Learning Machine

1 Introduction

1.1 Definition

> **Machine learning** explores the construction and study of algorithms that can learn from and make predictions on data. Such algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions, rather than following strictly static program instructions.

1.2 History and Development

Machine learning has evolved from the study of pattern recognition and computational learning theory in Artificial Intelligence.

> Field of study that gives computers the ability to learn without being explicitly programmed
>
> Arthur Samuel, 1959

> 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
>
> Tom M. Mitchell

> Can machines think?
>
> Alan Turing, *Computing Machinery and Intelligence*

1.3 Categorisation

Based on the nature of the learning signal,

- In **supervised learning**, the goal is to learn a general rule that maps inputs to outputs based on a training dataset (input and output).
- With **unsupervised learning**, no labels are available and the goal is to find structure from input only.
- **Reinforcement learning**, aims to take actions in an dynamic environment so as to maximize some notion of cumulative reward (such as driving a vehicle).

Among other small categories, **semi-supervised learning** is dealing with incomplete training dataset, **learning to learn** learns its own inductive bias based on previous experience and **developmental learning** generates its own sequences of learning situations to cumulatively acquire novel skills.

Another categorization arises when considering outputs:

- In **classification** (supervised) and **clustering** (unsupervised), inputs are divided into classes
- **In regression** (supervised), the outputs are continuous rather than discrete.
- **Density estimation** finds the distribution of inputs in some space.
- **Dimensionality reduction** simplifies inputs by mapping them into a lower-dimensional space

2 Tool

2.1 Tree

- A **graph** is mathematical structures used to model pairwise relations (*edge* or line) between objects (*vertices* or nodes). The edges may be *directed* or *undirected* for symmetric or asymmetric relation.
- In *graph theory*, a **tree** is an undirected graph in which any two vertices are connected by exactly one path.
- An **ordered tree** is a rooted tree (one vertex has been designated as the root) for which an ordering is specified for the children of each vertex.
- In *computer science*, a **tree data structure** is an ordered tree which has a value associated to each node.
- A **decision tree** is a decision support tool that uses a tree-like graph of decisions and their possible consequences (leaf), including chance event outcomes, resource costs, and utility.

2.2 Bayes

- **Bayes' theorem** relates an updated probability $P(A|B)$ to its prior probability $P(A)$ in relation to an event B and its relation to it $P(B|A)$

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

- **Bayes' Rule** is an equivalent way to formulate Bayes' theorem but relating only a proportional relation.

$$P(A|B) \propto P(A)P(B|A)$$

- **Statistical inference** is the process of deducing properties of an underlying distribution by analysis of data
- **Bayesian inference** is a method of inference in which Bayes' rule is used to update the probability for a hypothesis as evidence is acquired.

$$p(\theta \mid \mathbf{X}) = \frac{p(\mathbf{X} \mid \theta)p(\theta)}{p(\mathbf{X})} \propto p(\mathbf{X} \mid \theta)p(\theta)$$

where:

  - θ, the parameter
  - $\mathbf{X}$, a set of n observed data points
  - $p(\theta)$, the prior distribution
  - $p(\theta \mid \mathbf{X})$, posterior distribution
  - $p(\mathbf{X} \mid \theta)$, sampling distribution or likelihood
  - $p(\mathbf{X}) = \int_\theta p(\mathbf{X} \mid \theta)p(\theta)\,\mathrm{d}\theta$, marginal likelihood

2.3 Other

In geometry, a **hyperplane** is a subspace of one dimension less than its ambient space.

3 List of Algorithm

3.1 Decision tree learning

The goal is to create a model that predicts the value of a target variable based on several input variables. Each interior node corresponds to one of the input variables and each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

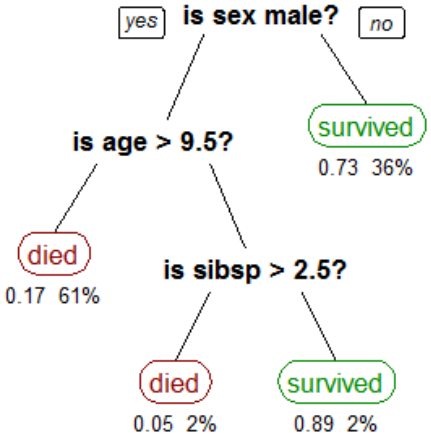


Figure 1: A tree showing survival of passengers on the Titanic ("sibsp" is the number of relative aboard). The figures under the leaves show the probability of survival and the percentage of observations in the leaf.

3.2 Support vector machines (SVMs)

A SVM is a supervised non-probabilistic binary linear classifier which constructs a hyperplane able to separate the two categories.

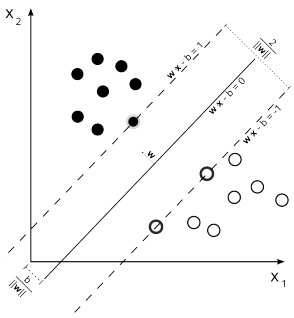
Given some training data $\mathcal{D}$, a set of n points of the form

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p,\, y_i \in \{-1, 1\}\}_{i=1}^n$$

We want to find the maximum-margin hyperplane that divides the class. Any hyperplane can be written as the set of points $\mathbf{x}$ satisfying

$$\mathbf{w} \cdot \mathbf{x} - b = 0$$

Where $\mathbf{w}$ is the normal vector to the hyperplane and $\frac{b}{\|\mathbf{w}\|}$ is the hyperplane offset from the origin along $\mathbf{w}$.



Graphicaly, the solution is fully determined by the nearest points which are called *support vectors*. Mathematically, we

are looking at a function

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b = \begin{cases} \geq 1 & \text{class 1} \\ 0 & \text{hyperplan} \\ \leq -1 & \text{class 2} \\ else & \text{margin} \end{cases}$$

The maximum-margin is achived when $\|\mathbf{w}\|$ is minimized, the optimization problem becomes

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{minimize}} && \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to} && y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 \end{aligned}$$

For not linearly separable data, the hinge loss function penalazied data $\mathbf{x}_i$ on the wrong side of the hyperplan with

$$\max\{0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i - b)\}$$

.

The optimization problem become

$$\underset{\mathbf{w},b}{\text{minimize}} \quad \frac{1}{n}\sum_i \max\{0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i - b)\} + \frac{1}{2}\lambda\|\mathbf{w}\|^2,$$

where the parameter λ determines the tradeoff between increasing the margin-size and ensuring that the $\mathbf{x}_i$ lie on the correct side of the margin.

A non-linear adaptation of SVM has been proposed by applying the kernel trick: every dot product are replaced by a nonlinear kernel function.

The classical approach to solve this problem is see it as a quadratic programming problem (Lagrange multiplier for linearly separable data or Augmented Lagrangian method else). More modern algorithm solves this problem with gradient problem: sub-gradient descent (if there are many training examples) or coordinate descent (if the dimension of the feature space is high).

3.3 Relevance vector machine (RVM)

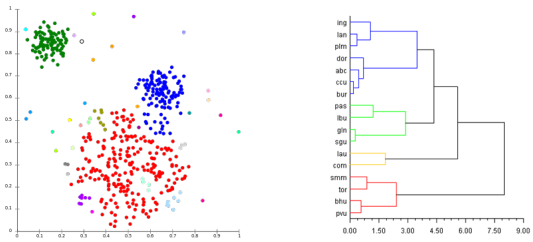
The RVM has an identical functional form to SVM, but provides probabilistic classification using Bayesian inference.

3.4 Cluster analysis

Clustering is the task of grouping objects such that objects in the same group are more both similar to each other and different to those in other groups. Cluster Analysis can also be formulated as a multi-objective optimization.

Hierarchical clustering

This algorithm base similitude on a measure of distance (euclidian distance or others). Dendrogram is the best way to visualised the "hierarchical" aspect of this algorithm: the bottom axis is the distance where two cluster merge (or divide for top-down approch)

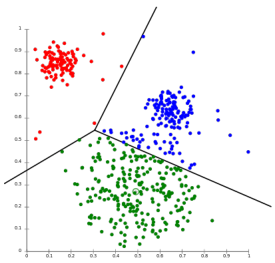


k-means algorithm

K-means clustering gives a formal definition as an optimization problem: find the k cluster centers and assign the objects to the nearest cluster center.

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \mathbf{c}_i\|^2$$

with $\mathbf{x}$ being the observations point, c_i the cluster center and $\mathbf{S}$ the k-set.



Fuzzy

In fuzzy clustering, data elements belong to a cluster according to a membership levels.

The widely used Fuzzy C-Means Algorithm (Bezdek 1981) is an adaptation of the k-mean algorithm where each element is compared to each center (not only one) and a partition matrix w_i is introduced to describe the degree of membership.

$$\arg\min_{C} \sum_{i=1}^{k} \sum_{j=1}^{n} w_{ij}^m \|\mathbf{x}_j - \mathbf{c}_i\|^2$$

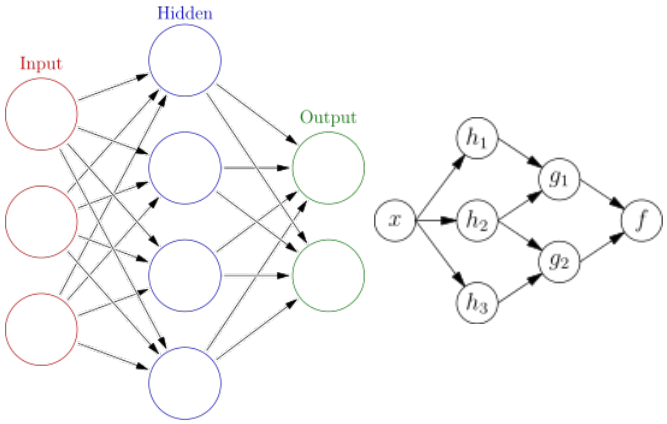
The fuzzifier m determines the level of cluster fuzziness.

$$w_{ij} = \frac{1}{\sum_{u=1}^{k} \left(\frac{\|\mathbf{x}_j - \mathbf{c}_i\|}{\|\mathbf{x}_j - \mathbf{c}_u\|} \right)^{\frac{2}{m-1}}} \qquad \mathbf{c}_i = \frac{\sum_j {w_{ij}}^m x_j}{\sum_j {w_{ij}}^m}$$

The algorithm converge by computing iteratly the weight and center.

3.5 Artificial neural network (ANNs)

There is no single formal definition, however ANNS are capable of approximating a non-linear functions with a large number of inputs by mimicking biological neural networks. Each nodes (neurons) consist of adaptive weights, parameters tuned by a learning algorithm.



3.5.1 Neurons Model

- Linear neurons: output is a weighted some of the imput plus a bais

$$z(x) = \sum_i w_i x_i + b$$

- Binary threshold neurons: output is a step function for the weighted input exeeding a threshold b.

$$y(x) = 1 \text{ if } z(x) > b \text{ else } 0$$

- Rectified Linear Neurons: combine a linear outpout with a threshold (low pass or high pass filter)

- Sigmoid neurons: real-value output of a smooth and bounded function (eg: logistic function). Their nice derivative make learning easy.

$$y(x) = \frac{1}{1 + e^{-z}}$$

- Stochastic binary neurons Transform linear neurons or sigmoid neurons as a stocastic process:

$$P(s = 1) = y(x)$$

3.5.2 Neural Network Architecture

- Feed-forward neural networks: organized by layers: input layer -¿ hidden layers -¿ output layer

- Recurrent neural networks: have cycles in their connection graph, difficult to train but much powerfull

- Symmetricall connected networks are recurrent network with weight equal in both direction. much easier to analyze. They are called Hopfield nets if no hidden unit are present and Boltzmann machines otherwise.

3.5.3 Perceptron

supervised learning, binary, linear classifiers

3.6 Bayesian networks

3.7 Genetic algorithms

3.8 Bootstrap

3.9 ANOVA

3.10 Logistic regression

3.11 k-nearest neighbor