Bayesian Inversion in Geophysics

Cedric Travelletti

Idiap Research Institute & University of Bern

December 13, 2019

Probing the interior of the Stromboli volcano from the outside

Consider the Stromboli island

Can we say anything about the internal structure of the volcano by only collecting data from the surface?

Use Gravimetry

Use Gravimetry

Use Gravimetric Measurements

- Go to several different locations on the surface of the volcano.
- Measure gravitational field there.
- Use it to infer the mass density distribution in the inside.

Mathematical Setup

Model density field as an unknown scalar function on the volcano interior.

Mathematical Setup

Model density field as an unknown scalar function on the volcano interior.

- Don't have access to it (not even pointwise evaluations).
- Can only measure gravity field.
  - At discrete set of locations only.
  - Depends linearly on mass distribution.
  - Can be modelled as measuring a linear operator applied to the unknown function.

Mathematical Setup

Model density field as an unknown scalar function on the volcano interior.

- Don't have access to it (not even pointwise evaluations).
- Can only measure gravity field.
  - At discrete set of locations only.
  - Depends linearly on mass distribution.
  - Can be modelled as measuring a linear operator applied to the unknown function.
- So called **Linear Inverse Problem**.

How can we solve it?

The traditional Approach...

Our volcano can be modelled as a bounded, closed region $D \subset \mathbb{R}^3$. The mass density field is then an unknown function

$$\mathfrak{u}_0 : D \to \mathbb{R}$$

The traditional Approach...

Our volcano can be modelled as a bounded, closed region $D \subset \mathbb{R}^3$. The mass density field is then an unknown function

$$\mathfrak{u}_0 : D \to \mathbb{R}$$

and we only have acces to a linear operator applied to the function

$$y_{obs} = G(\mathfrak{u}_0) \in \mathbb{R}^d$$

The traditional Approach...

Our volcano can be modelled as a bounded, closed region $D \subset \mathbb{R}^3$. The mass density field is then an unknown function

$$\mathfrak{u}_0 : D \to \mathbb{R}$$

and we only have acces to a linear operator applied to the function

$$y_{obs} = G(\mathfrak{u}_0) \in \mathbb{R}^d$$

Geophysicists solve the problem by minimizing some regularized misfit function on a discretized model space

$$u^* = \arg\min_{u \in \mathbb{R}^m} ||G(u) - y_{obs}|| + \lambda ||u - u_{ref}||$$

Here the models are discretized in $\mathbb{R}^m$ and $u_{ref} \in \mathbb{R}^m$ is some reference model.

...and its drawbacks

$$u^* = \underset{u \in \mathbb{R}^m}{\arg\min} ||G(u) - y_{obs}|| + \lambda ||u - u_{ref}||$$

- No procedure to choose regularization weight λ.
- Choice of reference model u_{ref} is arbitrary.
- No estimation of the uncertainty left in the solution.
- Only static design of experiments.
- Purely discrete, even if problem is intrinsically functional.

...and its drawbacks

$$u^* = \arg\min_{u \in \mathbb{R}^m} ||G(u) - y_{obs}|| + \lambda ||u - u_{ref}||$$

- No procedure to choose regularization weight λ.
- Choice of reference model u_{ref} is arbitrary.
- No estimation of the uncertainty left in the solution.
- Only static design of experiments.
- Purely discrete, even if problem is intrinsically functional.

This motivates us to use a probabilistic framework.

Probabilistic framework (sketch)

- Put a prior over possible solutions to the inverse problem (have to restrict to some class of functions).
- Get a posterior conditional on the observed data

Probabilistic framework (sketch)

- Put a prior over possible solutions to the inverse problem (have to restrict to some class of functions).
- Get a posterior conditional on the observed data

Prior → Observe data → Bayes theorem → Posterior

Probabilistic framework (sketch)

- Put a prior over possible solutions to the inverse problem (have to restrict to some class of functions).
- Get a posterior conditional on the observed data

Prior → Observe data → Bayes theorem → Posterior

- Principled treatment of regularization through hyperparam estimation
- Measure of residual uncertainty
- Access to full posterior distribution paves the way optimal experimental design.

Probabilistic framework allows to bring the latest advances in ML to Geophysics (and to the Inverse Problem community in general).

...and its drawbacks

These advantages come at a cost.

- Probabilistic framework is way heavier than the usual one.
- Typically, if computational cost of traditional model is $\mathcal{O}(n)$ for some n, then cost of probabilistic version is roughly $\mathcal{O}(n^2)$.

Goals of the Thesis

Use probabilistic inversion framework to bring the following to the inverse problem community:

Goals of the Thesis

Use probabilistic inversion framework to bring the following to the inverse problem community:

- **Set Estimation** What if we are not interested in the solution itself, but in regions where the solution has certain properties (excursion, steep variations, ...).

Goals of the Thesis

Use probabilistic inversion framework to bring the following to the inverse problem community:

- **Set Estimation** What if we are not interested in the solution itself, but in regions where the solution has certain properties (excursion, steep variations, ...).
- **UQ** Once we have an estimate for those region, can we say how confident we are in our estimate?

Goals of the Thesis

Use probabilistic inversion framework to bring the following to the inverse problem community:

- **Set Estimation** What if we are not interested in the solution itself, but in regions where the solution has certain properties (excursion, steep variations, ...).
- **UQ** Once we have an estimate for those region, can we say how confident we are in our estimate?
- **Experimental Design** Can we use this confidence measure to guide the data acquistion process.

Goals of the Thesis

Use probabilistic inversion framework to bring the following to the inverse problem community:

- **Set Estimation** What if we are not interested in the solution itself, but in regions where the solution has certain properties (excursion, steep variations, ...).
- **UQ** Once we have an estimate for those region, can we say how confident we are in our estimate?
- **Experimental Design** Can we use this confidence measure to guide the data acquistion process.
    - In particular: can we select measurements that will improve our estimate of the region of interest?.
- **Functional Inversion** Does a function space formulation provide better estimates?

Goals of the Thesis

Use probabilistic inversion framework to bring the following to the inverse problem community:

- **Set Estimation** What if we are not interested in the solution itself, but in regions where the solution has certain properties (excursion, steep variations, ...).
- **UQ** Once we have an estimate for those region, can we say how confident we are in our estimate?
- **Experimental Design** Can we use this confidence measure to guide the data acquistion process.
    - In particular: can we select measurements that will improve our estimate of the region of interest?.
- **Functional Inversion** Does a function space formulation provide better estimates?
- **Big Data** Extend the usual Bayesian Inversion techniques to models that are bigger than memory using recent advances in computing (goal discovered along the way).

Section 1

Inversion with Gaussian Process Priors

Remember we want to recover an unknown function

$$\mathfrak{u}_0 : D \to \mathbb{R}$$

from indirect measurements $y_{obs} = G(\mathfrak{u}_0)$.

Remember we want to recover an unknown function

$$\mathfrak{u}_0 : D \to \mathbb{R}$$

from indirect measurements $y_{obs} = G(\mathfrak{u}_0)$.

To do this in a Bayesian way, we need to be able to define a **prior on functions**.

Remember we want to recover an unknown function

$$\mathfrak{u}_0 : D \to \mathbb{R}$$

from indirect measurements $y_{obs} = G(\mathfrak{u}_0)$.

To do this in a Bayesian way, we need to be able to define a **prior on functions**.

Use Gaussian Processes

Gaussian Processes 101

Definition

Given a set D, a Gaussian Process on D is a real-valued stochastic process Z_x on D, such that for any finite number of points $x_1, ..., x_n \in D$, the distribution of $(Z_1, ..., Z_n)$ is gaussian.

Gaussian Processes 101

Definition

Given a set D, a Gaussian Process on D is a real-valued stochastic process Z_x on D, such that for any finite number of points $x_1, ..., x_n \in D$, the distribution of $(Z_1, ..., Z_n)$ is gaussian.

Such a process is entirely characterized by its mean and covariance function

$$\mu_0 : D \to \mathbb{R};\ x \mapsto \mathbb{E}[Z_x]$$
$$k : D \times D \to \mathbb{R};\ (x, y) \mapsto Cov[Z_x, Z_y]$$

Notation: $Z_x \sim Gp(\mu_0, k)$.

Gaussian Processes 101

Definition

Given a set D, a Gaussian Process on D is a real-valued stochastic process Z_x on D, such that for any finite number of points $x_1, ..., x_n \in D$, the distribution of $(Z_1, ..., Z_n)$ is gaussian.

Such a process is entirely characterized by its mean and covariance function

$$\mu_0 : D \to \mathbb{R};\ x \mapsto \mathbb{E}[Z_x]$$
$$k : D \times D \to \mathbb{R};\ (x, y) \mapsto Cov[Z_x, Z_y]$$

Notation: $Z_x \sim Gp(\mu_0, k)$.

Provides a neat way to define priors on functions.

Gaussian Processes 101 (contd.)

Conversely, given a function $\mu_0 : D \to \mathbb{R}$ and a positive definite function $k : D \times D \to \mathbb{R}$, we can define a Gaussian Process $Z_x \sim Gp(\mu_0, k)$.

Usually, k is chosen to belong a some class of kernels:

- Gaussian: $k(x, y) = \sigma_0^2 \exp\left(-\frac{||x-y||^2}{2\lambda^2}\right)$
- Matérn 3/2: $k(x, y) = \sigma^2 \left(1 + \sqrt{3}\frac{||x-y||}{\lambda}\right) \exp\left(-\sqrt{3}\frac{||x-y||}{\lambda}\right)$

The kernel determines the *regularity* of the realizations of the process.

Discrete Version

In practice, will only be interested in value of the process at finite number of points $x_1, ..., x_m$.

- Gaussian process $\mathcal{Z} \sim Gp(\mu_0, k)$ becomes gaussian random vector $Z = (\mathcal{Z}_{x_1}, ..., \mathcal{Z}_{x_m})$.
- Fully characterized by mean vector $\vec{\mu_0}$ and covariance matrix K

$$\vec{\mu_0} = (\mu_0(x_1), ..., \mu_0(x_m))$$
$$K = (K_{i,j})_{i,j=1,...,m}\,,\ K_{ij} = k(x_i, x_j)$$

From now on, drop vector arrow. Meaning of μ_0 will be clear from context.

Gaussian Process Conditioning for Inversion

Say we have the gaussian random vector $Z = (Z_{x_1}, ..., Z_{x_m})$ and we have an $m \times d$ measurement matrix G.

Gaussian Process Conditioning for Inversion

Say we have the gaussian random vector $Z = (Z_{x_1}, ..., Z_{x_m})$ and we have an $m \times d$ measurement matrix G.
Say we observe

$$y_{obs} = GZ + \eta$$

Where $\eta \sim \mathcal{N}(0, \Delta)$ independent of Z.

Gaussian Process Conditioning for Inversion

Say we have the gaussian random vector $Z = (Z_{x_1}, ..., Z_{x_m})$ and we have an $m \times d$ measurement matrix G.
Say we observe

$$y_{obs} = GZ + \eta$$

Where $\eta \sim \mathcal{N}(0, \Delta)$ independent of Z.

Then, the law of the vector, conditional on the data is gaussian with mean and covariance matrix

$$\tilde{\mu} = \mu_0 + KG^T\left(GKG^T + \Delta\right)^{-1}(y_{obs} - G\mu_0)$$

$$\tilde{K} = K - KG^T\left(GKG^T + \Delta\right)^{-1}GK$$

$$
\begin{aligned}
\tilde{\mu} &= \mu_0 + KG^T\left(GKG^T + \Delta\right)^{-1}(y_{obs} - G\mu_0)\\
\tilde{K} &= K - KG^T\left(GKG^T + \Delta\right)^{-1}GK
\end{aligned}
$$

The conditional mean $\tilde{\mu}$ can then be used as an approximation of the unkown u_0.

$$\tilde{\mu} = \mu_0 + KG^T\left(GKG^T + \Delta\right)^{-1}(y_{obs} - G\mu_0)$$
$$\tilde{K} = K - KG^T\left(GKG^T + \Delta\right)^{-1}GK$$

The conditional mean $\tilde{\mu}$ can then be used as an approximation of the unkown u_0.

Inversion performed by updating the mean and covariance function.

Section 2

Inverting the Stromboli

Recovering mass density from gravimetric measurements

- Measure (relative) vertical component of the gravitational field at discrete locations on the surface of the volcano.
- Solve the inverse problem to reconstruct mass density inside.
- Reconstruct density at 50m resolution (∼200'000 cells) from 543 measurements,
- Data provided by N. Linde's group [LBR+14].

Mathematical Formulation

- Discretize volcano in cubic cells.
- Only want to reconstruct the value of the density field at the cells centroid $x_1, ...x_m$.
- Unknown function is now a vector $u_0 \in \mathbb{R}^m$.
- We have access to the (vertical component of) the gravity field at locations $z_1, ..., z_d$ on the surface.

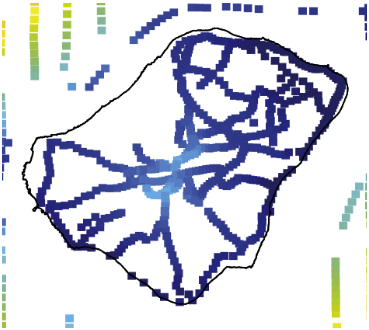


Figure: Location of measurement sites $z_1, ..., z_d$

Discretization mapping: $\mathfrak{u} \mapsto (\mathfrak{u}(x_1), ..., \mathfrak{u}(x_m))^t =: (u_1, ..., u_m)^t =: u$

Measurement Operator

We have to define the measurement operator G corresponding to measuring the gravitational field of the unknown function.

Proposition

For a given location on the boundary $z_0 \in \partial D$, the vertical component of the gravitational field generated by a mass distribution $u \in L^2_0(D)$ is given by:

$$g[\mathfrak{u}](z_0) = \gamma \int_D \mathfrak{u}(x)\phi(x, z_0)dx \tag{1}$$

Where γ is Newton's constant and we have the Green function:

$$\phi(x, z) = \frac{x_3 - z_3}{\|x - z\|^3},\ x = (x_1, x_2, x_3) \tag{2}$$

and the subscripts denote the components in the canonical basis of $\mathbb{R}^3$.

Measurement Operator (discrete case)

In the discrete case, we replace $\mathfrak{u}$ in the the integral by the piecewise constant approximation defined by its value at the cell centroids.

This give us a linear operator

$$\tilde{g} : \mathbb{R}^m \to \mathbb{R}$$

Stack the different measurements to obtain a matrix.

(discretized) Measurement Operator

Given measurement sites $z_1, ..., z_d \in \partial D$ scattered on the surface of the volcano, define the measurement matrix G to be the matrix representing the linear operator

$$\begin{aligned} &G : \mathbb{R}^m \to \mathbb{R}^d \\ &\mathbf{u} = (u_1, ..., u_m) \mapsto (\tilde{g}[\mathbf{u}](z_1), ..., \tilde{g}[\mathbf{u}](z_d)) \end{aligned}$$

We now have all the ingredients to *solve* our inverse problem. Recall the conditional mean and covariance are given by

$$\tilde{\mu} = \mu_0 + KG^T\left(GKG^T + \Delta\right)^{-1}\left(d_{obs} - G\mu_0\right)$$

$$\tilde{K} = K - KG^T\left(GKG^T + \Delta\right)^{-1}GK$$

Here the dimensions involved are $d \times m$ for G and $m \times m$ for K.

We now have all the ingredients to *solve* our inverse problem. Recall the conditional mean and covariance are given by

$$\tilde{\mu} = \mu_0 + KG^T\left(GKG^T + \Delta\right)^{-1}\left(d_{obs} - G\mu_0\right)$$
$$\tilde{K} = K - KG^T\left(GKG^T + \Delta\right)^{-1}GK$$

Here the dimensions involved are $d \times m$ for G and $m \times m$ for K. This seems innocent, but ...

The inversion grid we have to use contains $200'000$ cells.

We now have all the ingredients to *solve* our inverse problem. Recall the conditional mean and covariance are given by

$$\tilde{\mu} = \mu_0 + KG^T\left(GKG^T + \Delta\right)^{-1}\left(d_{obs} - G\mu_0\right)$$
$$\tilde{K} = K - KG^T\left(GKG^T + \Delta\right)^{-1}GK$$

Here the dimensions involved are $d \times m$ for G and $m \times m$ for K.
This seems innocent, but ...

The inversion grid we have to use contains $200'000$ cells.

The matrix K hence contains 40 **billion** elements.

We now have all the ingredients to *solve* our inverse problem. Recall the conditional mean and covariance are given by

$$\tilde{\mu} = \mu_0 + KG^T\left(GKG^T + \Delta\right)^{-1}\left(d_{obs} - G\mu_0\right)$$
$$\tilde{K} = K - KG^T\left(GKG^T + \Delta\right)^{-1}GK$$

Here the dimensions involved are $d \times m$ for G and $m \times m$ for K.
This seems innocent, but ...

The inversion grid we have to use contains $200'000$ cells.

The matrix K hence contains 40 **billion** elements.

This would take up 160 GB of memory on a computer

A closer look at the dimensions

The Stromboli inversion problem involves the following:

- 500 datapoints
- $2 \cdot 10^5$ inversion cells

Then the matrices and storage requirements (assuming single precision floating point numbers) at play are

$$\underbrace{K^{\#}}_{2\cdot 10^5\times 500 = 400MB} := \underbrace{K}_{2\cdot 10^5\times 2\cdot 10^5 = 160GB} \times \underbrace{G^T}_{2\cdot 10^5\times 500 = 400MB}$$

A closer look at the dimensions

The Stromboli inversion problem involves the following:

- 500 datapoints
- $2 \cdot 10^5$ inversion cells

Then the matrices and storage requirements (assuming single precision floating point numbers) at play are

$$\underbrace{K^{\#}}_{2\cdot 10^5 \times 500 = 400MB} := \underbrace{K}_{2\cdot 10^5 \times 2\cdot 10^5 = 160GB} \times \underbrace{G^T}_{2\cdot 10^5 \times 500 = 400MB}$$

But computation of posterior mean only involce $K^{\#}$.

Solving the Dimensionality Problem I

A few useful observations

- Covariance matrix K too big to be stored.
- But only needed in product with *projections to lower dimension*, e.g. $K^{\#} = KG^t$.
- Each element of K is defined *implicitly* by a formula $K_{ij} = k(x_i, x_j)$.

Solving the Dimensionality Problem I

A few useful observations

- Covariance matrix K too big to be stored.
- But only needed in product with *projections to lower dimension*, e.g. $K^{\#} = KG^t$.
- Each element of K is defined *implicitly* by a formula $K_{ij} = k(x_i, x_j)$.

- Can build elements of K on the fly.
- Only need to compute matrix-matrix products of K.
- Matrix-Matrix products easy to parallelize (line by line).

Solving the Dimensionality Problem II

Rember we want to compute $K^{\#} = KG^t$.

Solving the Dimensionality Problem II

Rember we want to compute $K^{\#} = KG^t$.

Subdivide model space in chunks:

$$(x_1, ..., x_m) = \left(X_1, ..., X_{N_{chunks}}\right), \; X_1 = \left(x_1, ..., x_{n_1}\right) \text{ and so on.}$$

Solving the Dimensionality Problem II

Rember we want to compute $K^{\#} = KG^t$.

Subdivide model space in chunks:

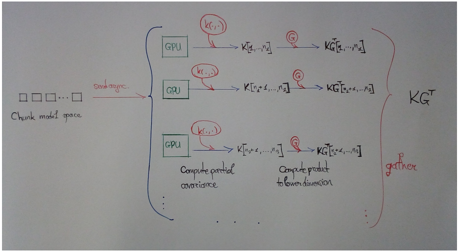
$$(x_1, ..., x_m) = (X_1, ..., X_{N_{chunks}}),\ X_1 = (x_1, ..., x_{n_1}) \text{ and so on.}$$

Algorithm

- *Distribute chunks among computational units.*
    - *Each unit builds corresponding lines of K (and all columns).*
    - *Each unit computes corresponding lines of the product KG^t.*
- *Gather results ans assemble complete product on main computational unit.*

Algorithm

- *Distribute chunks among computational units.*
  - *Each unit builds corresponding lines of K (and all columns).*
  - *Each unit computes corresponding lines of the product KG^t.*
- *Gather results ans assemble complete product on main computational unit.*



- Allows scaling in the model size.
- Specific to inverse problem setting (not valid for usual kriging).

Big number of datapoints has already been considered [WPG$^+$19], but big number of model points not treated in the litterature.

Inversion Result

Section 3

Hyperparameter Estimation

Hyperparameter Estimation

Results of preceding slide produced using some Gaussian Process prior.

Hyperparameter Estimation

Results of preceding slide produced using some Gaussian Process prior.

How can we choose the prior mean and covariance function?

Hyperparameter Estimation

Results of preceding slide produced using some Gaussian Process prior.

How can we choose the prior mean and covariance function?

- Not treated in traditional inversion schemes.
- Techniques developed here extends usual gaussian process methods to inverse setup.

Will restrict ourselve to constant prior mean

$$\mu_0 = m_0 \mathbb{1}_m,\ m_0 \in \mathbb{R},\ \mathbb{1}_m = (1, ..., 1)^t \in \mathbb{R}^m$$

And to stationary isotropic covariance kernels of the form

$$k(x, y) = \sigma_0^2 \tilde{k}\left(\frac{||x - y||}{\lambda_0}\right)$$

Where σ_0^2 is the prior variance, $\tilde{k}(0, 0) = 1$ and λ_0 is a lengthscale parameter.

Maximum Likelihood

Need to optimize 3 hyperparameters

- prior mean m_0
- prior variance σ_0^2
- lengthscale λ_0.

Maximum Likelihood

Need to optimize 3 hyperparameters

- prior mean m_0
- prior variance σ_0^2
- lengthscale λ_0.

Given observed data y, marginal data likelihood may be written as [RW06]:

$$-2\mathcal{L}(\mu_0, \lambda_0, \sigma_0; y) = n \log 2\pi - \log |R^{-1}| + \left(y - G\mu_0\right)^T R^{-1} \left(y - G\mu_0\right)$$

Where $R = R(\sigma_0^2, \lambda_0) = GK(\sigma_0^2, \lambda_0)G^T + \Delta$.

Maximum Likelihood

Need to optimize 3 hyperparameters

- prior mean m_0
- prior variance σ_0^2
- lengthscale λ_0.

Given observed data y, marginal data likelihood may be written as [RW06]:

$$-2\mathcal{L}(\mu_0, \lambda_0, \sigma_0; y) = n \log 2\pi - \log |R^{-1}| + \Big(y - G\mu_0\Big)^T R^{-1} \Big(y - G\mu_0\Big)$$

Where $R = R(\sigma_0^2, \lambda_0) = GK(\sigma_0^2, \lambda_0)G^T + \Delta$.

Choose hyperparameters that maximize the marginal data likelihood.

Maximizing the Likelihood in Practice

Optimal m_0 can be expressed analytically as function of the others

$$\hat{m}_0(\sigma_0^2, \lambda_0; y) = \left(I^T G^T R G I\right)^{-1} y^T R G I$$

Maximizing the Likelihood in Practice

Optimal m_0 can be expressed analytically as function of the others

$$\hat{m}_0(\sigma_0^2, \lambda_0; y) = \left(I^T G^T R G I\right)^{-1} y^T R G I$$

Remaining hyperparameters appear in (big) covariance matrix K, and hence in R.

Maximizing the Likelihood in Practice

Optimal m_0 can be expressed analytically as function of the others

$$\hat{m}_0(\sigma_0^2, \lambda_0; y) = \left(I^T G^T R G I\right)^{-1} y^T R G I$$

Remaining hyperparameters appear in (big) covariance matrix K, and hence in R.

- σ_0^2 may be factorized out of the matrix.
    - Can thus compute gradients for σ_0^2.
    - Implementation we use (PyTorch) gives us free gradients.
    - Optimize σ_0^2 by gradient descent.

Maximizing the Likelihood in Practice

Optimal m_0 can be expressed analytically as function of the others

$$\hat{m}_0(\sigma_0^2, \lambda_0; y) = \left(I^T G^T R G I\right)^{-1} y^T R G I$$

Remaining hyperparameters appear in (big) covariance matrix K, and hence in R.

- σ_0^2 may be factorized out of the matrix.
  - Can thus compute gradients for σ_0^2.
  - Implementation we use (PyTorch) gives us free gradients.
  - Optimize σ_0^2 by gradient descent.
- λ_0 cannot be factorized.
  - Appears in the full (not computable) covariance matrix.
  - Gradient-based approaches are hopeless
  - Fallback to brute-force search over discrete reasonable range.

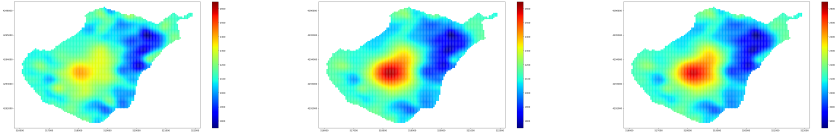
Inversion Results

	Hyperparameters				**Metrics**			
Kernel	λ	$\bar{\lambda}$	m_0	σ_0	$\mathcal{L}$	Train	Test	LOOCV
exponential	902	625	2046.6	197.1	-547.96	0.06799	12.7021	0.1651
Matérn 3/2	562	545	2112.5	221.4	-531.97	0.07180	12.6931	0.1648
Matérn 5/2	462	481	2133.8	221.6	-518.81	0.0705	12.6953	0.1730
Gaussian	342	403	2172.9	229.0	-478.31	0.0800	12.6959	0.1760

Here $\bar{\lambda}$ is the practical range, i.e. the distance at which the corresponding kernel function drops to half of its value at zero

Comparing Kernels

Can perform hyperparameter optimization for each class of kernel and compare posterior mean



Sea level slice of posterior mean [mGal] for different kernels. From left to right: squared exponential, Matérn 3/2, Matérn 5/2.

Section 4

Next Steps

Next Steps

- Fast inclusion of new datapoints.
- Functional formulation [Stu10].
    - Paves the way towards conditional simulations [**?**].
    - But need to solve eigenvalue problem over model space (big).
- Set estimation

Once the above are completed, we should have the ingredients to move towards

Sequential experimental design.

Next Steps: Fast Updating

Next Steps: Functional Formulation

Next Steps: Set Estimation

Thank You

Section 5

Set Estimation and Uncertainty Quantification on Sets

Set Estimation

We want to identify high density regions (excursion sets)

$$\Gamma^* = \{x \in X : u_0(x) \geq t_0\}$$

A simple plug-in estimate can be obtained using the posterior mean

$$\Gamma_{plug-in} = \{x \in X : \tilde{\mu}(x) \geq t_0\}.$$

Better estimates can be obtained by considering the full posterior distribution.

Azzimonti et al. (2016), Chevalier et al. (2013), Molchanov (2015)

Random Closed Sets (RACS)

The posterior distribution of the conditional field gives rise to a random closed set (RACS) Γ

$$\Gamma = \{x \in X : \tilde{Z}_x \geq t_0\}$$

Where $\tilde{Z}$ is any Gaussian Process whose law corresponds to the conditional law.
Can consider the pointwise probability to belong to the excursion set

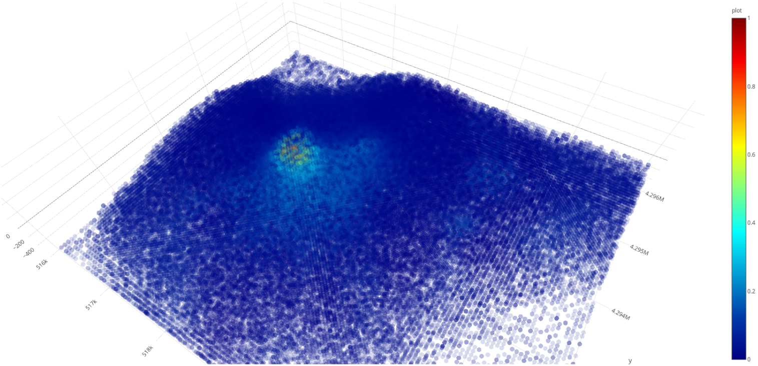
Coverage Function

$$p_\Gamma : X \to [0, 1]$$

$$p_\Gamma(x) := \mathbb{P}[x \in \Gamma]$$

Coverage function

Pointwise probability to belong the the excursion set above 2500 kg/m3.



Random Closed Sets Theory

The coverage function allows us to define a parametric family of set estimates for Γ

Vorob'ev Quantiles

$$Q_\alpha := \{x \in X : p_\Gamma \geq \alpha\}$$

The family of quantiles Q_α gives us a way to estimate Γ by controlling the (pointwise) probability α that the members of our estimate lie in Γ.

- Threshold α controls probability that points in our estimate lie in Γ.
- Can pick it such that the volume of the resulting set is equal to the expected volume of the excursion set

Vorob'ev Expectation

The Vorob'ev expectation is the quantile Q_{α_V} with threshold α_V chosen such that

$$\mu(Q_{\alpha_V}) = \mathbb{E}[\mu(\Gamma)]$$

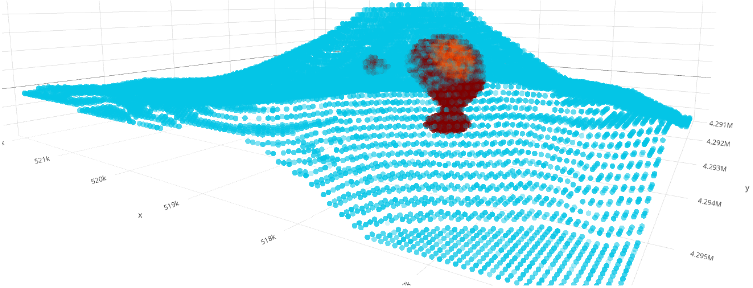
The expected volume of the excursion set can be computed using the coverage function

Robbins Theorem

$$\bar{V}_\Gamma := \mathbb{E}[\mu(\Gamma)] = \int_X p_\Gamma(x)dx$$

Vorob'ev Expectation

Plugin estimate and Vorob'ev expectation for excursion set above 2500.0 kg/m3.



Vorob'ev expectation: $\alpha = 0.22$, expected excursion measure $\mathbb{E}[\mu(\Gamma)] = 6678.16$ cells. Vorob'ev deviation: 7290.031 cells.

Vorob'ev Deviation

Can quantify uncertainty on an estimate Q for Γ by its Vorob'ev deviation

$$\mathcal{D}(Q) := \mathbb{E}[\mu(\Gamma \Delta Q)]$$

Theorem

$$\mathcal{D}(Q) = \int_Q \Big(1 - p_\Gamma(x)\Big) dx + \int_{Q^c} p_\Gamma(x) dx$$

This quantity will be the starting point for doing Bayesian optimal design, by selecting measurements that reduce the uncertainty.

Vorob'ev expectationt achieves the minimum deviation among all sets that have measure equal to the expected measure of Γ.

Theorem

The Vorob'ev expectation minimizes the deviation among closed set with volume $\bar{V}_\Gamma$.

$$Q_{\alpha_V} \in \arg\min\{\mathcal{D}(Q) | Q \subset X \text{ closed}, \ \mu(Q) = \bar{V}_\Gamma\}$$

Section 6

Functional Bayesian Approach to Inverse Problems

Solving the problem by discretization works, but it has some disadvantages

- Question of the discretization dependence
- Poor MCMC
- No regularity information. Hence wasted information for set estimation.

Thats why we want to take a functional approach

Ingredients

Functional approach to bayesian inversion was formalized by [Stu10]. Its main ingredients are:

- A separable Hilbert space $\mathcal{H}$ (**model space**).
- A Borel probability measure μ_0 on $\mathcal{H}$ (**prior**).
- A bounded linear operator $G : \mathcal{H} \to \mathbb{R}^d$ (**measurement operator**).
- Some data $y \in \mathbb{R}^d$ (call $\mathbb{R}^d$ the **data space**).

Then *Bayes Theorem* gives posterior

$$\frac{d\mu^y}{d\mu_0} = \frac{1}{Z} \exp\left(-\Phi(u; y)\right)$$

Rules of the Game

- There is an unknown function $u_0 \in \mathcal{H}$ which we would like to recover.
- We can only measure linear operators of the function, subject to some noise

$$y = G(u_0) + \eta$$

Where $\eta \sim \mathcal{N}(0, \Gamma)$ is a random vector on $\mathbb{R}^d$, independent of u_0 and y.

Bayesian View

In the Bayesian setting, we consider a random element $u \in \mathcal{H}$, distributed according to the prior μ_0 (technical details will follow, for the moment, just forget we are in a function space).

Then, conditional on the data $y = G(u) + \eta$, the random variable $u|y$ is distributed according to some measure μ^y which will serve as our posterior.

Theorem (Posterior Distribution)

Conditional on the data, the random variable $u|y$ is distributed according to a measure μ^y, whose Radon-Nikodym derivative is given by

$$\frac{d\mu^y}{d\mu_0}(u) = \frac{1}{Z}\exp\left(-\Phi(u; y)\right)$$

With $\Phi(u; y) = \frac{1}{2}||\Gamma^{-\frac{1}{2}}(y - G(u))||^2$ and Z a normalization constant.

Inversion Assumptions

In order for the posterior measure to exist and be well-defined, the measurement operator should satisfy some technical conditions:

Let the operator $\mathbf{G} \rightarrow \mathbb{R}^m$ satisfy:

i) For every $\epsilon > 0$ there exists $M = M(\epsilon) \in \mathbb{R}$ such that:

$$\forall u \in X : \ ||\Gamma^{-\frac{1}{2}}\mathbf{G}(u)||^2 \leq \exp\left(\epsilon ||u||^2_{\mathcal{H}} + M\right)$$

ii) For every $r > 0$ there exists $K = K(r) \in \mathbb{R}$ such that:

$$\forall u_1, u_2 \in B_r^X(0) : \ ||\Gamma^{-\frac{1}{2}}\left(\mathbf{G}(u_1) - \mathbf{G}(u_2)\right)|| \leq K||u_1 - u_2||_{\mathcal{H}}$$

Continuity in the Data

Theorem (Continuity in the data)

Provided the inversion assumptions are satisfied, the posterior measure μ^y is Lipschitz in the data on any bounded domain:

$$\forall r > 0 : \exists C_r > 0 : \forall y_1, y_2 \in B_r(0) \subset \mathcal{H}$$

$$d_{Hell}(\mu^{y_1}, \mu^{y_2}) \leq C_r ||\Gamma^{-\frac{1}{2}}(y_1 - y_2)||$$

Section 7

Model Selection

Cross-Validation

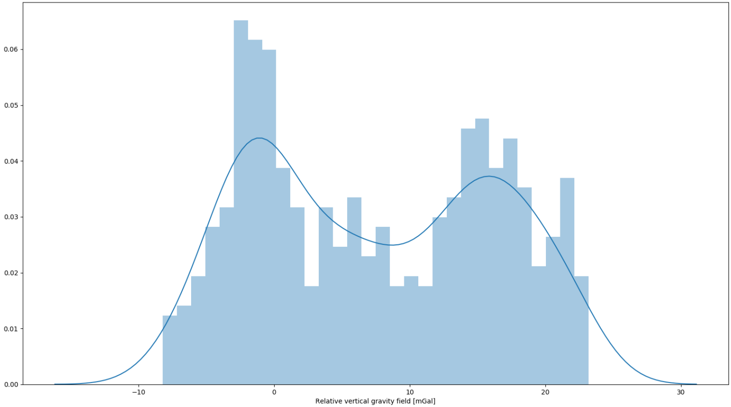
- Model selection by minimization of leave-one out root mean squared error.
- Remove one data point at a time and predict by conditioning on the remaining ones, average error over whole dataset.

Computation by brute force would be too expensive, fortunately, we have the fast-leave one out formula (adapted from Dubrule (1983)):

$$\hat{Z}_{x_{n+1}} = \mu_0(x_{n+1}) - \frac{1}{{}^{(n+1)}\mathbf{R}^{-1}_{n+1,n+1}} \sum_{i=1}^{n} {}^{(n+1)}\mathbf{R}^{-1}_{n+1,i}\Big(y_i - \mu_0(x_i)\Big)$$

- Working on extending to k-fold cross-validation.

Data Distribution



Sources

https://www.itij.com/story/115685/tourists-flee-stromboli-volcano-eruption

- Niklas Linde, Ludovic Baron, Tullio Ricci, Anthony Finizola, André Revil, Filippo Muccini, Luca Cocchi, and Cosmo Carmisciano, *3-d density structure and geological evolution of stromboli volcano (aeolian islands, italy) inferred from land-based and sea-surface gravity data*, Journal of volcanology and geothermal research **273** (2014), 58–69.

- Carl Edward Rasmussen and Christopher K. I. Williams, *Gaussian processes for machine learning*, The MIT Press, 2006.

- A. M. Stuart, *Inverse problems: A bayesian perspective*, Acta Numerica **19** (2010), 451559.

- Ke Alexander Wang, Geoff Pleiss, Jacob R. Gardner, Stephen Tyree, Kilian Q. Weinberger, and Andrew Gordon Wilson, *Exact gaussian processes on a million data points*, 2019.