Non-Parametric Methods

Chapter 7 Machine Learning

- This is any method that doesn't attempt to estimate distribution parameters, because the underlying distribution itself is hard to model/ unknown
- So we want to estimate a probability distribution p(x)
  - For the 1d case, the probability, P(x1<x<x2) (or P shorthand), that k of m samples drawn from a window ranging from x1 to x2 of window size h is given by

    $$P(x_1 < x < x_2) = P = \frac{k}{m}$$

    (this makes intuitive sense- probability of x being in a class is the number of observations of that class / total number of observations)
  - Also, assuming the window size/ bandwidth, h (= x2-x1), is small and hence p(x) doesn't vary much within the window, we can say that the probability of falling within in the window is

    $$P = p(x)h$$

  - Rearranging for p(x), we get

    $$p(x) = \frac{k}{mh}$$

    i.e. the instantaneous probability of a value x in a given window is just uniformly distributed across this window's total probability so just divide total probability in the window by h :).
  - In high dimensional space, where there are more than 1 parameters, we sub h for V = h^n and hence are finding the volume of a hypercube which we use as the divisor for p(x)

    $$p(x) = \frac{k}{mV}$$

Parzen windows

- These execute the above principles to formally classify (i.e. assign a probability to a value x)
  - We introduce a variable u for each window, where x' is a given window centre, such that

    $$u(x) = \frac{x - x'}{h}$$

- We can classify based on whether |u| < 1/2 i.e the point x lies in the window or not. Our classification function φ(u) = 1 or 0.
- We can then estimate k for each window by

$$k = \sum_i \phi(\frac{x_i - x'}{h})$$

  i.e. the total number of points in a window is the total number of points that are classified as 1 by our window function for this window
- And hence our probability distribution simply becomes

$$p(x') = \frac{k}{mh} = \frac{1}{hm} \sum_i \phi(\frac{x_i - x'}{h})$$

- This approach leads to a step function-esc probability distribution using 'top hat' windows, others exist that aim to smooth things, such as Gaussian windows

Nearest Neighbours

- Do what they say on the tin- classify using the majority vote from k-nearest neighbours to a point, with neighbours decided by smallest distance