MIE223 Data Science
Lab and Assignment 11:
Bayesian Modeling for Data Science

In this lab and assignment, you will learn about Bayesian modeling and its applications in data science.

- Programming language: Python (Google Colab Environment)
- Due Date: Posted in Syllabus

Marking scheme and requirements: You are required to use the Jupyter notebook named `Assignment_11.ipynb` to provide appropriate answers to the questions in this assignment. All changes to the `Assignment_11.ipynb` notebook should be committed and pushed to your assignment repository on GitHub.

This assignment has *4 points* in total and the point allocation is shown below:

- Solution to Questions (1 point):
  - Q1(a): 0.1 point
  - Q1(b): 0.1 point
  - Q1(c): 0.2 point
  - Q1(d): 0.2 point
  - Q1(e): 0.1 point
  - Q2(a): 0.1 point
  - Q2(b): 0.1 point
  - Q2(c): 0.1 point
- Code review (1 point)
- Quiz (2 points)

What/how to submit your work:

- All your code should be included in the provided notebook named `Assignment_11.ipynb`.
- Commit and push your work to your GitHub repository in order to submit it. Your last commit and push before the assignment deadline will be considered to be your submission. You can check your repository online to make sure that all required files have actually been committed and pushed to your repository.
- A link to create a personal repository for this assignment is posted on QUERCUS.

Notes that you should pay attention to

1. All your code should be written in the provided notebook named `Assignment_11.ipynb`.
2. All functions must *return* the specified return type.
3. Commit and push your work to your GitHub repository in order to submit it. **Your last commit and push will be considered to be your submission.** You can check your repository online to make sure that all required code has actually been committed and pushed to your repository.
4. **Your code should have output for all assignment questions unless otherwise stated**. If any of the assignment questions asks you to explain your outputs, please use markdown cells to clearly explain them.
5. As a backup plan, also upload a PDF copy of your assignment notebook to GitHub. This is useful especially when GitHub cannot render some images or outputs.
6. During the code review, TAs will go over your answer submissions to assess your understanding of the concepts relating to the assignment and the logic behind your codes. Please, ensure your codes are readable and well commented using python comments and markdown cells when necessary. Not have readable code may impact your code review score.
7. Please note the plagiarism policy in the syllabus.

1 Main Assignment

Please answer the questions below in the provided notebook named `Assignment_11.ipynb` using Google Colab. **Clearly give titles and labels to all plots.**

You are provided with two datasets: (1) visit counts to a website over the past 80 days, and (2) preprocessed document corpus for topic modeling. Some questions you might want to consider are:

- Given new observational data of website visits over time:
  - How would you design a generative Bayesian model for the observed data?
  - How would you verify from posterior samples whether your model is a good representation of the observed data?
- Given a large set of text documents on unknown topics:
  - How would you apply the Latent Dirichlet Allocation (LDA) topic modeling algorithm?
  - How would you interpret the topics generated by LDA to understand what is being covered in the documents?

Use the pre-loaded **website_visits_data** NumPy array and **topic_df** DataFrame for the questionss.

Q1. Bayesian Modeling for Multiple Switchpoints

(a) Plot a bar chart (not a histogram) of the count of website visits per day (same as lab). Comment on any trend you observe from the visualization (hint: think switchpoints).

(b) Use PyMC to write code for the following generative Bayesian model that has two switchpoints τ_1 and τ_2 and three Poisson mean parameters: λ_1 when the time index $idx < \tau_1$, λ_2 when $\tau_1 \leq idx < \tau_2$, and λ_3 when $idx \geq \tau_2$. Name each parameter of the PyMC model according to the **param_names** list provided. Include code to sample 1000 points from the posterior using a random seed of 45.

$$\tau_1, \tau_2 \sim \text{uniform between 0 and 79 (inclusive)}$$
$$\lambda_1, \lambda_2, \lambda_3 \sim \text{exponential with } \alpha \text{ set to the mean of the dataset}$$
$$\lambda_{12} = \text{choose } \lambda_1 \text{ or } \lambda_2 \text{ based on } \tau_1 \text{ and } idx$$
$$\lambda = \text{choose } \lambda_{12} \text{ or } \lambda_3 \text{ based on } \tau_2 \text{ and } idx$$
$$visits \sim Poisson(\lambda)$$

(c) Complete the **plot_posterior** function by writing the code to plot a histogram of posterior samples of a model parameter, and print the mean and 94% HDI of the posterior samples. Set the range of the x-axis from 0 to the maximum value of the posterior samples of the parameter. Answer the following questions: (i) Is the current model a good fit for the model (hint: refer to HDI ranges)? (ii) Justify whether or not the histograms of the posterior samples align with *your expectations* from the observed trends in (a)? (Hint: they shouldn't.)

(d) There is one modeling deficiency in the current generative Bayesian model definition, specifically related to the prior bounds of the switchpoints (note that one prior may condition on another – this is known as a *hyperprior* in Bayesian modeling). Identify what it is and propose a fix for it. Rewrite the PyMC model implementing the fix. Draw 1000 posterior samples from the updated models using a random seed of 45.

(e) Plot histograms of the posterior samples of the parameters of the revised model from (d). Again, set the range of the x-axis from 0 to the maximum value of the posterior samples of the parameter. Answer the same questions from (c) and whether these results are more sensible than (c)?

Q2. Latent Dirichlet Allocation (LDA) Topic Modeling

(a) Pre-process the tokenized documents using the pre-processing code in the lab. Remove words that appear in less than 2 documents or greater than 50% of the documents. Display the number of unique tokens before and after pre-processing. Hint: there are two minor changes you need to make to the pre-processing code given in the lab: (1) initialize the **texts** variable to the already pre-processed tokens (2) add and set the **no_above** argument of **dictionary.filter_extremes**.

(b) Train a gensim LDA model using 10 topics, 100 iterations, and a random state of 100. Keep the remaining hyperparameters the same as the model trained in the lab. Use pyLDAvis to visualize the topics.

(c) Pick 3 reasonably distinct topics from the visualization, and list the top 5 most salient words of each topic. To do this, try to choose topics that are non-overlapping in the inter-topic distance map. Provide a brief overall topic description for each topic and justify your choice.