Scientific Machine Learning

(Summer semester 2021)

Bayesian optimization for an expensive function, dimensionality reduction of high-dimensional output of a dataset

Available online: 30th July 2021, 18:00

Submission due: 3rd September 2021, 23:59

1. Bayesian optimization for an expensive function:
   Your task is to optimize the settings of a basketball playing robot (see Fig. 1), which is supposed to throw a ball into a basket for a showcase at the 2020 Olympics in Tokyo. For this purpose you can select the angle and the initial velocity of the ball as input parameters and simulate the path of the ball through the air, while trying to throw it as close to the target as possible. To simplify things, we will only consider the 2D case and the start and target position of the ball will be set to the same elevation. The return value of the target function *simulator* will be the squared distance between the target and the landing position of the ball. Your goal is to efficiently build a machine learning model of this function f and find input settings f that minimize its return value y.

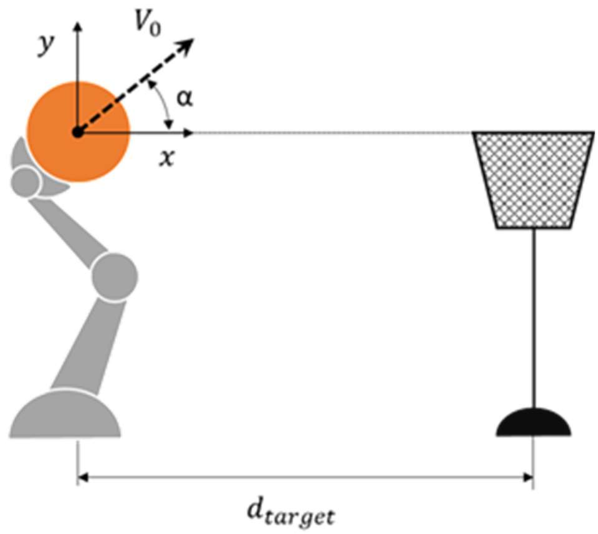

Figure 1. Problem sketch.

You are supposed to minimize the expensive function $f: \boldsymbol{x} \rightarrow y$ efficiently. Direct optimization with evaluating the function iteratively is expected to be expensive in terms of both the time and cost. For simplicity of this task, evaluating the function f for a given input is substituted for the above-mentioned simple simulation. This simulation to carry out $f: \boldsymbol{x} \rightarrow y$ is in the provided code whose module name is *simulator*.

Tasks:

(a) Carry out the Design of Experiments (DoE) to obtain an input dataset $\mathbf{X} = \{\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(N)}\}$, where N is the sample size and to be fixed as 30.

(b) Run the simulation code to obtain the corresponding output y to complete the dataset $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$.

(c) Make a Gaussian process (GP) model by using the dataset $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$, then visualize the predicted mean on the input space $\mathbf{X}$ as a 3D graph. Make two different GP models by the following conditions:

  i. Select an arbitrary set of values on the hyperparameter θ in the kernel function.

  ii. Optimize the values of the hyperparameter $\boldsymbol{\theta}$ by Maximum Likelihood Estimation (MLE).

  You need to modify the code to complete the learning and the prediction processes.

(d) Carry out the optimization (for minimizing the output value y) available in the code on the created GP model (in c.i). Then indicate the global minimum y_{min} with the corresponding input $\hat{\boldsymbol{x}}$.

(e) Improve the accuracy of the created GP model for the purpose of obtaining the global minimum y_{min} by the following two different approaches:

  A) by increasing the sample size N using the DoE to $N = 50$,

  B) by adding sample points by using the adaptive sampling techniques

    (A) 40 (the 30 initial DoE + 10 by the adaptive sampling

    (B) 50 (the 30 initial DoE + 20 by the adaptive sampling

    using the uncertainty information of the GP model, which comes down to the Bayesian optimization.

(f) Carry out the same optimization process as (d) on the improved GP model obtained by (e). Then compare and consider the results by (d) and (f).

(g) We specified to use GP models for the Bayesian optimization. Discuss why the GP models were selected for this purpose, which is to find the optimum solution of the output function, by taking pros and cons of other supervised learning techniques.

2. Dimensionality reduction of high-dimensional output of a dataset:

In this topic, a dataset $\mathbf{Y} = \{\boldsymbol{y}^{(1)}, \cdots, \boldsymbol{y}^{(N)}\}$, ($N = 160$ as the sample size) which represents a set of output data of an aerodynamic simulation and is provided by us. Each data point is composed of 192 scalar values, i.e. is a 192-dimensional vector as $\boldsymbol{y}^{(i)} = \left(y_1{}^{(i)}, \cdots, y_{192}{}^{(i)}\right)^{\mathrm{T}}$ where $i = 1, \cdots, N$. Figure 2 shows the airfoil configuration and the mesh to carry out the aerodynamic simulation. The number of the surface nodes of the mesh on the airfoil is 192. The pressure coefficient (C_p) is evaluated by the aerodynamic simulation, which yields the 192-dimensional vector output $\boldsymbol{y}^{(i)}$ at 160 different flow conditions. Figure 3 shows two examples of the 160 sample points. It can be observed that each sample contains 192 output values.

Each vector $\boldsymbol{y}^{(i)}$ represents the pressure distribution around the airfoil at some flow conditions $\boldsymbol{x}^{(i)} = ({M_\infty}^{(i)}, AoA^{(i)})^{\mathrm{T}}$ where $i = 1, \cdots, N$, which was generated by DoE. In the spaces 0.7 to 0.8 of the Mach number M_∞ and 0 to 3 degrees of the angle of attack AoA. The vectors $\boldsymbol{y}^{(i)}$ contain the values of C_p at the surface mesh nodes of the airfoil.

(a) Overview (b) Near field

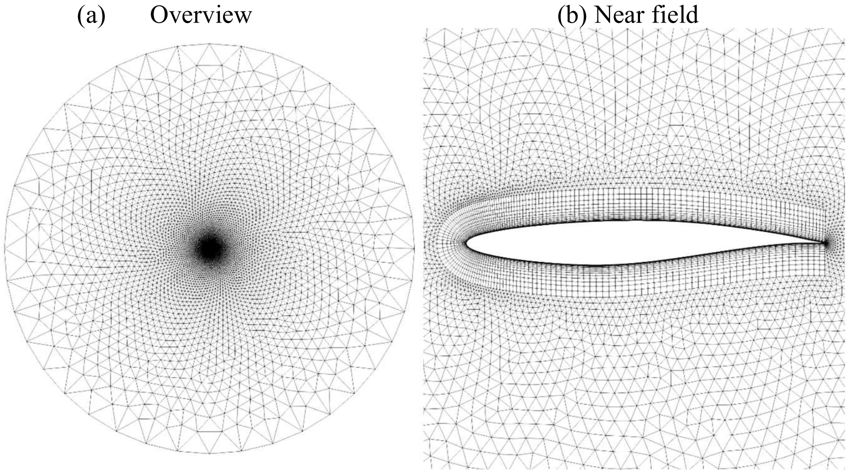

Figure 2. Mesh overview of the RAE2822 airfoil used for aerodynamics computations.

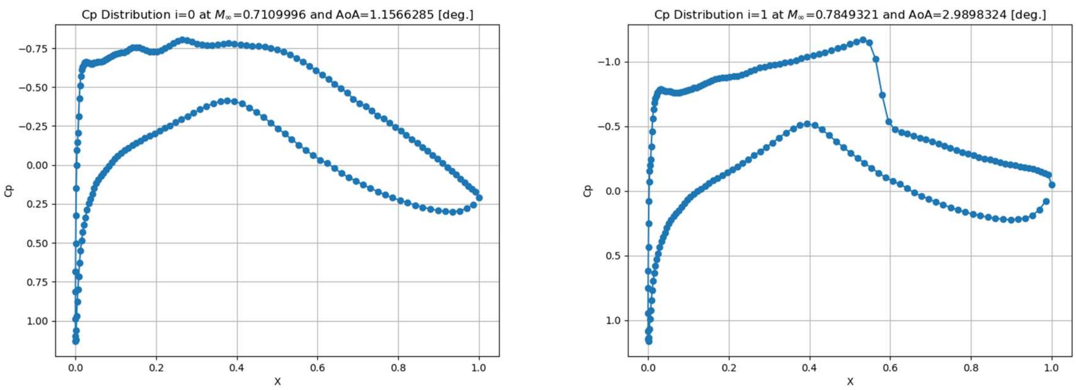

Figure 3. Two examples of the C_p distributions as two sample points. Each C_p distribution as one sample point is composed of 192-dimensional output.

Tasks:

(a) Carry out the principal component analysis (PCA/POD) on the given dataset. The PCA/POD can be carried out by using the eigenvalue decomposition of the covariance of the dataset.

  i. Check the eigenvalues and reduce the dimensionality from 192 to an arbitrary integer number.

  ii. Specify the new dataset $\mathbf{Y}'$ after the process of the dimensionality reduction by taking 99% of the total energy.

  iii. Compare the original C_p distributions given by $\mathbf{Y}$ and the new dataset $\mathbf{Y}'$. How much information has been lost? Pick up just one C_p distribution of $\mathbf{Y}$ for the comparison.

(b) We ask you to create a regression model for predicting new C_p distributions for a new flow condition. The regression model as a supervised learning technique needs a dataset $\mathcal{D} = (\mathbf{X}, \mathbf{Y}')$, where $\mathbf{X}$ is the given dataset of the flow conditions available with the corresponding dataset $\mathbf{Y}$.

  i. Create the regression model (any kinds of supervised learning techniques are fine) on the Training Dataset ("flow_conditions.csv", "surface_flow_sim_results.csv", $N = 160$).

  ii. Predict the Cp distribution for the flow conditions given by the test dataset ("flow_conditions_test.csv", N=80) and output the results as ".csv" file.

  iii. Discuss quantitatively the expected errors of your model.

Appendix A:

The covariance matrix of the GP model is $\sigma^2\mathbf{K}(\mathbf{x}, \boldsymbol{\theta})$, where $\mathbf{K}(\mathbf{x}, \boldsymbol{\theta})$ is the correlation matrix whose component is represented by the following kernel function:

$$k(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{\theta}) = exp\left(-\sum_{i=1}^{D} \theta_i \left\| x^{(i)} - x'^{(i)} \right\|^2\right) \tag{1}$$

where $\boldsymbol{\theta}$ is a vector as $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_D)^{\mathrm{T}}$, D is the dimensionality of the input parameter $\boldsymbol{x} = (x_1, \cdots, x_D)$. $\mathbf{x}$ is arbitrary set of the input $\boldsymbol{x}$. $\boldsymbol{\theta}$ and σ are treated as hyperparameters to be determined by the learning process as $\hat{\sigma}$ and $\hat{\boldsymbol{\theta}}$. Note that $\hat{\sigma}$ is analytically represented by $\boldsymbol{\theta}$ as $\hat{\sigma}(\boldsymbol{\theta})$:

$$\hat{\sigma}(\boldsymbol{\theta})^2 = \frac{\mathbf{Y}^{\mathrm{T}}\mathbf{K}(\mathbf{X}, \boldsymbol{\theta})^{-1}\mathbf{Y}}{N} \tag{2}$$

where $\mathbf{X}, \mathbf{Y}$ and N is the output data and sample size, respectively, as $\mathbf{X} = \left(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(N)}\right)^{\mathrm{T}}$, $\mathbf{Y} = \left(y^{(1)}, \cdots, y^{(N)}\right)^{\mathrm{T}}$. Eventually, only $\boldsymbol{\theta}$ needs to be numerically searched by the optimizer when evaluating the negative log likelihood-function:

$$E(\boldsymbol{\theta}) = \frac{N}{2} ln\, \hat{\sigma}(\boldsymbol{\theta})^2 + \frac{1}{2} ln|\mathbf{K}(\boldsymbol{\theta})| + C \tag{3}$$

where the fixed dataset $\mathbf{X}, \mathbf{Y}$ are omitted to be described for simplicity, e.g. $\mathbf{K}(\boldsymbol{\theta}) = \mathbf{K}(\mathbf{X}, \boldsymbol{\theta})$. Please review the lectures.

The predictive distribution $p(y|\boldsymbol{x}, \mathbf{X}, \mathbf{Y})$can be simply derived by taking the formula of the conditional Gaussian process as $p(y|\boldsymbol{x}, \mathbf{X}, \mathbf{Y}) = \mathcal{N}\left(y\middle|\mu_N(\boldsymbol{x}), {\sigma_N}^2(\boldsymbol{x})\right)$. When the covariance of the GP model is $\sigma^2\mathbf{K}(\boldsymbol{\theta})$, the mean $\mu_N(\boldsymbol{x})$ and the variance ${\sigma_N}^2(\boldsymbol{x})$ for arbitrary input $\boldsymbol{x}$ is represented as:

$$\mu_N(\boldsymbol{x}) = \boldsymbol{k}(\boldsymbol{x})^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{Y} \tag{4}$$
$${\sigma_N}^2(\boldsymbol{x}) = \sigma^2\left(1 - \boldsymbol{k}(\boldsymbol{x})^{\mathrm{T}}\mathbf{K}^{-1}\boldsymbol{k}(\boldsymbol{x})\right) \tag{5}$$

where the dataset $\mathbf{X}, \mathbf{Y}$ and the learned hyperparameter $\widehat{\boldsymbol{\theta}}$ are omitted to avoid the complexity of the description, e.g. $\mathbf{K}^{-1} = \mathbf{K}^{-1}\left(\mathbf{X}, \widehat{\boldsymbol{\theta}}\right)$ in Eqs. (4) and (5) since they are now all fixed.

Please try to avoid direct computations of the inverse $\mathbf{K}^{-1}$ and the determinant $|\mathbf{K}|$ appeared in the computations of the negative log likelihood-function $E(\boldsymbol{\theta})$ and the prediction. As presented in the lectures, the LU decomposition is the most promising approach to compute them. Please consider the common part in Eqs. (3) and (4), which is $\mathbf{K}^{-1}\mathbf{Y}$. $\mathbf{K}^{-1}\boldsymbol{k}(\boldsymbol{x})$ in Eq. (5) can be also computed by the LU decomposition. $|\mathbf{K}|$ can be computed as $|\mathbf{K}| = |\mathbf{L}||\mathbf{U}|$. Please specify the double precision to avoid numerical errors in these computations. The topics here are not in the machine learning category but in the linear algebra and the computer science. Please review the lectures.

Appendix B:

The so-called acquisition functions $a(\boldsymbol{x})$, which are composed of the uncertainty in Bayesian regression models in general and all the other information in the regression models, have some variations dependent on purposes. The new input $\boldsymbol{x}$ is actively determined by the acquisition function $a(\boldsymbol{x})$ in general:

$$\boldsymbol{x}_{new} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, a(\boldsymbol{x}) \tag{6}$$

There are various acquisition functions dependent on your purposes:

1. To improve the overall accuracy of the regression model (Using only the information of the uncertainty):

$$a(\boldsymbol{x}) = {\sigma_N}^2(\boldsymbol{x}) \tag{7}$$

2. To improve the accuracy of the model to predict the global minimum value (Using the uncertainty and the predicted mean and the current sample information)

$$a(\boldsymbol{x}) = EI(\boldsymbol{x}; \mu_N(\boldsymbol{x}), \sigma_N(\boldsymbol{x}), y_{min}) \tag{8a}$$

$$EI(\boldsymbol{x}; \mu_N(\boldsymbol{x}), \sigma_N(\boldsymbol{x}), y_{min}) = \begin{cases} (y_{min} - \mu_N(\boldsymbol{x}))\mathrm{cdf}(p_{std}(\boldsymbol{x})) + \sigma_N(\boldsymbol{x})\mathrm{pdf}(p_{std}(\boldsymbol{x})) & if \sigma_N(x) > 0 \\ 0 & if \sigma_N(x) = 0 \end{cases}$$

$$where, \; p_{std}(\boldsymbol{x}) = \frac{y_{min} - \mu_N(x)}{\sigma_N(x)}$$

(8b)

where y_{min} is the minimum of the current sample as $y_{min} = \min \mathbf{Y}$.

Appendix C:

The total energy to be considered in the dimensionality reduction using the PCA/POD is taking the ratio of the sum of the total eigenvalues and the sum of the arbitrary number of the eigenvalues in the descending order. Please review the Lecture 11.

The dimensionality reduction by the PCA/POD is a kind of linear transformation mapping. Replicating the predicted data $\left(\boldsymbol{x}^{(new)}, y'^{\;(new)}\right)$ obtained by a supervised learning technique on the space of $(\boldsymbol{x}, \boldsymbol{y}')$ on the original space $(\boldsymbol{x}, \boldsymbol{y})$ can be achieved by inversion of the linear transformation mapping.

Important notices:

- **Submission style**: The reports should be written in English, prepared in PowerPoint or pdf format and submitted together with your Python codes and csv files in a zipped file. Write your name and matriculation number on the file.
- **Deadline**: Submissions after the deadline will not be accepted in any case.
- **Submission portal**: Upload your files via StudIP.
- **Independent work policy**: The project, in all aspects such as developments, implementations and reporting, must be done individually! For that, print and fill in the form in the following page and **attach it** to your report when you submit.

Declaration of independent authorship

I hereby declare that the present work, which will be used for the topology optimization course evaluation, is solely and independently done by myself in all aspects, such as developments, code implementations, and writing of report. In addition, I confirm that I did not use any tools, materials or sources other than those explicitly specified.

Full name:

Date and place:

Signature: