Task 0 - Bayesian Inference

1. Task description

This is an **ungraded** dummy task to prepare you for the five graded tasks later in the semester. After this task, you should know how to:

- register for a task,
- form a team of up to three students,
- read the task and data description,
- download the provided files,
- make a submission with your predictions and source code,
- see how your submission scores with regards to the baselines and the other students, and
- hand in the task by both choosing which submission should be graded and writing an **individual** task description.

While this task is ungraded and the problem itself is easy, we strongly recommend that you complete it as we will **not extend the deadlines** for the graded tasks if you experience issues related to the points above.

Task

The task is to implement Bayesian inference in a simple setting. In particular, the setting is as follows. You are given a set of data points $X = \{x_1, ..., x_n\}$ which are sampled i.i.d. from one of the following three distributions:

- Normal distribution:
$$p_1(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2\,\sigma^2}x^2} \quad , \quad \sigma = \sqrt{2}$$
- Laplace distribution:
$$p_2(x) = \frac{1}{2b} e^{-\frac{1}{b}|x|} \quad , \quad b = 1$$
- Student's t-distribution:
$$p_3(x) = \frac{\Gamma\left((v+1)/2\right)}{\sqrt{v\pi}\;\;\Gamma\left(v/2\right)} \left(1 + \frac{x^2}{v}\right)^{-(v+1)/2} \quad , \quad v = 4$$

In 35 % of the cases, the dataset is drawn from the normal distribution, in 25 % of the cases from the Laplace distribution and in 40% of the cases from the Student's t-distribution. Let H_i denote the event that the data was sampled from p_i for $i = 1, 2, 3$. Your task is to implement a Bayes-optimal predictor that, given the dataset X, outputs the posterior probabilities $P(H_i \,|\, X)$ for $i = 1, 2, 3$.

Submission workflow

1. Install and start [Docker](). Understanding how Docker works and how to use it is beyond the scope of the project. Nevertheless, if you are interested, you could read about its [use cases]().
2. [Download handout]()
3. The handout contains the solution template `solution.py`. You should write your

code for calculating the log posterior probabilities right below the `# TODO: enter your code here` marker in the solution template.
4. You should use Python 3.8.5. You are free to use any other libraries that are not already imported in the solution template. Important: please make sure that you list these additional libraries together with their versions in the `requirements.txt` file provided in the handout.
5. Once you have implemented your solution, run the checker in Docker:
   - On Linux, run `bash runner.sh`. In some cases, you might need to enable Docker for your user if you see a Docker permission denied error.
   - On MacOS, run `bash runner.sh`. Docker might by default restrict how much memory your solution may use. If you encounter out-of-memory issues you can increase the limits as described in the Docker Desktop for Mac user manual. Running over the memory limit will result in docker writing "Killed" to the terminal. Note that some required Python packages do not support ARM-based MacBooks yet. On those machines, you can install a VM application such as UTM and emulate an *x86-based* OS such as Linux or Windows.
   - On Windows, open a PowerShell, change the directory to the handout folder, and run `docker build --tag task0 .; docker run --rm -v "$(pwd):/results" task0`.
6. If the checker fails, it will display an appropriate error message. If the check was successful, then a file called `results_check.byte` will be generated. You should upload this file together with your source code to the project server.
7. **HINT:** in order to avoid underflow, you should work with probabilities in the log space. You will also need calculate the model evidence, i.e., you will need to sum probabilities in the log space - for this, take a look at the LogSumExp trick and consider using its implementation from scipy.

Evaluation

We evaulate your posterior inference implementation on 50 random datasets (with different number of samples per dataset), sampled from the data generating process which is described above. In particular, we compute the Hellinger distance

$$H(P,Q) = \sqrt{\frac{1}{2}\sum_{i=1}^{3}(\sqrt{p_i} - \sqrt{q_i})^2}$$

between your posterior Q and the correct posterior P. The score of your submission is the average of $1 - H(P,Q)$ across the 50 datasets. If your implementation is correct, you should get a score close to 1.0. You pass this task with a score > 0.98.

Grading

This task is ungraded and will not count towards your project grade. However, in order to prepare you for the graded tasks, we provide you with the following information on how the subsequent tasks will be graded.

When handing in the task, you need to select which of your submissions will get graded and provide a short description of your approach. This has to be done **individually by each member** of the team. Your submisssion is graded as either **pass** or **fail**. A complete submission typically consists of the following **three components**:

- **Submission file**: The `results_check.byte` file generated by the `runner.sh` script which tries to execute your code and checks whether it fulfills the requirements of the task.

- Your **code** in form of a .py or .zip file. The source code must be runnable and able to reproduce your uploaded `results_check.byte` file.
- A **description** of your approach that is consistent with your code. If you do not hand in a description of your approach, you may obtain zero points regardless of how well your submission performs.

To pass the task, your submission needs to be complete and **outperform the baseline** in terms of PUBLIC score. Some tasks only have a single score on which you have to improve upon the baseline. Other tasks have a PUBLIC and PRIVATE score. The PRIVATE score determines your position on the server leaderboard but is irrelevant for grading.

Make sure that you properly hand in the task, otherwise you may obtain zero points for this task. If you successfully completed the hand-in, you should see the respective task on the overview page shaded in green.

Frequently asked questions

Which programming language am I supposed to use? What tools am I allowed to use?

You should implement your solutions in Python 3.8. You can use any publicly available code, but you should specify the source as a comment in your code.

Am I allowed to use methods that were not taught in the class?

Yes. Nevertheless, the baselines were designed to be solvable based on the material taught in the class up to the second week of each task.

In what format should I submit the code?

If you changed only the solution file, you can submit it alone. If you changed other files too, then you should submit all changed files in a zip of size max. 1 MB. You can assume that all files from the handout that you have not changed will be available to your code.

Will you check / run my code?

We will check your code and compare it with other submissions. If necessary, we will also run your code. Please make sure that your code is runnable and your results are reproducible (fix the random seeds, etc.). Provide a readme if necessary.

Should I include the handout data in the submission?

No. You can assume the data will be available under the same path as in the handout folder.

Can you help me solve the task? Can you give me a hint?

As the tasks are a graded part of the class, **we cannot help you solve them**. However, feel free to ask general questions about the course material during or after the exercise sessions.

Can you give me a deadline extension?

We do not grant any deadline extensions!

Can I post on Moodle as soon as have a question?

This is highly discouraged. Remember that collaboration with other teams is prohibited. Instead,

- Read the details of the task thoroughly.
- Review the frequently asked questions.
- If there is another team that solved the task, spend more time thinking.
- Discuss it with your team-mates.

When will I receive the project grades?

We will publish the project grades before the exam the latest.