Bayesian Decision Theory

Chapter 2 Machine Learning
This is a stats-based decision framework that aims to identify the probability of a state of nature, **ω**, given a certain observation and then use this to inform the best action, **α**, to take. Let's break this down:

- In a classification problem, the classes are known as the **states of nature**
- The total probability of each state of nature is intuitively 1 (i.e. the actual state of nature can only take up one of the potential states, so their combined probability ha gotta be 1):

  $$\sum P(\omega_i) = 1$$

  where P(**ω** i) is known as the **prior probability** of state of nature i which is independent of any observations.
- The probability of an observation, x, given a state of nature, w_i, is called the **likelihood** expressed as:

  $$Likelihood = P(x|\omega_i)$$

- What we actually want though is the probability of a class itself given an observation known as the **posterior**:

  $$Posterior = P(\omega_i|x)$$

  - From Bayesian stats, this can be derived from the typical Bayes Theorem equation:

  $$P(\omega_i|x) = \frac{P(x|\omega_i)p(\omega_j)}{p(x)} = \frac{Likelihood \times Prior}{Evidence}$$

  where the evidence, p(x) can be found from the total probability formula:

  $$p(x) = \sum P(x \cap \omega_i) = \sum P(x|\omega_i) \times p(\omega_i)$$

- So based on the posterior we then need to make a decision. The best approach isn't to necessarily choose the highest posterior as the class, so we introduce the idea of a loss function which we aim to minimise based on the available actions that we can take.

  $$loss = \lambda(\alpha_i|\omega_i)$$

  $$Risk = R(\alpha_i|x) = \sum_j \lambda(\alpha_i|\omega_j) \times P(\omega_j|x)$$

- So the risk of a particular action given an observation is basically the **expected loss**: it's the total loss associated with each action given the state of nature, multiplied by the

posterior of that state of nature (i.e. it's chance of being the true state of nature given this observation).

Practical Usage

The whole notion of this thing depends on us knowing the likelihoods and priors- these of course need to be estimated themselves:

- The MLE is used for this. It revolves around maximising the likelihood of achieving our dataset given it was derived from certain parameters. More formally:

$$likelihood = P(D;\boldsymbol{\theta}) = \prod_k P(\boldsymbol{x_k};\boldsymbol{\theta})$$

- The log likelihood, l(θ), is used for easier math downstream:

$$l(\boldsymbol{\theta}) = lnP(D;\boldsymbol{\theta}) = \prod_k lnP(x_k;\boldsymbol{\theta})$$

- And this is of course maximised by finding the stationary points of this vector of log likelihoods. We also often can assume that our sample data follows a normal distribution such that:

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

  where d denotes the number of dimensions.
- In the bivariate parameter case (d = 2), letting θ_1 = μ and θ_2 = σ^2 this becomes:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\theta_2}}\exp\left(-\frac{1}{2\theta_2}(\mathbf{x}-\theta_1)^2\right)$$

- Subbing this into the max log likelihood formulae you get a system of equations: 1 for each parameter that you can use to solve for unknowns. For this bivariate case you end up with

$$\hat{\theta}_1 = \frac{1}{n}\sum_{i=1}^{n} x_i$$

  which makes a lot of intuitive sense given this is meant to be the MLE for μ, and

$$\hat{\theta}_2 = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2$$

  which interestingly has a divisor of n instead of n-1 (as expected for the sample variance) but this is because we are technically estimating the distribution that would produce the of our dataset as opposed to estimating population stats from our sample