HW4

CS4720 Machine Learning

By HG King

Question 1:

19) Assume we have training data from a Gaussian distribution of known covariance **Σ** but unknown mean **μ**. Suppose further that this mean itself is random, and characterized by a Gaussian density having mean **μ0**and covariance **Σ0**

a. What is the MAP estimator **μ**?

In the case where the **μ** is unknown, the sample **μ** can be determined by:

**μ** = 1/n sum(**x**k), the sum being from k = 1 to n.

b. Suppose we transform our coordinates by a linear transformation **x**’ = **Ax**, for non-singular matrix **A**, and accordingly for other terms. Determine whether your MAP estimator gives the approximate estimate for the transformed mean **μ**’.

In this case, the formula would still be the same, as it is just a simple average over the data set. The actual value of the sample mean will change though since the data points will change.

31) In the following, suppose *a* and *b* are positive constants greater than 1 and *n* a variable parameter.

a) Is an+1 = O(an)?

No, say n is equal to 2, then n+1 is equal to three. In terms of big O notation, O(a2) is much smaller than O(a3).

b) Is abn = O(an)?

Yes, for a sufficiently large n, any constant multiplier will not affect the order of the function. If b is close in size to n, then the big O will be closer to O(n2).

c) Is an+b = O(an)?

No, since b is a constant which is greater than 1, looking at part a above, it shows that adding a constant of 1 will make the order higher than O(an).

d) Prove f(n) is O(f(n)).

Suppose f(n) = a0 + a1n + a2n2. With a sufficiently large n, approaching infinity, the constant a terms won’t have much effect on the value of the function. Additionally, the n’s with a lower order of growth will not have much effect on the value of the function either. This leaves us with the term of highest order, in this case it is n2. This can be written using big O notation: f(n) = O(n2).

Question 2:

See python file.