1. TASK DESCRIPTION

TASK

In this task, you will use Bayesian optimization to tune the structural features of a drug candidate, which affects its absorption and distribution. The features should be optimized subject to a constraint on how difficulty the candidate is to synthesize. Let $x \in \mathcal{X}$ be a parameter that quantifies such structu features. We want to find a candidate with x that is 1) bioavailable enough to reach its intended target, and 2) easy to synthesize. We use logP as c objective - a coarse proxy for bioavailability. To this end, for a specific x, we simulate the candidate's corresponding logP as well as its synthetic acessibl score (SA), which is a proxy for how difficult it is to synthesize. Our goal is to find the structural features x^* that induce the highest possible logP wh satisfying a constraint on synthesizability.

More formally, let us denote with $f : \mathcal{X} \to [0, 1]$ the mapping from structural features to the candidate's corresponding bioavailibility (logP), which v assume is bounded. Given a candidate with x, we observe $y_f = f(x) + \varepsilon_f$, where $\varepsilon_f \sim \mathcal{N}(0, \sigma_f^2)$ is zero mean Gaussian i.i.d. noise. Moreover, we denote w $v : \mathcal{X} \to \mathbb{R}^+$ the mapping from structural features to the corresponding synthetic accessiblity of a candidate (SA). Similar to our objective logP, we observe noisy value of this synthesizability score, which we denote with $y_v = v(x) + \varepsilon_v$, with $\varepsilon_v \sim \mathcal{N}(0, \sigma_v^2)$ zero mean Gaussian i.i.d. noise. The problem is formaliz as,

$$x^* \in \text{argmax}_{x \in \mathcal{X}, v(x) < \kappa} f(x),$$

where κ is the maximum tolerated synthetic accessiblity (SA). Compounds with higher SA are more difficult to synthesize. The objective of this problem do not have an analytical expression, is computationally expensive to evaluate, and is only accessible through noisy evaluations. Therefore, it is well suited Bayesian optimization (see [1] (https://pubs.rsc.org/en/content/articlepdf/2020/sc/c9sc04026a) for further reading on an example of constrained Bayesi optimization in drug discovery).

You need to solve the drug discovery problem presented above with Bayesian optimization. Let x_i be evaluated at the i^{th} iteration of the Bayesi optimization algorithm. While running the Bayesian optimization algorithm, you have a fixed budget for trying out hyperparameters for which t synthesizability constraint is violated, i.e. $v(x_i) \geq \kappa$. Furthermore, the final solution must satisfy $v(x) < \kappa$.

Remarks: In the motivating example above, x takes discrete values (for example, the number of functional groups would a natural number) and t objective and the constraint can be evaluated independently. However, to keep the problem simple, we let x be continuous and we evaluate f anc simultaneously. Moreover, to avoid unfair advantages due to differences in computational power, the physiochemical properties of the molecule c simulated, therefore, the time required for this step is platform-independent. This task does not have a private score.

Below, you can find the quantitative details of this problem.

- The domain is $\mathcal{X} = [0, 10]$.
- The noise perturbing the observation is Gaussian with standard deviation $\sigma_f = 0.15$ and $\sigma_v = 0.0001$ for logP and SA, respectively.
- The mapping f can be effectively modeled with a Matérn (https://en.wikipedia.org/wiki/Mat%C3%A9rn_covariance_function) with smoothne parameter $\nu = 2.5$ or a RBF (https://en.wikipedia.org/wiki/Radial_basis_function_kernel) kernel with variance 0.5, lengthscale 10, 1 or 0.5. To achie the best result, we recommend tuning the kernel and the lengthscale parameter.
- The mapping v can be effectively modeled with an additive kernel composed of a Linear kernel and a Maté (https://en.wikipedia.org/wiki/Mat%C3%A9rn_covariance_function) with smoothness parameter $\nu = 2.5$ or a R (https://en.wikipedia.org/wiki/Radial_basis_function_kernel) kernel with variance $\sqrt{2}$, lengthscale 10, 1 or 0.5. And the prior mean should be 4. achieve the best result, we recommend tuning the kernel and the lengthscale parameter.
- The maximum tolerated SA is $\kappa = 4$.

We specify the following three baselines. Passing the EASY baseline constitutes as passing grade.

- EASY = 0.6
- MEDIUM = 0.685
- HARD = 0.785

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 proje Nevertheless, if you are interested, you could read about Docker's use cases (https://www.docker.com/use-cases).
2. Download handout (/static/task3_handout.zip)

3. The handout contains the solution template `solution.py`. You need to implement all sections marked with `# TODO: ...`. You are free to implement t `plot` method for visualization. Please make sure that your implementation is entirely self-contained within the `BO_algo` class and preserves t existing names and signatures of all methods. However, you are free to introduce new methods and attributes. Note that the `main()` method in t solution template is *for illustrative purposes only* and is *completely ignored* by the checker! You *may play around* with the toy example provided in t `solution` file.
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: if you do u additional libraries, please make sure that you list all additional libraries together with their versions in the Dockerfile 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/linu postinstall/#manage-docker-as-a-non-root-user) if you see a Docker permission denied error. Please also first refer to the provided guide *I permission-errors.md* or this thread (https://moodle-app2.let.ethz.ch/mod/moodleoverflow/discussion.php?d=5429#p14778) before posti questions to the forum.
   - On MacOS, run `bash runner.sh`. Docker might by default restrict how much memory your solution may use. Running over the memory limit v result in docker writing "Killed" to the terminal. If you encounter out-of-memory issues you can increase the limits as described in the Dock Desktop for Mac user manual (https://docs.docker.com/desktop/mac/). Running over the memory limit will result in docker writing "Killed" to t terminal. You could alternatively run your solutions on the ETH euler cluster. Please follow the guide specified by *euler-guide.md* in the handout.
   - On Windows, open a PowerShell, change the directory to the handout folder, and run `docker build --tag task3 .; docker run --rm "$(pwd):/results" task3`.
   - If you are having trouble running your solution using docker locally, consider using the ETH Euler cluster to run your solution. Please follow t guide specified by *euler-guide.md* in the handout. The setup time of using the cluster means that this option is only worth doing if you really ca run your solution locally.
6. If the checker fails, it will display an appropriate error message. If the checker runs successfully, it will show you your PUBLIC score, tell you whether yc solution passes this task, and generate a `results_check.byte` file. The `results_check.byte` file constitutes your submission and needs to be upload to the project server along with your code and text description to pass this task.
7. You pass this task if your score is above the EASY baseline. More details are given in the Evaluation section below.
8. We limit submissions to the server to 40 per team, with at most 20 in 24 hours.

EVALUATION

We are interested in minimizing the normalized regret for not knowing the best hyperparameter value. Let $\tilde{x}_j$ be the final optimal solution suggested by t algorithm for a randomly intialized drug discovery task j. Let N_j be the number of unsafe evaluations that violate the constraint $\nu(x) < \kappa$. For our evaluati we do not penalize staying within the safe set around the initial safe point $x^{\blacktriangle}$, and use the local optimum $x^{\bullet}$. Define the normalized regret as follows:

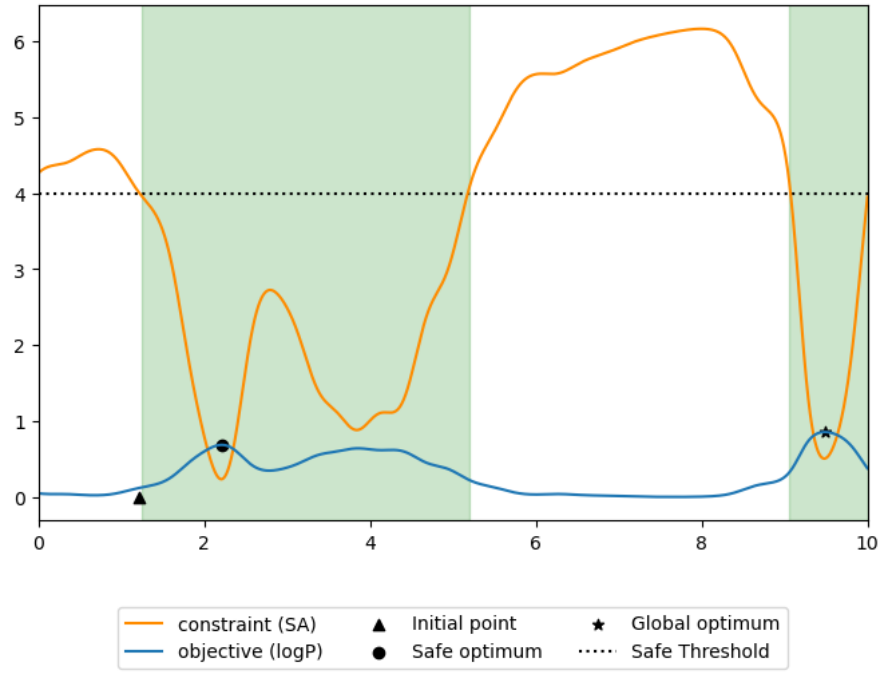
$$r_j = \max\left(\frac{f(x^{\bullet}) - f(\tilde{x}_j)}{f(x^{\bullet})}, 0\right)$$

To evaluate your algorithm, $j \in [100]$ random drug discovery tasks as the one described above are generated and your final score is computed as

$$S_j = 1 - 0.4 * r_j - 0.6 * h(3N_j - 3.25) - 0.15\mathbb{I}[|\tilde{x}_j - x^{\blacktriangle}| \leq 0.1]$$

where $h(\cdot)$ is the sigmoid function. The third term penalizes trivial solutions, i.e. returning the first safe point. You successfully pass the project if your me score over 100 random tasks is above the EASY baseline $\bar{S} \geq 0.6$. Note that $\bar{S}$ is a random variable, however, the baseline has been set taking into accou the variance of $\bar{S}$. In other words, you don't need to worry about this randomness.

The following is an example of the safe BO problem:



HINT

1. This task is designed such that a correct implementation of a standard Bayesian optimization algorithm (i.e. unaware of the SA constraint) might r be sufficient to pass. We strongly suggest taking into account the constraint on synthesizability. For example, by restricting the choice of x_i

parameters that are safe with high probability, you should obtain a better score. For further reading on the topic see (https://las.inf.ethz.ch/files/sui15icml-long.pdf), [3] (https://www.cs.princeton.edu/~rpa/pubs/gelbart2014constraints.pdf), (https://www.dynsyslab.org/wp-content/papercite-data/pdf/duivenvoorden-ifac17.pdf), and (https://proceedings.neurips.cc/paper/2019/file/4f398cb9d6bc79ae567298335b51ba8a-Paper.pdf).

2. To perform a constrained optimization over x, i.e.,

$$\max_x f(x) \text{ s.t. } v(x) < 0,$$

you may use a Lagrangian relaxation:

$$\max_x f(x) - \lambda \max(v(x), 0),$$

where λ is a hyperparameter used to penalize constraint violation.

GRADING

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 a 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 a 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 submissi to our baseline. This project task is graded with a grade between **6.0 and 2.0** based upon your public score. Failing to beat the easy baseline's score grar you a 2.0 grade. For scores better than the easy baseline but worse than the hard baseline, we will grant a grade between 4.0 to 6.0 depending on how clo you are to the hard vs easy baseline. We emphasise that this will likely not be a simple linear interpolation between the hard and easy baselines. In additi for the pass/fail decision, we consider the code and the description of your solution that you submitted. That is, your code should be runnable and reprodu your predictions (see faq) and you should include a short writeup of your submitted solution. We emphasize that the public score leaderboard is just for fu the scores of other teams will not effect the baseline or your own grade.

⚠ 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 publi available code that was not produced directly for the purposes of this course, 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 mentioned in the project description or taught in the class up to t 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 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 a 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 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 exam at the latest.

REFERENCES

[1] Ryan-Rhys Griffiths, José Miguel Hernández-Lobato "Constrained Bayesian optimization for Automatic Chemical Design using Variational Autoencode Chemical Science. 2019. (https://pubs.rsc.org/en/content/articlepdf/2020/sc/c9sc04026a)
[2] Yanan Sui, Alkis Gotovos, Joel W. Burdick, and Andreas Krause. "Safe Exploration for Optimization with Gaussian Processes." In Proceedings of the 32 International Conference on Machine Learning. 2015. (https://las.inf.ethz.ch/files/sui15icml-long.pdf)
[3] Michael Gelbart, Jasper Snoek, Ryan P. Adams. "Bayesian Optimization with Unknown Constraints" Conference On Uncertainty In Artificial Intelligence. 20 (https://www.cs.princeton.edu/~rpa/pubs/gelbart2014constraints.pdf)
[4] Rikky R.P.R. Duivenvoorden, Felix Berkenkamp, Nicolas Carion, Andreas Krause, and Angela P. Schoellig. "Constrained Bayesian optimization with Parti

Swarms for Safe Adaptive Controller Tuning." In Proceedings of the International Federation of Automatic Control World Congress. 20
(https://www.dynsyslab.org/wp-content/papercite-data/pdf/duivenvoorden-ifac17.pdf)
[5] Turchetta, Matteo, Felix Berkenkamp, and Andreas Krause. "Safe exploration for interactive machine learning." Advances in Neural Information Processi
Systems 32 (2019). (https://proceedings.neurips.cc/paper/2019/file/4f398cb9d6bc79ae567298335b51ba8a-Paper.pdf)