

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

29 January 2020

## Summary

A critical decision process in geophysical data collection is how to efficiently combine a variety of sensor types and where to collect new observations. This may include drill-site placement or the layout of a particular geophysical survey, in particular if measurements are very costly or resources are limited. Bayesian Optimisation (BO) solves this decision making problem by finding global optimal solutions using a probabilistic framework given multiple measurements, prior knowledge, and model uncertainties. One of the major challenges for applying BO in geoscience still lies in constructing a 3D model from sparse, typically 2D sensor data that can be computationally efficiently evaluated and that transforms the combined data and uncertainties from multiple sensor measurements coherently into a posterior function approximating the true model; this is also known as an inversion problem.

**GeoBO** is a Python package that solves both, Multi-Objective Optimisation and Joint Inversion, within one probabilistic framework: from prior selection and input, data fusion and inversion, to the final sensor optimisation and real world model output. Fig.1 shows a graphical overview model of **GeoBO**; the two main process steps are summarized in the following.

First, **GeoBO** jointly solves multi-linear forward models of 2D survey data (e.g. magnetics and gravity) to 3D-geophysical properties using Gaussian Process kernels (see e.g. (Rasmussen and Williams 2006), (Melkumyan and Ramos 2009)). The joint inversion provides a better constrained combined solution of multiple, distinct sensor types, rather than taking their individual solutions. The key feature here is to consider all cross-variances between geophysical properties by solving simultaneously multiple forward models ((Melkumyan and Ramos 2011), (Reid et al. 2013)). Thus, this approach can take full advantage of additional sensor information to reveal properties of the reconstructed geophysical model at higher accuracy. By using non-parametric Gaussian Process priors, the solution of the inverse problem provides a complete posterior distribution for all geophysical properties which are described by their mean and variance value at each location (voxel).

In a second step, **GeoBO** allows the user to define objectives in an acquisition function ((Brochu, Cora, and De Freitas 2010)) which typically has to balance between a) exploration, i.e., querying points that maximise the information gain and minimize the uncertainty of a geophysical site 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), and c) minimize the number of samples given a cost function for any new measurement (e.g. costly drill-cores). The reconstructed 3D model output of the joint inversion model is then used to query for the next most promising point based on the

acquisition function, which guides the search for a user-defined optimum.

GeoBO allows applications to specify priors as additional options, such as a) the typical correlation lengthscale of the geological structure via GP hyperparameters, b) the gain/cost parameters in the BO acquisition function, and c) the correlation coefficients between different geophysical properties. This package includes forward models for gravity and magnetics surveys (Yaoguo and Oldenburg 1998), drillcores, and test-sets of geological models and data. While it has been tested on solving the problem of allocating iteratively new expensive drill-core samples, the same method can be applied to a wide range of optimisation and sensor fusion problems.

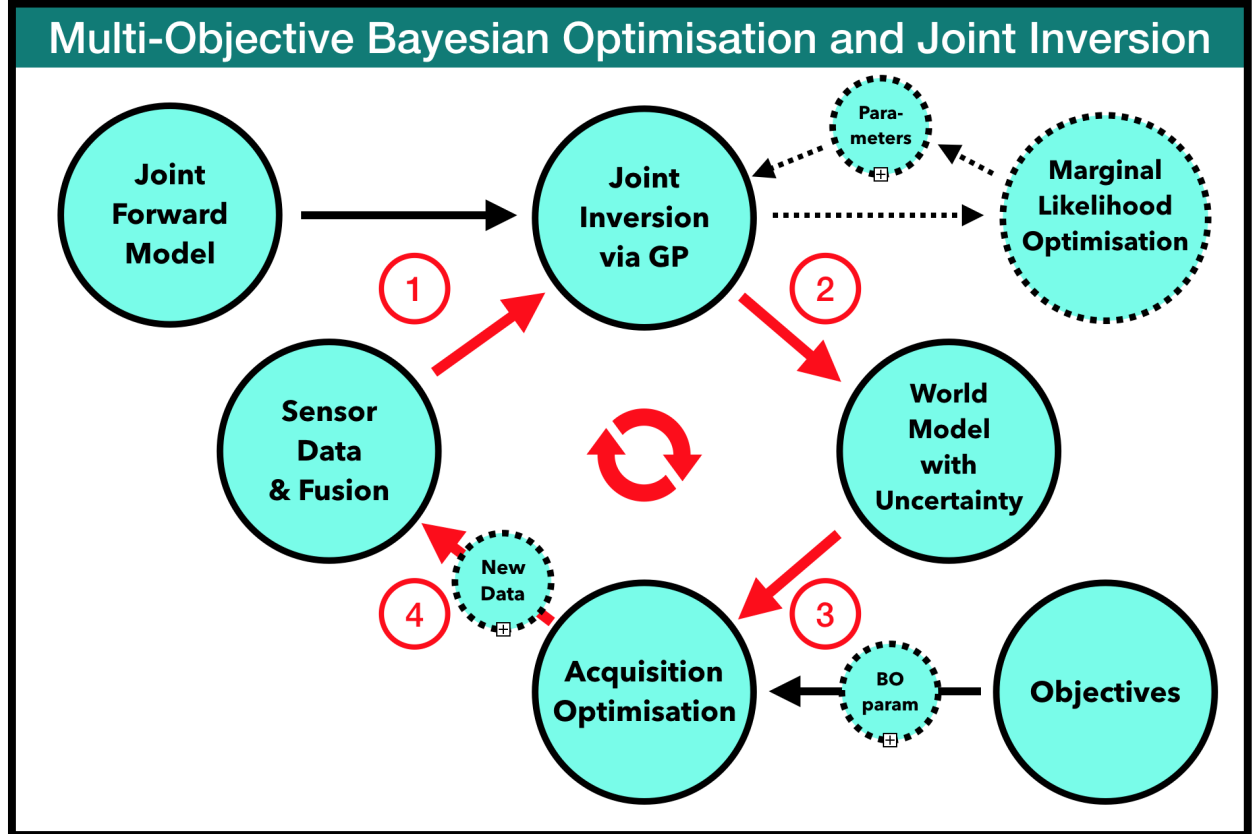


Figure 1: Overview of the probabilistic framework for multi-objective optimisation and joint inversion; the process stages are: 1) Joint inversion via Gaussian Processes based on multiple sensor data fusion and forward model, 2) Posterior model generation of 3D multi-geophysical properties. 3) Maximisation of acquisition function to allocate optimal new sensor location and type. 4) New acquired data is combined with existing data; process repeats until maximum number of iterations is achieved.

## Installation, Dependencies and Usage

GeoBO requires the following python packages: numpy, matplotlib, scikit\_image, scipy, rasterio, pandas, pyvista, skimage, PyYAML(for details see requirements.txt). The installation can be tested by running the script with included synthetic data and default settings.

To use GeoBO, change the main settings such as filenames and parameters in settings.yaml (see description there) and run python run\_geobo.py.

Core features of **GeoBO**:

- 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 use in settings.yaml)
- Computation of complete posterior distribution for all geophysical properties (described by their mean and variance value at each location/voxel)
- Generation of ranked proposal list for new most promising drillcores (optional: non-vertical drillcores) based on global optimisation of acquisition function (see settings.yaml)
- Templates for acquisition function to use in Bayesian Optimisation (see ‘futility’ functions in run\_geobo.py)
- Flexible parameter settings (settings.yaml) for exploration-exploitation trade-off and inclusion of local 3D cost function in acquisition function (see function ‘create\_costcube’ in run\_geobo.py)

Other features are:

- 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 (see plot settings in settings.yaml)
- 3D Cube export in VTK format (for subsequent analysis, e.g., with ParaView)
- Options to include any pre-existing drillcore data (see settings.yaml for feature names as well as format of sample drilldata)
- Library of Gaussian Process (GP) kernels including sparse GP kernels (kernels.py)
- Flexible settings for any cube geometry and resolution (cube length/width/height and voxel resolutions)
- Optional output: marginal GP likelihood for subsequent optimization of GP inversion process (not automated here)

## Acknowledgements

Sebastian Haan would like to acknowledge Dietmar Muller, Ben Mather, and Fabio Ramos for their valuable contributions. This research was supported by the Sydney Informatics Hub, a Core Research Facility of the University of Sydney.

## References

- Brochu, Eric, Vlad M Cora, and Nando De Freitas. 2010. “A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning.” *arXiv Preprint arXiv:1012.2599*.
- Melkumyan, Arman, and Fabio Ramos. 2009. “A Sparse Covariance Function for Exact Gaussian Process Inference in Large Datasets.” In *IJCAI*, 9:1936–42.
- Melkuyman, Armon, and Fabio Ramos. 2011. “Multi-Kernel Gaussian Processes.” In *IJCAI*, 22:1408. 1.
- Rasmussen, Carl Edward, and Christopher KI Williams. 2006. *Gaussian Process for Machine Learning*. MIT press.

Reid, Alistair, O’Callaghan Simon Timothy, Edwin V Bonilla, Lachlan McCalman, Tim Rawling, and Fabio Ramos. 2013. “Bayesian Joint Inversions for the Exploration of Earth Resources.” In *IJCAI*, 2877–84.

Yaoguo, Li, and Douglas W Oldenburg. 1998. “3D-Inversion of Gravity Data.” *Geophysics* 63 (1): 109–19.