# Machine Learning

# Machine Learning

## Theory

Chapter 1

## Supervised Learning

- Where your training data includes labels for each sample of input parameters:
  - **Classification**: A subset of supervised learning where your labels are discrete classes
  - **Regression**: Your labels are continuous

## Unsupervised Learning

- Labels are not available and the goal is to identify patterns in the data:
  - Clustering
  - Principal Component Analysis

## Rules-based techniques

- Instead of having a function based model y = f(**x**), you use a set of predefined rules to reach a classification
  - Random forest's (using the result of multiple decision trees to make an overall decision)

## Covariance

Just as in the univariate case, it's variance can be determined by the average sum of squared differences between instances of the variable and it's mean:

$$(\sigma^2) = \frac{\sum (X_i - \mu)^2}{m}$$

The same is the case for a multivariate problem, where the output is now a matrix with non-diagonal elements representing interdependence between variables:

$$C = \frac{(X_i - \mu)^t (X_i - \mu)}{m}$$

where m is just the number of points in the dataset. Note that **C** is always symmetric.

Chapter 11

# General Concepts

## Bias/ Variance Trade-off

- Bias pertains to the error of a model when compared to its training dataset - high bias suggests it poorly fits to the dataset that it was trained on, very low bias suggests potential overfitting of data.
- Variance pertains to how much the error changes when the model is evaluated on a different dataset to the one it was trained on. High variance suggests that the model has overfit the training dataset and the error changes drastically when compared to a non-training dataset. Low variance suggests the model has a consistent error rate when evaluated on other datasets
- Total error = Bias + Variance. Ideally we therefore want to minimise both
- Usually complex models lead to large variance (overfit), and simple models lead to large bias (underfit)

## Validation

- Validation is used during training to assess how well a model is performing on unseen data - it's essentially intermittent testing during the training phase.
- K-fold cross validation is commonly used
    - Here the dataset is split into K folds, and for each fold, the other k-1 folds are used for training and the remaining fold is used for testing.

> **Note**
>
> The idea here isn't to find the best model after trying all k folds- no, its to identify suitable model parameters such as regularisation parameters etc. by looking at average performance and seeing if results are suitable

## Scaling

- It is common practice to scale (e.g. normalise) the feature space, so that each feature is treated equally by the model.
- In the case where you had a 2 input model with features O(10^6) and O(10^1) respectively, a change of 1 in feature 2 will lead to significantly more change than the same change in feature 1, and as such the gradients will look to change feature 2 more than 1 and neglect (/converge much more slowly to) the potential improvements that could be seen by changing feature 1, by say 10^5.

> **Note**

Always save transformation parameters used on the training data so that when used on unseen data, the same transformations can be applied to map the input space correctly

## Metrics

- These are used to assess similarity and must have the following properties:
  - Non-negativity:

$$D(a, b) \geq 0$$

  - Symmetry:

$$D(a, b) = D(b, a)$$

  - Reflexivity:

$$D(a, b) = 0, \text{ if and only if a } ==\text{b}$$

  - Triangle inequality:

$$D(a, c) \leq D(a, b) + D(b, c)$$

- These can be shown to hold for metrics such as Euclidian distance, lp, etc.

Contents