7.6 Parameter ESTIMATION: Map

MAP : Maximum a posteriori Estimation

We saw that while calculating MLE we do not consider any prior knowledge which, in some situations, might produce undesirable result.

For example, let’s toss an unbiased coin 4 times. We repeat this experiment twice and got following results.

$$T, H, T, T \rightarrow \theta = 0.25$$
$$T, T, T, T \rightarrow \theta = 0.0$$
$$where, \theta\, is\, parameter\, and\, represents\; Probability\, of\, occurance\, of\; Head$$

We all know that in an unbiased coin, the probability of getting head is 0.5, but in first case θ_{MLE} is 0.25 and in second case it is 0.0, these two values are way off from the actual value. This happened because we did not consider the prior knowledge of probability of getting head which is 0.5.

MAP is a Bayesian approach which reflects our belief, prior knowledge, about θ.

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$
$$where,$$
$$P(\theta|D) = posterior\, probability$$
$$P(\theta) = Prior$$
$$P(D|\theta) = likelihood\, or\, class\, conditional$$

Thus, Bayes' law converts our prior belief about the parameter θ (before seeing data) into a posterior probability, by using the likelihood function. The maximum a-posteriori (MAP) estimate is defined as

$$\theta_{MAP} = \underset{\theta}{argmax}\, P(\theta|D) = \underset{\theta}{argmax}\, \frac{P(D|\theta)\, P(\theta)}{P(D)}\,.$$

As $P(D)$ is independent of θ, we can remove $P(D)$ from the above equation and get

$$\theta_{MAP} = \underset{\theta}{argmax}\, P(D|\theta)\, P(\theta) = \underset{\theta}{argmax}\, P(\theta) \prod_{i=1}^{n} P(X_i|\theta)$$

$$\log\left(\underset{\theta}{argmax}\, P(D|\theta)\, P(\theta)\right) = \underset{\theta}{argmax} \log(P(\theta)) + \sum_{i=1}^{n} \log(P(X_i|\theta))\,.$$

Next, we do partial differentiation of above equation w.r.t, θ equate it to zero, and then solve it to get value of θ_{MAP}.

Now we are going to see what is the θ_{MAP} of uni-variate normal distribution under a specific setup.

Setup :

Data generation: n data has been generated, X_1 to X_n, all these data points are independent given θ and they are drawn according to

$$N(\theta, \sigma^2)$$
$$normal\ distribution$$
$$mean = \theta$$
$$variance = \sigma^2$$

The prior distribution of the mean, θ, itself is a normal distribution, i.e., ν is also normally distributed.

$$N(\nu, \beta^2)$$
$$normal\ distribution$$
$$mean = \nu$$
$$variance = \beta^2$$

Under this setup

$$\theta_{MAP} = \frac{\sigma^2\nu + \beta^2 \sum_{i=1}^{i=n} X_i}{n\beta^2 + \sigma^2}$$

let's generate 1000 data with following parameter values

$$\nu = 0.6$$
$$\beta^2 = 1$$
$$\sigma^2 = 2.5$$
$$\theta = N(\nu, \beta^2)$$

```
import numpy as np
nu = 0.6
betaSquare = 1
sigmaSquare = 2.5
theta = np.random.normal(nu, betaSquare, 1000)  # generate 1000 value of theta first
data = [np.random.normal(i, sigmaSquare, 1) for i in theta]  # then generate data using theta and sigmaSquare
```

Next, we will calculate θ_{MAP} of the generated data. We will write a function to do that.

```
def calThetaMAP(nu, beta, sigma, data):
    numerator = sigma * nu + beta * sum(data)
    denominator = len(data) * beta + sigma
    thetaMAP = float(numerator) / float(denominator)
    print "theta MAP ", thetaMAP

calThetaMAP(nu, betaSquare, sigmaSquare, data)
print "Actual mean ", np.asarray(data).mean()  # first convert data which is a list into ndarray then calculate the mean
```

In the function calThetaMAP, following formula has been implemented.

$$\theta_{MAP} = \frac{\sigma^2\nu + \beta^2 \sum_{i=1}^{i=n} X_i}{n\beta^2 + \sigma^2}$$