Introduction

Machine Learning is at the heart of some of today's most exciting developments in computer science. Some examples of machine learning applications include

- Internet search engine (intelligent information retrieval)
- Handwritten character recognition
- Speech recognition
- Object/Face/Fingerprint recognition (computer vision)

Machine Learning is currently a hot research topic and many new techniques are emerging very rapidly.

1. Definition of Machine Learning

Mitchell used the following to define Machine Learning

The field of Machine Learning seeks to answer the question "How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?" [http://www.cs.cmu.edu/~tom/pubs/MachineLearning.pdf]

From this we can divide Machine Learning into two parts. One is the building of adaptive computer systems that automatically improve with experience and the other is the understanding of the laws that govern learning processes/systems.

The emphasis of the course is on the first – how to build computer systems that automatically improve with experiences – we will learn this through several exemplars.

2. A Formal Definition of Learning

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. (Mitchell, Machine Learning, McGraw-Hill, 1997)

Handwriting recognition problem (also see class slide)

- Task ***T***: Recognizing hand written characters
- Performance measure ***P***: percent of characters correctly classified
- Training experience ***E***: a database of handwritten characters with given classifications

3. The Procedures of Building a Learning System

Step 1: Collect training examples (experience)

Step2: Representing experience (sometimes called data processing, or feature extraction). Normally we use a pair of vectors E = (X, D), where X represents the sensory measurements of the objects or events the system tries to recognize or understand, D is the known classification or identity of X. Often, X is called a feature vector, or an input vector; D is called desired output or class identity of X.

The shape and form of X and D will be problem dependant and it requires domain knowledge to design effective and efficient representation of these. As a principle, we want X to be compact, i.e., we want to use as small set of numbers as possible and at the same time we want X to be discriminative, i.e., we want X's belong to the same object to be the same or similar and X's belong to different objects different (sounds obvious but in practice it is not always easy to achieve!).

Step 3 Choose a learning paradigm. Over the years, many different machine learning models have been developed. Some are simple, some are very complex, some are effective for one type of problems and others are effective for another type of problems. We can have different models capable of solving the same problem. Exactly which model is best for what application is not so straight forward, often this has to be determined through trial and error, or through empirical testing.

In this course, we will first introduce some very simple models, and then introduce more complex ones. We will study models based on statistical theories and we will study models based on optimizing a suitably defined objective functions.

Step 4 Learning. Depending on the models chosen in Step 3, model specific learning algorithms will now be applied to build or learn a system.

The process can be thought of as follows: At the beginning, the model is ignorant – it will get things wrong, then as learning progress, as the system is exposed to the experience (training samples) more and more, the system will get better at solving the problem (think the process of how you would learn to do something).

The learning algorithm is a formal mathematical procedure the chosen model uses to adjust the system's adjustable parameters so that as learning progresses, the system will get better and better at solving the problem – i.e., the system *improves with experience*!

Learning will stop when learning has reached its limit – further process will not improve the performance anymore. After learning stops, all adjustable parameters will be fixed at their final values and the system is ready to be used in real world application.

Step 5 Once the learning is completed, the model can now be used. For an unknown input X, we present it to the model and follow the rules of the model to produce an output which will be taken as the identity and an interpretation of X.

4. Feature Vector and Feature Space

An object or event we are trying to recognize or process have to be represented in some way, normally by a set of numbers which together will form a vector – we call this the feature vector. This can be the pixels to represent a handwritten character when it is scanned in image form, or the recording of speech signals for a speech recognition system, or a histogram of words in a document for a document retrieval application. In general we say X is an n-dimensional feature vector.

Normally, X lies in a Euclidean space and corresponds to a point in the n-dimensional space. If $n = 2$, then X is a point on a plan, if $n = 3$, then X is a point in a 3 dimensional space. For $n > 3$, we cannot visualize the point, but we can imagine X being a point in a n-dimensional hyperspace.

By imagining X as a point in the n-dimensional feature space is sometime helpful for intuitively understanding some of the problems we try to solve and some of the algorithms we will be using. In particular, imagining a given object/event as a point in the n-dimensional feature space introduces the notions of neighbours and neighbourhood and similarity.