TASK 0

Ungraded Dummy Task

You are in the group Banana consisting of umbacht (umbacht@student.ethz.ch (mailto://umbacht@student.ethz.ch)).

1. READ THE TASK DESCRIPTION

2. SUBMIT SOLUTIONS

3. HAND IN FINAL SOLUTION

1. TASK DESCRIPTION

This is an **ungraded** dummy task to prepare you for the four 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 $X = \{x_1, \ldots, x_n\}$ 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:

$$p_3(x) = \frac{\Gamma((v+1)/2)}{\sqrt{v\pi}\,\Gamma(v/2)} \left(1 + \frac{x^2}{v}\right)^{-(v+1)/2} \quad , \quad v = 4$$

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 (https://www.docker.com/get-started). 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 (https://www.docker.com/use-cases).
2. Download handout (/static/task0_handout_5slq29.zip)
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. Tasks can be by passed by changing only sections marked TODO.
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 (https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-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 (https://docs.docker.com/desktop/mac/). 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 (https://mac.getutm.app/) 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 (https://en.wikipedia.org/wiki/LogSumExp) and consider using its implementation from scipy (https://docs.scipy.org/doc/scipy/reference/generated/scipy.special.logsumexp.html?highlight=logsumexp#scipy.special.logsumexp).

EVALUATION

We evaulate your posterior inference implementation on 100 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.

Some tasks have a public and private score. This task has **only a public score**. Your algorithm will make predictions on a held out test set, or (for tasks 3 and 4) obtain a single score for its interactions with the environment. 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. We will then compare your selected submission to our baseline. This project task is graded with either **pass (6.0) or fail (2.0)**. To pass the project, you need to achieve a better score than

the baseline. In addition, for the pass/fail decision, we consider the code and the description of your solution that you submitted. We emphasize that the public score leaderboard is just for fun: the scores of other teams will not effect the baseline or your own grade. The following **non-binding** guidance provides you with an idea on what is expected to pass the project: If you hand in a properly-written description, your source code is runnable and reproduces your predictions, and your submission performs better than the baseline, you can expect to have passed the assignment.

⚠ Make sure that you properly hand in the task, otherwise you may obtain zero points for this task.

FREQUENTLY ASKED QUESTIONS

⦿ WHICH PROGRAMMING LANGUAGE AM I SUPPOSED TO USE? WHAT TOOLS AM I ALLOWED TO USE?

You are free to choose any programming language and use any software library. However, **we strongly encourage you to use Python**. You can use publicly available code, but you should specify the source as a comment in your code.

⦿ AM I ALLOWED TO USE MODELS 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?

You can submit it as a single file (main.py, etc.; you can compress multiple files into a .zip) having max. size of 1 MB. If you submit a zip, please make sure to name your main file as *main.py* (possibly with other extension corresponding to your chosen programming language).

⦿ WILL YOU CHECK / RUN MY CODE?

We will check your code and compare it with other submissions. We also reserve the right to run your code. Please make sure that your code is runnable and your predictions are reproducible (fix the random seeds, etc.). Provide a readme if necessary (e.g., for installing additional libraries).

⦿ SHOULD I INCLUDE THE DATA IN THE SUBMISSION?

No. You can assume the data will be available under the path that you specify in your code.

⦿ 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 PRIVATE SCORES? AND THE PROJECT GRADES?

We will publish the private scores, and corresponding grades before the at exam the latest.