CS F320 – Foundations of Data Science

Assignment – 1

Submission deadline: 24-10-2020, 23:59

General Instructions:

- This assignment is a coding project and is expected to done in groups. Each group can contain at most three members. Make sure that all members in the group are registered to this course and please try to maintain the same group for all the assignments.
- This assignment is expected to be done in Python using standard libraries like NumPy, Matplotlib and Scipy. You can use Jupyter Notebook. No other ML library like scikit/sklearn, TensorFlow, Torch etc. should be used.
- Refrain from directly copying codes/snippets from other groups or the internet as all codes will be put through a plagiarism check.
- All deliverable items (ex. `.py` files, `.ipynb` files, reports, images) should be put together in a single `.zip` file. Rename this file as `A3_<id-of-first-member>_<id-of-second-member>_<id-of-third-member>` before submission.
- Submit the zip file on CMS on or before the aforementioned deadline. Please note that this is a hard deadline and no extensions/exemptions will be given. The demos for this assignment will be held on a later date which shall be conveyed to you by the IC. All group members are expected to be present during the demo.

Problem Statement:

- Bob and Lisa would like to find out the probability of getting head, μ, of a biased coin. They are excited to get a probability distribution of μ but not just a point estimate. You will have to randomly generate a dataset for this problem wherein the size of the dataset should be around 160 and $\mu_{ML} \notin (0.4, 0.6)$. μ_{ML} is maximum likelihood estimator of the data that is being generated by you. As we know, $osterior \propto likelihood \times prior$, i.e. $p(\mu \mid D, a, b) \propto p(D \mid \mu)\, p(\mu \mid a, b)$. Our goal is to find the distribution followed by the mean of the coin tosses after observing the data points. As we know, coin tossing follows a Bernoulli distribution. Its probability density function is given by $Bern(x \mid \mu) = \mu^x (1 - \mu)^{1-x}$ where μ is the mean of the Bernoulli distribution. Thus, for a dataset D of N points, we get the likelihood function as:

$$p(D \mid \mu) = \prod_{n=1}^{N} p(x_n \mid \mu) = \prod_{n=1}^{N} \mu^{x_n} (1 - \mu)^{1-x_n}$$

We will take the prior to be a beta distribution. The PDF for a beta distribution is given by:

$$Beta(\mu \mid a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \mu^{a-1} (1 - \mu)^{b-1}$$

where a and b are the parameters and Γ is the gamma function (The gamma function is available in the Scipy library as `scipy.special.gamma`). Choose appropriate a and b such that the mean of the prior is 0.4. As seen in class, we know how to find out the posterior distribution given prior distribution and likelihood function. There are two approaches to find out the posterior distribution – one is to use all 160 data points at one go and the other is to use each data point sequentially.

What needs to be documented in your report:

- Bob would like to solve this problem using the following approach. One example will be taken at a time and hence the likelihood function will have only one term in its product. As with any sequential learning problem, this posterior after viewing the first example becomes the prior for the next example. Thus, you start off with a prior and will observe how examples coming in change the appearance of the prior. You will have to make plots of the prior at each stage (similar to Fig 2.2 in Bishop). Combine all the plots together to make a GIF.
- This is Lisa's view of the problem. In this case, the likelihood will be over the entire dataset. Compute the posterior after viewing the entire dataset at once and make plot of the posterior for the same. Comment on the differences/similarities between the two aforementioned models obtained.

Questions to ponder on:

- The size of the dataset has been restricted to 160 data points. What happens if more points are added (say ~10^5)? What would the posterior distribution look like if $\mu_{ML} = 0.5$? Which model, Bob's or Lisa's, would be more helpful and easier while working with large real time data and why?
- What if another distribution like Gamma, Guassian or Pareto were to be chosen as the prior? Would the posterior computation be easier or difficult and why?

For further references/understanding, you may go through pages 71 – 74 of Bishop's book on Pattern Recognition and Machine Learning and your class notes.