OptBayesExpt Overview

R. D. McMichael `rmcmichael@nist.gov`
National Institute of Standards and Technology
May 6, 2019

What is it for?

It's for making smarter measurements. This repository offers an implementation of optimized Bayesian experimental design (OBED). It is an adaptive strategy for controlling experiments that chooses measurement settings based on accumulated data. It is for cases with

- an experiment (possibly computational)
  - that yields measurements and uncertainty estimates,
  - and that can be controlled on the fly by one or more experimental settings, and
- a parametric model, i.e. an equation that relates unknown parameters and experimental settings to measurement predictions. If you would normally fit a function to the data to get fit parameters, that's the parametric model.

The real benefit of OBED methods is that they direct measurements toward settings that have the best chance of making our parameter estimates more precise. This feature is very helpful in situations where the measurements are expensive.

It is not for fitting existing data, but if you're thinking about making more measurements, you might be interested.

Note that *Bayesian optimization* addresses a different problem: finding a maximum or minimum of an unknown function. *Bayesian optimization* has proven to be useful for machine learning.

What does it do?

The optimal Bayesian experimental design algorithms play the role of an impatient experimenter who monitors data from a running experiment and changes the measurement settings in order to get better, more meaningful data. Note the two steps here. The first step, looking at the data, is really an act of extracting meaning from the numbers, learning something about the system from the existing measurements. The second step is using that knowledge to improve the measurement strategy.

In the "looking at the data" role, the OBED routines use a user-supplied model function and Bayesian inference to extract and update information about model parameters as new measurement data arrives. Then, in the decision making role, the OBED routines use the updated parameter knowledge to select settings that have the best chance of refining the parameters.

The most important role is the responsibility of the user. As delivered, the BayesOptExpt is ignorant of the world, and it's the user's responsibility to describe the world in terms of a model, reasonable parameters, and reasonable experimental settings. As with most computer programs, "the garbage in, garbage out" rule applies.

What's included?

Core files

The core files provide Python classes that implement optimal Bayesian experimental design.

- **obe.py** provides the `OptBayesExpt` class with methods for the learning and deciding steps described above. This class inherits all methods and data from both ProbDistFunc and ExptModel, and it is the only class that a user will need to interact with directly. Manual: [notebook][html]
- **probdistfunc.py** provides the `ProbDistFunc` class with methods for handling probability distribution functions. Its methods are used to define the parameters and their respective ranges, perform basic mathematical functions, to supply random draws, basic statistics of the distribution, and the distribution itself. Manual: [notebook][html]
- **exptmodel.py** provides the `ExptModel` class with methods to define the experimental settings and to evaluate a model function. The model function itself must be provided by the user, preferably by adding it to an instance of the BayesOptExpt class. Manual: [notebook][html]

TCP communications

These files add a TCP socket interface to OptBayesExpt so that non-python data acquisition code can interface with OptBayesExpt using JSON-formatted strings sent over TCP sockets.

- **obe_server.py** provides the `OBE_Server` class, which adds TCP communication to the `optBayesExpt` class. A mini-language uses label-value commands to configure an OptBayesExpt instance, to update probability distribution functions with new data, and to request efficient measurement settings.
- **obe_socket.py** provides the `Socket` class, which handles opening and closing TCP connections and the JSON string encoding and decoding for the OBE_Server.

What's next?

Tutorials

- **Docs/sequentialLorentzian.ipynb** [notebook] [html] provides a tutorial introduction to the OptBayesExpt software including setup and use of `OptBayesExpt` in a simulated "measurement" of a Lorentzian peak. The `ipynb` file is live code that can be run from inside the Jupyter Notebook environment. See also **sequentialLorentzian.py**.

- **Docs/manual.ipynb** [notebook][html] outlines the "if it works good, it is good" philosophy of the project and provides a tutorial-level description of the theory behind optimal Bayesian experimental design.

Demos

A brief discussion of these demos is included in the manual. [notebook] [html]

- **demoLorentzian.py** demonstrates how to incorporate a simple model into a BayesOptExpt. A simulation takes the place of a real measurement, supplying noisy "measurement results" that the BayesOptExpt uses to locate and measure a randomly placed Lorentzian peak. One setting, several model parameters. Also see **Docs/sequentialLorentzian.ipynb** [notebook][html] is a Jupyter Notebook with a step-by-step walk-through of the `sequentialLorentzian.py` code.

- **demoLorentzian2.py** uses the Lorentzian peak experimental model to demonstrate 10 × improved measurement efficiency of BayesOptExpt relative to an average & fit method.

- **pipulse.py** is a slightly more complicated demo featuring multiple (two!) experimental settings and the process for including pre-existing information, a *prior*.

- **slopeIntercept.py** demonstrates measurement of straight lines, $y = m x + B$. The demonstration presents options in the decision-making part of the code.
- **LabView/OBE_ODMR_demo.vi** demonstrates how OBETCP.py can be used from Labview. Like the sequentialLorentzian.py example, it's about finding and measuring a Lorentzian peak.

Legal stuff

Disclaimer

Certain commercial firms and trade names are identified in this document in order to specify the installation and usage procedures adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that related products are necessarily the best available for the purpose.

Terms of Use

OTHERWISE, WHETHER OR NOT INJURY WAS SUSTAINED BY PERSONS OR PROPERTY OR OTHERWISE, AND WHETHER OR NOT LOSS WAS SUSTAINED FROM, OR AROSE OUT OF THE RESULTS OF, OR USE OF, THE SOFTWARE OR SERVICES PROVIDED HEREUNDER.