Bayesian Optimisation with Dimension Scheduling Algorithm: Application to Biological Systems

Doniyor Ulmasov\textsuperscript{a}, Dr Caroline Baroukh\textsuperscript{b}, Dr Benoit Chachuat\textsuperscript{c}, Dr Marc Peter Deisenroth\textsuperscript{a}, and Dr Ruth Misener\textsuperscript{a},

\textsuperscript{a} Department of Computing, Imperial College London, UK | \textsuperscript{b} INRA, Laboratory of Environmental Biotechnology, France | \textsuperscript{c} Department of Chemical Engineering, Imperial College London, UK
Presentation Overview

- Algae Model
- Bayesian Optimisation
- Problem
- Bayesian Optimisation with Dimension Scheduling Algorithm
- Framework
- Questions?
Algae Models
Algae Models

- Learn about reactions of micro-algae metabolism to, e.g., light exposure. Measure lipids, carbohydrates, chlorophyll
- Simulations based on experimental data (Baroukh et al., 2014)
- The objective to minimize the squared error between the simulations and the experimental data

Figure: Simplified central carbon metabolic network of a unicellular photoautotrophic microalga (Baroukh et al., 2014)
Bayesian Optimisation
Bayesian Optimisation

- Bayesian Optimisation is a data-efficient asymptotically complete global black box optimisation technique. That uses all available data, prior and the evidence to calculate the posterior distribution.

- Also known as Kriging

Legged Locomotion (Calandra et al. (AMAI, 2015))
Pharmaceutical Processes (F. Boukouvala 2010)
CO2 Capture Technologies (M. M. Faruque 2012)
Integrated gasification combined cycle (Y. Lang 2011)
Objective: max \( f(x) \)

\[
\begin{align*}
f(x) &= \cos(x) + \sin(3 \times x) \\
\text{Expected Improvement Policy} \\
\text{Matérn 5/2 Kernel}
\end{align*}
\]
$f(x) = \cos(x) + \sin(3 \times x)$

- Expected Improvement Policy
- Matérn 5/2 Kernel
- High Variance during start
Objective: \( \max f(x) \)

- \( f(x) = \cos(x) + \sin(3 \times x) \)
- Expected Improvement Policy
- Matérn 5/2 Kernel
- High Variance during start
Objective: max $f(x)$

- $f(x) = \cos(x) + \sin(3 \times x)$
- Expected Improvement Policy
- Matérn 5/2 Kernel
- High Variance during start
Objective: max $f(x)$

- $f(x) = \cos(x) + \sin(3 \times x)$
- Expected Improvement Policy
- Matérn 5/2 Kernel
- High Variance during start
\[ f(x) = \cos(x) + \sin(3 \times x) \]

- Expected Improvement Policy
- Matérn 5/2 Kernel
- High Variance during start
- Over time the variance is reduced
\( f(x) = \cos(x) + \sin(3 \times x) \)

- Expected Improvement Policy
- Matérn 5/2 Kernel
- High Variance during start
- Over time the variance is reduced
Objective: $\max f(x)$

$\mathbf{f(x) = \cos(x) + \sin(3 \times x)}$

- Expected Improvement Policy
- Matérn 5/2 Kernel
- High Variance during start
- Over time the variance is reduced
Objective: $\max f(x)$

- $f(x) = \cos(x) + \sin(3 \times x)$
- Expected Improvement Policy
- Matérn 5/2 Kernel
- High Variance during start
- Over time the variance is reduced
\( f(x) = \cos(x) + \sin(3 \times x) \)

- Expected Improvement Policy
- Matérn 5/2 Kernel
- High Variance during start
- Over time the variance is reduced

**Objective:** \( \max f(x) \)
Doniyor Ulmasov | Bayesian Optimisation with Dimension Scheduling Algorithm | ESCAPE26 2016

- \( f(x) = \cos(x) + \sin(3 \times x) \)
- Expected Improvement Policy
- Matérn 5/2 Kernel
- High Variance during start
- Over time the variance is reduced
- As number of iterations increase we converge to a global optimum
Bayesian Optimization Algorithm

1. Choose several initial points $x$ and evaluate $f(x)$ or $g(x, \omega)$. The points can be randomly generated or used from data samples collected prior.

2. While the stopping criterion is not met:
   2a. Calculate the posterior from the GP on $f$ from the points observed.
   2b. Use an acquisition function to decide where to evaluate next.

3. Based on the most recent posterior distribution, decide on the point with the best estimated value or best observed value

   The stopping criteria can be arbitrary (e.g. number of iterations or till certain value is reached.)

(E. Brochu 2010)
Gaussian Process

- Gaussian Processes construct a probabilistic model of the function in a form of:
  \[ f(x) \sim \mathcal{GP}(m(x), k(x, x')). \]
- A GP is specified by a mean function, \( m \) and a covariance function, \( k \) (kernel).
- The kernels of a GP incorporate smoothness assumptions made of the data.
Problem
Iteration Times of the Algae Model with Bayesian Optimisation

- Iteration time increasing at $n^3$
- Impractical for our case where the simulation takes under a second
- At 2000\textsuperscript{th} iteration BO spends 1500 seconds to decide the next point of evaluation.

**Figure Long run experiment: iteration time in seconds with a fitted $n^3$ trendline**
Objective Value of the Algae Model with Bayesian Optimisation

- Objective Value did not improve further after a certain point.
- There is known optimum value not achieved during 9 day processing period.
- Stuck at local minimum.

![Graph showing Best Running Objective Value over iterations](figure.png)
Analysing the Call Stack

- Majority of the computational resources is used by the GP posterior calculations.
- Each call to the acquisition function requires calculations from the GP posterior.
- Posterior calculations are $O(n^3)$ operation
- As number of data points increase, the performance degrades due to the $O(n^3)$ operation.
Current and Alternative Solutions

- We attempted to achieve an optimal solution by trying out all the combinations of kernels and policies over the Algae models, none yielded satisfactory results.
- Additive Bayesian Optimisation (Kandasamy et al., 2015), which breaks the problem into multiple additive functions did not perform well either.
- Prior solution involved commercial software suite to optimise with manually added initial points.
- Depending on the initial points, the performance of the software suite varied.
- Global Optimisation took up to 24 hours to process.
Dimension Scheduling Algorithm
The Algorithm
Dimension Scheduling Algorithm

Initialize the GPs with initial data, and set the best objective value from the observations

Initialize with $n$ Random Samples:
\[
\{x_{n}, y_{n}\}_{n=1}^{N}
\]

Add the observations to all GPs

Set $y^b = \max f(x_n)$

$\hat{x}^b = \arg\max_{x_n} f(x_n)$

while termination condition is not met:

Select dimension set $Z$ from distribution $P$

If $y^b < y_{n+1}$ then:

$y^b = y_{n+1}$

$\hat{x}^b = (x_{n+1}^Z \cap \hat{x}^b)$

Solve

\[
x_{n+1}^Z = \arg\max a(GP_Z)
\]

Add $(y_{n+1}, \hat{x}_{n+1}^Z)$ to the GP

Evaluate

\[
y_{n+1} = f(x_{n+1}^Z \cap \hat{x}^b)
\]
DSA Illustration

\[
\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 = x^b = \arg\max f(x_n)
\]
DSA Illustration

\[ \theta_1, \theta_2, \theta_3, \theta_4 = x^* = \text{argmax } f(x_n) \]

Select \( GP_{23} \)
DSA Illustration

\[
\begin{align*}
\theta_1 & \quad \theta_2 & \quad \theta_3 & \quad \theta_4 \\
\Rightarrow & \quad \Rightarrow & \quad \Rightarrow & \quad \Rightarrow \\
\theta_2^* & \quad \theta_3^* & \quad \Rightarrow & \quad \Rightarrow \\
\text{Select } \text{GP}_{23} & \quad \Rightarrow & \quad \Rightarrow \\
x_{n+1}^{2,3} & = \arg\max a(\text{GP}_{23})
\end{align*}
\]
DSA Illustration

\[ \theta_1, \theta_2, \theta_3, \theta_4 \]

\[ x^b = \arg\max f(x_n) \]

Select \( GP_{23} \)

\[ x_{n+1}^{2,3} = \arg\max a(GP_{23}) \]

Evaluate the global objective function

\[ f(\theta_1, \theta_2^*, \theta_3^*, \theta_4) < f(\theta_1, \theta_2, \theta_3, \theta_4) \]
DSA Illustration
DSA Properties

- The training data is distributed across $\frac{d!}{k!(d-k)!}$ GPs.
- Each GP contains the data corresponding to its dimension, plus the initialisation data.
- Each new iteration select a new subset of the dimensions from a probability distribution, which reflects the importance of the corresponding parameter.
- Thus we reduce the computation times by keeping less data in each GP.
Comparison with Traditional BO

Experiment Setup:
- 10 different variants of Algae yield models.
- 3 with 10 dimensions, 3 with 11 dimensions, 4 with 12 dimensions
- Online probability distribution for the DSA
- Squared Exponential kernel
- Expected Improvement policy
- Only 500 iterations
- One set of initial sampled data per experiment
- DSA built on top of pybo, Bayesian Optimisation library.
  ➢ *Therefore same underlying code for the kernels, acquisitions functions and the Gaussian Processes.*
Results: Objective value

- Best achieved objective in four experiments using standard BO and DSA. Each experiment ran for 500 iterations.
- The lowest objective value achieved lets us compare the different Algae models.

<table>
<thead>
<tr>
<th>Model</th>
<th>d</th>
<th>BO:1</th>
<th>DSA:1</th>
<th>BO:2</th>
<th>DSA:2</th>
<th>BO:3</th>
<th>DSA:3</th>
<th>BO:4</th>
<th>DSA:4</th>
</tr>
</thead>
<tbody>
<tr>
<td>M19</td>
<td>10</td>
<td>54.62</td>
<td>32.32</td>
<td>52.66</td>
<td>47.20</td>
<td>39.35</td>
<td>30.46</td>
<td>41.70</td>
<td>24.48</td>
</tr>
<tr>
<td>M19o</td>
<td>10</td>
<td>77.78</td>
<td>66.09</td>
<td>114.78</td>
<td>354.81</td>
<td>78.64</td>
<td>133.36</td>
<td>80.05</td>
<td>69.54</td>
</tr>
<tr>
<td>M26</td>
<td>11</td>
<td>87.91</td>
<td>35.86</td>
<td>32.56</td>
<td>26.66</td>
<td>55.40</td>
<td>31.53</td>
<td>57.58</td>
<td>26.54</td>
</tr>
<tr>
<td>M29G</td>
<td>11</td>
<td>46.32</td>
<td>59.77</td>
<td>72.19</td>
<td>35.16</td>
<td>86.61</td>
<td>77.40</td>
<td>73.62</td>
<td>64.59</td>
</tr>
<tr>
<td>M29C</td>
<td>11</td>
<td>36.50</td>
<td>24.78</td>
<td>44.71</td>
<td>37.25</td>
<td>41.47</td>
<td>34.59</td>
<td>52.74</td>
<td>33.77</td>
</tr>
<tr>
<td>M31</td>
<td>11</td>
<td>39.24</td>
<td>24.37</td>
<td>60.39</td>
<td>29.75</td>
<td>49.52</td>
<td>26.26</td>
<td>44.18</td>
<td>48.53</td>
</tr>
<tr>
<td>M32C</td>
<td>12</td>
<td>43.33</td>
<td>33.14</td>
<td>52.19</td>
<td>31.15</td>
<td>50.62</td>
<td>28.02</td>
<td>51.13</td>
<td>26.18</td>
</tr>
<tr>
<td>M32G</td>
<td>12</td>
<td>42.51</td>
<td>27.12</td>
<td>47.55</td>
<td>26.78</td>
<td>52.38</td>
<td>31.45</td>
<td>53.21</td>
<td>23.54</td>
</tr>
<tr>
<td>M33</td>
<td>12</td>
<td>44.07</td>
<td>24.82</td>
<td>38.78</td>
<td>24.64</td>
<td>40.26</td>
<td>29.09</td>
<td>46.68</td>
<td>20.61</td>
</tr>
<tr>
<td>M35</td>
<td>12</td>
<td>50.16</td>
<td>25.86</td>
<td>57.72</td>
<td>22.86</td>
<td>57.06</td>
<td>28.97</td>
<td>57.00</td>
<td>40.11</td>
</tr>
</tbody>
</table>
Results: Objective value

- Best achieved objective in four experiments using standard BO and DSA. Each experiment ran for 500 iterations.
Average computation time of DSA and traditional BO for all models. DSA results in a substantial computational speed-up.
Further Work and Limitations

- Current state of the algorithm does not attempt to reduce the total regret accumulated throughout the function evaluations.

- The algorithm dimension selection is based on randomness, as a result there is no 100% guarantee on the performance of the algorithm in terms of the objective value. Further work is required to define a deterministic dimension selection algorithm.

- Global Gaussian Process potentially can be used to determine if the solution generated by the solver on reduced dimension should be evaluated or not to reduce the total regret.

- Parallelised and Distributed implementation of the Algorithm
  - Combined with Distributed Gaussian Processes for further Performance Boost (M. P. Deisenroth 2015)
The Framework
Bayesian Optimisation Framework GUI
Available at github.com/udoniyor/projectb

- Project B.(ayes) is a Graphical and Command Line interface on top of a Bayesian Optimisation framework pybo by Mathew Hoffman (2014).
- The framework allows any command line simulation to be optimised via BO or DSA techniques.
- The interface provides:
  - Graphical user interface to configure BO/DSA hyper-parameters
  - Visual feedback of the optimisation process
  - Easy sharing file format
  - Start and Pause and Continue of the optimisation process
The graphical user interface is divided into three phases.

Phase 1: Selection - user sets all the parameters for the model to be optimised.
- Users can set the parameters from a file (as specified prior)
- Provides the user with details on requirements, and performs validation as user enters the parameters
- Can set multiple parameter values

Phase 2: Observation – user can view current state of the optimisation
- Provides with a overview of the current state of the optimisation
- Shows information on current policy, kernel, iteration being optimised
- Can continue from previous file

Phase 3: Evaluation – provides with a complete overview of the optimization results
- User may query the GPs created during the optimisation process for the mean and variance of selected points

Available at github.com/udoniyor/projectb
Framework: [github.com/udoniyor/projectb](https://github.com/udoniyor/projectb)
Framework: github.com/udoniyor/projectb
ProjectB: Evaluation

Model

Command:
scilex -args C:/Users/doniyor/Desktop/Scilab_scripts_model/-f C:/Users/doniyor/Desktop/Scilab_scripts_model/Chlamy_Kliphus_19_reduced_resized_params.sce -nwm

Model Input:
C:/Users/doniyor/Desktop/Scilab_scripts_model/input/params_resized_19.txt

Model Output:
C:/Users/doniyor/Desktop/Scilab_scripts_model/output/objective_value.txt

Results

<table>
<thead>
<tr>
<th>#</th>
<th>Kernel</th>
<th>Policy</th>
<th>Iterations</th>
<th>Best Result</th>
<th>Time (s):</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SE</td>
<td>EI</td>
<td>45</td>
<td>-66.09366</td>
<td>129.47</td>
</tr>
</tbody>
</table>

Query

Console:
- Saving the data...
- Finished experiment
- Ended Bayesian Optimization
- Now you can interact with the model(s)

Save Console Logs  Output Directory  Exit

Framework: github.com/udoniyor/projectb
Bayesian Optimisation with Dimension Scheduling Algorithm

Doniyor Ulmasov\textsuperscript{a}, Dr Caroline Baroukh\textsuperscript{b}, Dr Benoit Chachuat\textsuperscript{c}, Dr Marc Peter Deisenroth\textsuperscript{a}, and Dr Ruth Misener\textsuperscript{a},

Questions?

Framework: https://github.com/udoniyor/projectb

Project Partners: 

Funding Sources: 

EPSRC
Engineering and Physical Sciences Research Council
EP/M028240/1