neural network that is attempting to learn a pattern (perhaps in images) will be given as input a series of images, where some include the learning target and some do not.

When the Neural Network has been shown each positive or negative example once, then all the examples may be shown again in randomised order.
More on reinforcement learning

Reinforcement learning is often used where the human programmer does not know the correct actions to take,

Robotics and games, such as Backgammon, are well known examples of reinforcement learning

Reinforcement learning is often implemented in actor-critic systems

TD-gammon is a world beating backgammon program that learnt by reinforcement learning
More on unsupervised learning

Kohonen networks learn input output mappings without a supervision or reinforcement signal

Unsupervised learning often involves clustering
What is a Decision Tree?

Classifying items within a taxonomy (like you did in GCSE science classes with living creatures)
Multivariate versus univariate testing

How many features of an organism do you need to test to make classifications (how many kinds of question).

In this biological taxonomy, testing is multivariate (and we have multiple outcomes for each test)

Is it a living organism

- Is it a plant
- Is it a fungus
- Is it an animal
- Is it a prototist
- Is it a bacteria

- Is it an invertebrate
  - Is it a vertebrate
    - Is it a homo sapien
    - Is it an arthropod
    - Is it a mollusc
    - Is it a nematode
Multiple outcomes versus binary trees

Is it a living organism
- Is it a plant
- Is it a fungus
- Is it an animal
- Is it a prototist
- Is it a bacteria
  - Is it a vertebrate
    - Is it an invertebrate
      - Is it a Homo sapien
      - Is it an arthropod
      - Is it a mollusc
      - Is it a nematode
Problems that Decision Tree learning is a good choice of solution for:

Instances are represented by attribute value pairs (i.e. attributes might be images in the form of bit maps and values might be is there a tank in the image (or not))

The target function has discrete output values (i.e. booleans such as there is, or is not, a tank, rather than there is a 10% probability of there being a tank)

Disjunctive descriptions (energy use goes up when it is very cold and very hot)

The training data may contain errors

The training data may contain missing attribute values
What is decision tree learning?

How to construct a decision tree? Which node to choose as the first node?

<table>
<thead>
<tr>
<th>day</th>
<th>outlook</th>
<th>temperature</th>
<th>humidity</th>
<th>wind</th>
<th>play tennis?</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>weak</td>
<td>no</td>
</tr>
<tr>
<td>d2</td>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>strong</td>
<td>no</td>
</tr>
<tr>
<td>d3</td>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>d4</td>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>d5</td>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>d6</td>
<td>rain</td>
<td>cool</td>
<td>normal</td>
<td>strong</td>
<td>no</td>
</tr>
<tr>
<td>d7</td>
<td>overcast</td>
<td>cool</td>
<td>normal</td>
<td>strong</td>
<td>yes</td>
</tr>
<tr>
<td>d8</td>
<td>sunny</td>
<td>mild</td>
<td>high</td>
<td>weak</td>
<td>no</td>
</tr>
<tr>
<td>d9</td>
<td>sunny</td>
<td>cool</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>d10</td>
<td>rain</td>
<td>mild</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>d11</td>
<td>sunny</td>
<td>mild</td>
<td>normal</td>
<td>strong</td>
<td>yes</td>
</tr>
<tr>
<td>d12</td>
<td>overcast</td>
<td>mild</td>
<td>high</td>
<td>strong</td>
<td>yes</td>
</tr>
<tr>
<td>d13</td>
<td>overcast</td>
<td>hot</td>
<td>normal</td>
<td>weak</td>
<td>yes</td>
</tr>
<tr>
<td>d14</td>
<td>rain</td>
<td>mild</td>
<td>high</td>
<td>strong</td>
<td>no</td>
</tr>
</tbody>
</table>
What is decision tree learning?

How to construct a decision tree?

Which node to choose as the first node?

the ID3 algorithm - choosing which node to construct according to how much information gain it allows

entropy = - $p_1 \log_2 p_2$

Occams razor - why choose the most parsimonious solution?

(Incorrect simple solutions may be less likely than incorrect complicated solutions.)
Conclusion

Overview of Machine Learning
  Rote Learning
  Supervised Learning
  Reinforcement Learning
  Unsupervised Learning

In-depth case study on Decision Tree Learning
  What is a decision tree?
  What kinds of problem are suitable for Decision Tree Learning?
  What is a parsimonious solution?
  Entropy and the ID3 algorithm