Machine Learning

Semester 1, 2024

Dr Simon Denman

Tarang Janawalkar

Contents

1 Introduction

Machine learning is a field of computer science concerned with the development of statistical algorithms that enable computer systems to learn insights from data. Machine learning is a multi-disciplinary field that is related to statistics, pattern recognition, data mining, and artificial intelligence.

1.1 Machine Learning Process

Broadly speaking, there are three steps to enable machines to learn.

1. **Input Data**: A model must be provided with input data. This data consists of audio, images, text, or any other form of data, which enables us to build a model. Ideally, it is desirable to have a large amount of data, as this produces a more accurate model, and removes the possibility of overfitting when many features are present.
2. **Data Abstraction**: To effectively train a model on this data, it needs to be abstracted into a consistent format that can be understood by a model. This includes pre-processing the data, removing any noise or irrelevant features, and ensuring that the dimensionality (or the number of features) is the same across all samples.
3. **Generalisation**: When data is ready, we can use a machine learning technique to determine whether a model generalises well to new data.

1.2 Machine Learning Paradigms

There are three main types of machine learning paradigms:

- **Supervised Learning**: In this type of learning, a model is provided both input and expected output data. The purpose here is to learn a mapping from input to output, to predict an output for new unseen data.

  For example, we might provide a model a video of pedestrians with a person count in each frame. The model then regresses features such as size, texture, and shape to predict the number of people in unseen frames.

  Another related example may include object tracking, where a model learns to track objects in a video, given a pre-labelled set of frames with tracked objects.

- **Unsupervised Learning**: In this type of learning, a model is provided only input data, and is mainly used for knowledge discovery—to find order and characteristics in data.

  An example of this is clustering, where a model might group people who are walking together, based on their motion characteristics.

  Another example is the detection of anomalies or abnormal behaviour in data. For example, a model may be trained on pedestrian motion, and use unsupervised learning to track vehicles and bicycles in a pedestrian zone.

- **Reinforcement Learning**: In this type of learning, a model learns to make decisions by interacting and responding to an environment. The model is rewarded or penalised based on its actions, and its goal is to learn the best sequence of actions to maximise its reward.

An example of this may be a robot learning to walk. The objective might be to maximise the distance travelled in one direction.

1.3 Importance of Data

Data is the most important aspect of machine learning, both in terms of **quality and quantity**. A high-quality dataset that is well prepared and which covers all possible cases considered in a model, leads to better outcomes.

- a model cannot produce predictions that are better than the data it is trained on
- a model cannot compensate for errors in the annotation of data

It is also important to consider **data diversity** to ensure that a model does not learn **biases** present in the data. For example, a male dominated dataset may lead to a model that is biased against women.

Data also suffers from the *curse of dimensionality*. Datasets often have:

- a large number of dimensions or features (i.e., columns, observed variables)
- but a small number of samples (i.e., rows, observations)

In such cases, the feature space is sparsely populated and the model may **overfit**. Every new feature increases the complexity of the model, and also necessitates more data, to produce an accurate model—this number depends on the type of classifier or learning algorithm used, and therefore a model should be kept simple.

1.4 Data Splitting

To efficiently train and evaluate a model, we can split the data into three sets:

- **Training Set**: This set is used to train the model.
- **Validation Set**: This set is used to tune model hyperparameters to evaluate a model's performance. The training set will be trained using different hyperparameters to evaluate which set of hyperparameters produce the best model.
- **Test Set**: This set is used to evaluate the model's performance on unseen data.

This approach uses *holdout validation* where data is *held out* for validation and/or testing, so that the training, validation, and testing stages are completely separate. Note that this requires datasets to be large enough to ensure that a model has enough data for training.

When **insufficient data** is available, *cross-validation* can be used to dynamically split the dataset into training and validation sets (i.e., a 80% training/20% validation split). This may be repeated 5 times, so that each split is used as a validation set. The best model is then selected based on the average performance across all splits.

In some cases, machine learning may not be possible if the dataset is too small. This may be because the data does not contain the patterns and relationships that we are trying to learn. For some problems, this threshold may be a few hundred samples, while for others, it may be a few million. Extrapolation beyond the data may also be problematic in such cases.

1.5 Traditional Machine Learning to Deep Learning

1.5.1 Traditional Machine Learning

The traditional machine learning pipeline typically consists of:

- **Pre-processing**: Preparing data for a model using normalisation, scaling, noise reduction, etc.
- **Feature Extraction**: Extracting features from data; MFCC (audio), HOG (image/video), and bag-of-words (text).

  This step is often informed by the task. For example, if we need to recognise shapes in images, we might use a technique that captures edge detection. In the case of audio, we might use a technique that captures frequency information.
- **Machine Learning**: Passing the features to a model using a machine learning technique.

In this model, features are **hand-crafted**, and based on domain-specific knowledge. Choosing the optimal features for a given problem can be a tedious and iterative process. This leads to different formulations for different tasks, and may significantly increase the complexity of a problem, especially when we need to extract multiple sets of features.

1.5.2 An Aside on Neural Networks

A neural network is modelled after the human brain and nervous system. It is built on a single unit or **node** called a *neuron*, in which data is passed through a series of **layers**, each of which is connected to other neurons through **edges**. Neural networks typically consist of a large number of parameters and many interconnections.

Choosing the appropriate number of layers in a neural network (or the *depth* of a neural network), is problem dependent. A higher depth allows us to learn *higher level features*, and contains more parameters, but this is often harder to learn and also increases the likelihood of overfitting. Larger networks also require more data, and therefore more memory and computational resources.

1.5.3 Deep Learning

Deep learning is a subset of machine learning that uses neural networks to learn features from data. It differs from the traditional machine learning pipeline in that it does not require manual feature extraction, but instead learns it's own representation from pre-processed data. This makes deep learning more **adaptable** to different tasks, and also reduces the complexity of a problem.

A downside of deep learning is that models are often extremely large compared to traditional approaches;

- a traditional machine learning model may have a few hundred parameters
- a deep learning model may have millions of parameters

This leads to an increase in computational resources required to train and evaluate a model, and also makes a model difficult to interpret.

In deep learning, feature engineering is replaced with **network engineering**, where we must carefully choose the appropriate architecture for a problem. This includes:

- the number of layers,
- the number of neurons in each layer, and
- the type of activation function used.

This is often an iterative process, and it may not be feasible to find the optimal architecture for a given problem, through exhaustive testing.

2 Linear Regression

A simple model for predicting the relationship between a dependent variable y (**response**), and an independent variable x (**predictor**), is a straight line. Given a set of n input-output pairs $\left(x^{(i)},\, y^{(i)}\right)$, we can model a linear relationship between x and y as:

$$y^{(i)} = \beta_0 + \beta_1 x^{(i)}$$

where β_0 is the y-**intercept** and β_1 is the **gradient** of the line. In practice, data rarely fits a model perfectly and therefore, we will assume the relationship:

$$\hat{y}\,(x) = \hat{\beta}_0 + \hat{\beta}_1 x$$

where $\hat{y}$ is known as the **hypothesis function** which **predicts** value of y, and $\hat{\beta}_0$ and $\hat{\beta}_1$ are the **estimated** values of the **parameters** β_0 and β_1, respectively. This model adds a **residual** term ϵ to account for the error in each observation:

$$\begin{aligned} y^{(i)} &= \hat{y}\left(x^{(i)}\right) + \epsilon^{(i)} \\ &= \hat{\beta}_0 + \hat{\beta}_1 x^{(i)} + \epsilon^{(i)} \end{aligned}$$

here ϵ is assumed to be normally distributed with zero mean and σ^2 variance:

$$\epsilon \overset{\text{iid}}{\sim} \mathrm{N}\left(0,\, \sigma^2\right).$$

As y depends on ϵ, we can also show that

$$y \sim \mathrm{N}\left(\beta_0 + \beta_1 x,\, \sigma^2\right)$$

where y is normally distributed for fixed values of x. The values of $\hat{\beta}_0$ and $\hat{\beta}_1$ arise when we consider the **loss function**:

$$L\left(\boldsymbol{\beta}\right) = \frac{1}{2}\epsilon^2 = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(y - (\beta_0 + \beta_1 x))^2.$$

which we wish to minimise. As these estimators apply to the entire training set, we will consider the **cost function**:

$$J\left(\boldsymbol{\beta}\right) = \frac{1}{n}\sum_{i=1}^{n} L\left(\beta\right).$$

which averages the loss across all observations. The goal of this cost function is to find the best parameters $\hat{\boldsymbol{\beta}}$ that minimise the loss across all observations:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} J\left(\boldsymbol{\beta}\right).$$

In this model, we must ensure that there are at least as many observations as there are parameters to estimate.

2.1 Correlation

Correlation is a statistical relationship between two variables. This measure allows us to determine the strength of a linear relationship between two variables. One common measure of correlation is the **Pearson correlation coefficient**, which is given by:

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

where σ_{xy} is the **covariance** between x and y, and σ_x and σ_y are the variances of x and y, respectively. Using the sample covariance[1]:

$$r_{xy} = \frac{s_{xy}}{s_x s_y}.$$

The sample covariance s_{xy} has the following characteristics:

- $s_{xy} > 0$: y increases as x increases
- $s_{xy} < 0$: y decreases as x increases
- $s_{xy} \approx 0$: no linear relationship between x and y

To generalise this measure across datasets, we often consider the *normalised* correlation coefficient r_{xy}, which tells use the strength of a linear relationship between two variables. The following diagram shows various datasets and their correlation coefficients:

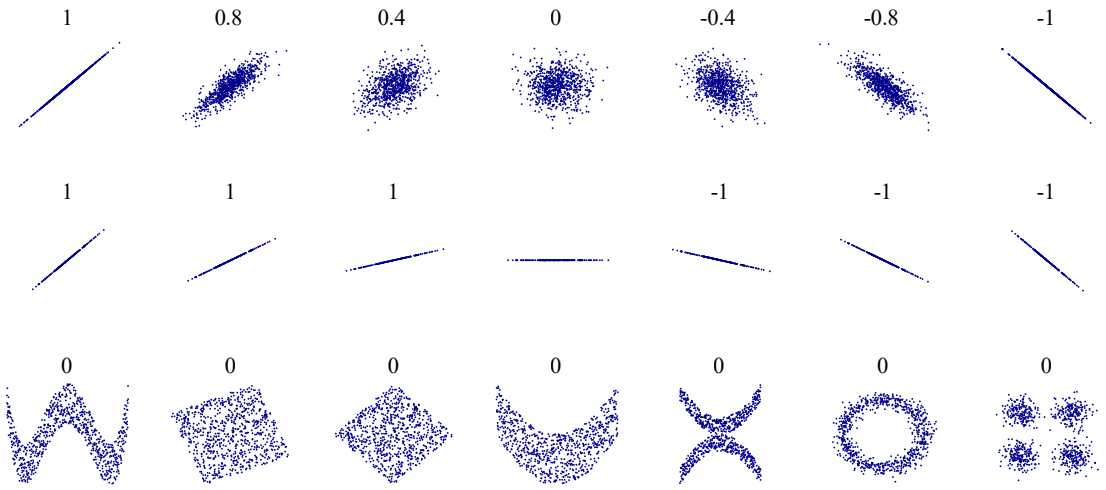

Figure 1: Pearson correlation coefficient of x and y for several sets of points.

There are a number of important points to note about the correlation coefficient:

[1] See Appendix A for more details.

- The correlation coefficient does not contain any information about the gradient, as is seen in the second row.
- A correlation coefficient of 0 indicates *no linear relationship*, it does not imply that there is *no relationship*, as is seen in the third row.
- The correlation coefficient does not imply *causation*. Care must be taken when assuming that a relationship between two variables is causal.

To minimise redundancy in a model, it is crucial for predictors to be correlated with response variables, and uncorrelated to other predictors.

2.2 Simple Linear Regression

Simple linear regression is a special case of linear regression, where we have only one input variable x_1, and we can use the method of maximum likelihood estimation to estimate the parameters β_0 and β_1:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} \left(x^{(i)} - \bar{x}\right)\left(y^{(i)} - \bar{y}\right)}{\sum_{i=1}^{n} \left(x^{(i)} - \bar{x}\right)^2}$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

where $\bar{x}$ and $\bar{y}$ are the sample means of x and y, respectively.

2.3 Multiple Linear Regression

In multiple linear regression, we consider a p-dimensional feature space, or p input variables, with $p+1$ parameters to estimate. Here hypothesis function generalises to:

$$\hat{y}\left(x\right) = \hat{\beta}_0 x_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p = \sum_{i=0}^{p} \hat{\beta}_i x_i = \boldsymbol{\beta}^\top \mathbf{x}.$$

where we have introduced the feature $x_0 \coloneqq 1$ to simplify our notation. There are several ways to estimate $\hat{\boldsymbol{\beta}}$, and here we will use matrix calculus. The cost function $J\left(\boldsymbol{\beta}\right)$ can be written as:

$$J\left(\boldsymbol{\beta}\right) = \frac{1}{2n} \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\beta}^\top \mathbf{x}^{(i)}\right)^2 = \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = \frac{1}{2n} \boldsymbol{\epsilon}^\top \boldsymbol{\epsilon}.$$

where $\mathbf{X}$ is an $n \times (p+1)$ matrix of input variables:

$$\mathbf{x} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_p^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_p^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(n)} & x_2^{(n)} & \cdots & x_p^{(n)} \end{bmatrix}$$

Consider the vector derivative of the cost function with respect to $\boldsymbol{\beta}$:

$$\begin{aligned}
\frac{\partial J(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} &= \frac{1}{2n}\frac{\partial}{\partial \boldsymbol{\beta}}\left[\boldsymbol{\epsilon}^\top \boldsymbol{\epsilon}\right] \\
&= \frac{1}{2n}\frac{\partial}{\partial \boldsymbol{\beta}}\left[(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right] \\
&= \frac{1}{2n}\frac{\partial}{\partial \boldsymbol{\beta}}\left[(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top \mathbf{y} - (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top \mathbf{X}\boldsymbol{\beta}\right] \\
&= \frac{1}{2n}\frac{\partial}{\partial \boldsymbol{\beta}}\left[\mathbf{y}^\top \mathbf{y} - (\mathbf{X}\boldsymbol{\beta})^\top \mathbf{y} - \mathbf{y}^\top \mathbf{X}\boldsymbol{\beta} + (\mathbf{X}\boldsymbol{\beta})^\top \mathbf{X}\boldsymbol{\beta}\right] \\
&= \frac{1}{2n}\frac{\partial}{\partial \boldsymbol{\beta}}\left[\mathbf{y}^\top \mathbf{y} - \boldsymbol{\beta}^\top (\mathbf{X}^\top \mathbf{y}) - (\mathbf{y}^\top \mathbf{X})\boldsymbol{\beta} + \boldsymbol{\beta}^\top (\mathbf{X}^\top \mathbf{X})\boldsymbol{\beta}\right] \\
&= \frac{1}{2n}\left(-\mathbf{X}^\top \mathbf{y} - (\mathbf{y}^\top \mathbf{X})^\top + 2\mathbf{X}^\top \mathbf{X}\boldsymbol{\beta}\right) \\
&= \frac{1}{n}\left(\mathbf{X}^\top \mathbf{X}\boldsymbol{\beta} - \mathbf{X}^\top \mathbf{y}\right).
\end{aligned}$$

This implies that the optimal parameters $\boldsymbol{\beta}$ are given by:

$$\begin{aligned}
\frac{\partial J(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} &= \mathbf{0} \\
\mathbf{X}^\top \mathbf{X}\boldsymbol{\beta} - \mathbf{X}^\top \mathbf{y} &= \mathbf{0} \\
\mathbf{X}^\top \mathbf{X}\boldsymbol{\beta} &= \mathbf{X}^\top \mathbf{y} \\
\boldsymbol{\beta} &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}.
\end{aligned}$$

2.4 Assumptions of Linear Regression

There are several assumptions that must be satisfied for linear regression to be valid:

1. **Linearity**: The relationship between the responses and predictors is linear.
2. **Independence**: Responses are linearly independent.
3. **No Multicollinearity**: Predictors are not linearly dependent.
4. **Normality**: Residuals are normally distributed.
5. **Homoscedasticity**: Residuals have constant variance.
6. **Exogeneity**: There is no correlation between predictors and residuals.

A Numerical Summaries of Data

A.1 Mean

The mean of a set of n observations $\left(x^{(i)}\right)$ is given by:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x^{(i)}$$

A.2 Variance

The variance of a set of n observations $\left(x^{(i)}\right)$ is given by:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} \left(x^{(i)} - \bar{x}\right)^2$$

For a sample, the variance is given by:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left(x^{(i)} - \bar{x}\right)^2$$

A.3 Covariance

The covariance between two sets of n observations $\left(x^{(i)}\right)$ and $\left(y^{(i)}\right)$ is given by:

$$\sigma_{xy} = \frac{1}{n} \sum_{i=1}^{n} \left(x^{(i)} - \bar{x}\right)\left(y^{(i)} - \bar{y}\right)$$

For a sample, the covariance is given by:

$$s_{xy} = \frac{1}{n-1} \sum_{i=1}^{n} \left(x^{(i)} - \bar{x}\right)\left(y^{(i)} - \bar{y}\right)$$