☰ TASK 3

Hyperparameter tuning with Bayesian optimization

> You are in the group 👥 Geniva consisting of 👤 durnovv (durnovv@student.ethz.ch (mailto://durnovv@student.ethz.ch)), 👤 growlands (growlands@student.ethz.ch (mailto://growlands@student.ethz.ch)) and 👤 nbalke (nbalke@student.ethz.ch (mailto://nbalke@student.ethz.ch)).

📖 1. READ THE TASK DESCRIPTION

💻 2. SUBMIT SOLUTIONS

✉ 3. HAND IN FINAL SOLUTION

📖 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. These features should be optimized subject to a constraint on how difficult the candidate is to synthesize. Let $x \in \mathcal{X}$ be a parameter that quantifies such structural 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 our 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 acessiblity 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 while 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 bioavailability (logP), which we 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 with $v : \mathcal{X} \to \mathbb{R}^+$ the mapping from structural features to the corresponding synthetic accessibility of a candidate (SA). Similar to our objective logP, we observe a 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 formalized as,

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

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

You need to solve the drug discovery problem presented above with Bayesian optimization. Let x_i be evaluated at the ith iteration of the Bayesian optimization algorithm. While running the Bayesian optimization algorithm, you have a fixed budget for trying out hyperparameters for which the 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 be a natural number) and the objective and the constraint can be evaluated independently. However, to keep the problem simple, we let x be continuous and we evaluate f and v simultaneously. Moreover, to avoid unfair advantages due to differences in computational power, the physiochemical properties of the molecule are 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 smoothness 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 achieve 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 Matérn (https://en.wikipedia.org/wiki/Mat%C3%A9rn_covariance_function) with smoothness parameter $\nu = 2.5$ or a RBF (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. To achieve the best result, we recommend tuning the kernel and the lengthscale parameter.
- The maximum tolerated SA is $\kappa = 4$.

We specify the following baseline, which needs to be beaten to achieve a passing grade: 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 project. 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 the `plot` method for

visualization. Please make sure that your implementation is entirely self-contained within the `BO_algo` class and preserves the existing names and signatures of all methods. However, you are free to introduce new methods and attributes. Note that the `main()` method in the solution template is *for illustrative purposes only* and is *completely ignored* by the checker! You *may play around* with the toy example provided in the `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 use 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/linux-postinstall/#manage-docker-as-a-non-root-user) if you see a Docker permission denied error. Please also first refer to the provided guide *fix-permission-errors.md* before posting 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 will result in docker writing "Killed" to the terminal. 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. 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 -v "$(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 the 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 can't 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 your solution passes this task, and generate a `results_check.byte` file. The `results_check.byte` file constitutes your submission and needs to be uploaded to the project server along with your code and a link to a one-minute video to pass this task.

7. You pass this task if your score is above the 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 the algorithm for a randomly initialized drug discovery task j. Let N_j be the number of unsafe evaluations that violate the constraint $\nu(x) < \kappa$. For our evaluation, 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

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

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 mean score over 100 random tasks is above the baseline $\bar{S} \ge 0.785$. Note that $\bar{S}$ is a random variable, however, the baseline has been set taking into account 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:
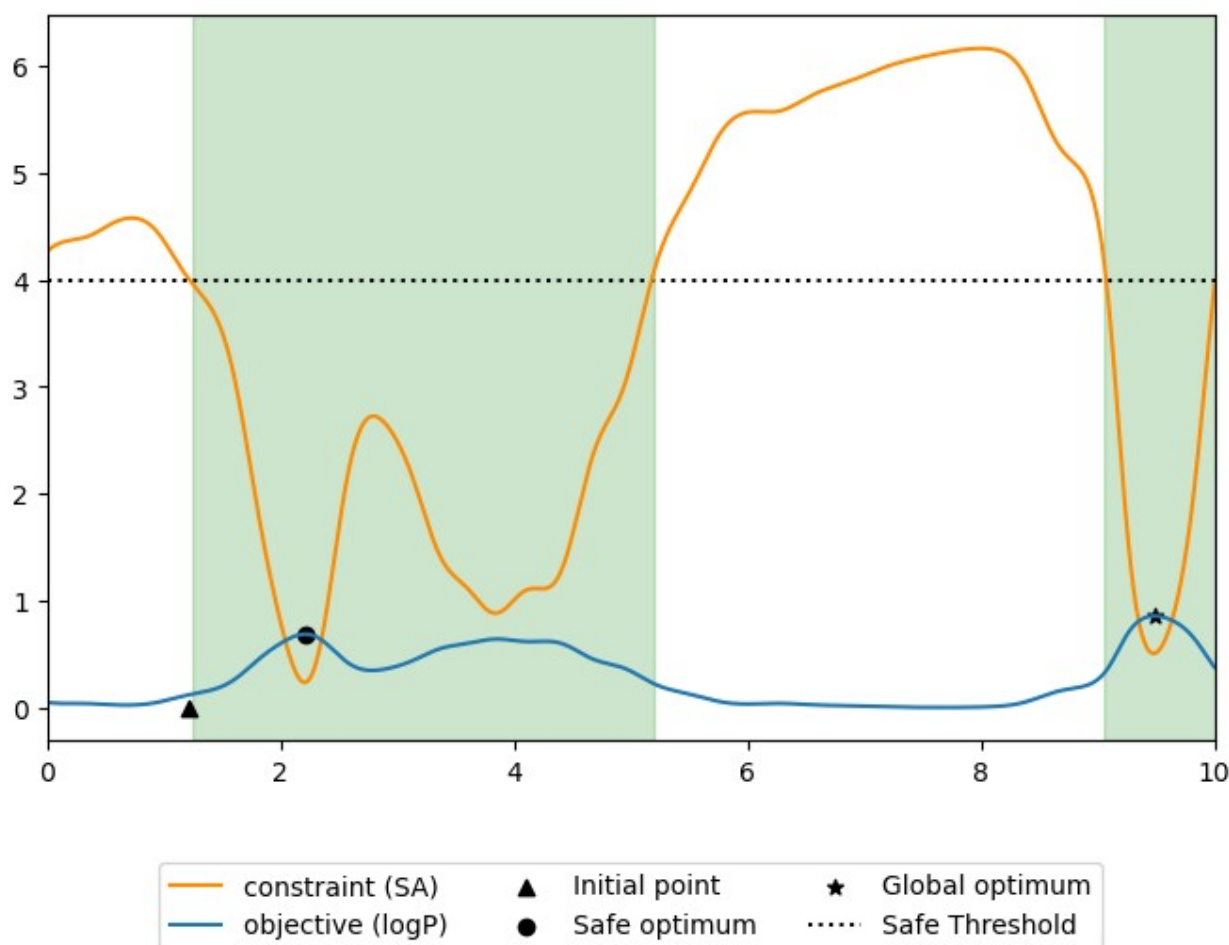
HINT

1. This task is designed such that a correct implementation of a standard Bayesian optimization algorithm (i.e. unaware of the SA constraint) might not be sufficient to pass. We strongly suggest taking into account the constraint on synthesizability. For

example, by restricting the choice of x_i to parameters that are safe with high probability, you should obtain a better score. For further reading on the topic see [2] (https://las.inf.ethz.ch/files/sui15icml-long.pdf), [3] (https://www.cs.princeton.edu/~rpa/pubs/gelbart2014constraints.pdf), [4] (https://www.dynsyslab.org/wp-content/papercite-data/pdf/duivenvoorden-ifac17.pdf), and [5] (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

This is a pass — fail project and you will not receive a grade for this task. You need to pass 3 / 4 projects in the PAI course to be eligible for taking the exam. 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 public link to a maximum one minute long video explaining your approach. Make sure that we can access the link till the first of February of the following year. Every student has to submit an individual 1 minute long video. This has to be done **individually by each member** of the team. For generating the video link, please use the ETH polybox (https://polybox.ethz.ch/index.php/login/) service, upload your video there and generate a shareable link (see this documentation (https://unlimited.ethz.ch/en/help/storage/manuals-and-documentation/manuals-polybox/share-with-polybox#Sharewithpolybox-SharingwithpeopleoutsideETH) for more detail). Submissions that are not handed-in, do not have a video that we can access till the first of February of the following year or have a video exceeding the one minute threshold will not be graded.

Note that handing in your task is mandatory: submitting a solution and a video, but not selecting the submission to hand it will result in failure.

We will compare your selected submission to our public baseline, which is visible on the public leaderboard. You pass this project if your submitted solution beats our public baseline. At the end of the semester, we will award one team with a certificate and prize for their performance on this task. The selection criteria are based on the team's performance on our public and private leaderboards and the creativity of their solution. The prize will be disclosed at the end of the semester. The private leaderboards are based on a separate test score on an undisclosed test set or (for task 3 and 4) environment. You only receive feedback about your performance on the public part in the form of the public score, while the private leaderboard remains secret. The purpose of this division is to prevent overfitting to the public score. Your model should generalize well to the private part of the test set. The creativity of your solution will be evaluated by our TAs based on your video submission.

> ⚠ Make sure that you properly hand in the task, otherwise you will fail this task.

PLAGIARISM

The use of **open-source** libraries is allowed and encouraged, except code that could reasonably be considered a solution to this or previous years' PAI projects. We do not allow copying the work of other groups / students outside the group (including work produced by students in previous versions of this course). **Publishing project solutions online is not allowed and use of solutions from previous years in any capacity is considered plagiarism**. Among the code, including those of previous years, we search for similar solutions in order to detect plagiarism. If we find strong evidence for plagiarism, we reserve the right to let the respective students or the entire group fail in the PAI 2025 course and take further disciplinary actions. Although not strictly forbidden, we discourage the use of Github Copilot or similar code/language generation tools for writing code. **We expect that if such tools are used, the tools used are stated in the video submission explaining the solution (see above). While it will have no effect on your grade or if a solution passes or fails, it may affect the awarding of prizes for best solutions**. We discourage these tools because we feel that the best way to understand the material is to write the code yourself referring to just the lecture material, source papers and documentation of any libraries used. The projects are designed in a way that you should be able to complete them in a reasonable amount of time using this approach. For the purposes of disclosing what generative AI tools you used to write code, we don't need you to disclose using e.g. basic code autocompletion such as those used in the default setup of Sublime Text 3. By submitting the solution, you agree to abide by the plagiarism guidelines of PAI2025.

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 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 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 (pass/fail) 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 exam at the latest.

REFERENCES

[1] Ryan-Rhys Griffiths, José Miguel Hernández-Lobato "Constrained Bayesian optimization for Automatic Chemical Design using Variational Autoencoders." 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 32nd 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. 2014. (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 Particle Swarms for Safe Adaptive Controller Tuning." In Proceedings of the International Federation of Automatic Control World Congress. 2017. (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 Processing Systems 32 (2019). (https://proceedings.neurips.cc/paper/2019/file/4f398cb9d6bc79ae567298335b51ba8a-Paper.pdf)