GeoBO: A Python package for Multi-Objective Bayesian Optimisation and Joint Inversion in Geosciences

`GeoBO` is build upon a probabilistic framework using Gaussian Process (GP) priors to jointly solve multi-linear forward models. This software generates multi-output 3D cubes of geophysical properties (e.g. density, magnetic susceptibility, mineral concentrations) and their uncertainties from 2D survey data (e.g. magnetics and gravity) and any pre-existing drillcore measurements. The reconstructed 3D model is then used to query the next most promising measurement location given an expensive cost function (e.g. for drillcores). A ranked list of new measurements is proposed based on user-defined objectives as defined in the acquisition function which typically aims to optimize exploration (reducing global model uncertainty) and exploitation (focusing on highly promising regions) while minimizing costs.

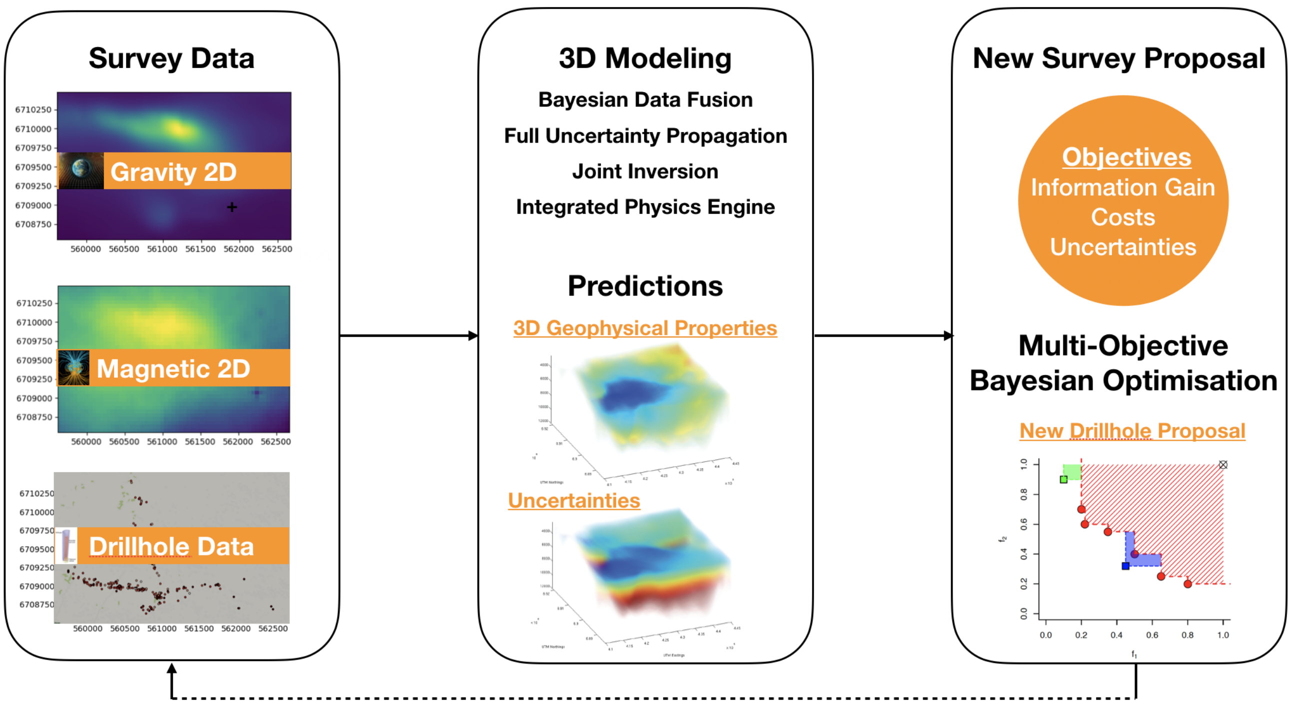

Figure 1: GeoBO Framework

Table of Contents

Definitions

Bayesian Optimisation (BO) is a powerful framework for finding the extrema of objective functions that are noisy, expensive to evaluate, do not have a closed-form (e.g. black-boxfunctions), or have no accessible derivatives. The model used for approximating the objective function is called surrogate model, which is typically based on a Gaussian Process models for tractability. Gaussian Processes define a prior over functions (typically given by a kernel function) and is used to propose points in the search space where sampling is likely to yield an improvement. The specific set of objectives for the improvement are defined in an acquisition function, which guides the search for a user-defined optimum.

Acquisition function

The key of BO is the acquisition function, which typically has to balance between a) exploration, i.e., querying points that maximise the information gain and minimize the uncertainty of a model b) exploitation, i.e. querying points that maximise the reward (e.g. concentrating search in the vicinity locations with high value such as minerals) c) minimize the number of samples given an expensive cost function for any new measurement.

Forward Models and Joint Inversion

In geology and geophysics, inversion problems occur whenever the goal is to reconstruct the geological conditions, i.e. the 3D distribution of physical rock properties, that give rise to a set of (2D) geophysical observations. Since the number of possible geological configurations is typically greater than the number of observational constraints, the problem is nearly always under-determined. Forward models transform the localized measurement of a remote sensor grid into a 3D representation of geophysical properties of a region. The most common geophysical linear forward model are gravity and magnetic forward models, which are computed using Li's tractable approximation. Joint inversion is simultaneously interpreting multiple (distinct) sensor measurements using a single model to provide a better constrained joint solution rather than taking individual solutions that only satisfy their aspect of data on their own.

Functionality

GeoBO's probabilistic framework includes all steps from prior selection, data fusion and inversion, to sensor optimisation and real world model output. The main functionalities of GeoBO are summarised in the following:

- Joint probabilistic inversion tool by solving simultaneously multi-linear forward models (e.g. gravity, magnetics) using cross-variances between geophysical properties (cross-variance terms can be specified by user)
- Output 1: Generation of cubes and computation of complete posterior distribution for all geophysical properties (described by their mean and variance value at each location (cubecell aka voxel).
- Output 2: Generation of ranked proposal list for new most promising drillcores based on global optimisation of acquisition function
- Templates for acquisition function to use in Bayesian Optimisation
- Flexible parameter settings for exploration-exploitation trade-off and inclusion of local 3D cost function in acquisition function

Other features are:

- Generation of simulated geophysical data with a choice of three different models
- Package includes geological survey/drillcore sample as well as synthetic data and functions for synthetic data generation
- Generation of 2D/3D visualisation plots of reconstructed cubes and survey data
- 3D Cube export in VTK format (for subsequent analysis, e.g., in Python or with ParaView)
- Options to include any pre-existing drillcore data
- Library of Gaussian Process (GP) kernels including sparse GP kernels
- Flexible settings for any cube geometry and resolution
- (Optional) Optimization of GP hyperparameters and cross-correlation coefficients via computation of marginal GP likelihood

Example outputs can be found in the directory `examples/results/`.

Installation And Requirements

Installation

To install GeoBO locally using setuptools:

```
python setup.py build
python setup.py install
```

or using pip:

```
pip3 install geobo
```

The installation can be tested by running the example with included synthetic data and default settings:

```
cd geobo/
python main.py tests/settings_example1.yaml
```

Requirements

- python >=3.6
- numpy
- matplotlib
- scikit_image
- scipy
- rasterio
- pandas
- pyvista
- skimage
- PyYAML

Documentation

Documentation conversion is generated using pandoc. The README markdown file can be converted to PDF:

```
pandoc -V geometry:margin=1.0in README.md -o README.pdf
```

A complete API documentation for all modules can be found here:

- `run_geobo.py`
- `inversion.py`
- `kernels.py`
- `cubeshow.py`
- `sensormodel.py`
- `simcube.py`
- `utils.py`

Usage and Settings

1) Change the main settings such as filenames and parameters in `settings.yaml`. These settings specify:

- directory, filenames, and geophysical drillcore properties
- the generated cube's geometry, size, and resolution
- Gaussian Process settings (lengthscale, input data uncertainty, correlation coefficients, kernel function)
- local Earth's magnetic field vector
- Bayesian Optimisation Settings (vertical/non-vertical drillcores, the exploration/exploitation and cost weighting)
- plotting settings
- optional generation of simulated data

2) Then run geobo

```
cd geobo/
python main.py settings.yaml
```

The main functions for the acquisition function can be found in `run_geobo.py`; visualisation functions and VTK export are defined in `cubeshow.py`; inversion functions are defined in `inversion.py`.

Examples and Tests

Synthetic Models

Synthetic geophysical models can be created by setting switching on `gen_simulation` in the settings yaml file. Three different models are so far implemented:

- two-cylindric dipping bodies (`modelname: 'cylinders'`)
- two layer model (`modelname: 'layers2'`)
- three layer model (`modelname: 'layers3'`) For each model a 3D voxel cube with geological structures is generated with density and magnetic susceptibility properties, plus the corresponding 2D gravity and magnetic remote sensor measurements. Other custom models can be included by adding a new model in function `create_syncube()` in `simcube.py`

Result examples of the synthetic models are stored in the subfolder `examples/testdata/synthetic/`. An example settings file is given in `settings_example1.yaml` and can be run by

```
cd geobo/
python main.py tests/settings_example1.yaml
```

The output results include the generated reconstructed density and magnetic susceptibility cubes and their corresponding uncertainty cubes, visualisations of original survey data and reconstructed properties, and list of new drillcore proposals. The output figure 'newdrill_proposals.png' shows the location of the already existing drills (black points), proposed new drill positions (white), and the best new drill location (red).

Drillcore Test Example

Another examples includes drillcore and gravity/magnetic survey data (`examples/testdata/sample/`). This example can be run with

```
cd geobo/
python main.py tests/settings_example2.yaml
```

and creates the reconstructed density and magnetic susceptibility cubes, uncertainty cubes

Literature

Sebastian Haan, Fabio Ramos, Dietmar Muller, "Multi-Objective Bayesian Optimisation and Joint Inversion for Active Sensor Fusion", GEOPHYSICS

Carl Edward Rasmussen and Christopher KI Williams, Gaussian process for machine learning, MIT press, 2006.

Li Yaoguo and Douglas W Oldenburg, "3d-inversion of gravity data," Geophysics, vol. 63, no. 1, pp. 109–119, 1998.

Arman Melkumyan and Fabio Ramos, "A sparse covariance function for exact gaussian process inference in large datasets.," in IJCAI, 2009, vol. 9, pp. 1936–1942

Armon Melkuyman and Fabio Ramos, "Multi-kernel gaussian processes," in IJCAI, 2011, vol. 22, p. 1408

Reid, A., O. Simon Timothy, E. V. Bonilla, L. McCalman, T. Rawling, and F. Ramos, 2013, Bayesian joint inversions for the exploration of earth resources.: IJCAI, 2877

Eric Brochu, Vlad M Cora, and Nando De Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," arXiv preprint arXiv:1012.2599, 2010.

Related Software

For the inversion part, GeoBO uses a direct inversion method via transformation of Gaussian Process priors, which enables joint inversion but is limited to linear forward models (e.g. gravity, magnetics, drillcores). For solving more complex non-linear forward models (e.g., seismic, or prior geological knowledge), the following bayesian inversion methods can potentially be applied to generate 3D geophysical surrogate models or to further refine GeoBo's 3D posterior model:

- hIPPYlib: an Extensible Software Framework for Large-scale Deterministic and Bayesian Inverse Problems. Publication Link; the software code is available at hippylib.github.io
- Obsidian: a flexible software platform for MCMC sampling of 3-D multi-modal geophysical models on distributed computing clusters. Publication Link; the code for version 0.1.2 of Obsidian is available at https://doi.org/10.5281/zenodo.2580422
- GemPy: open-source stochastic geological modeling and inversion; geoscientific model development. See gempy.org

Attribution and Acknowledgments

Acknowledgments are an important way for us to demonstrate the value we bring to your research. Your research outcomes are vital for ongoing funding of the Sydney Informatics Hub.

If you make use of this code for your research project, please include the following acknowledgment:

"This research was supported by the Sydney Informatics Hub, a Core Research Facility of the University of Sydney."

Project Contributors

Key project contributors to the GeoBO project are:

- Dr. Sebastian Haan (USYD, Sydney Informatics Hub): Expert in machine learning and physics, main contributor and software development of GeoBO.
- Prof. Fabian Ramos (USYD): Computational scientist and research expert in machine learning and bayesian computational techniques.
- Prof. Dietmar Muller (USYD, School of Geoscience): Research expert in geophysics and geoscience applications.
- Dr. Ben Mather (USYD, Sydney Informatics Hub/School of Geoscience): Computational Geophysicist, GeoBO testing.

License