1 Introduction to Machine Learning

Contents

Definition 1.1 (Machine Learning (Arthur Samuel, 1959))**.** "Field of study that gives computer the ability to learn **without** being explicitly programmed."

Remark. Samuel wrote a computer program that played 10,000 games of checkers against itself - one of the world's first self-learning program.

1.1 Machine learning algorithms

1.1.1 Supervised Learning

Definition 1.2. In **supervised learning** we teach the computer how to solve some problem, and then let it use its new found knowledge to solve similar problems.

Example. ***Regression vs. classification***

Figure 1: TOP: ***regression*** *problem (****continuous valued*** ***output****)*: What is the best predictor of house price - straight or quadratic?
BOTTOM: ***classification*** *problem (****discrete valued*** ***output****)*: What is the likelihood of my friend with a tumour of size x having a malignant or benign cancer?

Example. *Classification under multiple attributes*

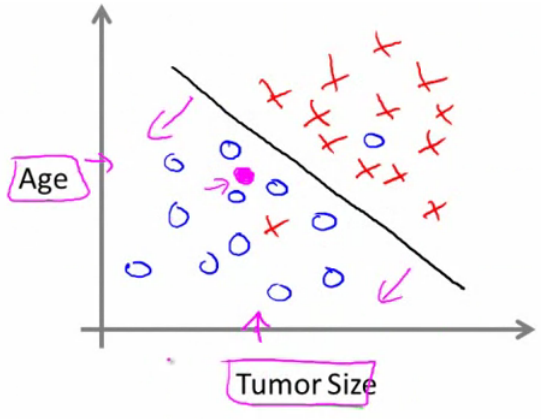

Figure 2: What is the likelihood of my friend with a tumour of size x **and** of age y having a malignant of benign cancer?

Note. We can deal with problems that have an infinite number of attributes (see support vector machine in chapter).

1.1.2 Unsupervised Learning

Definition 1.3. In **unsupervised** learning, we let the computer determine its own structure and patterns in the data.

Example 1.1. ***Clustering (*relationships among the variables in the data*)***

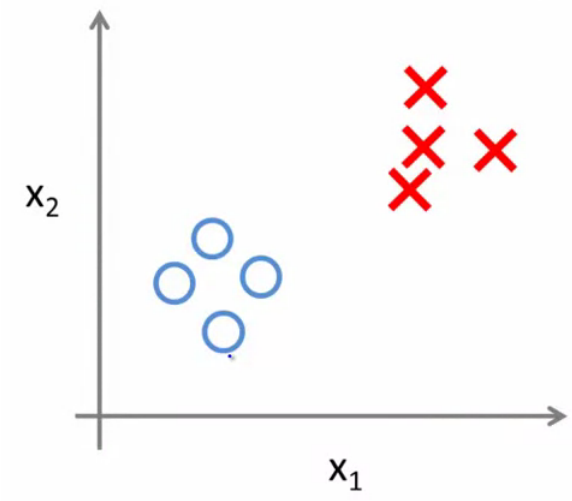

Figure 3: Given an **unknown** data set, can you find some structure to the data? E.g. Google News - given a set of new articles, group them into cohesive groups.

1.2 LEARNING = REPRESENTATION + EVALUATION + OPTIMISATION[1]

There are literally thousands of machine learning algorithms available, and hundreds more are published each year. The key to not getting lost in this huge space is to realise that it consists of combinations of just three components:

1. **Representation:** The model must be represented in some formal language that the computer can handle. In particular, how should we represent the inputs (or features) to be used, as well as the mapping from features to predicted output.

[1] Information obtained from `https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf`

2. **Evaluation:** An evaluation function is needed to distinguish good mapping from inputs to outputs.

3. **Optimisation:** what learner best maps the inputs to the target output (i.e. which parameters minimises the cost function).

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Figure 4: Common examples of each of the 3 components of learning algorithms.